MELSEC A/Q-Serie

Speicherprogrammierbare Steuerungen

Programmieranleitung

Art.No.: 87432
201004
Version F

Programmieranleitung der
MELSEC-A- und Q-Serie
Artikel-Nr.: 87432

Version			Änderungen / Ergänzungen / Korrekturen
A	09/98	pdp	Programmieranleitung der MELSEC-A- und Q-Serie basierend auf Melsec Medoc *plus*
B	06/99	pdp	Abs. 5.1.2: Geändertes Bitmap für den Kontaktplan der ORP-Anweisung Geändertes Bitmap in den Funktionsweisen der LDP-Anweisung Abs. 7.6.3: Ergänzung eines Hinweises zur Verwendung der CALL-Anweisung Abs. 7.14: Ergänzung um die Anweisungen RSET_K_MD und RSET_K_P_MD
C	06/00	pdp	Abs. 7.11.13: Ergänzung eines Hinweises zur Verwendung der ASC- und ASCP-Anweisung
D	10/00	pdp	Darstellung der Anweisungen in GX Developer Ergänzungen für die Q-Serie (Q02CPU, Q02HCPU, Q06HCPU, Q12HCPU und Q25HCPU)
E	05/02	pdp-dk	Namensänderungen: Melsec Medoc *plus* in GX IEC Developer, GPP/WIN in GX Developer Ergänzungen für die CPU-Module Q00JCPU, Q00CPU und Q01CPU Neue Anweisungen für Multi-CPU-Betrieb: S.TO und FROM Ergänzungen bei den Anweisungen FREAD und FWRITE Erweiterung der Sondermerker und -register für CPUs des System Q ab Version B Getrennte Listen der Verarbeitungszeiten für A-Serie und QnA-Serie/System Q Abs. 6.5.1: Korrektur bei den Fehlerquellen Abs. 6.5.2: Korrektur bei den Fehlerquellen Abs. 6.7.3: Erweiterung der Beschreibung der COM-Anweisung für den Multi-CPU-Betrieb Abs. 6.8.9: Korrektur der Zeitangaben für n1 Abs. 7.1.1: Korrektur der Operanden für MELSEC Q Abs. 7.1.3: Korrektur der Operanden für MELSEC Q Abs. 7.1.5: Korrektur der Operanden für MELSEC Q Abs. 7.1.7: Korrektur der Operanden für MELSEC Q Abs. 7.5.12: Korrektur der Fehlerquellen Kap. 9: Darstellung der Anweisungen in GX IEC Developer Abs. 9.5.1: Korrektur der Zeiten beim Vergleich zwischen RBMOV- und BMOV-Anweisung
F	09/04	pdp-dk	Neues Kapitel 10: Anweisungen für Q4ARCPU Neues Kapitel 11: Anweisungen für Sondermodule Abs. 2.8: Kurzbeschreibung der Anweisungen für Q4ARCPU Abs. 2.9: Kurzbeschreibung der Anweisungen für Sondermodule Neue CPU-Module Q12PHCPU und Q25PHCPU

Zu diesem Handbuch

Die in diesem Handbuch vorliegenden Texte, Abbildungen, Diagramme und Beispiele dienen ausschließlich der Erläuterung, Bedienung, Programmierung und Anwendung der speicherprogrammierbaren Steuerungen der Serien A und Q und des MELSEC System Q.

Sollten sich Fragen zur Programmierung und Betrieb der in diesem Handbuch beschriebenen Geräte ergeben, zögern Sie nicht, Ihr zuständiges Verkaufsbüro oder einen Ihrer Vertriebspartner (siehe Umschlagseite) zu kontaktieren.
Aktuelle Informationen sowie Antworten auf häufig gestellte Fragen erhalten Sie über das Internet (www.mitsubishi-automation.de).

Die MITSUBISHI ELECTRIC EUROPE BV behält sich vor, jederzeit technische Änderungen oder Änderungen dieses Handbuchs ohne besondere Hinweise vorzunehmen.

Inhaltsverzeichnis

1 Einleitung

Dieses Handbuch beschreibt die Programmierung und Verarbeitung der Grundbefehle und Applikationsanweisungen, die in den CPUs der MELSEC A- und QnA-Serie sowie den CPUs des MELSEC System Q zur Verfügung stehen.

1.1 Weitere Handbücher

Programmieranleitung MELSEC QnA-Serie und System Q (Regelungsanweisungen)
- Beschreibung der Anweisungen zur Realisierung von PID-Regelungen

Programming Manual (AD57/58)
- Beschreibung spezieller Anweisungen für die Sondermodule AD57/58

Programming Manual MELSEC QnA Series and MELSEC System Q (SFC)
- Beschreibung der SFC-Anweisungen zur Programmierung von Schrittketten in Ablaufsprache

GX Developer Handbuch
- Beschreibung der Online-Funktionen des GX Developer inklusive Programmierung und Fehlersuche

GX IEC Developer Beginner's Manual/ Einsteigerhandbuch
- Grundlagen zur Programmierung mit dem GX IEC Developer

GX IEC Developer Reference Manual/ Benutzerhandbuch
- Detaillierte Beschreibung zur Programmierung mit dem GX IEC Developer
- Beschreibung der IEC-Anweisungen (IEC-Standardbibliothek)

HINWEIS *Alle Handbücher sind in unserer aktuellen SPS-Preisliste aufgeführt und stehen auf der MITSUBISHI ELECTRIC-Homepage (www.mitsubishi-automation.de) als PDF-Dokument für den Download zur Verfügung.*

1.2 CPU-Typen

Die in diesem Handbuch beschriebenen Funktionen lassen sich durch die aktuellen Versionen von GX Developer und GX IEC Developer auf alle CPU-Typen übertragen, solange diese die benutzten Anweisungen unterstützen.

Die beschriebenen Anweisungen gelten für die folgenden MELSEC SPS- und CPU-Typen:

SPS-Typ		CPU-Typen
A-Serie	AnA/AnU	A2A, A2A-S1, A2U, A2U-S1, A3A, A3U
	AnAS/AnUS	A2AS, A2AS-S1, A2AS-S30, A2AS-S60, A2US, A2US-S1
	AnN	A1, A2, A2C A3M, A3N
	AnS	A1S, A1S-S1, A2S, A2S-S1
Q-Serie	QnA	Q2A/Q2AS, Q2A-S1/Q2AS-S Q3A Q4A, Q4AR
System Q	Q (Single-Prozessor-CPU)	Q00J
	Q (Multi-Prozessor-CPUs)	Q00, Q01 (eingeschränkt Multi-Prozessor-tauglich) Q02, Q02H, Q06H, Q12H, Q12PH, Q25H, Q12PH PC-CPU-Module: PPC-CPU686(MS)-64 PPC-CPU686(MS)-128

Wenn generell von MELSEC A, MELSEC Q oder - z. B. in den Tabellen- von A und Q gesprochen wird, sind damit alle CPU-Typen der A-Serie bzw. der Q-Serie und des MELSEC System Q gemeint. Auf Ausnahmen wird hingewiesen.

1.3 Software

Alle beschriebenen Anweisungen können – abgesehen von wenigen Ausnahmen – in den zur Verfügung stehenden Software-Paketen verwendet werden:
- GX Developer
- GX IEC Developer

Die Beispiele in diesem Handbuch wurden mit dem GX IEC Developer erstellt. Die Darstellung in der MELSEC-Anweisungsliste entspricht grundsätzlich der in GX Developer.
Alle in diesem Handbuch beschriebenen Anweisungen gehören beim GX IEC Developer zur Herstellerbibliothek.
Je nach eingestellter CPU erscheinen im GX IEC Developer-Dialogfenster zur Auswahl einer Anweisung immer nur die Anweisungen, die in der aktuellen CPU auch tatsächlich verarbeitet werden können.

1.4 Finden einer Anweisung

Für Geübte

Wenn Sie mit der Programmierung der Anweisungen in der MELSEC A- und Q-Serie sowie im System Q bereits vertraut sind, schlagen Sie in den Anweisungskapiteln 5 bis 9 nach. In der Kopfzeile erscheint der Name der Anweisung, wie er beim GX Developer und im MELSEC-Editor des GX IEC Developer verwendet wird.

Für Einsteiger

Wenn Sie mit der Handhabung der Anweisungen noch nicht so vertraut sind, gehen Sie wie folgt vor:

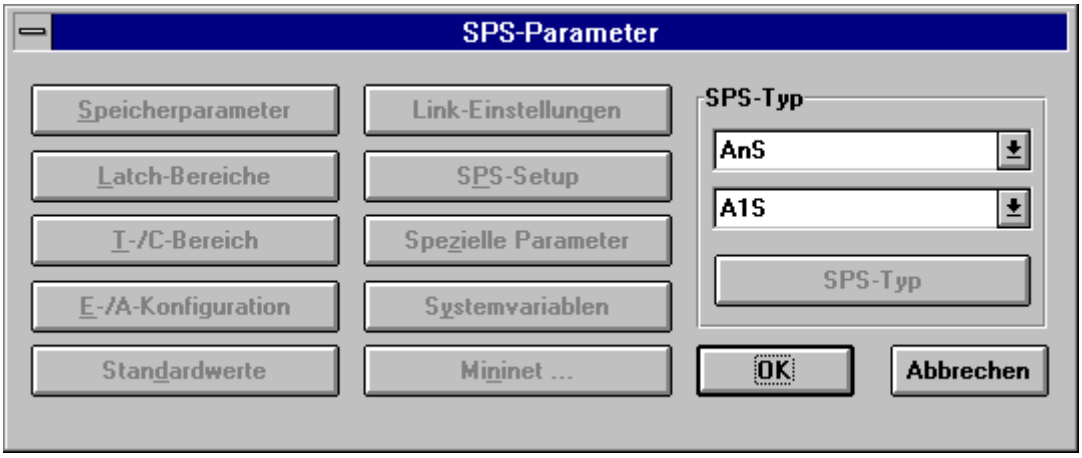
- Lesen Sie die Hinweise in Kap. 3 zu den unterschiedlichen Schreibweisen der Anweisungen im MELSEC- und IEC-Editor.
- Lesen Sie die Hinweise in Kap. 4, um den gleichbleibenden Aufbau jeder Anweisungsbeschreibung zu verstehen.
- Nutzen Sie
  - die tabellarische Übersicht der Anweisungsgruppen mit Kurzbeschreibungen in Kap. 2.
  - den Index, in dem alle Anweisungen aufgeführt sind.

HINWEIS *Alle in diesem Handbuch beschriebenen Anweisungen befinden sich genauso detailliert in der Online-Hilfe des* GX IEC Developer*.*

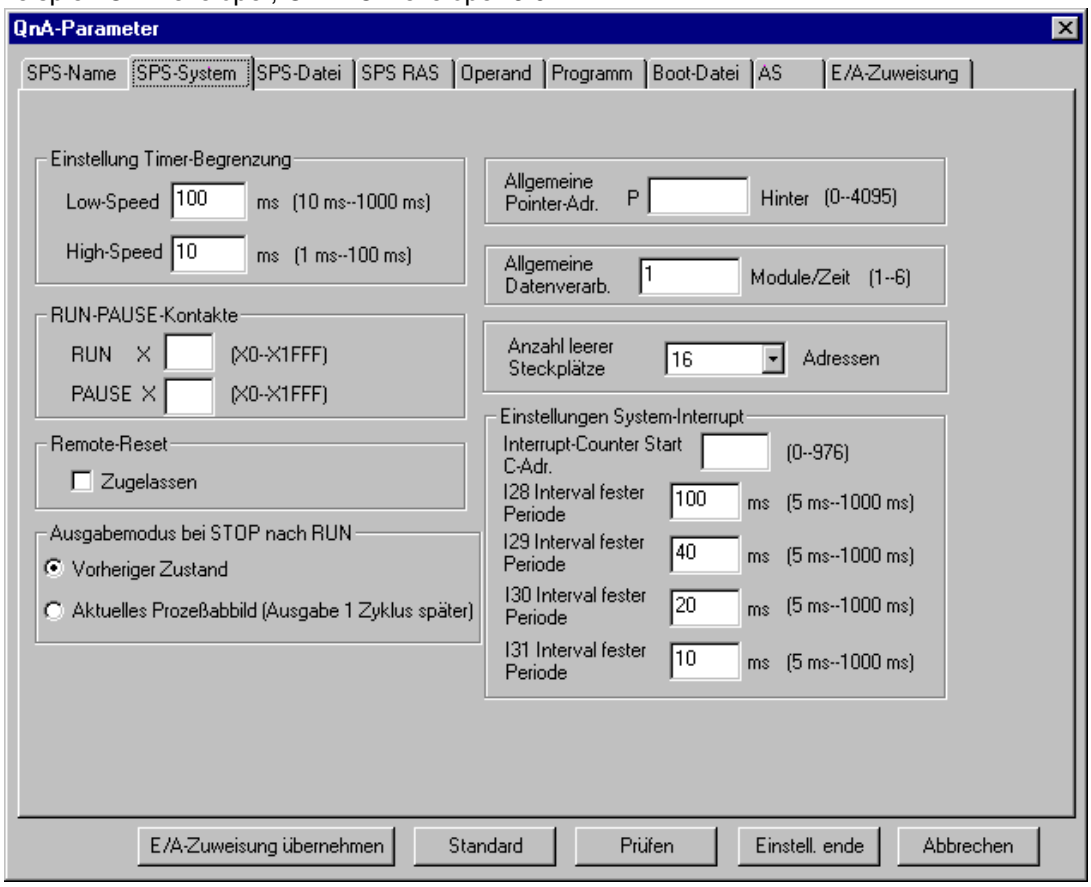
1.5 SPS-Parameter

Die Parameter dienen dazu, bestimmte Funktionen, Operandenbereiche usw. festzulegen. Zur Programmierung der in diesem Handbuch beschriebenen Funktionen können die Parameter unverändert bleiben oder entsprechend den Anwenderbedürfnissen geändert werden. Nähere Hinweise zur Einstellung der SPS-Parameter enthalten die entsprechenden Hardware-Beschreibungen der CPUs und Programmier-Handbücher.

Beispiel : GX IEC Developer



Beispiel: GX Developer, GX IEC Developer 6.0



1.6 Gegenüberstellung: GX IEC Developer und GX Developer

Die folgende Tabelle gibt eine Übersicht über die wichtigsten Merkmale der Software-Pakete GX IEC Developer und GX Developer.

GX IEC Developer	GX Developer
Strukturierte Benutzung	Einfache Benutzung
Programmierung nach IEC 1131	—
Verfügbare Editioren: Anweisungsliste, Kontaktplan, Strukturierter Text, SFC, FUB	Verfügbare Editioren: Anweisungsliste, Kontaktplan, SFC
Funktionen und Funktionsbausteine	Funktionsbausteine (ab V. 7)
Programmänderungen im Online-Betrieb	Programmänderungen im Online-Betrieb Austausch von Programmen online möglich
Diagnosefunktionen für die SPS	Diagnosefunktionen für die SPS
Diagnosefunktionen für Netzwerke	Diagnosefunktionen für Netzwerke

2 Anweisungen

2.1 Einteilung der Anweisungen

Die Anweisungen lassen sich in mehrere Hauptgruppen einteilen, die in der folgenden Tabelle aufgeführt sind:

Anweisungsgruppen		Beschreibung	Referenz
Grund-befehlssatz	Eingangsanweisungen	Beginn einer Verknüpfung, Reihen- und Serienschaltung von Kontakten	Abs. 5.1
	Verknüpfungs-anweisungen	Serielle und parallele Blockverknüpfung, Speichern und Verarbeiten eines Verknüpfungsergebnisses, Signalumkehr von Operationsergebnissen, Umwandlung von Operationsergebnissen in gepulste Ergebnisse, Setzen von Flankenmerkern	Abs. 5.2
	Ausgangs-anweisungen	Bitoperanden, Zähler- und Zeitkontakte, Ausgabe, Setzen und Rücksetzen von Fehlermerkern, Setzen und Rücksetzen von Operanden, flankengesteuerte Differenzausgabe, Umkehr des Schaltzustandes eines Operanden, Erzeugung von Schaltimpulsen	Abs. 5.3
	Verschiebe-anweisungen	Verschieben von Bit-Operanden	Abs. 5.4
	Master-Control-Anweisungen	Aktivieren und Deaktivieren einzelner Programmbereiche	Abs. 5.5
	Programmende-anweisungen	Ende eines Programmbereiches, Ende von Haupt- und Unterprogrammen	Abs. 5.6
	Sonstige Anweisungen	Unterbrechung der Verarbeitung, Leerschritt im Programm	Abs. 5.7
Applikations-anweisungen Teil I	Vergleichs-anweisungen	Datenvergleich, wie z.B. =, >, ≥ usw.	Abs. 6.1
	Arithmetik-anweisungen	Addition, Subtraktion, Multiplikation, Division, von BIN- und BCD-Daten, Gleitkommazahlen und BIN-Datenblöcken, Verknüpfung von Zeichenfolgen, Inkrement, Dekrement	Abs. 6.2
	Konvertierungs-anweisungen	Datenkonvertierung wie z.B. BCD → BIN und BIN → BCD	Abs. 6.3
	Transferanweisungen	Übertragung, Austausch und Negation von Daten	Abs. 6.4
	Programmverzwei-gungsanweisungen	Sprung, Unterprogrammaufruf	Abs. 6.5
	Interrupt-Programmaufruf	Interrupt-Programmaufruf	Abs. 6.6
	Datenaktualisierungs-anweisungen	Link-Refresh und E/A-Schnittstellenauffrischung	Abs 6.7
	Weitere Anweisungen	Ein-/Zweiphasiger Auf-/Abwärtszähler, programmierbare Timer, Sonderfunktionstimer, Positionieranweisung, Rampensignal, Impulszähler, Impulsausgang, Puls-Weiten-Modulation, Eingabematrix	Abs. 6.8

Anweisungsgruppen		Beschreibung	Referenz
Applikations-anweisungen Teil II	Logik-anweisungen	UND-/ ODER-Logik, Exklusiv-ODER-/ NOR-Logik	Abs. 7.1
	Rotations-anweisungen	Datenrotation rechts/links mit 16- und 32-Bit	Abs. 7.2
	Verschiebe-anweisungen	Bit- oder blockweises Verschieben innerhalb eines Datenwortes	Abs. 7.3
	Bit-Verarbeitungs-anweisungen	Setzen und Rücksetzen von Bits, Bitabfrage	Abs. 7.4
	Datenverarbeitungs-anweisungen	Daten in definierten Bereichen suchen, Daten kodieren und dekodieren, Datenwerte auftrennen und zusammenführen	Abs. 7.5
	Strukturierte Programmanweisung	Wiederholungsanweisung, Unterprogrammaufruf, Unterprogrammaufruf zwischen Programmdateien, Umschaltung zwischen Main- und Sub-Programmbereich, Mikrocomputer-Programmaufruf, indizierte Adressierung eines gesamten Programmbereichs, Speicherung indizierter Operandenadressen in einer Index-Liste	Abs. 7.6
	Verarbeitungs-anweisungen für Datenlisten	Daten zur weiteren Verarbeitung in und aus einer Datenliste schreiben und lesen, bestimmte Datenblöcke in der Datenliste löschen und einfügen	Abs. 7.7
	Anweisungen für den Pufferspeicherzugriff	Zugriff auf den Pufferspeicher eines Sondermoduls oder eines Remote-Moduls	Abs. 7.8
	Display-Anweisungen	Ausgabe von ASCII-Zeichen an die Ausgänge eines Moduls oder die LED-Anzeige einer CPU	Abs. 7.9
	Anweisungen zur Fehlerdiagnose und Fehlerbeseitigung	Fehlerkontrolle, Status Latch, Abtastüberwachung (Sampling Trace), Programmüberwachung (Program Trace)	Abs. 7.10
	Verarbeitungs-anweisungen für Zeichenfolgen	Zeichenfolgenverarbeitung (ASCII-Code)	Abs. 7.11
	Anweisungen für Sonderfunktionen	Anweisungen zu Winkelfunktionen, Wurzel- und Exponentialrechnungen mit BCD-Daten und Gleitkommazahlen	Abs. 7.12
	Datenkontroll-anweisungen	Überprüfung von Eingangsdaten bezüglich vorgegebener Wertebereiche und Speicherung der überprüften Daten	Abs. 7.13
	Umschalt-anweisungen für Datenregisterblöcke und Dateien	Umschaltung zwischen File-Registerblöcken und Dateien	Abs. 7.14
	Uhr-Anweisungen	Schreiben und Lesen von Uhr-Daten	Abs. 7.15
	Anweisungen für periphere Baugruppen	Ausgabe von Meldungen und Tastatureingabe an peripheren Baugruppen	Abs. 7.16
	Programm-anweisungen	Anweisungen zum Wechsel der Programm-ausführungsmodi	Abs. 7.17
	Weitere Anweisungen	WDT zurücksetzen, Übertragstelle (Carry) setzen und rücksetzen, Impulsgeber, direktes Lesen und Schreiben von Bytes, Tastatureingabe, Sichern und Wiederherstellen von Index-Registerinhalten, Schreiben von Daten in EEPROM-Register	Abs. 7.18

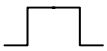
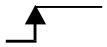
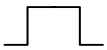
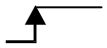
Anweisungsgruppen		Beschreibung	Referenz
Data-Link-Anweisungen	Netzwerk-Daten-aktualisierungs-Anweisungen (Refresh-Anweisungen)	Anweisungen für Datenaktualisierungen in Netzwerk-modulen.	Abs. 8.5
	Erweiterte Data-Link-Anweisungen	Lesen und Schreiben von Daten in und aus Ziel-Stationen in Ziel-Netzwerken, Senden von Daten an Netzwerkmodule in Ziel-Stationen in Ziel-Netzwerken, Lesen mittels SEND-Anweisung gesendeter Daten, Datenanforderung an andere Stationen (Schreib-/Lese-Operationen mit Uhrdaten, RUN-/STOP-Operation), Lesen und Schreiben von Daten in und aus Sondermodulen in ausgelagerten Ein-/Ausgabe-Stationen.	Abs. 8.6
	Zur A-Serie kompatible Data-Link-Anweisungen	Lesen und Schreiben von Daten in und aus Ziel-Stationen in Ziel-Netzwerken, Lesen und Schreiben von Daten in und aus lokalen Stationen (nur an Master-Stationen), Lesen und Schreiben von Daten in und aus Sondermodulen in ausgelagerten Ein-/Ausgabe-Stationen.	Abs. 8.7
	Lesen und Schreiben von Routing-Informationen	Lesen und Schreiben der Routing-Parameter (Netzwerks- und Stationsnummer der Relais-Station, Stationsnummer der Routing-Station)	Abs. 8.8
Anweisungen für eine System Q-CPU	Modul-Informationen auslesen	Direktes Lesen der Modulspeicher	Abs. 9.1
	Fehlererkennung und -beseitigung	Überwachung (Trace) setzen/rücksetzen	Abs. 9.2
	Transfer von Daten in und aus Dateien	Daten in eine Datei schreiben, Daten aus einer Datei lesen	Abs. 9.3
	Programm-anweisungen	Programm aus Speicher laden, Programm im Speicher löschen	Abs. 9.4
	Transferanweisungen	Übertragung von Daten	Abs. 9.5
	Anweisungen zum Datenaustausch im Multi-CPU-Betrieb	Daten in den gemeinsamen Speicherbereich eintragen, Daten aus dem gemeinsamen Speicherbereich einer anderen CPU lesen	Abs. 9.6
Anweisungen für eine Q4ARCPU	Betriebsart einstellen	Anlaufverhalten der CPU und das Verhalten beim Umschalten auf die Reserve-CPU eines redundanten Systems wählen	Abs. 10.1
	Transferanweisungen	Daten von der aktiven CPU zur Reserve-CPU transferieren und Datenaustausch mit dem Pufferspeicher eines oder auch mehrerer Sondermodule	Abs. 10.2
Anweisungen für Sondermodule	Anweisungen für serielle Schnittstellenmodule	Empfangene Daten in einem Interrupt-Programm in die CPU übertragen, anwenderdefinierte Datenrahmen lesen, festlegen oder löschen, Daten mittels anwenderdefinierter Datenrahmen versenden	Abs. 11.1
	Anweisungen für PROFIBUS/DP-Module	Datenaustausch mit dem Pufferspeicher eines PROFIBUS-Moduls	Abs. 11.2
	Anweisungen für ETHERNET-Module	Empfangene Daten aus feste Puffer lesen, Daten in feste Puffer eintragen, Auf- und Abbau einer Verbindung, Fehlerspeicher löschen und „ERR.“-LED ausschalten, Fehlercode aus dem ETHERNET-Modul lesen, ETHERNET-Modul erneut initialisieren	Abs. 11.3
	Anweisung für MELSECNET/10	Stationen festlegen, die beim Duplexbetrieb verbunden sind	Abs. 11.4
	Anweisungen für CC-Link	Netzwerkparameter einstellen, Parameter für automatische Aktualisierung festlegen, Daten aus dem Pufferspeicher eines CC-Link-Moduls oder der SPS-CPU dieser Station lesen oder Daten schreiben, Daten unter Verwendung eines Handshake aus dem Pufferspeicher einer intelligenten Station lesen oder in den Pufferspeicher eintragen, Daten aus dem automatisch aktualisierten Speicherbereich lesen oder Daten in diesen Bereich eintragen	Abs. 11.5

2.2 Übersicht der Anweisungen

2.2.1 Erläuterungen der Übersichtstabelle

Die nachfolgenden Abschnitte 2.3 bis 2.6 enthalten eine Übersicht aller in diesem Handbuch beschriebenen Anweisungen.

Im folgenden wird die Aufteilung der Übersichtstabelle näher beschrieben:

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Addition und Subraktion von 16-Bit-Binärdaten	+	s, d	(d)+(s) → (d)		3	5	6.2.1
	+P						6.2.1
	+	s1, s2, d1	(s1)+(s2) → (d1)		4	7	6.2.1
	+P						6.2.1

(1) (2) (3) (4) (5) (6) (7) (8)

Erläuterung der einzelnen Spalten:

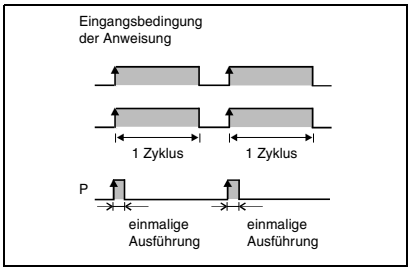
(1) Anweisungsgruppe

(2) Angabe des Anweisungsnamens („Befehl“) für die Programmierung

Die Anweisungsnamen werden in der MELSEC-Schreibweise dargestellt (Erläuterung der Schreibweise siehe Kapitel 3.2).

Grundsätzlich werden 16-Bit-Anweisungen dargestellt. Bei 32-Bit-Anweisungen ist dem Anweisungsnamen ein „D“ vorangestellt.
- 16-Bit-Anweisung: +
- 32-Bit-Anweisung: D+

Puls-Anweisungen, d.h. Anweisungen, die nur bei steigender Flanke eines Signals ausgeführt werden, wird ein „P“ angefügt.
- normale Anweisung: +
- Puls-Anweisung: +P



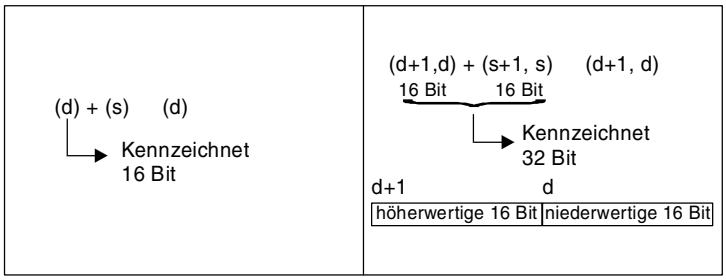
Anweisungen, die Zeichenfolgen verarbeiten, werden mit einem vorangestellten „$“ gekennzeichnet:
- normale Anweisung: +
- Zeichenfolgen-Anweisung: $+

(3) Angabe der zu verwendenden Variablen

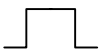
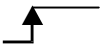
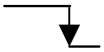
Hier werden die Variablen angegeben. Die Datenquelle wird mit einem „s“ (Source), das Datenziel wird mit einem „d“ (Destination) gekennzeichnet.

Beispiel: s = wenn nur eine Datenquelle vorhanden ist
s1, s2 = wenn mehrere Datenquellen vorhanden sind
s+0, s+1, (s1)+0, (s1)+1 = bei 32-Bit-Anweisungen
z. B. s1 = Datenregister D0, (s1)+1 =Datenregister D1
s+0, s+1, s+2, s+3 = 4 aufeinanderfolgende Operanden z. B. bei einem Array

(4) Bedeutung und Verarbeitung der kompletten Steuerungsanweisung



(5) Anzeige der Ausführungsbedingung entsprechend der folgenden Tabelle

Symbol	Ausführungsbedingung
keine Angabe	Die Anweisung wird ständig und unabhängig vom Zustand der vorgeschalteten Bedingung ausgeführt. Ist die Eingangsbedingung nicht gesetzt, wird die Anweisung nicht ausgeführt.
	Die Anweisung wird so lange ausgeführt, wie die Eingangsbedingung vorliegt. Fällt die Eingangsbedingung ab, wird die Anweisung nicht weiter ausgeführt und verarbeitet.
	Bei dieser Anweisung handelt es sich um eine gepulste Anweisung. Sie wird nur bei ansteigender Flanke des Eingangssignals einmalig ausgeführt (z.B. wenn die Eingangsbedingung von AUS nach EIN wechselt). Anschließend wird die Anweisung auch bei bestehendem Eingangssignal nicht weiter ausgeführt und verarbeitet.
	Bei dieser Anweisung handelt es sich ebenfalls um eine gepulste Anweisung. Sie wird nur bei abfallender Flanke des Eingangssignals einmalig ausgeführt (z.B. wenn die Eingangsbedingung von EIN nach AUS wechselt). Anschließend wird die Anweisung auch bei bestehendem Eingangssignal nicht weiter ausgeführt und verarbeitet.

(6+7) Angabe der Programmschritte

Es wird die Anzahl von Schritten angegeben, die bis zur vollständigen Ausführung der Anweisung erforderlich sind. Es wird hier unterschieden zwischen der MELSEC A- und Q-Serie/System Q. Nähere Erläuterungen hierzu enthält der Abschnitt 3.9.

(8) Anzeige des Referenzabschnittes

Zeigt die Nummer des Handbuch-Kapitels, in dem die Anweisung beschrieben wird.

2.3 Grundbefehlssatz

2.3.1 Eingangsanweisungen

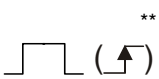
Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Eingangs-anweisungen	LD		Beginn einer Verknüpfung (Lade)		*1	1	5.1.1
	LDI		Beginn einer Verknüpfung (Lade Öffnerkontakt)				
	AND		Reihenschaltung von Eingangskontakten (Schließer)				
	ANI		Reihenschaltung von Eingangskontakten (Öffner)				
	OR		Parallelschaltung von Eingangskontakten (Schließer)				
	ORI		Parallelschaltung von Eingangskontakten (Öffner)				
	LDP		Beginn einer flankengesteuerten Verknüpfung		*1	2	5.1.2
	LDF		Beginn einer flankengesteuerten Verknüpfung				
	ANDP	s	Reihenschaltung, flankengesteuert (ansteigende Flanke)				
	ANDF	s	Reihenschaltung, flankengesteuert (abfallende Flanke)				
	ORP	s	Parallelschaltung, flankengesteuert (ansteigende Flanke)				
	ORF	s	Parallelschaltung, flankengesteuert (abfallende Flanke)				

*: Die Anzahl der Programmschritte ist abhängig von den verwendeten Operanden.
- Bei Verwendung von internen Operanden oder File-Registern (R0 bis R32767): 1
- Bei Verwendung eines direkt adressierbaren Eingangs (DX) : 2
- Bei Verwendung anderer Operanden : 3
- Bei Verwendung von Fileregistern 2R auf den Speicherkarten kann sich die Anzahl der Schritte verdoppeln

2.3.2 Verknüpfungsanweisungen

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Verknüpfungsanweisungen	ANB	—	Serielle Blockverknüpfung (Reihenverknüpfung von Parallelschaltungen)		1	1	5.2.1
	ORB		Parallele Blockverknüpfung (Parallelverknüpfung von Reihenschaltungen)				
	MPS	—	Ergebnisverarbeitung (Ergebnis speichern)		1	1	5.2.2
	MRD		Ergebnisverarbeitung (Ergebnis lesen)				
	MPP		Ergebnisverarbeitung (Ergebnis lesen und löschen)				
	INV	—	Signalumkehr von Operationsergebnissen		1		5.2.3
	MEP	—	Umwandlung von Operationsergebnissen in gepulste Ergebnisse (bei ansteigender Flanke)		1		5.2.4
	MEF		Umwandlung von Operationsergebnissen in gepulste Ergebnisse (bei abfallender Flanke)				
	EGP	d	Setzen des Flankenmerkers bei ansteigender Flanke des Operationsergebnisses		1		5.2.5
	EGF		Setzen des Flankenmerkers bei abfallender Flanke des Operationsergebnisses				

2.3.3 Ausgangsanweisungen

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte Q	Schritte A	Referenz
Ausgangs-anweisungen	OUT	d	Setzen von Ausgängen		* 1	* 1	5.3.1
	SET	d	Setzen eines Operanden	** 	* 1		5.3.5
	RST	d	Rücksetzen eines Operanden	** 	2	* 1	5.3.6
	PLS	d	Ausgabe bei steigender Signalflanke		2	* 3	5.3.8
	PLF		Ausgabe bei fallender Signalflanke				
	FF	s	Invertierung eines Bit-Ausgangsoperanden		2		5.3.9
	DELTA	d	Erzeugung eines Schaltimpuls an einem direkt adressierbaren Ausgang		2		5.3.11
	DELTAP						

*: Die Anzahl der Programmschritte ist abhängig von den verwendeten Operanden. Die genaue Anzahl entnehmen Sie dem Abschnitt, in dem die einzelnen Anweisungen beschrieben sind.

**: Diese  Ausführungsbedingung wird nur bei Verwendung des Fehlermerkers (F) angewendet.

2.3.4 Verschiebeanweisungen

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte Q	Schritte A	Referenz
Verschiebe-anweisungen	SFT	d	Verschiebung von Bit-Operanden		2	* 3	5.4.1
	SFTP						

*: Die Anzahl der Schritte bei Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.9.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

2.3.5 Master-Control-Anweisungen

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Master-Control-Anweisungen	MC	n, d	Aktivierung einzelner Programmbereiche		2	* 3/5	5.5.1
	MCR	n	Deaktivierung einzelner Programmbereiche		1		

*: Die Anzahl der Schritte beträgt 5 für die MC-Anweisung und 3 für die MCR-Anweisung. Näheres zur Anzahl der Schritte bei Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.9.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

2.3.6 Programmendeanweisungen

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Programm-ende-anweisungen	FEND	—	Beenden eines Programmbereichs		1		5.6.1
	END		Beenden eines Programms				5.6.2

2.3.7 Sonstige Anweisungen

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Sonstige Anweisungen	STOP	—	Unterbrechungs-anweisung		1		5.7.1
	NOP	—	Leerschritt im Programm				5.7.2

2.4 Applikationsanweisungen I

2.4.1 Vergleichsanweisungen

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
16-Bit-Datenvergleich	LD=	s1, s2	Ausgang schaltet, wenn s1 = s2		3	* 5/7	6.1.1
	AND=						
	OR=						
	LD<>	s1, s2	Ausgang schaltet, wenn s1 ≠ s2		3	* 5/7	6.1.1
	AND<>						
	OR<>						
	LD>	s1, s2	Ausgang schaltet, wenn s1 > s2		3	* 5/7	6.1.1
	AND>						
	OR>						
	LD<=	s1, s2	Ausgang schaltet, wenn s1 <= s2		3	* 5/7	6.1.1
	AND<=						
	OR<=						
	LD<	s1, s2	Ausgang schaltet, wenn s1 < s2		3	* 5/7	6.1.1
	AND<						
	OR<						
	LD>=	s1, s2	Ausgang schaltet, wenn s1 >= s2		3	* 5/7	6.1.1
	AND>=						
	OR>=						

*: Die Anzahl der Programmschritte ist abhängig von den verwendeten Operanden.
Die genaue Anzahl entnehmen Sie dem Abschnitt, in dem die einzelnen Anweisungen beschrieben sind.

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
32-Bit-Datenvergleich	LDD=	s1, s2	Ausgang schaltet, wenn s1 = s2		* 3	** 11	6.1.2
	ANDD=						
	ORD=						
	LDD<>	s1, s2	Ausgang schaltet, wenn s1 ≠ s2		* 3	** 11	6.1.2
	ANDD<>						
	ORD<>						
	LDD>	s1, s2	Ausgang schaltet, wenn s1 > s2		* 3	** 11	6.1.2
	ANDD>						
	ORD>						
	LDD<=	s1, s2	Ausgang schaltet, wenn s1 <= s2		* 3	** 11	6.1.2
	ANDD<=						
	ORD<=						
	LDD<	s1, s2	Ausgang schaltet, wenn s1 < s2		* 3	** 11	6.1.2
	ANDD<						
	ORD<						
	LDD>=	s1, s2	Ausgang schaltet, wenn s1 >= s2		* 3	** 11	6.1.2
	ANDD>=						
	ORD>=						

*: Die Anzahl der Programmschritte ist abhängig vom Typ der CPU und von den verwendeten Operanden.
- Bei Verwendung einer QnA-CPU oder einer Single-Prozessor-Q-CPU: 3
- Bei Verwendung einer Multi-Prozessor-Q-CPU und
  interner Wortoperanden (außer File-Register ZR): 5
  Konstanten: 5
  Bit-Operanden, deren Adresse ein Vielfaches von 16 ist,
  die die Bit-Blockbezeichnung K8 haben und die nicht durch Index-Vergabe bearbeitet werden: 5
- Bei Verwendung einer Multi-Prozessor-Q-CPU und anderer Operanden als oben aufgeführt: 3

Obwohl bei der Q-CPU mehr Schritte als bei der QnA-CPU gebraucht werden, ist die Verarbeitungsgeschwindigkeit höher.

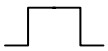
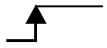
**: Die Anzahl der Programmschritte ist abhängig von den verwendeten Operanden.
Die genaue Anzahl entnehmen Sie dem Abschnitt, in dem die einzelnen Anweisungen beschrieben sind.

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Vergleich von Gleitkomma-zahlen	LDE=	s1, s2	Ausgang schaltet, wenn s1 = s2		3		6.1.3
	ANDE=						
	ORE=						
	LDE<>	s1, s2	Ausgang schaltet, wenn s1 ≠ s2		3		6.1.3
	ANDE<>						
	ORE<>						
	LDE>	s1, s2	Ausgang schaltet, wenn s1 > s2		3		6.1.3
	ANDE>						
	ORE>						
	LDE<=	s1, s2	Ausgang schaltet, wenn s1 <= s2		3		6.1.3
	ANDE<=						
	ORE<=						
	LDE<	s1, s2	Ausgang schaltet, wenn s1 < s2		3		6.1.3
	ANDE<						
	ORE<						
	LDE>=	s1, s2	Ausgang schaltet, wenn s1 >= s2		3		6.1.3
	ANDE>=						
	ORE>=						

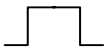
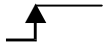
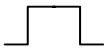
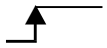
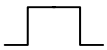
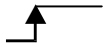
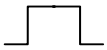
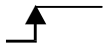
Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Vergleich von Zeichenfolgen	LD$=	s1, s2	* Die Zeichenfolgen in s1 und s2 werden zeichenweise verglichen. Der Ausgang schaltet, wenn s1 = s2		3		6.1.4
	AND$=						
	OR$=						
	LD$<>	s1, s2	* Die Zeichenfolgen in s1 und s2 werden zeichenweise verglichen. Der Ausgang schaltet, wenn s1 ≠ s2		3		6.1.4
	AND$<>						
	OR$<>						
	LD$>	s1, s2	* Die Zeichenfolgen in s1 und s2 werden zeichenweise verglichen. Der Ausgang schaltet, wenn s1 > s2		3		6.1.4
	AND$>						
	OR$>						
	LD$<=	s1, s2	* Die Zeichenfolgen in s1 und s2 werden zeichenweise verglichen. Der Ausgang schaltet, wenn s1 <= s2		3		6.1.4
	AND$<=						
	OR$<=						
	LD$ <	s1, s2	* Die Zeichenfolgen in s1 und s2 werden zeichenweise verglichen. Der Ausgang schaltet, wenn s1 < s2		3		6.1.4
	AND$<						
	OR$<						
	LD$>=	s1, s2	* Die Zeichenfolgen in s1 und s2 werden zeichenweise verglichen. Der Ausgang schaltet, wenn s1 >= s2		3		6.1.4
	AND$>=						
	OR$>=						

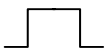
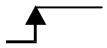
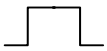
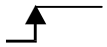
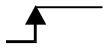
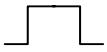
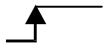
*: Die Bedingungen, unter denen Zeichenfolgenvergleiche durchgeführt werden können, werden unten beschrieben.

- Identisch: Alle Zeichen in den Zeichenfolgen sind identisch.
- Größere Zeichenfolge: Sind die Zeichenfolgen unterschiedlich, wird die Folge mit den meisten Zeichen festgelegt.
- Kleinere Zeichenfolge: Sind die Zeichenfolgen unterschiedlich, wird die Folge mit den wenigsten Zeichen festgelegt.

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Blockweiser Vergleich von Binärdaten	BKCMP=	s1, s2, n, d1	Verglichen werden die Zeichen aus n 16-Bit-Blöcken von s1 mit den Zeichen aus n 16-Bit-Blöcken aus s2. Das Vergleichsergebnis wird in n 16-Bit-Blöcke von d1 gespeichert.		5		6.1.5
	BKCMP<>	s1, s2, n, d1					
	BKCMP>	s1, s2, n, d1					
	BKCMP<=	s1, s2, n, d1					
	BKCMP<	s1, s2, n, d1					
	BKCMP>=	s1, s2, n, d1					
	BKCMP=P	s1, s2, n, d1					
	BKCMP<>P	s1, s2, n, d1					
	BKCMP>P	s1, s2, n, d1					
	BKCMP<=P	s1, s2, n, d1					
	BKCMP<P	s1, s2, n, d1					
	BKCMP>=P	s1, s2, n, d1					

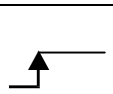
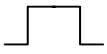
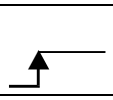
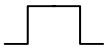
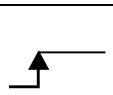
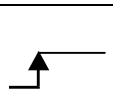
2.4.2 Arithmetikanweisungen

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Addition und Subraktion von 16-Bit-Binärdaten	+	s, d	(d)+(s) → (d)		3	5	6.2.1
	+P						6.2.1
	+	s1, s2, d1	(s1)+(s2) → (d1)		4	7	6.2.1
	+P						6.2.1
	-	s, d	(d)-(s) → (d)		3	5	6.2.1
	-P						6.2.1
	-	s1, s2, d1	(s1)-(s2) → (d1)		4	7	6.2.1
	-P						6.2.1

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Addition und Subraktion von 32-Bit-Binärdaten	D+	s, d	(d+1, d)+(s+1, s) → (d+1, d)		* 3	9	6.2.2
	D+P						6.2.2
	D+	s1, s2, d1	((s1)+1, s1)+((s2) +1, s2) → ((d1)+1, d1)		** 4	11	6.2.2
	D+P						6.2.2
	D-	s, d	(d+1, d)-(s+1, s) → (d+1, d)		* 3	9	6.2.2
	D-P						6.2.2
	D-	s1, s2, d1	((s1)+1, s1)-((s2)+1,s2) → ((d1)+1, d1)		** 4	11	6.2.2
	D-P						6.2.2

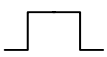
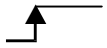
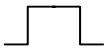
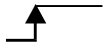
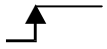
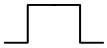
*: Die Anzahl der Programmschritte ist abhängig vom Typ der CPU und von den verwendeten Operanden.
- Bei Verwendung einer QnA-CPU oder einer Single-Prozessor-Q-CPU: 3
- Bei Verwendung einer Multi-Prozessor-Q-CPU und interner Wortoperanden (außer File-Register ZR): 5
  Konstanten:5
  Bit-Operanden, deren Adresse ein Vielfaches von 16 ist, die die Bit-Blockbezeichnung K8 haben und die nicht durch Index-Vergabe bearbeitet werden: 5
- Bei Verwendung einer Multi-Prozessor-Q-CPU und anderer Operanden als oben aufgeführt: 3

**: Die Anzahl der Programmschritte ist abhängig vom Typ der CPU und von den verwendeten Operanden.
- Bei Verwendung einer QnA-CPU oder einer Single-Prozessor-Q-CPU: 4
- Bei Verwendung einer Multi-Prozessor-Q-CPU und interner Wortoperanden (außer File-Register ZR): 6
  Konstanten: 6
  Bit-Operanden, deren Adresse ein Vielfaches von 16 ist, die die Bit-Blockbezeichnung K8 haben und die nicht durch Index-Vergabe bearbeitet werden: 6
- Bei Verwendung einer Multi-Prozessor-Q-CPU und anderer Operanden als oben aufgeführt: 4

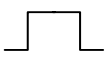
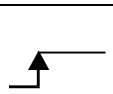
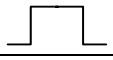
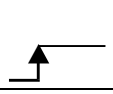
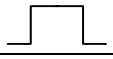
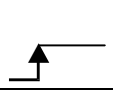
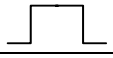
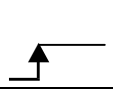
Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Multiplikation und Division von 16-Bit-Binärdaten	x	s1, s2, d1	(s1)x(s2) → ((d1)+1, d1)		* 4	** 7	6.2.4
	xP						6.2.4
	/	s1, s2, d1	(s1)/(s2) → Quotient (d1), Rest ((d1)+1)		* 4	** 7	6.2.4
	/P						6.2.4
Multiplikation und Division von 32-Bit-Binärdaten	Dx	s1, s2, d1	((s1)+1, s1)x((s2)+1, s2) → ((d1)+3, (d1)+2, (d1)+1, d1)		* 4	** 11	6.2.4
	DxP						6.2.4
	D/	s1, s2, d1	((s1)+1, s1)/((s2)+1, s2) → Quotient ((d1)+1, d1), Rest ((d1)+3, (d1)+2)		* 4	** 11	6.2.4
	D/P						6.2.4

*: Die Anzahl der Programmschritte ist abhängig vom Typ der CPU und von den verwendeten Operanden.
- Bei Verwendung einer QnA-CPU oder einer Single-Prozessor-Q-CPU: 4
- Bei Verwendung einer Multi-Prozesor-Q-CPU und
  interner Wortoperanden (außer File-Register ZR): 3
  Konstanten: 3
  Bit-Operanden, deren Adresse ein Vielfaches von 16 ist,
  die die Bit-Blockbezeichnung K4 haben und die nicht durch Index-Vergabe bearbeitet werden: 3
- Bei Verwendung einer Multi-Prozessor-Q-CPU und anderer Operanden als oben aufgeführt: 4

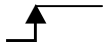
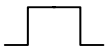
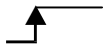
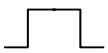
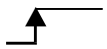
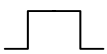
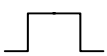
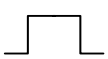
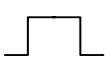
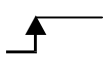
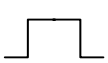
**: Die Anzahl der Programmschritte ist abhängig von den verwendeten Operanden.
Die genaue Anzahl entnehmen Sie dem Abschnitt, in dem die einzelnen Anweisungen beschrieben sind.

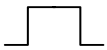
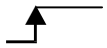
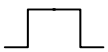
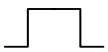
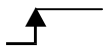
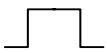
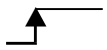
Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Addition und Subtraktion von BCD-Daten (4-stellig)	B+	s, d	(d)+(s) → (d)		3	* 7	6.2.5
	B+P						6.2.5
	B+	s1, s2, d1	(s1)+(s2) → (d1)		4	* 9	6.2.5
	B+P						6.2.5
	B-	s, d	(d)-(s) → (d)		3	* 7	6.2.5
	B-P						6.2.5
	B-	s1, s2, d1	(s1)-(s2) → (d1)		4	* 9	6.2.5
	B-P						6.2.5
Addition und Subtraktion von BCD-Daten (8-stellig)	DB+	s, d	(d+1, d)+(s+1,s) → (d+1, d)		3	* 9	6.2.6
	DB+P						6.2.6
	DB+	s1, s2, d1	((s1)+1, s1)+((s2)+1,s2) → ((d1)+1, d1)		4	* 11	6.2.6
	DB+P						6.2.6
	DB-	s, d	(d+1, d)+(s+1,s) → (d+1, d)		3	* 9	6.2.6
	DB-P						6.2.6
	DB-	s1, s2, d1	((s1)+1, s1)+((s2)+1,s2) → ((d1)+1, d1)		4	* 11	6.2.6
	DB-P						6.2.6

*: Die Anzahl der Programmschritte ist abhängig von den verwendeten Operanden.
Die genaue Anzahl entnehmen Sie dem Abschnitt, in dem die einzelnen Anweisungen beschrieben sind.

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Multiplikation und Division von BCD-Daten (4-stellig)	B¥	s1, s2, d1	(s1)x(s2) → ((d1)+1, d1)		4	* 9	6.2.7
	B×P						6.2.7
	B/	s1, s2, d1	(s1)/(s2) → Quotient (d1), Rest ((d1)+1)		4	* 9	6.2.7
	B/P						6.2.7
Multiplikation und Division von BCD-Daten (8-stellig)	DB×	s1, s2, d1	((s1)+1, s1)x((s2)+1, s2) → ((d1)+3, (d1)+2, (d1)+1, d1)		4	* 11	6.2.8
	DB×P						6.2.8
	DB/	s1, s2, d1	((s1)+1, s1)/((s2)+1, s2) → Quotient ((d1)+1, d1), Rest ((d1)+3, (d1)+2)		4	* 11	6.2.8
	DB/P						6.2.8

*: Die Anzahl der Programmschritte ist abhängig von den verwendeten Operanden.
Die genaue Anzahl entnehmen Sie dem Abschnitt, in dem die einzelnen Anweisungen beschrieben sind.

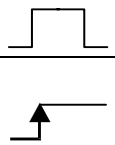
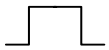
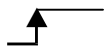
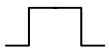
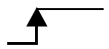
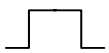
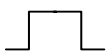
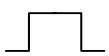
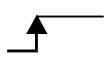
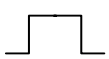
Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Addition und Subraktion von Gleitkomma-zahlen	E+	s, d	(d+1, d)+(s+1, s) → (d+1, d)		3		6.2.9
	E+P						6.2.9
	E+	s1, s2, d1	((s1)+1, s1)+((s2)+1, s2) → ((d1)+1, d1)		4		6.2.9
	E+P						6.2.9
	E-	s, d	(d+1, d)-(s+1, s) → (d+1, d)		3		6.2.9
	E-P						6.2.9
	E-	s1, s2, d1	((s1)+1, s1)-((s2)+1, s2) → ((d1)+1, d1)		4		6.2.9
	E-P						6.2.9
Multiplikation und Division von Gleitkomma-zahlen	Ex	s1, s2, d1	((s1)+1, s1)x((s2)+1, s2) → ((d1)+1, d1)		4		6.2.10
	ExP						6.2.10
	E/	s1, s2, d1	((s1)+1, s1)/((s2)+1, s2) → Quotient ((d1)+1, d1)		4		6.2.10
	E/P						6.2.10
Blockweise Addition und Subtraktion von Binärdaten	BK+	s1, s2, d, n	Addiert wird der n-te 16-Bit-Block aus s1 mit dem n-ten 16-Bit-Block aus s2.		5		6.2.11
	BK+P						6.2.11
	BK-	s1, s2, d, n	Subtrahiert wird der n-te 16-Bit-Block aus s2 von dem n-ten 16-Bit-Block aus s1.		5		6.2.11
	BK-P						6.2.11

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Verknüpfung von Zeichenfolgen	$+	s, d	Die Zeichenfolge in s wird an die Zeichenfolge in d angehängt. Die verknüpfte Zeichenfolge wird in d gespeichert.		3		6.2.12
	$+P						6.2.12
	$+	s1, s2, d1	Die Zeichenfolge in s wird an die Zeichenfolge in d angehängt. Die verknüpfte Zeichenfolge wird in d gespeichert.		4		6.2.12
	$+P						6.2.12
Inkrementieren und von Binärdaten	INC	d	(d)+1→ (d)		2	** 3	6.2.13
	INCP						6.2.13
	DINC	d	(d+1, d)+1 → (d+1, d)		* 2	** 3	6.2.13
	DINCP						6.2.13
Dekrementieren von Binärdaten	DEC	d	(d)-1→ (d)		2	** 3	6.2.14
	DECP						6.2.14
	DDEC	d	(d+1, d)-1 → (d+1, d)		* 2	** 3	6.2.14
	DDECP						6.2.14

*: Die Anzahl der Programmschritte ist abhängig vom Typ der CPU und von den verwendeten Operanden.
- Bei Verwendung einer QnA-CPU oder einer Single-Prozessor-Q-CPU: 2
- Bei Verwendung einer Multi-Prozessor-Q-CPU und
  interner Wortoperanden (außer File-Register ZR): 3
  Konstanten: 3
  Bit-Operanden, deren Adresse ein Vielfaches von 16 ist,
  die die Bit-Blockbezeichnung K8 haben und die nicht durch Index-Vergabe bearbeitet werden: 3
- Bei Verwendung einer Multi-Prozessor-Q-CPU und anderer Operanden als oben aufgeführt: 2

**: Die Anzahl der Programmschritte ist abhängig von den verwendeten Operanden.
Die genaue Anzahl entnehmen Sie dem Abschnitt, in dem die einzelnen Anweisungen beschrieben sind.

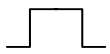
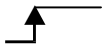
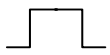
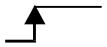
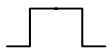
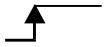
2.4.3 Konvertierungsanweisungen

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
BCD-Konvertierung	BCD	s, d	(s) BCD-Konvertierung (d); BIN (0 bis 9999)		3	* 5	6.3.1
	BCDP						6.3.1
	DBCD	s, d	(s+1, s) BCD-Konvertierung (d+1, d); BIN (0 bis 99999999)		3	* 9	6.3.1
	DBCDP						6.3.1
BIN-Konvertierung	BIN	s, d	(s) BIN-Konvertierung (d); BCD (0 bis 9999)		3	* 5	6.3.2
	BINP						6.3.2
	DBIN	s, d	(s+1, s) BIN-Konvertierung (d+1, d); BCD (0 bis 99999999)		3	* 9	6.3.2
	DBINP						6.3.2
Konvertierung von Binärzahlen in Gleitkommazahlen	FLT	s, d	(s+1, s) Gleitkommazahl-Konvertierung (d); Binärzahl (-32768 bis 32767)		3		6.3.3
	FLTP						6.3.3
	DFLT	s, d	(s+1, s) Gleitkommzahl-Konvertierung (d+1, d); Binärzahl (-2147483648 bis 2147483647)		3		6.3.3
	DFLTP						6.3.3
Konvertierung von Gleitkommazahlen in Binärzahlen	INT	s, d	(s+1, s) BIN-Konvertierung (d); Gleitkommazahl (-32768 bis 32767)		3		6.3.4
	INTP						6.3.4
	DINT	s, d	(s+1, s) Gleitkommzahl-Konvertierung (d+1, d); Binärzahl (-2147483648 bis 2147483647)		3		6.3.4
	DINTP						6.3.4

*: Die Anzahl der Programmschritte ist abhängig von den verwendeten Operanden.
Die genaue Anzahl entnehmen Sie dem Abschnitt, in dem die einzelnen Anweisungen beschrieben sind.

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte Q	Schritte A	Referenz
Konvertierung von 16-Bit-Binärdaten in 32-Bit-Binärdaten	DBL	s, d	Konvertierung (s) → (d+1, d); BIN-(-32768 bis 32767)		3		6.3.5
	DBLP						6.3.5
Konvertierung von 32-Bit-Binärdaten in 16-Bit-Binärdaten	WORD	s, d	Konvertierung (s+1, s) → (d); BIN-(-32768 bis 32767)		3		6.3.6
	WORDP						6.3.6
Konvertierung von 16-/32-Bit-Binärdaten in den Gray-Code	GRY	s, d	Konvertierung in den Gray-Code (s) → (d); Binärzahl (-32768 bis 32767)		3		6.3.7
	GRYP						6.3.7
	DGRY	s, d	Konvertierung in den Gray-Code (s+1, s) → (d+1, d); Binärzahl (-2147483648 bis 2147483647)		3		6.3.7
	DGRYP						6.3.7
Konvertierung von Gray-Code-Daten in 16-/32-Bit-Binärdaten	GBIN	s, d	BIN-Konvertierung (s) → (d); Gray-Code (-32768 bis 32767)		3		6.3.8
	GBINP						6.3.8
	DGBIN	s, d	BIN-Konvertierung (s+1, s) → (d+1, d); Gray-Code (-2147483648 bis 2147483647)		3		6.3.8
	DGBIN						6.3.8
Zweierkomplementbildung von 16-/32-Bit-Binärdaten (Vorzeichenumkehr)	NEG	d	$\overline{(d)}$ → (d); BIN-Daten		2	* 3	6.3.9
	NEGP						6.3.9
	DNEG	d	$\overline{(d+1, d)}$ → (d+1, d); BIN-Daten		2	* 3	6.3.9
	DNEGP						6.3.9

*: Die Anzahl der Programmschritte ist abhängig von den verwendeten Operanden.
Die genaue Anzahl entnehmen Sie dem Abschnitt, in dem die einzelnen Anweisungen beschrieben sind.

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Vorzeichen-umkehr bei Gleitkomma-zahlen	ENEG	d	$\overline{(d+1, d)} \rightarrow (d+1, d)$ Gleitkommazahl		2		6.3.10
	ENEGP						6.3.10
Blockweise Umwandlung von BIN-Daten in BCD-Daten	BKBCD	s, d, n	Konvertiert den n-ten 16-Bit-Block in s und speichert den entsprechenden umgewandelten n-ten 4-stelligen BCD-Datenblock in d.		4		6.3.11
	BKBCDP	s, d, n					6.3.11
Blockweise Umwandlung von BCD-Daten in BIN-Daten	BKBIN	s, d, n	Konvertiert den n-ten 4-stelligen BCD-Datenblock in s und speichert den entsprechenden umgewandelten n-ten 16-Bit-Block in d.		4		6.3.12
	BKBINP	s, d, n					6.3.12

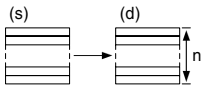
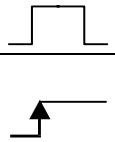
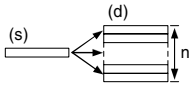
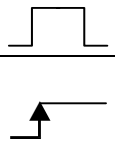
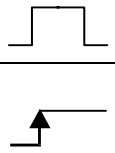
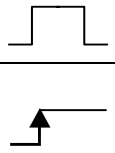
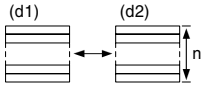
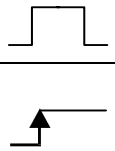
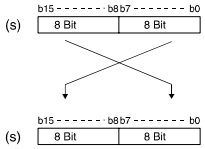
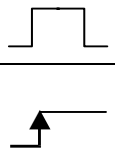
2.4.4 Transferanweisungen

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte Q	Schritte A	Referenz
16-Bit-Datenübertragung	MOV	s, d	(s) → (d)		* 3	*** 5	6.4.1
	MOVP	s, d					
32-Bit-Datenübertragung	DMOV	s, d	(s+1, s) → (d+1, d)		** 3	*** 7	6.4.1
	DMOVP	s, d					
Übertragung von Gleitkommazahlen	EMOV	s, d	(s+1, s) → (d+1, d); Gleitkommazahl		3		6.4.2
	EMOVP	s, d					
Übertragung von Zeichenfolgen	$MOV	s, d	Überträgt die in s gespeicherten Bytes der Zeichenfolge nach d.		3		6.4.3
	$MOVP	s, d					
Inversion von 16-Bit-Binärdaten	CML	s, d	$\overline{(s)}$ → (d)		* 3	*** 5	6.4.4
	CMLP	s, d					
Inversion von 32-Bit-Binärdaten	DCML	s, d	$\overline{(s+1, s)}$ → (d1+1, d1)		** 3	*** 7	6.4.4
	DCMLP	s, d					

*: Die Anzahl der Programmschritte ist abhängig vom Typ der CPU und von den verwendeten Operanden.
- Bei Verwendung einer QnA-CPU oder einer Single-Prozessor-Q-CPU: 3
- Bei Verwendung einer Multi-Prozessor-Q-CPU und
  interner Wortoperanden (außer File-Register ZR): 2
  Konstanten: 2
  Bit-Operanden, deren Adresse ein Vielfaches von 16 ist,
  die die Bit-Blockbezeichnung K4 haben und die nicht durch Index-Vergabe bearbeitet werden: 2
- Bei Verwendung einer Multi-Prozessor-Q-CPU und anderer Operanden als oben aufgeführt: 3

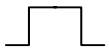
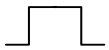
**: Die Anzahl der Programmschritte ist abhängig vom Typ der CPU:
- Bei Verwendung einer Single-Prozessor-Q-CPU: 2
- Bei Verwendung einer QnA-CPU oder einer Multi-Prozessor-Q-CPU: 3

***: Die Anzahl der Programmschritte ist abhängig von den verwendeten Operanden.
Die genaue Anzahl entnehmen Sie dem Abschnitt, in dem die einzelnen Anweisungen beschrieben sind.

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
Übertragung von Binärdaten-blöcken	BMOV	s, n, d			4	* 9	6.4.5
	BMOVP	s, n, d					
Übertragung eines Binärdaten-blocks	FMOV	s, n, d			4	* 9	6.4.6
	FMOVP	s, n, d					
Austausch von 16-Bit-Binärdaten	XCH	d1, d2	(d1) ◄——► (d2)		3	* 5	6.4.7
	XCHP	d1, d2					
Austausch von 32-Bit-Binärdaten	DXCH	d1, d2	((d1)+1, d1) ◄——► ((d2)+1, d2)		3	* 7	6.4.7
	DXCHP	d1, d2					
Blockweiser Austausch von Binärdaten-blöcken	BXCH	n, d1, d2			4		6.4.8
	BXCHP	n, d1, d2					
Austausch der Bytes innerhalb einer Binärzahl	SWAP	s			3		6.4.9
	SWAPP	s					

*: Die Anzahl der Programmschritte ist abhängig von den verwendeten Operanden.
Die genaue Anzahl entnehmen Sie dem Abschnitt, in dem die einzelnen Anweisungen beschrieben sind.

2.4.5 Programmverzweigungsanweisungen

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Sprung-anweisungen	CJ	p	Bedingter Sprung (p = Sprungziel)		2	* 3	6.5.1
	SCJ	p	Bedingter Sprung im nächsten Zyklus (p = Sprungziel)				
	JMP	p	Sprunganweisung (p = Sprungziel)		2	* 3	6.5.1
	GOEND		Sprung zum Programmende		1		6.5.2

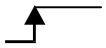
*: Die Anzahl der Programmschritte ist abhängig von den verwendeten Operanden.
Die genaue Anzahl entnehmen Sie dem Abschnitt, in dem die einzelnen Anweisungen beschrieben sind.

2.4.6 Anweisung zum Interrupt-Programmaufruf

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Interrupt verhindert	DI		Verhindert die Abarbeitung eines Interrupt-Programms		1	* 1	6.6.1
Interrupt ermöglicht	EI		Ermöglicht den Aufruf eines Interrupt-Programms		1	* 1	6.6.1
Bit-Schemen der Ausführungs-bedingungen der Interrupt-Programme	IMASK	s	Das in s angegebene Bit-Schema enthält Bits, die den einzelnen Interrupt-Adressen zugeordnet sind.		2	* 1	6.6.1
Rücksprung aus dem Interrupt-Programm ins Haupt-programm	IRET		Ende des Interrupt-Programms		1	* 1	6.6.2

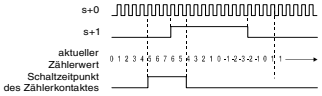
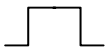
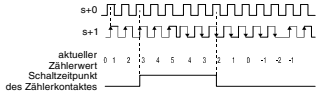
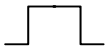
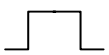
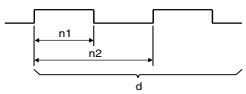
*: Die Anzahl der Programmschritte ist abhängig von den verwendeten Operanden.
Die genaue Anzahl entnehmen Sie dem Abschnitt, in dem die einzelnen Anweisungen beschrieben sind.

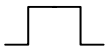
2.4.7 Datenaktualisierungsanweisung

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
E/A-Teilaktualisierung	RFS	s, n	Die RFS-Anweisung aktualisiert die Ein- und Ausgänge des ausgewählten Bereichs für einen Programmzyklus.		3		6.7.1
E/A-Teilaktualisierung	SEG	s, d	Die SEG-Anweisung ermöglicht bei gegebener Eingangsbedingung die Aktualisierung eines bestimmten E/A-Adressenbereiches.			* 9	6.7.2
Aktualisierungs-anweisung für Netzwerk- und Schnittstellen-daten	COM		Ist der SM775 (bei MELSEC Q) nicht gesetzt (0), wird eine Aktualisierung der Netzwerk- und Schnittstellen-Daten (Link Refresh) und eine Gesamtdatenverarbei-tung (END-Verarbeitung) ausgeführt.		1	* 3	6.7.3
Verhindern der Link-Refresh-Ausführung	DI		Die DI-Anweisung verhindert die Ausführung eines Link-Refreshs so lange, bis eine EI-Anweisung in der Programmfolge erscheint.		1		6.7.4
Ermöglichen der Link-Refresh-Ausführung	EI		Die Ausführung eines Link-Refreshs wird nach dem Setzen einer EI-Anweisung möglich.		1		6.7.4

*: Die Anzahl der Programmschritte ist abhängig von den verwendeten Operanden.
Die genaue Anzahl entnehmen Sie dem Abschnitt, in dem die einzelnen Anweisungen beschrieben sind.

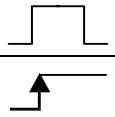
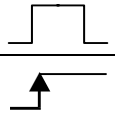
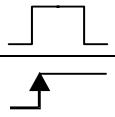
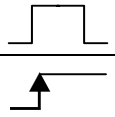
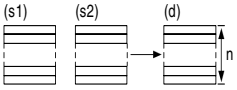
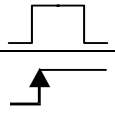
2.4.8 Weitere Anweisungen

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Einphasiger Auf-/ Abwärtszähler	UDCNT1	s, n, d			4		6.8.1
Zweiphasiger Auf-/Abwärts-zähler	UDCNT2	s, n, d			4		6.8.2
Programierbarer Timer	TTMR	d, n	(Zeit, die der TTMR gesetzt ist) x n → (d) n=0:1, n=1:10, n=2:100		3		6.8.3
Sonderfunktions-Timer (langsame Timer)	STMR	s,n, d	Die STMR-Anweisung nutzt die in d+0 bis d+3 angegebenen Ausgänge, um vier unterschiedliche Timerfunktionen zu realisieren: d+0:Ausschaltverzögerter Timerausgang d+1:Timerausgang mit Setzimpuls (Setzen bei abfallender Flanke) d+2:Timerausgang mit Setzimpuls (Setzen bei ansteigender Flanke) d+3:Einschaltverzögerter Timerausgang		3		6.8.4
Sonderfunktions-Timer (schnelle Timer)	STMRH	s,n, d	s.o.		3		6.8.4
Positionier-anweisung für Rotationstische	ROTC	s, n1, n2, d	Mit der ROTC-Anweisung ist es möglich, einen Rotationstisch mit der in n1 angegebenen Anzahl von Sektoren (Teilung) derart zu positionieren, dass ein in s+2 angegebener Sektor an eine in s+1 angegebene Position gefahren wird.		5		6.8.5
Rampensignal	RAMP	n1, n2, n3, d1, d2	Die RAMP-Anweisung erhöht stufenweise den Inhalt in (d1)+0 von dem in n1 angegebenen Startwert bis zu dem in n2 angegebenen Endwert.		6		6.8.6
Impulszähler	SPD	s, n, d	Die SPD-Anweisung zählt die Impulse des in s angegebenen Eingangs für die in n angegebene Dauer und speichert das Ergebnis der Messung in d.		4		6.8.7
Impulsausgang mit einstellbarer Anzahl von Impulsen	PLSY	s1, s2, d	Die PLSY-Anweisung gibt Impulse mit der in s1 angegebenen Frequenz und der in s2 angegebenen Anzahl an den in d angegebenen Ausgang aus.		4		6.8.8
Puls-Weiten-Modulation	PWM	n1, n2, d			4		6.8.9

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Bildung einer Eingabe-Matrix	MTR	s, n , d1, d2	Die MTR-Anweisung liest, beginnend bei s 16 Informationen ein. Die Anzahl der Wiederholungen dieses Vorgangs (Reihen) ist in n angegeben. Die jeweiligen Zustände der eingelesenen Informationen werden beginnend in d2 gespeichert.		5		6.8.10

2.5 Applikationsanweisungen Teil II

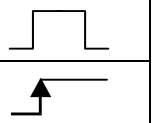
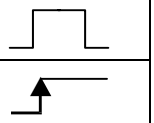
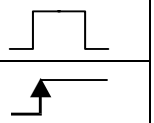
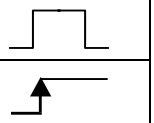
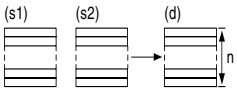
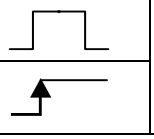
2.5.1 Logikanweisungen

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Logisches Produkt	WAND	s, d	(d) ∧ (s) ⟶ (d)		3	*** 5	7.1.1
	WANDP						
	WAND	s1, s2, d1	(s1) ∧ (s2) ⟶ (d1)		4	*** 7	7.1.1
	WANDP						
	DAND	s, d	(d+1,d) ∧ (s+1, s) ⟶ (d+1, d)		* 4	*** 9	7.1.1
	DANDP						
	DAND	s1, s2, d	((s1)+1, s1) ∧ ((s2)+1, s2) ⟶ (d+1, d)		** 4		7.1.1
	DANDP						
	BKAND	s1, s2, n, d			5		7.1.2
	BKANDP						

*: Die Anzahl der Programmschritte ist abhängig vom Typ der CPU und von den verwendeten Operanden.
- Bei Verwendung einer QnA-CPU: 4
- Bei Verwendung einer Single-Prozessor-Q-CPU: 3
- Bei Verwendung einer Multi-Prozessor-Q-CPU und interner Wortoperanden (außer File-Register ZR): 6
  Konstanten: 6
  Bit-Operanden, deren Adresse ein Vielfaches von 16 ist, die die Bit-Blockbezeichnung K8 haben und die nicht durch Index-Vergabe bearbeitet werden: 6
- Bei Verwendung einer Multi-Prozessor-Q-CPU und anderer Operanden als oben aufgeführt: 4

**: Die Anzahl der Programmschritte ist abhängig vom Typ der CPU und von den verwendeten Operanden.
- Bei Verwendung einer QnA-CPU: 4
- Bei Verwendung einer System Q-CPU und interner Wortoperanden (außer File-Register ZR): 6
  Konstanten: 6
  Bit-Operanden, deren Adresse ein Vielfaches von 16 ist, die die Bit-Blockbezeichnung K8 haben und die nicht durch Index-Vergabe bearbeitet werden: 6
- Bei Verwendung einer System Q-CPU und anderer Operanden als oben aufgeführt: 4

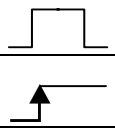
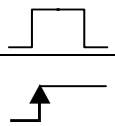
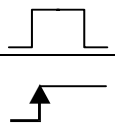
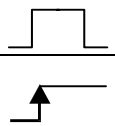
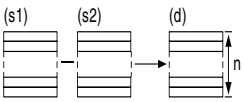
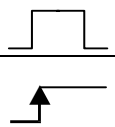
***: Die Anzahl der Programmschritte ist abhängig von den verwendeten Operanden.
Die genaue Anzahl entnehmen Sie dem Abschnitt, in dem die einzelnen Anweisungen beschrieben sind.

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Logische Addition	WOR	s, d	(d) ∨ (s) ⟶ (d)		3	*** 5	7.1.3
	WORP						
	WOR	s1, s2, d1	(s1) ∨ (s2) ⟶ (d1)		4	*** 7	7.1.3
	WORP						
	DOR	s, d	(d+1, d) ∨ (s+1, s) ⟶ (d+1, d)		* 4	*** 9	7.1.3
	DORP						
	DOR	s1, s2, d	((s1)+1,s1) ∨ ((s2)+1, s2) ⟶ (d+1, d)		** 4		7.1.3
	DORP						
	BKOR	s1, s2, n, d			5		7.1.4
	BKORP						

*: Die Anzahl der Programmschritte ist abhängig vom Typ der CPU und von den verwendeten Operanden.
- Bei Verwendung einer QnA-CPU: 4
- Bei Verwendung einer Single-Prozessor-Q-CPU: 3
- Bei Verwendung einer Multi-Prozessor-Q-CPU und
  interner Wortoperanden (außer File-Register ZR): 6
  Konstanten: 6
  Bit-Operanden, deren Adresse ein Vielfaches von 16 ist,
  die die Bit-Blockbezeichnung K8 haben und die nicht durch Index-Vergabe bearbeitet werden: 6
- Bei Verwendung einer Multi-Prozessor-Q-CPU und anderer Operanden als oben aufgeführt: 4

**: Die Anzahl der Programmschritte ist abhängig vom Typ der CPU und von den verwendeten Operanden.
- Bei Verwendung einer QnA-CPU: 4
- Bei Verwendung einer System Q-CPU und
  interner Wortoperanden (außer File-Register ZR): 6
  Konstanten: 6
  Bit-Operanden, deren Adresse ein Vielfaches von 16 ist,
  die die Bit-Blockbezeichnung K8 haben und die nicht durch Index-Vergabe bearbeitet werden: 6
- Bei Verwendung einer System Q-CPU und anderer Operanden als oben aufgeführt: 4

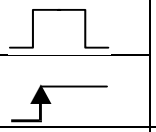
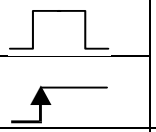
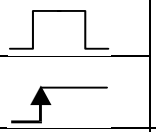
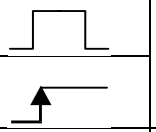
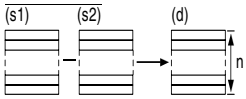
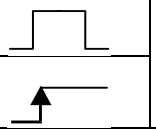
***: Die Anzahl der Programmschritte ist abhängig von den verwendeten Operanden.
Die genaue Anzahl entnehmen Sie dem Abschnitt, in dem die einzelnen Anweisungen beschrieben sind.

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Exclusiv-ODER-Logik	WXOR	s, d	(d) ⩛ (s) ⟶ (d)		3	*** 5	7.1.5
	WXORP						
	WXOR	s1, s2, d1	(s1) ⩛ (s2) ⟶ (d1)		4	*** 7	7.1.5
	WXORP						
	DXOR	s, d	(d+1, d) ⩛ (s+1, s) ⟶ (d+1, d)		* 4	*** 9	7.1.5
	DXORP						
	DXOR	s1, s2, d	((s1)+1, s1) ⩛ ((s2)+1, s2) ⟶ (d+1, d)		** 4		7.1.5
	DXORP						
	BKXOR	s1, s2, n, d			5		7.1.6
	BKXORP						

*: Die Anzahl der Programmschritte ist abhängig vom Typ der CPU und von den verwendeten Operanden.
- Bei Verwendung einer QnA-CPU: 4
- Bei Verwendung einer Single-Prozessor-Q-CPU: 3
- Bei Verwendung einer Multi-Prozessor-Q-CPU und
  interner Wortoperanden (außer File-Register ZR): 6
  Konstanten: 6
  Bit-Operanden, deren Adresse ein Vielfaches von 16 ist,
  die die Bit-Blockbezeichnung K8 haben und die nicht durch Index-Vergabe bearbeitet werden: 6
- Bei Verwendung einer Multi-Prozessor-Q-CPU und anderer Operanden als oben aufgeführt: 4

**: Die Anzahl der Programmschritte ist abhängig vom Typ der CPU und von den verwendeten Operanden.
- Bei Verwendung einer QnA-CPU: 4
- Bei Verwendung einer Q-CPU und
  interner Wortoperanden (außer File-Register ZR): 6
  Konstanten: 6
  Bit-Operanden, deren Adresse ein Vielfaches von 16 ist,
  die die Bit-Blockbezeichnung K8 haben und die nicht durch Index-Vergabe bearbeitet werden: 6
- Bei Verwendung einer Q-CPU und anderer Operanden als oben aufgeführt: 4

***: Die Anzahl der Programmschritte ist abhängig von den verwendeten Operanden.
Die genaue Anzahl entnehmen Sie dem Abschnitt, in dem die einzelnen Anweisungen beschrieben sind.

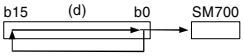
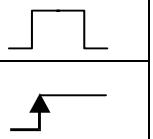
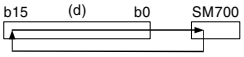
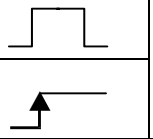
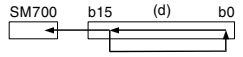
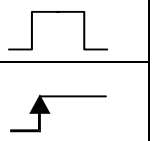
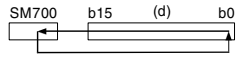
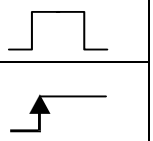
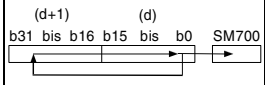
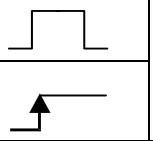
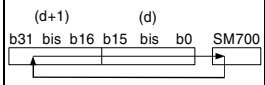
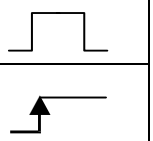
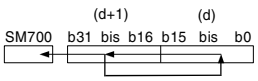
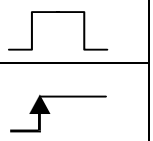
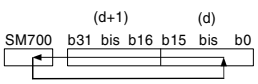
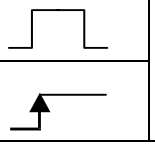
Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Exklusiv-NOR-Logik	WNXR	s, d	(d) ∨ (s) ⟶ (d)		3	*** 5	7.1.7
	WNXRP						
	WNXR	s1, s2, d1	$\overline{(s1)} - \overline{(s2)}$ ⇳ (d1)		4	*** 7	7.1.7
	WNXRP						
	DNXR	s, d	$\overline{(d+1, d) - (s+1, s)}$ ⇳ (d+1, d)		* 3	*** 9	7.1.7
	DNXRP						
	DNXR	s1, s2, d	$\overline{((s1)+1, s1) - ((s2)+1, s2)}$ ⇳ (d+1, d)		** 4		7.1.7
	DNXRP						
	BKXNR	s1, s2, n, d			5		7.1.8
	BKXNRP						

*: Die Anzahl der Programmschritte ist abhängig vom Typ der CPU und von den verwendeten Operanden.
- Bei Verwendung einer QnA-CPU: 4
- Bei Verwendung einer Single-Prozessor-Q-CPU: 3
- Bei Verwendung einer Multi-Prozessor-Q-CPU und
  interner Wortoperanden (außer File-Register ZR): 6
  Konstanten: 6
  Bit-Operanden, deren Adresse ein Vielfaches von 16 ist,
  die die Bit-Blockbezeichnung K8 haben und die nicht durch Index-Vergabe bearbeitet werden: 6
- Bei Verwendung einer Multi-Prozessor-Q-CPU und anderer Operanden als oben aufgeführt: 4

**: Die Anzahl der Programmschritte ist abhängig vom Typ der CPU und von den verwendeten Operanden.
- Bei Verwendung einer QnA-CPU: 4
- Bei Verwendung einer Q-CPU und
  interner Wortoperanden (außer File-Register ZR): 6
  Konstanten: 6
  Bit-Operanden, deren Adresse ein Vielfaches von 16 ist,
  die die Bit-Blockbezeichnung K8 haben und die nicht durch Index-Vergabe bearbeitet werden: 6
- Bei Verwendung einer Q-CPU und anderer Operanden als oben aufgeführt: 4

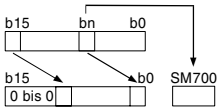
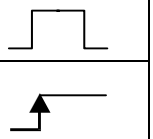
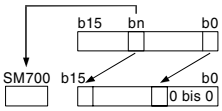
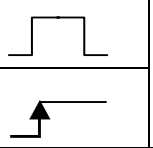
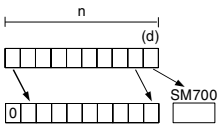
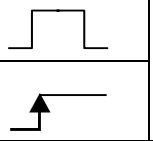
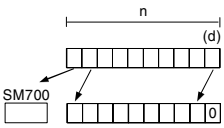
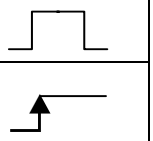
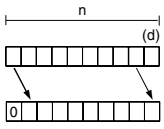
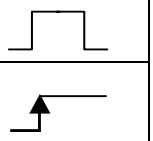
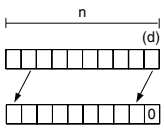
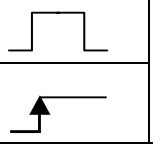
***: Die Anzahl der Programmschritte ist abhängig von den verwendeten Operanden.
Die genaue Anzahl entnehmen Sie dem Abschnitt, in dem die einzelnen Anweisungen beschrieben sind.

2.5.2 Rotationsanweisungen

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte Q	Schritte A	Referenz
Datenrotation rechts (16 Bit)	ROR	n, d	 Rotiert n Bit nach rechts		3	* 3	7.2.1
	RORP						
	RCR	n, d	 Rotiert n Bit nach rechts		3	* 3	7.2.1
	RCRP						
Datenrotation links (16 Bit)	ROL	n, d	 Rotiert n Bit nach links		3	* 3	7.2.2
	ROLP						
	RCL	n, d	 Rotiert n Bit nach links		3	* 3	7.2.2
	RCLP						
Datenrotation rechts (32 Bit)	DROR	n, d	 Rotiert n Bit nach rechts		3	* 3	7.2.3
	DRORP						
	DRCR	n, d	 Rotiert n Bit nach rechts		3	* 3	7.2.3
	DRCRP						
Datenrotation links (32 Bit)	DROL	n, d	 Rotiert n Bit nach links		3	* 3	7.2.4
	DROLP						
	DRCL	n, d	 Rotiert n Bit nach links		3	* 3	7.2.4
	DRCLP						

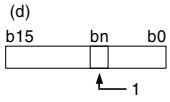
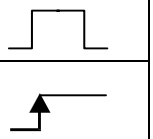
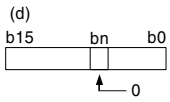
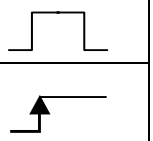
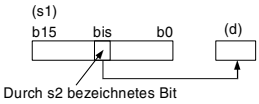
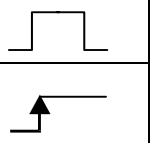
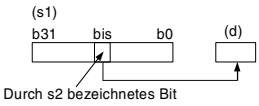
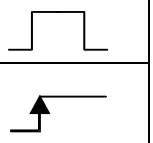
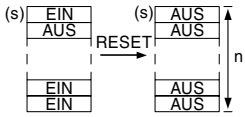
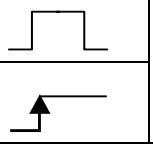
*: Die Anzahl der Programmschritte ist abhängig von den verwendeten Operanden.
Die genaue Anzahl entnehmen Sie dem Abschnitt, in dem die einzelnen Anweisungen beschrieben sind.

2.5.3 Verschiebeanweisungen

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte Q	Schritte A	Referenz
Verschiebung eines 16-Bit-Datenwortes um n Bit	SFR	n, d			3	* 3	7.3.1
	SFRP						
	SFL	n, d			3	* 3	7.3.1
	SFLP						
Verschiebung von n Bit-Operanden um 1 Bit	BSFR	n, d			3	* 7	7.3.2
	BSFRP						
	BSFL	n, d			3	* 7	7.3.2
	BSFLP						
Verschiebung von n Wort-Operanden um 1 Adresse	DSFR	n, d			3	* 7	7.3.3
	DSFRP						
	DSFL	n, d			3	* 7	7.3.3
	DSFLP						

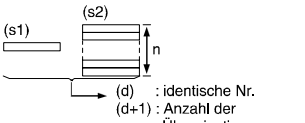
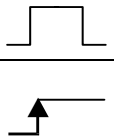
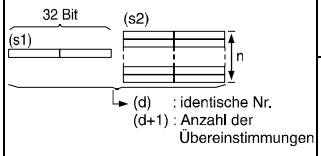
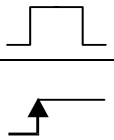
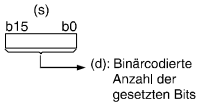
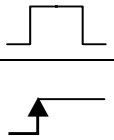
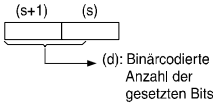
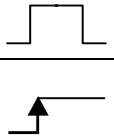
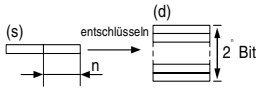
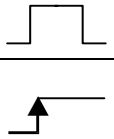
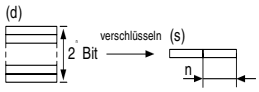
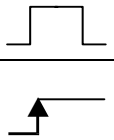
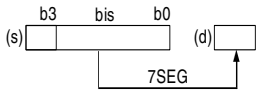
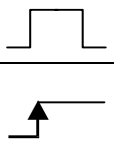
*: Die Anzahl der Programmschritte ist abhängig von den verwendeten Operanden.
Die genaue Anzahl entnehmen Sie dem Abschnitt, in dem die einzelnen Anweisungen beschrieben sind.

2.5.4 Bit-Verarbeitungsanweisungen

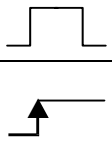
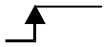
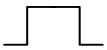
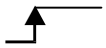
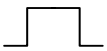
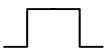
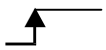
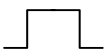
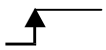
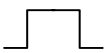
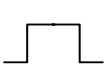
Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte Q	Schritte A	Referenz
Setzen/ Rücksetzen einzelner Bits	BSET	n, d			3	* 3	7.4.1
	BSETP						
	BRST	n, d			3	* 7	7.4.1
	BRSTP						
Zustandsabfrage einzelner Bits in 16-/32-Bit-Datenwörtern	TEST	s1, s2, d			4		7.4.2
	TESTP						
	DTEST	s1, s2, d			4		7.4.2
	DTESTP						
Zurücksetzen von Bitbereichen	BKRST	s, n			3		7.4.3
	BKRSTP						

*: Die Anzahl der Programmschritte ist abhängig von den verwendeten Operanden.
Die genaue Anzahl entnehmen Sie dem Abschnitt, in dem die einzelnen Anweisungen beschrieben sind.

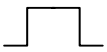
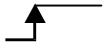
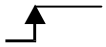
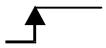
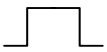
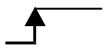
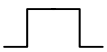
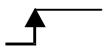
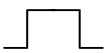
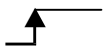
2.5.5 Datenverarbeitungsanweisungen

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte Q	Schritte A	Referenz
Suchen von 16-Bit-Daten	SER	s1, s2, n (A) s1, s2, n, d (Q)			5	* 9	7.5.1
	SERP						
	DSER	s1, s2, n (A) s1, s2, n, d (Q)			5	* 9	7.5.1
	DSERP						
Datenbit-Kontrolle (16-/32-Bit)	SUM	s (A) s, d (Q)			3	* 3	7.5.2
	SUMP						
	DSUM	s (A) s, d (Q)			3	* 3	7.5.2
	DSUMP						
Daten decodieren	DECO	s, d, n	Decodierung von 8 nach 256 Bit 		4	* 9	7.5.3
	DECOP						
Daten codieren	ENCO	s, d, n	Codierung von 256 nach 8 Bit 		4	* 9	7.5.4
	ENCOP						
7-Segment-Decodierung	SEG	s, d			3	7	7.5.5
	SEGP						

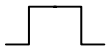
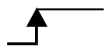
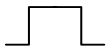
*: Die Anzahl der Programmschritte ist abhängig von den verwendeten Operanden.
Die genaue Anzahl entnehmen Sie dem Abschnitt, in dem die einzelnen Anweisungen beschrieben sind.

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
16-Bit-Datenworte auftrennen, zusammenführen	DIS	s, n, d	Die 16-Bit-Datenwerte werden in Gruppen zu 4 Bits aufgetrennt. Die aufzutrennenden Datenwerte werden in s, die Anzahl der 4-Bit-Gruppen in n und die erste Zieladresse in d festgelegt.		4	* 9	7.5.6
	DISP						
	UNI	s, n, d	Die jeweils 4 niedrigstwertigen Bits von bis zu vier 16-Bit-Datenwerten werden aufgetrennt und zusammen mit den Zuständen in einem 16-Bit-Datenwert gespeichert.		4	* 9	7.5.7
	UNIP						
	NDIS	s1, s2, d	Die Daten ab s1 werden in Bit-Gruppen mit der in s2 angegebenen Größe aufgetrennt. Die aufgetrennten Bit-Gruppen werden einzeln ab d gespeichert.		4		7.5.8
	NDISP						
	NUNI	s1, s2, d	Die ab s2 angegebenen Größen der Bit-Gruppen werden aus s1 herausgetrennt und zu einem Datenwert zusammengeführt. Die Bit-Gruppen werden aufeinanderfolgend ab d gespeichert.				
	NUNIP						
	WTOB	s, n, d	In der Anweisung werden die aufzutrennenden Datenwerte in s, die Anzahl der Byte-Gruppen in n und die erste Zieladresse in d festgelegt.		4		7.5.9
	WTOBP						
	BTOW	s, n, d	Die Startadresse der zusammenzuführenden Datenwerte wird in s, die Anzahl der Byte-Gruppen in Folge in n und die Zieladresse in d festgelegt.				
	BTOWP						
Suchen von Maximalwerten in 16-/32-Bit-Daten	MAX	s, n, d	Sucht in 16-Bit-Datenblöcken nach dem größten Wert. Die Anzahl der zu durchsuchenden Datenblöcke wird in n angegeben. Der größte in s bis s+(n-1) gefundene Wert wird in d gespeichert.		4		7.5.10
	MAXP						
	DMAX	s, n, d	Sucht in 32-Bit-Datenblöcken nach dem größten Wert. Die Anzahl der zu durchsuchenden Datenblöcke wird in n angegeben. Der größte in s bis s+(n-1) gefundene Wert wird in d gespeichert.				
	DMAXP						

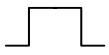
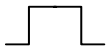
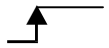
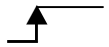
*: Die Anzahl der Programmschritte ist abhängig von den verwendeten Operanden.
Die genaue Anzahl entnehmen Sie dem Abschnitt, in dem die einzelnen Anweisungen beschrieben sind.

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Suchen von Minimalwerten in 16-/32-Bit-Daten	MIN	s, n, d	Sucht in 16-Bit-Datenblöcken nach dem kleinsten Wert. Die Anzahl der zu durchsuchenden Datenblöcke wird in n angegeben. Der kleinste in s bis s+(n-1) gefundene Wert wird in d gespeichert.		4		7.5.11
	MINP						
	DMIN	s, n, d	Sucht in 32-Bit-Datenblöcken nach dem kleinsten Wert. Die Anzahl der zu durchsuchenden Datenblöcke wird in n angegeben. Der kleinste in s bis s+(n-1) gefundene Wert wird in d gespeichert.				
	DMINP						
Sortieren von 16-/32-Bit-Daten	SORT	s1, n, s2, d1, d2	Sortiert die mit n angegebene Anzahl der in s1 angegebenen 16-Bit-Daten in steigender oder fallender Reihenfolge.		6		7.5.12
	SORTP						
	DSORT		Sortiert die in n angegebene Anzahl der in s1 angegebenen 32-Bit-Daten in steigender oder fallender Reihenfolge.				
	DSORTP						
Summenbildung von 16-/32-Bit-Binärdatenblöcken	WSUM	s, n, d	Addiert die n 16-Bit-Binärwerte ab s und speichert sie in d und d+1.		4		7.5.13
	WSUMP						
	DWSUM	s, n, d	Addiert die n 32-Bit-Binärwerte ab s und s+1 und speichert sie in d bis d+3.		4		7.5.14
	DWSUMP						

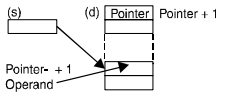
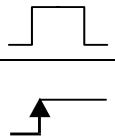
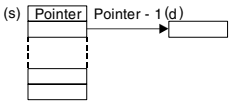
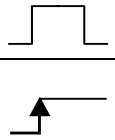
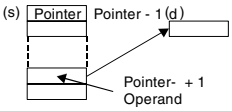
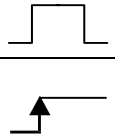
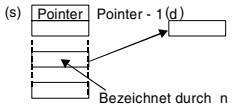
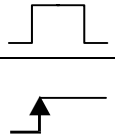
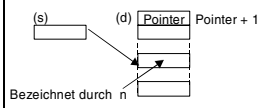
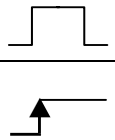
2.5.6 Strukturierte Programmanweisungen

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte Q	Schritte A	Referenz
Wiederholungsanweisungen	FOR	n	Die FOR-NEXT-Anweisungskombination ermöglicht es, einzelne Programmsequenzen ohne Vorgabe einer Eingangsbedingung zu wiederholen. Die Anzahl der Wiederholungen wird in n festgelegt.		2	* 3	7.6.1
	NEXT				1	* 1	
	BREAK	p, d	Die BREAK-Anweisung unterbricht die FOR-NEXT-Schleife während der Ausführung und springt zu dem in p angegebenen Pointer/ Label.		3		7.6.2
	BREAKP						
Unterprogrammaufruf	CALL	p	Der Aufruf einer Unterprogrammroutine mit Hilfe einer CALL-Anweisung erfolgt durch Angabe des Labels der Unterprogrammroutine.		2	* 3	7.6.3
	CALLP						
	RET		Mit Hilfe einer RET-Anweisung wird das Ende einer Unterprogrammroutine gekennzeichnet.		1	* 1	7.6.4
	FCALL	p	Mit dem Rücksetzen der Ausführungsbedingung der FCALL-Anweisung werden die Kontakte und Spulen der in p (Pointer/Label) angegebenen Unterprogrammroutine in den Zustand versetzt, als ob die entsprechende Ausführungsbedingung nicht gesetzt ist.		2		7.6.5
	FCALLP						
Unterprogrammaufruf zwischen Programmdateien	ECALL	Dateiname, p	Der Aufruf einer Unterprogrammroutine in einer Programmdatei mit Hilfe einer ECALL-Anweisung erfolgt durch Angabe des Dateinamens der Programmdatei und durch Angabe des Labels der Unterprogrammroutine.		3		7.6.6
	ECALLP						

*: Die Anzahl der Programmschritte ist abhängig von den verwendeten Operanden.
Die genaue Anzahl entnehmen Sie dem Abschnitt, in dem die einzelnen Anweisungen beschrieben sind.

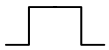
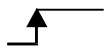
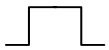
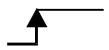
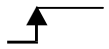
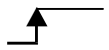
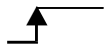
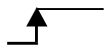
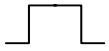
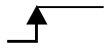
Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Unterprogrammaufruf zwischen Programmdateien	EFCALL	Dateiname, p	Mit dem Rücksetzen der Ausführungsbedingung der EFCALL-Anweisung werden die Kontakte und Spulen der in p (Pointer/Label) angegebenen Unterprogrammroutine in der mit dem Dateinamen angegebenen Programmdatei in den Zustand versetzt, als ob die entsprechende Ausführungsbedingung der jeweiligen Anweisung, die einen Kontakt oder eine Spule anspricht, nicht gesetzt ist.		3		7.6.7
	EFCALLP						
Umschaltung zwischen MAIN- und SUB-Programmbereich	CHG		Die Anweisung schaltet bei gegebener Eingangsbedingung zwischen dem MAIN- und dem SUB-Programmbereich um.			1	7.6.8
Mikrocomputer-Programmaufruf	SUB	n	Bei gegebener Eingangsbedingung ruft die SUB-Anweisung das an der Adresse "n" befindliche Mikrocomputer-Programm auf.		3		7.6.9
	SUBP						
Indizierte Adressierung eines gesamten Programmteils	IX	s	Die IX- und IXEND-Anweisungen nehmen eine indizierte Adressierung der Operandenadressen in dem Programmteil vor, der zwischen der IX- und IXEND-Anweisung programmiert wird.		2		7.6.10
	IXEND				1		
Speicherung indizierter Operandenadressen in einer Index-Liste	IXDEV		Die IXDEV- und IXSET-Anweisung lesen die Adressen der im Offset-Bereich befindlichen Operanden aus und schreiben diese Offset-Werte als Index-Liste in den in d angegebenen Operanden.		1		7.6.11
	IXSET	p, d			3		

2.5.7 Verarbeitungsanweisung für Datenlisten

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Daten in eine Datenliste schreiben	FIFW	s, d			3	* 7	7.7.1
	FIFWP						
Lesen zuerst eingegebener Daten aus der Datenliste	FIFR	s, d			3	* 7	7.7.2
	FIFRP						
Lesen zuletzt eingegebener Daten aus der Datenliste	FPOP	s, d			3		7.7.3
	FPOPP						
Löschen bestimmter Datenblöcke in der Datenliste	FDEL	s, n, d			4		7.7.4
	FDELP						
Einfügen bestimmter Datenblöcke in die Datenliste	FINS						
	FINSP						

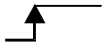
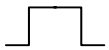
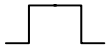
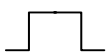
*: Die Anzahl der Programmschritte ist abhängig von den verwendeten Operanden.
Die genaue Anzahl entnehmen Sie dem Abschnitt, in dem die einzelnen Anweisungen beschrieben sind.

2.5.8 Anweisungen für den Pufferspeicherzugriff

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte Q	Schritte A	Referenz
Daten aus einem Sondermodul lesen	FROM	n1, n2, n3, d	Liest 1-Wort-Daten (16 Bits) aus dem Pufferspeicherbereich eines Sondermoduls.		5	* 9	7.8.1
	FROMP						
	DFRO		Liest 2-Wort-Daten (32 Bits) aus dem Pufferspeicher eines Sondermoduls.				
	DFROP						
Daten in ein Sondermodul schreiben	TO	s, n1, n2, n3	Schreibt 1-Wort-Daten (16 Bits) aus dem Speicher der CPU in den Pufferspeicher eines Sondermoduls.		5	* 9	7.8.2
	TOP						
	DTO		Schreibt 2-Wort-Daten (32 Bits) in den Pufferspeicher eines Sondermoduls.				
	DTOP						
Daten aus einer Remote-Station lesen	FROM, PRC	n1, n2, n3, d (FROM(P)/ DFRO(P)) s, d PRC	Liest 1-Wort-Daten (16 Bit) aus einer Remote-Station			* 7/9	7.8.3
	FROMP, PRC						
	DFRO, PRC		Liest 2-Wort-Daten (32 Bit) aus einer Remote-Station				
	DFROP, PRC						
Daten in eine Remote-Station schreiben	TO, PRC	s, n1, n2, n3 (TO(P)/DTO(P)) s, d (PRC)	Schreibt 1-Wort-Daten (16 Bit) in eine Remote-Station			* 7/9	7.8.4
	TOP, PRC						
	DTO, PRC		Schreibt 2-Wort-Daten (32 Bit) in eine Remote-Station				
	DTOP, PRC						

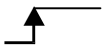
*: Die Anzahl der Programmschritte ist abhängig von den verwendeten Operanden.
Die genaue Anzahl entnehmen Sie dem Abschnitt, in dem die einzelnen Anweisungen beschrieben sind.

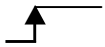
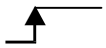
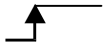
2.5.9 Display-Anweisungen

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
ASCII-Ausgabe	PR	s, d	SM701 (Q-Serie/ System Q) gesetzt (1): Ausgabe einer ASCII-Zeichenfolge mit 16 Zeichen an ein Ausgangsmodul. Die Zeichenfolge wird, aufgeteilt in 2 mal 8 Zeichen, aus dem Adressenbereich s gelesen und an die in d vorgegebenen Ausgänge ausgegeben. SM701 (Q-Serie/ System Q) nicht gesetzt (0): Ausgabe einer ASCII-Zeichenfolge bis zu dem Zeichencode "00_H" aus dem Adressenbereich s als Hexadezimalcode an die in d vorgegebenen Ausgänge.		3	* 7	7.9.1
	PRC	s, d	Gibt einen Kommentar (in ASCII-Code) an ein Ausgangsmodul aus. Ist SM701 (Q-Serie/ System Q) gesetzt (1), werden 16 Zeichen ausgegeben. Ist SM701 nicht gesetzt (0), werden 32 Zeichen ausgegeben.		3	* 7	7.9.2
Anzeige von ASCII-Zeichen und Kommentaren	LED	s	Die LED-Anweisung ruft ASCII-Daten (16 Zeichen) aus 8 Adressen eines vorgegebenen Adressenbereichs auf und bringt sie auf dem LED-Display einer dafür geeigneten CPU zur Anzeige.		2	* 3	7.9.3
	LEDC	s (Q)	Die LEDC-Anweisung ruft Kommentare (16 Zeichen) aus einem vorgegebenen Adressenbereich in s auf und bringt sie auf dem LED-Display einer dafür geeigneten CPU zur Anzeige.		2	* 3	7.9.4
	LEDA	n	Anzeige von ASCII-Zeichen auf dem LED-Display der CPU			* 13	7.9.5
	LEDB						
Anzeige löschen	LEDR		Rücksetzen von Fehlermerkern und Displayanzeigen		1	* 1	7.9.6

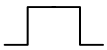
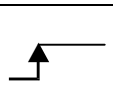
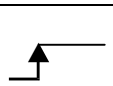
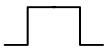
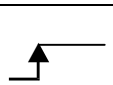
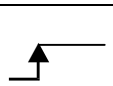
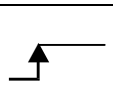
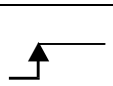
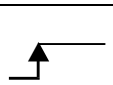
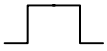
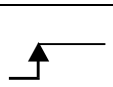
*: Die Anzahl der Programmschritte ist abhängig von den verwendeten Operanden.
Die genaue Anzahl entnehmen Sie dem Abschnitt, in dem die einzelnen Anweisungen beschrieben sind.

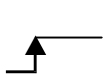
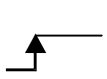
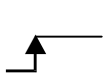
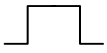
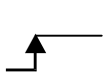
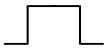
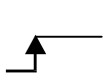
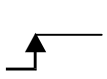
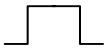
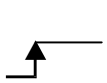
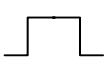
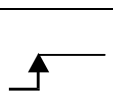
2.5.10 Fehlererkennung und Fehlerbeseitigung

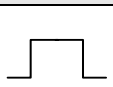
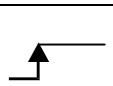
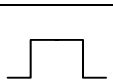
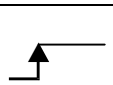
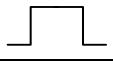
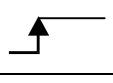
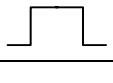
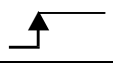
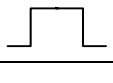
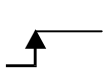
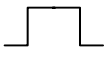
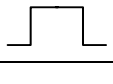
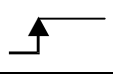
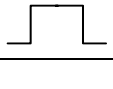
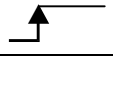
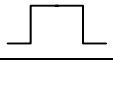
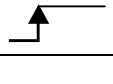
Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Fehlerkontrolle	CHKST		Startet die Ausführung der CHK-Anweisung. Ist die Ausführungsbedingung der CHKST-Anweisung nicht gesetzt (0), wird der nächste auf die CHK-Anweisung folgende Programmschritt ausgeführt.		1		7.10.1
	CHK (Q)		Ermöglicht eine Fehlerkontrolle in einer Kontaktanordnung mit Grenzschaltern. Sobald ein Fehler in dieser Anordnung vorliegt, wird der Operand d1 gesetzt und ein entsprechender Fehlercode im Operand d2 gespeichert.				
	CHK (A)	d1, d2	Ermöglicht eine Fehlerkontrolle in einer Kontaktanordnung mit Grenzschaltern. Sobald ein Fehler in dieser Anordnung vorliegt, wird der Operand d1 gesetzt und ein entsprechender Fehlercode im Operand d2 gespeichert.			5	7.10.2
	CHKCIR		Erzeugt Prüfnetzwerke für die CHK-Anweisung und startet den Programmbereich mit den erzeugten Prüfnetzwerken.		1		7.10.3
	CHKEND		End-Anweisung für den Programmbereich mit den erzeugten Prüfnetzwerken.				
Status Latch setzen/ rücksetzen	SLT		Führt die Zwischenspeicherung definierter Operanden-daten aus, die im Status-Latch-Speicher gesichert werden. Dort können sie überprüft und angezeigt werden.		1	1	7.10.4
	SLTR		Die im Status-Latch-Bereich zwischengespeicherten Daten werden gelöscht, und die SLT-Anweisung wird zurückgesetzt.				
Abtastüberwachung (Sampling Trace) setzen/ rücksetzen	STRA		Sampling Trace setzen		1	1	7.10.5
	STRAR		Sampling Trace rücksetzen				

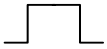
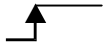
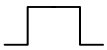
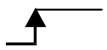
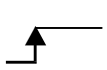
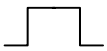
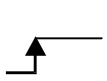
Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Programm-überwachung (ProgramTrace) ausführen/ setzen/ rücksetzen	PTRA		Program Trace setzen		1		7.10.6
	PTRAR		Program Trace rücksetzen				
	PTRAEXE		Program Trace ausführen				
	PTRAEXEP						

2.5.11 Verarbeitungsanweisungen für Zeichenfolgen

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Konvertierung von 16-/32-Bit-Binärdaten in Dezimalzahlen im ASCII-Code	BINDA	s, d	Konvertiert die in s angegebene 16-Bit-Binärzahl in eine 5-stellige Dezimalzahl im ASCII-Code und speichert sie in d.		3		7.11.1
	BINDAP						
	DBINDA		Konvertiert die in s angegebenen 32-Bit-Binärdaten in eine 10-stellige Dezimalzahl im ASCII-Code und speichert sie in d.				
	DBINDAP						
Konvertierung von 16-/32-Bit-Binärdaten in Hexadezimalzahlen im ASCII-Code	BINHA	s, d	Konvertiert die in s angegebenen 16-Bit-Binärdaten in eine 4-stellige Hexadezimalzahl im ASCII-Code und speichert sie in d.		3		7.11.2
	BINHAP						
	DBINHA		Konvertiert die in s angegebenen 32-Bit-Binärdaten in eine 8-stellige Hexadezimalzahl im ASCII-Code und speichert sie in d.				
	DBINHAP						
Konvertierung von 4-/8-stelligen BCD-Daten in den ASCII-Code	BCDDA	s, d	Konvertiert die in s angegebenen 4-stelligen BCD-Daten in das ASCII-Format und speichert sie in d.		3		7.11.3
	BCDDAP						
	DBCDDA		Konvertiert die in s angegebenen 8-stelligen BCD-Daten in das ASCII-Format und speichert sie in d.				
	DBCDDAP						
Konvertierung dezimaler ASCII-Daten in 16-/32-Bit-Binärdaten	DABIN	s, d	Konvertiert die in s angegebenen 5-stelligen dezimalen ASCII-Daten in das Format BIN-16-Bit und speichert sie in d.		3		7.11.4
	DABINP						
	DDABIN		Konvertiert die in s angegebenen 10-stelligen dezimalen ASCII-Daten in das Format BIN-32-Bit und speichert sie in d.				
	DDABINP						

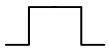
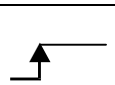
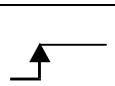
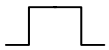
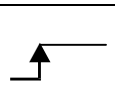
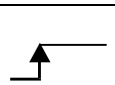
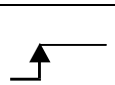
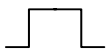
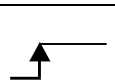
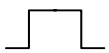
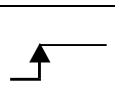
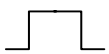
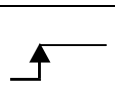
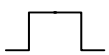
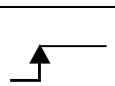
Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Konvertierung hexadezimaler ASCII-Daten in 16-/32-Bit-Binärdaten	HABIN	s, d	Konvertiert die 4-stelligen hexadezimalen ASCII-Daten in s in das BIN-16-Bit-Datenformat und speichert sie in d.		3		7.11.5
	HABINP						
	DHABIN		Konvertiert die 8-stelligen hexadezimalen ASCII-Daten in s in das BIN-32-Bit-Datenformat und speichert sie in d.				
	DHABINP						
Konvertierung dezimaler ASCII-Daten in 4-/8-stellige BCD-Daten	DABCD	s, d	Konvertiert die dezimalen ASCII-Daten in s in das 4-stellige BCD-Datenformat und speichert die Daten in d.		3		7.11.6
	DABCDP						
	DDABCD		Konvertiert die dezimalen ASCII-Daten in s in das 8-stellige BCD-Datenformat und speichert die Daten in d.				
	DDABCDP						
Auslesen von Kommentardaten	COMRD	s, d	Liest Kommentardaten aus s und speichert sie als ASCII-Code in d.		3		7.11.7
	COMRDP						
Erfassung der Länge von Zeichenfolgen	LEN	s, d	Erfasst die Länge einer Zeichenfolge, die in s angegeben ist, und speichert das Ergebnis in d.		3		7.11.8
	LENP						
Konvertierung von 16-/32-Bit-Binärdaten in Zeichenfolgen	STR	s1, s2, d	Fügt dem 16-Bit-Binärdatenwert in s2 ein Dezimalkomma an der Stelle hinzu, die in s1 angegeben ist. Das Ergebnis wird in eine Zeichenfolge konvertiert und in d gespeichert.		4		7.11.9
	STRP						
	DSTR		Fügt dem 32-Bit-Binärdatenwert in s2 ein Dezimalkomma an der Stelle hinzu, die in s1 angegeben ist. Das Ergebnis wird in eine Zeichenfolge konvertiert und in d gespeichert.				
	DSTRP						

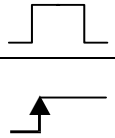
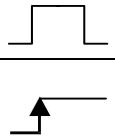
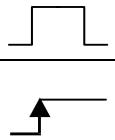
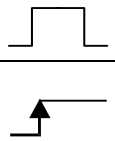
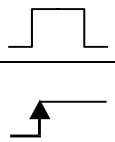
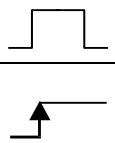
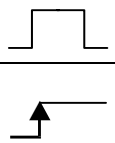
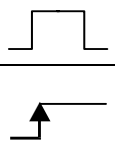
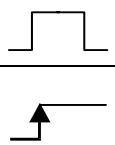
Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Konvertierung von Zeichenfolgen in 16-/32-Bit-Binärdaten	VAL	s, d1, d2	Konvertiert die in s gespeicherten Zeichenfolgen in 16-Bit-Binärdaten. Die Anzahl der Stellen und der Binärwert werden ab d1und d2 gespeichert.		4		7.11.10
	VALP						
	DVAL		Konvertiert die in s gespeicherten Zeichenfolgen in 32-Bit-Binärdaten. Die Anzahl der Stellen und der Binärwert werden in d1und d2 gespeichert.				
	DVALP						
Konvertierung von Gleitkommazahlen in Zeichenfolgen	ESTR	s1, s2, d	Konvertiert die Gleitkommazahlen (reelle Zahlen) in s1 in eine Zeichenfolge. Das Format dieser Zeichenfolge wird in s2 angegeben. Das Ergebnis wird in d gespeichert.		4		7.11.11
	ESTRP						
Konvertierung von Zeichenfolgen in dezimale Gleitkommazahlen	EVAL	s, d	Konvertiert die Zeichenfolge in s in eine dezimale Gleitkommazahl (reelle Zahl). Das Ergebnis wird in d gespeichert.		3		7.11.12
	EVALP						
Konvertierung von BIN-16-Bit-Daten in den ASCII-Code (Q)	ASC	s, n, d	Konvertiert die ab s gespeicherten 16-Bit-Binärdaten in das hexadezimale ASCII-Format und speichert das Resultat unter Berücksichtigung der in n angegebenen Anzahl von Zeichen in d.		4		7.11.13
	ASCP						
Konvertierung von alphanumerischen Zeichenfolgen in den ASCII-Code (A)	ASC	d	Konvertiert die angegebene alphanumerische Zeichenfolge in den ASCII-Code und speichert das Ergebnis in d.			* 13	7.11.14
Konvertierung von hexadezimalen ASCII-Werten in Binärwerte	HEX	s, n, d	Konvertiert die hexadezimalen ASCII-Zeichen in s in Binärwerte. Die Anzahl der zu konvertierenden Zeichen wird in n festgelegt. Das Ergebnis der Konvertierung wird in d gespeichert.		4		7.11.15
	HEXP						
Auszug der Zeichenfolgen-daten (rechter Teil der Zeichenfolge)	RIGHT	s, n, d	Speichert n Zeichen von der rechten Seite der Zeichenfolge (Ende der Zeichenfolge) in s und speichert die Zeichen in d.		4		7.11.16
	RIGHTP						
Auszug der Zeichenfolgen-daten (linker Teil der Zeichenfolge)	LEFT	s, n, d	Speichert n Zeichen von der linken Seite der Zeichenfolge (Anfang der Zeichenfolge) in s und speichert die Zeichen in d.				
	LEFTP						

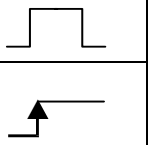
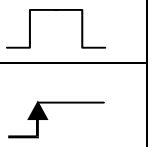
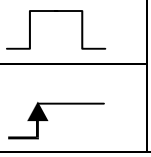
Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Speichern und Verschieben von Zeichenfolgenteilen	MIDR	s1, s2, d	Speichert einen definierten Teil der in s1 gespeicherten Zeichenfolge in d. Das erste Zeichen des zu speichernden Teils ist in s2 angegeben.		4		7.11.17
	MIDRP						
	MIDW	s1, s2, d	Speichert einen Teil in definierter Länge der in s1 gespeicherten Zeichenfolge in d. Die erste Adresse des Speicherbereichs in d ist in s2 angegeben.				
	MIDWP						
Suche von Zeichenfolgen	INSTR	s1, s2, n, d	Sucht in der Zeichenfolge in s2 die in s1 angegebene Zeichenfolge. Die Suche beginnt mit dem in n angegebenen Zeichen.		5		7.11.18
	INSTRP						
Gleitkommazahlumrechnung in das BCD-Format	EMOD	s1, s2, d1	Berechnet aus der Gleitkommazahl (reelle Zahl) in s1 unter Berücksichtigung der in s2 angegebenen Kommaverschiebung nach rechts das BCD-Format. Das Ergebnis wird in d1 gespeichert.		4		7.11.19
	EMODP						
Gleitkommazahlumrechnung in das Dezimal-Format	EREXP	s1, s2, d1	Berechnet aus der Gleitkommazahl im BCD-Format in s1 unter Berücksichtigung der in s2 angegebenen Nachkommastellen das Dezimal-Format der Gleitkommazahl (reelle Zahl). Das Ergebnis wird in d1 gespeichert.		4		7.11.20
	EREXPP						

*: Die Anzahl der Programmschritte ist abhängig von den verwendeten Operanden.
Die genaue Anzahl entnehmen Sie dem Abschnitt, in dem die einzelnen Anweisungen beschrieben sind.

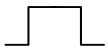
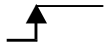
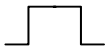
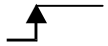
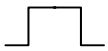
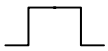
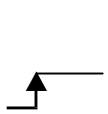
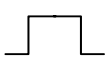
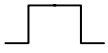
2.5.12 Anweisungen für Sonderfunktionen

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Sinus-berechnung	SIN	s, d	SIN(s+1, s) ⟶ (d+1, d)		3		7.12.1
	SINP						
Cosinus-berechnung	COS	s, d	COS(s+1, s) ⟶ (d+1, d)		3		7.12.2
	COSP						
Tangens-berechnung	TAN	s, d	TAN(s+1, s) ⟶ (d+1, d)		3		7.12.3
	TANP						
Arcussinus-berechnung	ASIN	s, d	SIN^{-1}(s+1, s) ⟶ (d+1, d)		3		7.12.4
	ASINP						
Arcuscosinus-berechnung	ACOS	s, d	COS^{-1}(s+1, s) ⟶ (d+1, d)		3		7.12.5
	ACOSP						
Arcustangens-berechnung	ATAN	s, d	TAN^{-1}(s+1, s) ⟶ (d+1, d)		3		7.12.6
	ATANP						
Umrechnung von Grad in Radiant	RAD	s, d	(s+1, s) ⟶ (d+1, d) Umrechnung von Grad in Radiant		3		7.12.7
	RADP						
Umrechnung von Radiant in Grad	DEG	s, d	(s+1, s) ⟶ (d+1, d) Umrechnung von Radiant in Grad		3		7.12.8
	DEGP						
Quadratwurzel-berechnung	SQR	s, d	$\sqrt{(s+1, s)}$ ⟶ (d+1, d)		3		7.12.9
	SQRP						

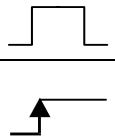
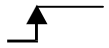
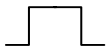
Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte Q	Schritte A	Referenz
Gleitkommazahlen als Exponent von e	EXP EXPP	s, d	$e^{(s+1, s)} \longrightarrow (d+1, d)$		3		7.12.10
Logorithmus-naturalis-Berechnung	LOG LOGP	s, d	LOG e(s+1, s) ⟶ (d+1, d)		3		7.12.11
Generierung von Zufallszahlen	RND RNDP	d	Speichern der generierten Zufallszahl in d.		2		7.12.12
Aktualisierung von Zufallszahlenserien	SRND SRNDP	s	Aktualisieren der Zufallszahl in s.		2		
Quadratwurzelberechnung aus 4-stelligen BCD-Daten	BSQR BSQRP	s, d	$\sqrt{(s)}$ → (d) +0 ganze Zahl; +1 Nachkommastelle		3		7.12.13
Quadratwurzelberechnung aus 8-stelligen BCD-Daten	BDSQR BDSQRP	s, d	$\sqrt{(s+1, s)}$ → (d) +0 ganze Zahl; +1 Nachkommastelle		3		
Sinusberechnung mit BCD-Daten	BSIN BSINP	s, d	sin (s) → (d) +0 Vorzeichen; +1 ganze Zahl; +2 Nachkommastelle		3		7.12.14
Cosinusberechnung mit BCD-Daten	BCOS BCOSP	s, d	cos (s) → (d) +0 Vorzeichen; +1 ganze Zahl; +2 Nachkommastelle		3		7.12.15
Tangensberechnung mit BCD-Daten	BTAN BTANP	s, d	tan (s) → (d) +0 Vorzeichen; +1 ganze Zahl; +2 Nachkommastelle		3		7.12.16

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Arcussinus-berechnung mit BCD-Daten	BASIN BASINP	s, d	$\sin^{-1}(s) \rightarrow$ (d) +0 Vorzeichen; +1 ganze Zahl; +2 Nachkomma-stelle		3		7.12.17
Arcuscosinus-berechnung mit BCD-Daten	BACOS BACOSP	s, d	$\cos^{-1}(s) \rightarrow$ (d) +0 Vorzeichen; +1 ganze Zahl; +2 Nachkomma-stelle		3		7.12.18
Arcustangens-berechnung mit BCD-Daten	BATAN BATANP	s, d	$\tan^{-1}(s) \rightarrow$ (d) +0 Vorzeichen; +1 ganze Zahl; +2 Nachkomma-stelle		3		7.12.19

2.5.13 Datenkontrollanweisungen

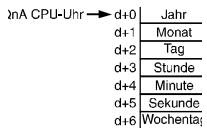
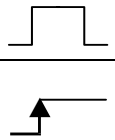
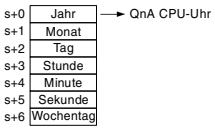
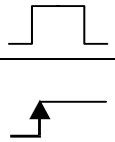
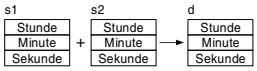
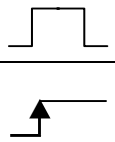
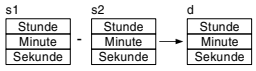
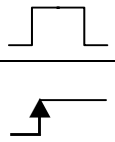
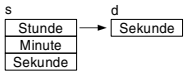
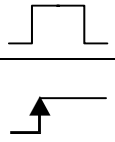
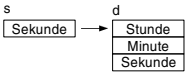
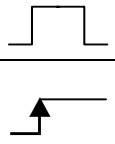
Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte Q	Schritte A	Referenz
Begrenzung des Ausgangswertebereichs von 16-/32-Bit-Binärdaten	LIMIT	s1, s2, s3, d	Wenn (s3)<(s1) Wird der Wert von s1 in d gespeichert. Wenn (s1)≤(s3)≤(s2) Wird der Wert von s3 in d gespeichert. Wenn (s2)<(s3) Wird der Wert von s2 in d gespeichert.		5		7.13.1
	LIMITP						
	DLIMIT	s1, s2, s3, d	Wenn ((s3)+1, s3)<((s1)+1, s1) Wird der Wert von ((s1)+1, s1) in (d+1, d) gespeichert. Wenn ((s1)+1, s1)≤ ((s3)+1, s3) <((s2)+1,s2) Wird der Wert von ((s3)+1, s3) in (d+1, d) gespeichert. Wenn ((s2)+1, s2)< ((s3)+1, s3)<((s2)+1, s2) Wird der Wert von ((s2)+1, s2) in (d+1, d) gespeichert.				
	DLIMITP						
Eingangsoffset-Wert von 16-/32-Bit-Binärdaten	BAND	s1, s2, s3, d	Wenn (s1)≤(s3)≤(s2) 0 → (d) Wenn (s3)<(s1) (s3)→ (s1)→ (d) Wenn (s2)<(s3) (s3)→ (s2)→ (d)		5		7.13.2
	BANDP						
	DBAND	s1, s2, s3, d	Wenn ((s1)+1, s1)≤((s3)+1, s3) ≤((s2)+1, s2) 0 → (d+1, d) Wenn ((s3)+1, s3)<(s1+1, s1) ((s3)+1, s3)-((s1)+1, s1) →(d+1, d) Wenn ((s2)+1, s2)<((s3)+1, s3) ((s3)+1, s3)-((s2)+1, s2) →(d+1, d)				
	DBANDP						
Ausgangsoffset-Wert von 16-/32-Bit-Binärdaten	ZONE	s1, s2, s3, d	Wenn s3=0: 0 → (d) Wenn s3>0: s3 + s2 → (d) Wenn s3<0: s3→ s1 → (d)		5		7.13.3
	ZONEP						
	DZONE	s1, s2, s3, d	Wenn ((s3)+1, s3)=0 0 → (d+1, d) Wenn ((s3)+1, s3)>0 ((s3)+1, s3)+((s2)+1, s2) →(d+1, d) Wenn ((s3)+1, s3)<0 ((s3)+1, s3)+((s1)+1, s1) →(d+1, d)				
	DZONEP						

2.5.14 Umschaltanweisungen für File-Registerblöcke

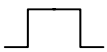
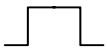
Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Umschaltung zwischen File-Register-blöcken	RSET	s	Schaltet von einem File-Registerblock, der im Programm benutzt wird, zu dem File-Register-block mit der in s angegebenen Adresse um.		2		7.14.1
	RSETP						
Umschaltung zwischen Dateien in File-Registern	QDRSET	s	Schaltet von einer im Programm benutzten File-Registerdatei in die in s angegebene File-Registerdatei um.		* 2 + n		7.14.2
	QDRSETP						
Umschaltung zwischen Dateien für Kommentar-daten in File-Registern	QCDSET	s	Schaltet von einer im Programm benutzten Kommentardatei in die in s angegebene Kommentardatei um.		* 2 + n		7.14.3
	QCDSETP						

*: n = (Anzahl Zeichen im Dateinamen/2) = Anzahl zusätzlicher Schritte. (Nachkommastellen werden aufgerundet).

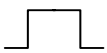
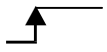
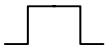
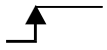
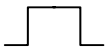
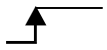
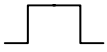
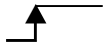
2.5.15 Uhr-Anweisungen

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte Q	Schritte A	Referenz
Lesen von Uhr-Daten	DATERD	d			2		7.15.1
	DATERDP						
Schreiben von Uhr-Daten	DATEWR	s			2		7.15.2
	DATEWRP						
Addition von Uhr-Daten	DATE+	s1, s2, d			4		7.15.3
	DATE+P						
Subtraktion von Uhr-Daten	DATE-	s1, s2, d			4		7.15.4
	DATE-P						
Wechsel des Datenformats der Uhr-Daten von Std., Min. und Sek. in Sekunden	SECOND	s, d			3		7.15.5
	SECONDP						
Wechsel des Datenformats der Uhr-Daten von Sekunden in Std., Min. und Sek.	HOUR	s, d					
	HOURP						

2.5.16 Anweisungen für Peripheriegeräte

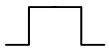
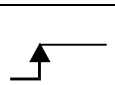
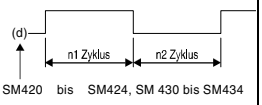
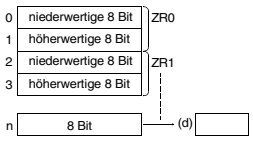
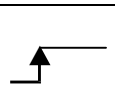
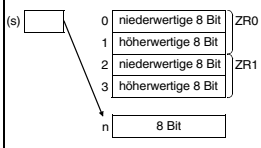
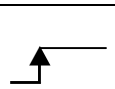
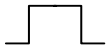
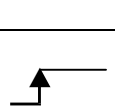
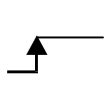
Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Ausgabe von Meldungen an Peripheriegeräte	MSG	s	Gibt die Zeichenfolge, die sich in s befindet, als Meldung an ein im Terminal-Modus angegebenes Peripheriegerät aus.		2		7.16.1
Tastatureingabe von Daten an Peripheriegeräten	PKEY	d	Die eingegebenen Tastaturdaten (Zeichen) werden aus dem im Terminal-Modus bestimmten Peripheriegerät gelesen und im ASCII-Format in d geschrieben.		2		7.16.2

2.5.17 Programmanweisungen

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Umschaltung von Programmen in den Standby-Modus	PSTOP	s	Schaltet die Programmdatei, die in s benannt wird, in den Standby-Modus um.		* 2 + n		7.17.1
	PSTOPP						
Umschaltung von Programmen in den Standby-Modus mit Rücksetzen der Ausgänge	POFF	s	Schaltet die Programmdatei, die in s benannt wird, in den Standby-Modus mit Rücksetzen der Ausgänge um.		* 2 + n		7.17.2
	POFFP						
Umschaltung von Programmen in den Modus einer Programmausführung pro Zyklus	PSCAN	s	Schaltet die Programmdatei, die in s benannt wird, in den Modus einer Programmausführung pro Zyklus um.		* 2 + n		7.17.3
	PSCANP						
Umschaltung von Programmen in den Modus niedriger Verarbeitungsgeschwindigkeit	PLOW	s	Schaltet die Programmdatei, die in s benannt wird, in den Modus niedriger Verarbeitungsgeschwindigkeit um.		* 2 + n		7.17.4
	PLOWP						

*: n = (Anzahl Zeichen im Programmnamen/2) = Anzahl zusätzlicher Schritte. (Nachkommastellen werden aufgerundet).

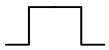
2.5.18 Weitere Anweisungen

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte Q	Schritte A	Referenz
Watch-Dog-Timer zurücksetzen	WDT		Setzt den Watch-Dog-Timer (WDT) in einem Ablaufprogramm zurück.		1	1	7.18.1
	WDTP						
Übertragstelle setzen und zurücksetzen	STC		Bei Rotations- und Schiebevorgängen wird der Übertrag (0 oder 1) im Carry Flag gespeichert.			1	7.18.2
	CLC		Nach Ausführung der Anweisung erfolgt das Rücksetzen des Carry Flags.				
Vorgabe von Ausführungszyklen	DUTY	n1, n2, d			4	7	7.18.3
Direktes Lesen eines Bytes	ZRRDB	n, d			3		7.18.4
	ZRRDBP						
Direktes Schreiben eines Bytes	ZRWRB	n, s			3		7.18.5
	ZRWRBP						
Speichern einer indirekten Adresse	ADRSET	s, d	Der in s angegebene Operand wird zur indirekten Adressierung in den unter d angegebenen Operanden gespeichert.		3		7.18.6
	ADRSETP						
Tastatureingabe numerischer Werte	KEY	s, n, d1, d2	Ermöglicht die Tastatureingabe von 8 ASCII-Zeichen an die in s angegebenen Eingänge (X). Die eingegebenen Werte werden hexadezimal codiert und in d1 gespeichert.		5		7.18.7
Sichern der Indexregisterinhalte in ein Register	ZPUSH	d	Sichert die Inhalte der Index-Register Z0 bis Z15 in d.		2		7.18.8
	ZPUSHP						

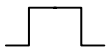
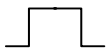
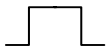
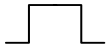
Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Wiederherstellen der Indexregisterinhalte aus einem Register	ZPOP	d	Die Inhalte der Index-Register Z0 bis Z15 in d werden wiederhergestellt.		2		7.18.9
	ZPOPP						
Schreiben von Daten in ein EEPROM-Register	EROMWR	s, n, d1,d2	Schreibt die mit n angegebene Anzahl der Datenwörter in s in das in d1 angegebene EEPROM-File-Register.		6		7.18.10
	EROMWRP						

2.6 Data-Link-Anweisungen

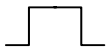
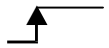
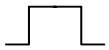
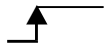
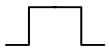
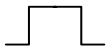
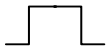
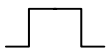
2.6.1 Netzwerk-Datenaktualisierungs-Anweisungen

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Netzwerk-Daten-aktualisierung	ZCOM	Jn Un	Datenaktualisierung in Netzwerkmodulen		5		8.5.1

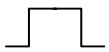
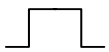
2.6.2 Erweiterte Data-Link-Anweisungen (QnA-Serie kompatibel)

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Data-Link (QnA-Serie kompatibel)	READ	Jn, s1, s2, d1, d2 Un, s1, s2, d1, d2	Lesen von CPU-Daten aus Zielstationen		9		8.6.1
	SREAD	Jn, s1, s2, d1, d2, d3 Un, s1, s2, d1, d2, d3			10		8.6.2
	WRITE	Jn, s1, s2, d1, d2 Un, s1, s2, d1, d2	Schreiben von CPU-Daten in Zielstationen		10		8.6.3
	SWRITE	Jn, s1, s2, d1, d2, d3 Un, s1, s2, d1, d2, d3			11		8.6.4
	SEND	Jn, s1, s2, d Un, s1, s2, d	Senden von Daten an Netzwerkmodule		8		8.6.5
	RECV	Jn, s, d1, d2 Un, s, d1, d2	Lesen mittels SEND-Anweisung gesendeter Daten		8		8.6.6
	REQ	Jn, s1, s2, d1, d2 Un, s1, s2, d1, d2	Datenanforderung anderer Stationen		8		8.6.7
	ZNFR	Jn, s1, s2, d Un, s1, s2, d	Lesen von Daten aus Sondermodulen in Remote-Stationen		8		8.6.8
	ZNTO	Jn, s1, s2, d Un, s1, s2, d	Schreiben von Daten in Sondermodule in Remote-Stationen		8		8.6.9

2.6.3 Data-Link-Anweisungen (A-Serie kompatibel)

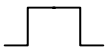
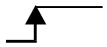
Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Data-Link (A-Serie kompatibel)	J.ZNRD	Jn, n1, s, n2, d1, d2	Lesen von QnA-Daten aus Zielstationen		32		8.7.1
	JP.ZNRD		Lesen von Daten aus lokalen Sationen				
	J.ZNWR	Jn, n1, s, n2, d1, d2	Schreiben von QnA-Daten in eine Ziel-Station		32		8.7.2
	JP.ZNWR		Schreiben von Daten in eine lokale Station				
	LRDP	s, n1, n2, d	Lesen von Daten aus einer lokalen Sation (nur A-Serie)			11	8.7.3
	LWTP	Jn, s, d1, d2	Schreiben von Daten in eine lokale Station (nur A-Serie)			11	8.7.4
	RFRP	n1, n2, n3, d	Lesen von Daten aus einem Sondermodul in einer Remote-Station		11	11	8.7.5
	G.RFRP	Un, n1, n2, d1, d2					
	RTOP	s, n1, n2, n3	Schreiben von Daten in einem Sondermodule in einer Remote-Station		11	11	8.7.6
	G.RTOP	Un, n1, s, n2, d1					

2.6.4 Routing-Informationen

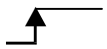
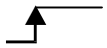
Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Lesen/ Schreiben von Routing-Informationen	Z.RTREAD	n, d	Lesen der Routing-Informationen des Netzwerks n und Speichern der Daten in d		7		8.8.1
	ZP.RTREAD						
	Z.RTWRITE	s, n	Schreiben der Routing-Informationen aus s in das Netzwerk n		8		8.8.2
	ZP.RTWRITE						

2.7 Anweisungen für CPUs des MELSEC System Q

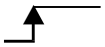
2.7.1 Modul-Informationen auslesen

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Informationen aus einem Modul lesen	UNIRD	n1, d, n2	Die Modulinformationen werden ab der E/A-Adresse, die in n1 angegeben ist, gelesen und ab der Adresse abgelegt, die in d angegeben wird. Die Länge der Daten wird in n2 angegeben.		4		9.1.1
	UNIRDP						

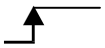
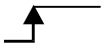
2.7.2 Fehlererkennung und Fehlerbeseitigung

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Überwachung (Trace) setzen/ rücksetzen	TRACE		Führt die Zwischenspeicherung der durch ein Programmiergerät definierter Operanden-daten auf der Speicher-karte aus, wenn SM800, SM801 und SM802 gesetzt sind.		1		9.2.1
	TRACER		Die durch die TRACE-Anweisung gespeicherten Daten werden gelöscht.		1		9.2.1

2.7.3 Transfer von Daten in und aus Dateien

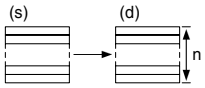
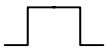
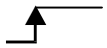
Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Daten in eine Datei schreiben	SP.FWRITE	u0, s0, d0, s1, s2, d1	Schreibt Daten in eine angegebene Datei		11		9.3.1
Daten aus einer Datei lesen	SP.FREAD	u0, s0, d0, s1, d1, d2	Liest Daten aus einer angegebene Datei		11		9.3.2

2.7.4 Programmanweisungen

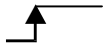
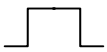
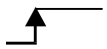
Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Programm aus Speicher laden	PLOADP	s, d	Lädt eine Programmdatei, aus dem Speicher (nicht Laufwerk 0) in Laufwerk 0 und schaltet das Programm in den Standby-Modus.		3		9.4.1
Programm, das im Standby-Modus ist, löschen	PUNLOADP	s, d	Die Programmdatei, die in Laufwerk 0 im Standby-Modus ist, wird gelöscht.		3		9.4.2
Programm, das im Standby-Modus ist, löschen und Programm aus Speicher laden	PSWAPP	s1, s2, d	Die Programmdatei s1, die in Laufwerk 0 im Standby-Modus ist, wird gelöscht. Anschliessend wird die Programmdatei s2, aus dem Speicher (nicht Laufwerk 0) in Laufwerk 0 transferiert und in den Standby-Modus geschaltet.		4		9.4.3

Diese Funktionen stehen nur im GX Developer zur Verfügung, da der GX IEC Developer das File-Konzept nicht unterstützt.

2.7.5 Transferanweisungen

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Übertragung von Datenblöcken mit hoher Geschwindigkeit	RBMOV	s, d, n			4		9.5.1
	RBMOVP						

2.7.6 Anweisungen zum Datenaustausch im Multi-CPU-Betrieb

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Daten in den gemeinsamen Speicherbereich eintragen	S.TO	s1, s2, s3, s4, d	Die Daten werden in den gemeinsamen Speicherbereich der CPU eingetragen, in der die S.TO-Anweisung ausgeführt wird.		5		9.6.1
	S.TOP				5		
Daten aus dem gemeinsamen Speicherbereich einer anderen CPU lesen.	FROM	n1, n2, n3, d	Aus dem gemeinsamen Speicherbereich einer anderen CPU werden Daten zu der CPU übertragen, in der die FROM-Anweisung ausgeführt wird.		5		9.6.2
	FROMP				5		
Gemeinsamen Speicherbereich aktualisieren	COM		Der gemeinsame Speicherbereich für den Multi-CPU-Betrieb wird aktualisiert.		1		6.7.3

2.8 Spezielle Anweisungen für eine Q4ARCPU

2.8.1 Anweisungen zur Einstellung der Betriebsart

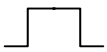
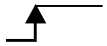
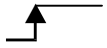
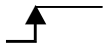
Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Anlaufverhalten der CPU einstellen	SGMODE	s1, s2	Auswahl zwischen Neustart und Wiederanlauf (Warmstart)				10.1.1
Verhalten bei Umschaltung	CGMODE	s	Einstellung des Verhaltens bei Umschaltung von der aktiven CPU zur Reserve-CPU				10.1.2

2.8.2 Transferanweisungen

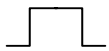
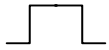
Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Datentransfer zur Standby-CPU	TRUCK	s	Operandendaten von der aktiven CPU zur Reserve-CPU übertragen				10.2.1
Datenaustauch mit Sondermodulen	SPREF	s	Daten blockweise aus dem Pufferspeicher von Sondermodulen lesen oder in den Speicher eintragen				10.2.2

2.9 Anweisungen für Sondermodule

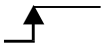
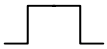
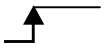
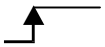
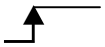
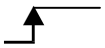
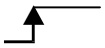
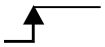
2.9.1 Anweisungen für serielle Schnittstellen-Module

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Daten vom Schnittstellenmodul in die SPS-CPU übertragen	BUFRCVS	„Un“, n1, d1	Empfangene Daten werden in einem Interrupt-Programm vom Schnittstellenmodul in die SPS-CPU übertragen.				11.1.1
Anwenderdefinierte Datenrahmen lesen	GETE	Un, s1, s2, d	Anwenderdefinierte Datenrahmen aus dem Schnittstellenmodul lesen				11.1.2
	GETEP						
Anwenderdefinierte Datenrahmen schreiben oder löschen	PUTE	Un, s1, s2, d	Anwenderdefinierte Datenrahmen in ein Schnittstellenmodul eintragen oder aus einem Schnittstellenmodul löschen				11.1.3
	PUTEP						
Daten versenden	PRR	Un, s, d	Daten über das Schnittstellenmodul mittels anwenderdefinierter Datenrahmen senden				11.1.4
	PRRP						

2.9.2 Anweisungen für PROFIBUS/DP-Module

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Daten lesen	BBLKRD	„Un“, n1, n2, d	Aus dem Pufferspeicher eines PROFIBUS/DP-Moduls werden Daten gelesen und in der CPU abgelegt.				11.2.1
	BBLKRDP						
Daten schreiben	BBLKWR	„Un“, n1, n2, s	Daten aus der CPU werden in den Pufferspeicher eines PROFIBUS/DP-Moduls eingetragen.				11.2.2
	BBLKWR						

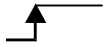
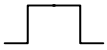
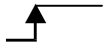
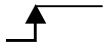
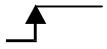
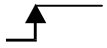
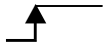
2.9.3 Anweisungen für ETHERNET-Module

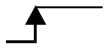
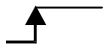
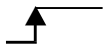
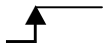
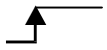
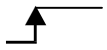
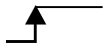
Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Daten aus festen Puffern lesen	BUFRCV	„Un“, s1, s2, d1, d2	Die bei der Kommunikation mit festen Puffern empfangenen Daten werden aus dem ETHERNET-Modul gelesen.				11.3.1
	BUFRCVS	„Un“, s1, d1					11.3.2
Daten in feste Puffer schreiben	BUFSND	„Un“, s1, s2, s3, d1	Daten aus der CPU werden zum ETHERNET-Modul übertragen				11.3.3
Verbindung öffnen	OPEN	„Un“, s1, s2, d1	Aufbau einer Verbindung				11.3.4
Verbindung schließen	CLOSE	„Un“, s1, s2, d1	Abbau einer Verbindung				11.3.5
Fehler löschen	ERRCLR	„Un“, s1, d1	Fehlercodes in Pufferspeicher löschen, LED „ERR.“ des ETHERNET-Moduls ausschalten				11.3.6
Fehlercode lesen	ERRRD	„Un“, s1, d1	Fehlercodes aus Pufferspeicher lesen				11.3.7
Erneute Initialisierung des ETHERNET-Moduls	UINI	„Un“, s1, d1	Das mit „Un“ angegebene ETHERNET-Modul wird nochmal initialisiert.				11.3.8

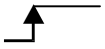
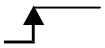
2.9.4 Anweisung für MELSECNET/10

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Stationen paarweise verbinden	PAIRSET	Jn, s1	Stationen festlegen, die beim Duplexbetrieb verbunden sind				11.4.1

2.9.5 Anweisungen für CC-Link

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Netzwerkparameter übertragen (A-Serie)	RLPA	n, d1, d2	Übertragung der Netzwerkparameter in die Master-Station des CC-Link		—	23	11.5.1
	RLPA_P						
Netzwerkparameter übertragen (System Q)	RLPASET	Un, s1 bis s5, d1				—	11.5.2
	RLPASET_P						
Parameter für automatisch Aktualisierung übertragen (A-Serie)	RRPA	n, d	Übertragung der Parameter für die automatische Aktualisierung von Operanden		—	20	11.5.3
	RRPA_P						
Daten aus Pufferspeicher oder anderer SPS-CPU lesen (A-Serie)	RIRD	n1, n2, d1, d2	Daten aus dem Pufferspeicher eines CC-Link-Moduls einer anderen Station oder aus der SPS-CPU dieser Station lesen		—	26	11.5.4
	RIRD_P						
Daten aus Pufferspeicher oder anderer SPS-CPU lesen (QnA-Serie und System Q)	RIRD	Un, s, d1, d2			8	—	11.5.5
	RIRD_P						
Daten in den Pufferspeicher oder in eine andere SPS-CPU eintragen (A-Serie)	RIWT	n1, n2, d1, d2	Daten in den Pufferspeicher eines CC-Link-Moduls einer anderen Station oder in die SPS-CPU dieser Station schreiben		—	26	11.5.6
	RIWT_P						

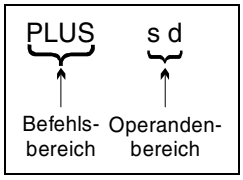
Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Daten in den Pufferspeicher oder in eine andere SPS-CPU eintragen (QnA-Serie, System Q)	RIWT	Un, s, d1, d2	Daten in den Pufferspeicher eines CC-Link-Moduls einer anderen Station oder in die SPS-CPU dieser Station schreiben		8	—	11.5.7
	RIWT_P						
Daten aus intelligenter Station lesen (A-Serie)	RICV	n1, n2, d1, d2, d3	Aus dem Pufferspeicher einer intelligenten CC-Link-Station werden Daten unter Verwendung eines Handshakes gelesen.		—	29	11.5.8
	RICV_P						
Daten aus intelligenter Station lesen (QnA-Serie, System Q)	RICV	Un, s1, s2, d1, d2			10	—	11.5.9
	RICV_P						
Daten in intelligente Station schreiben (A-Serie)	RISEND	n1, n2, d1, d2, d3	In den Pufferspeicher einer intelligenten CC-Link-Station werden Daten unter Verwendung eines Handshakes eingetragen.		—	29	11.5.10
	RISEND_P						
Daten in intelligente Station schreiben (QnA-Serie, System Q)	RISEND	Un, s1, s2, d1, d2			10	—	11.5.11
	RISEND_P						
Daten in den automatisch aktualisierten Speicher schreiben (A-Serie)	RITO	n1, n2, n3, n4, d1	Daten aus der SPS-CPU in den automatisch aktualisierten Bereich des Pufferspeichers der CC-Link-Master-Station eintragen. Anschließend werden diese Daten zu der angegebenen Station übertragen.		—	29	11.5.12
	RITO_P						
Daten in den automatisch aktualisierten Speicher schreiben (QnA-Serie, System Q)	RITO	Un, n1, n2, n3, d			9	—	11.5.13
	RITO_P						

Gruppe	Anweisung	Variablen	Bedeutung	Ausführung	Schritte		Referenz
					Q	A	
Daten aus dem automatisch aktualisierten Speicher lesen (A-Serie)	RIFR	n1, n2, n3, n4, d1	Daten lesen, die von einer anderen Station in den automatisch aktualisierten Bereich des Pufferspeichers der CC-Link-Master-Station eingetragen wurden.		—	29	11.5.14
	RIFR_P						
Daten aus dem automatisch aktualisierten Speicher lesen (QnA-Serie, System Q)	RIFR	Un, n1, n2, n3, d			9	—	11.5.15
	RIFR_P						

3 Konfiguration der Anweisungen

3.1 Aufbau einer Anweisung

Die meisten Anweisungen bestehen aus einem Befehlsbereich und einen Operandenbereich. Einige Anweisungen, die keinen Operanden benötigen, bestehen nur aus dem Befehlsbereich.



Befehlsbereich

Der Befehlsbereich beschreibt die Funktionalität der Anweisung.

PLUS $\hat{=}$ Addieren

Operandenbereich

Der Operandenbereich beschreibt die zu benutzenden Konstanten oder Variablen. Der Operandenbereich kann aus drei Teilen bestehen: der Datenquelle (s), dem Datenziel (d) und der Anzahl (n).

3.1.1 Datenquelle (s)

- Die Datenquelle bezeichnet die Operanden, die durch die Anweisung verarbeitet werden. In 16-Bit-Anweisungen wird die Datenquelle mit s bezeichnet. In 32-Bit-Anweisungen wird sie mit s+1 und s bezeichnet.
- In einer Datenquelle können Konstanten oder Variablen angegeben werden.

Konstanten

Bezeichnen einen konstanten numerischen Wert, der durch die Anweisung verarbeitet wird. Dieser Wert wird durch das Schreiben des Programms gesetzt. Er kann während der Ausführung des Programms nicht geändert werden. Es ist ratsam, jede Variable, die als Konstante verwendet werden soll, zu indizieren.

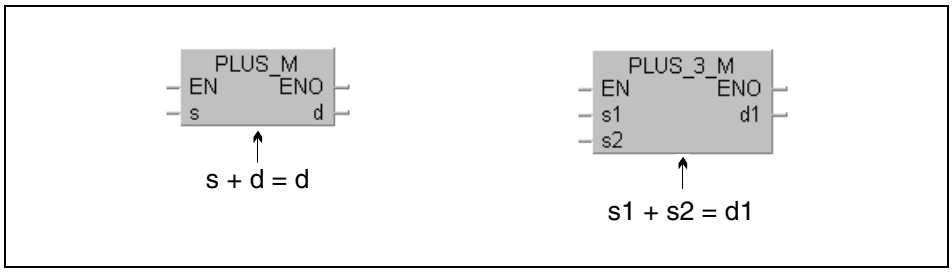
Variablen

Bezeichnen einen Operanden, in dem die Daten gespeichert sind, die durch die Anweisung verarbeitet werden (siehe auch Kap. 3.4).
Bevor die Anweisung ausgeführt wird, müssen die Daten in dem Operanden gespeichert sein. Es ist möglich, die gespeicherten Daten während der Ausführung des Programms zu ändern.

3.1.2 Datenziel (d)

- Das Datenziel bezeichnet die Operanden, in denen die Daten nach der Verarbeitung gespeichert werden. In 16-Bit-Anweisungen wird das Datenziel mit d bezeichnet. In 32-Bit-Anweisungen wird es mit d+1 und d bezeichnet.
  Trotzdem erfordern einige Anweisungen mit 2 Operanden, dass der zu verarbeitende Wert in einem Datenziel d gespeichert wird, bevor die Anweisung ausgeführt wird. Das Verarbeitungsergebnis wird dann wiederum in demselben Operanden d gespeichert.

  Beispiel: Die Additionsanweisung für BIN-16-Bit-Daten

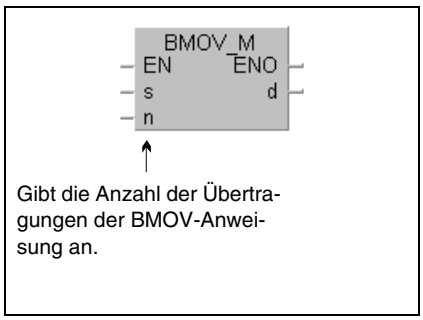


- Zur Speicherung der Daten muss immer ein Operand als Datenziel gesetzt sein.

3.1.3 Anzahl (n)

- Mit n wird angegeben, wieviel Operanden verwendet werden sollen oder wie oft eine Anweisung ausgeführt werden soll.

  Beispiel: Die Blockübertragungsanweisung



- Der Wert in n kann innerhalb des Bereichs von 0 und 32767 liegen. Ist die Anzahl auf 0 gesetzt, wird die Anweisung nicht ausgeführt.

3.2 Schreibweise der Anweisungen

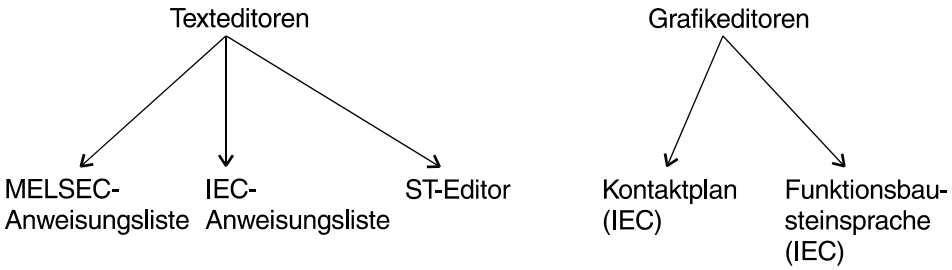
Aus der Schreibweise lassen sich einige Besonderheiten der Anweisung ablesen.

3.2.1 16-/ 32-Bit und Puls

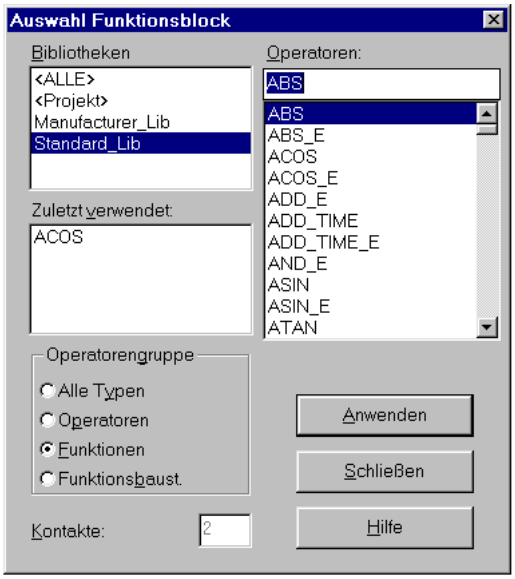
SORT	16-Bit-Verarbeitung
SORTP	16-Bit-Verarbeitung mit Puls
DSORT	32-Bit-Verarbeitung
DSORTP	32-Bit-Verarbeitung mit Puls

3.2.2 MELSEC und IEC

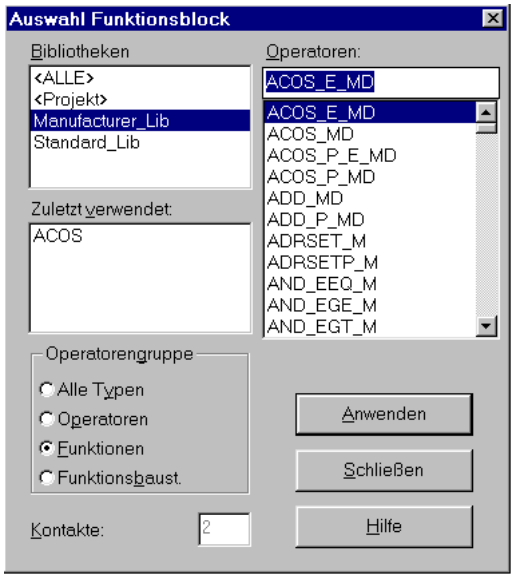
Im GX IEC Developer stehen für die Anweisungen verschiedene Editoren zur Verfügung:



In diesen Editoren erscheinen die Anweisungen in einer anderen Schreibweise.



Bei der Auswahl einer Anweisung im GX IEC Developer erscheint dieses Dialogfenster.
Im Auswahlfenster „Bibliotheken“ legen Sie fest, welche Anweisungen im Feld „Operatoren“ zur Auswahl angezeigt werden:
ALLE: MELSEC- und IEC-Anweisungen
Projekt: Selbsterstellte Funktionen u. FB
Manufacturer: MELSEC-Anweisungen
Standard: IEC-Anweisungen



Bei der Auswahl einer Anweisung im GX IEC Developer erscheint dieses Dialogfenster, wenn Sie sich in der Herstellerbibliothek (Manufacturer_Lib) befinden. Die Liste enthält die sogenannten „Angepassten“ MELSEC-Anweisungen.

Die Funktionalität der „Echten“ und „Angepassten“ Anweisungen ist identisch. Sie unterscheiden sich nur in der Schreibweise.

Bedeutung der Endungen im IEC-Editor:

Endung im IEC-Editor	Bedeutung
_M	MELSEC-Anweisung
_P_M	Gepulste Ausführung einer Anweisung
_MD	Erweiterte MELSEC-Applikationsanweisung (Dedicated Instruction) (siehe auch Kap. 3.3)
_P_MD	Gepulste Ausführung einer erweiterten Applikationsanweisung
_K_MD	Verwendung einer Konstanten in einer erweiterten Applikationsanweisung
_K_P_MD	Gepulste Ausführung und Verwendung einer Konstanten in einer erweiterten Applikationsanweisung
_S_MD	Erweiterte MELSEC-Applikationsanweisung für CPUs des System Q
_P_S_MD	Gepulste Ausführung einer erweiterten Applikationsanweisung für CPUs des System Q

3.2.3 Weitere Besonderheiten der Schreibweise

In der untenstehenden Tabelle sind die Symbole aufgeführt, mit denen einige Anweisungen im MELSEC-Editor bezeichnet sind. In der rechten Spalte sind die entsprechenden Namen im IEC-Editor aufgeführt.

Beispiel: MELSEC-Editor IEC-Editor

LD$> LD_STRING_GT_M

MELSEC-Editor	IEC-Editor
$	STRING
=	EQ
<>	NE
<=	LE
<	LT
>=	GE
>	GT
+	PLUS
-	MINUS
x	MULTI
/	DIVID

3.2.4 Festlegung der Schreibweise in diesem Handbuch

In den Kapiteln 5 bis 8, in denen die Anweisungen detailliert beschrieben werden, werden beide Editoren, d.h. beide Schreibweisen verwendet. In der Kopfzeile steht jeweils die „Echte“ MELSEC-Anweisung, wie sie in der MELSEC-Anweisungsliste erscheint.

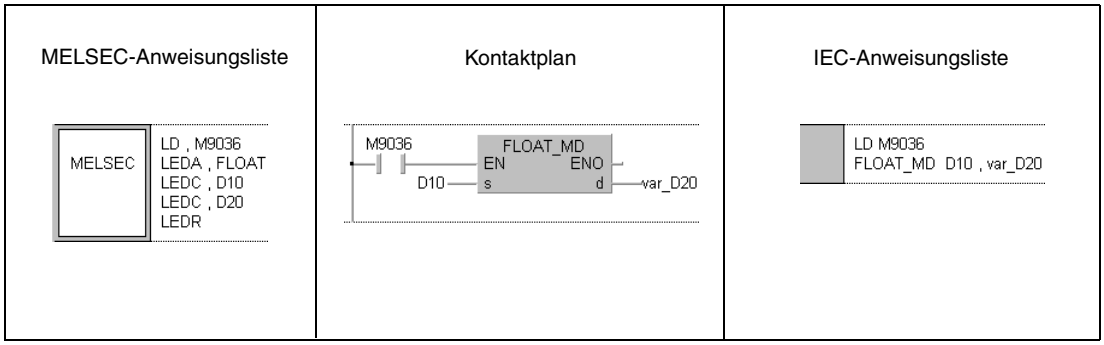
HINWEIS *Die tabellarische Übersicht am Anfang jeder Anweisungsgruppe zeigt immer beide Schreibweisen.*

3.3 Programmierung der erweiterten Anweisungen

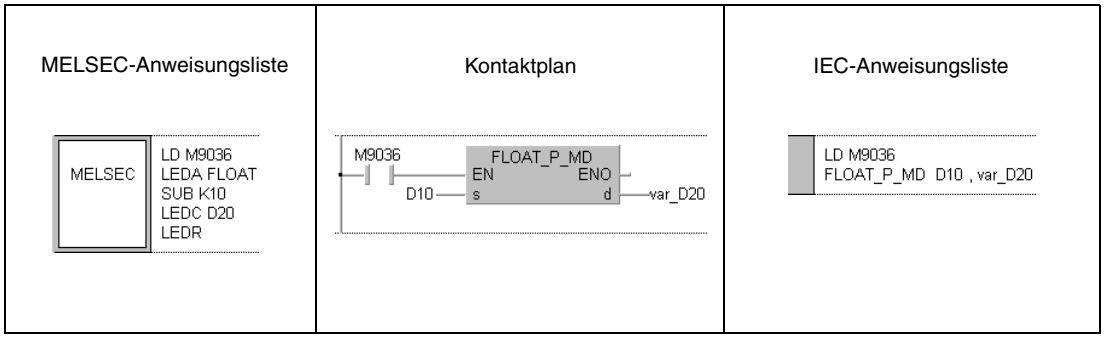
Die erweiterten Programmieranweisungen (Dedicated Instructions) sind „Angepasste“ Anweisungen, die sich nicht nur in ihrer Schreibweise von den „Echten“ MELSEC-Anweisungen unterscheiden, sondern die auch nach einer besondere Programmierweise für die unterschiedlichen CPUs verlangen.

Um z.B. die Funktion der FLOAT_MD-Anweisung auch im MELSEC-Editor einer A-Serie CPU zu erhalten, ist eine besondere Vorgehensweise erforderlich. Im MELSEC-Editor muss die FLOAT_MD Anweisung mit den Anweisungen LEDA, LEDC, LEDR zusammen programmiert werden. In den IEC-Editoren können die erweiterten Programmieranweisungen in gewohnter Weise programmiert werden.

Beispiel: Programmierung der FLOAT_MD-Anweisung (normale Ausführung 16 Bit)



Beispiel: Programmierung der FLOAT_P_MD-Anweisung (gepulste Ausführung 16 Bit, Verwendung einer Konstanten im Operanden s)



Weiterführende Informationen zur Programmierung der erweiterten Applikationsanweisungen finden Sie in den folgenden Handbüchern:

- GX IEC Developer Benutzerhandbuch
- Programming Manual (Dedicated Instructions)

3.4 Programmierung von Variablen

Der überwiegende Teil der Anweisungen erfordert neben dem Befehlsteil auch einen Operandenteil, in dem die Variablen angegeben werden. In diesen Variablen befinden sich die Werte für die Verarbeitung der Anweisung.

Je nach gewähltem Editor im GX IEC Developer ist eine unterschiedliche Programmierung der Variablen erforderlich.

Im MELSEC-Editor:

Die Datenregister D100 und D10 können mit der Variablenbezeichnung D100 und D10 direkt übergeben werden.

Die angeschlossene SPS erkennt automatisch, dass es sich um die folgenden Operanden handelt.

D100 = D100 und D101
D10 = D10, D11, D12, D13

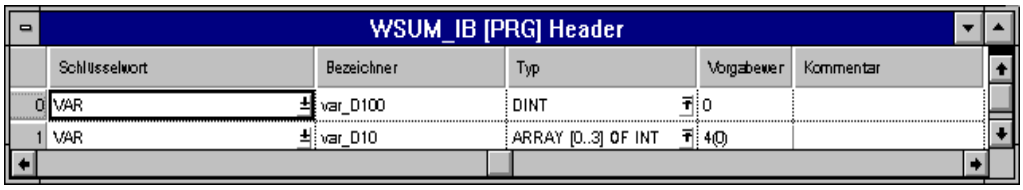
Im IEC-Editor:

Im IEC-Editor können direkte Operanden nur dann übergeben werden, wenn es sich tatsächlich nur um diesen Operanden handelt.

Beispiel: AND D10

Um die Anweisung DWSUMP_M einzusetzen, muss vorab im Header einer Programm-Organisationseinheit (POE) eine Variablendeklaration vorgenommen werden.

Beispiel**:** Header der IEC-AWL



„var_D100“ und „var_D10“ stehen hier als Bezeichner (Namen). Es handelt sich nicht direkt um die Operanden D100 und D10. Die SPS vergibt dafür intern freie Registerbereiche.

Beispiel: DWSUMP

DWSUMP	var_D100, s ↑ 32-Bit	4, n ↑ 16-Bit oder Konstante	var_D10 d ↑ Array

Die Variable var_D100 ist vom Typ DINT (32 Bit). Die Variable var_D10 ist vom Typ ARRAY. Das Array enthält vier 16-Bit-Register vom Typ INT (siehe auch Kap. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“).

Festlegung der Schreibweise

In den Screenshots weisen die Bezeichnungen var_D100 oder var_D10 sofort darauf hin, dass es sich nicht um direkte Operanden, sondern um Bezeichner handelt. In diesen Fällen ist immer eine Variablendeklaration erforderlich! Ist die Programmierung der Anweisung nur mit einer Variablendeklaration möglich, erscheint immer ein entsprechender Hinweis.

HINWEIS Als Bezeichner kann jeder beliebige Name eingesetzt werden (z.B. Motor 1, Lampe). Die Namen var_D100 oder var_D10 wurden gewählt, um den direkten Vergleich der Programmierung im MELSEC-Editor zu ermöglichen.

Einen Überblick über die Datentypen der Operanden der einzelnen Anweisungen gibt die Variablentabelle zu Beginn jeder Anweisung (das Beispiel zeigt die Variablentabelle der DWSUM-Anweisung 7.5.14).

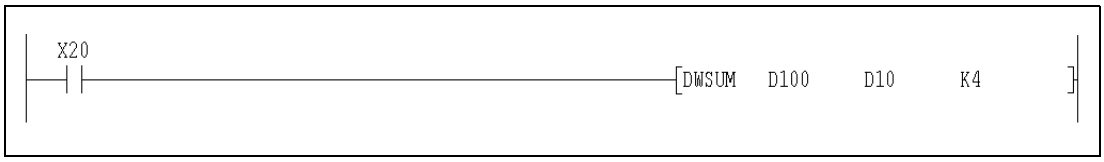
Operand	Befehlswert	Datentyp	
		MELSEC	IEC
s	Erste Adresse des Operanden, in dem die zu addierenden Daten gespeichert sind.	BIN-32-Bit	ANY32
d	Erste Adresse des Operanden, in dem das Ergebnis gespeichert wird.	BIN-64-Bit	Array [1..4] of ANY16
n	Anzahl der zu addierenden Datenblöcke.	BIN-16-Bit	ANY16

Im GX Developer:

Die Datenregister D100 und D10 können mit der Variablenbezeichnung D100 und D10 direkt übergeben werden.

Die angeschlossene SPS erkennt automatisch, dass es sich um die folgenden Operanden handelt.

D100 = D100 und D101
D10 = D10, D11, D12, D13



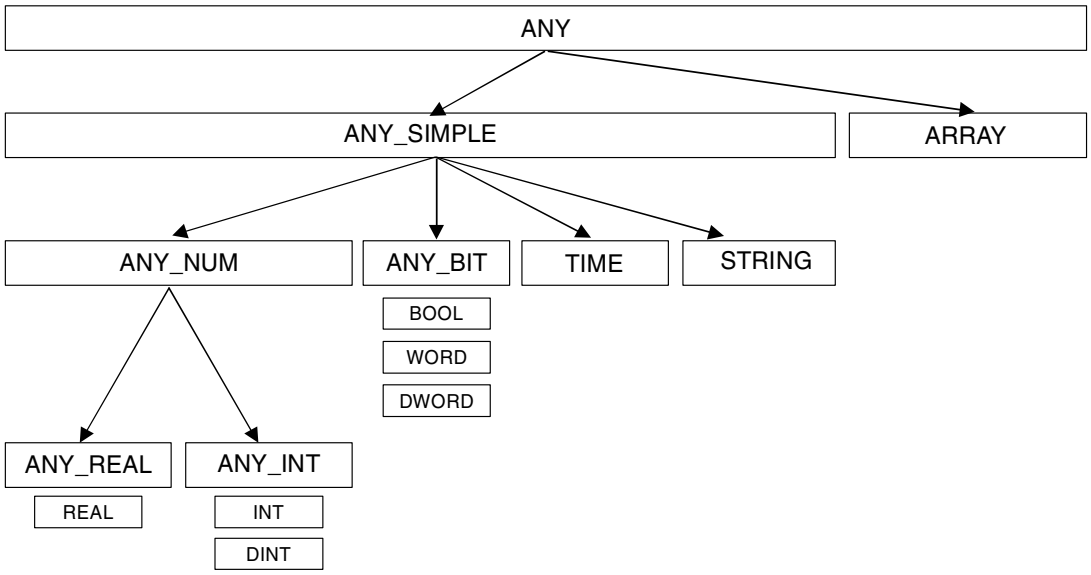
3.5 Datentypen

Mit dem Datentyp werden die Anzahl und die Bearbeitung der Bits sowie die Wertebereiche der Variablen festgelegt.

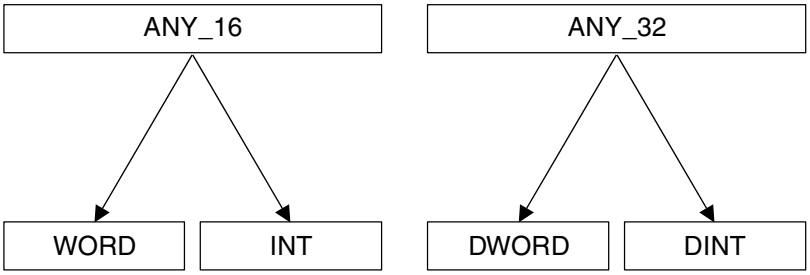
Es gibt folgende Datentypen:

Datentyp		Wertebereich	Größe	CPU
BOOL	Boolean	0 (FALSE), 1 (TRUE)	1 Bit	A-Serie Q-Serie System Q
INT	ganze Zahl (INTEGER)	-32.768 bis 32.767	16 Bit	
DINT	doppelte ganze Zahl	-2.147.483.648 bis 2.147.483.647	32 Bit	
WORD	Bitfolge 16	0 bis 65.535	16 Bit	
DWORD	Bitfolge 32	0 bis 4.294.967.295	32 Bit	
REAL	Gleitkommazahl	3.4 +/- 38 (7 Stellen)	32 Bit	
TIME	Zeitwert	T#-24d-0h31m23s648.00ms bis T#24d20h31m23s647.00ms	32 Bit	Q-Serie System Q
STRING	Zeichenfolge	max. 50 Zeichen		

Hierarchie der Datentypen ANY



Hierarchie der Datentypen ANY16 und ANY32



Datentyp	Bedeutung
ANY	jeder Datentyp
ANY_SIMPLE	einfacher Datentyp
ANY_NUM	numerischer Datentyp
ANY_REAL	Gleitkommazahl
ANY_INT	Integer-Datentyp
ANY_BIT	Bitverarbeitender Datentyp
ANY_16	jeder 16-Bit-Datentyp
ANY_32	jeder 32-Bit-Datentyp
TIME	Zeit
STRING	Zeichenfolge
REAL	Gleitkommazahl
INT	Integer-Wert
DINT	Double-Integer-Wert
BOOL	Boolescher Wert
WORD	Wort (16 Bit)
DWORD	Doppelwort (32 Bit)
ARRAY	Feld

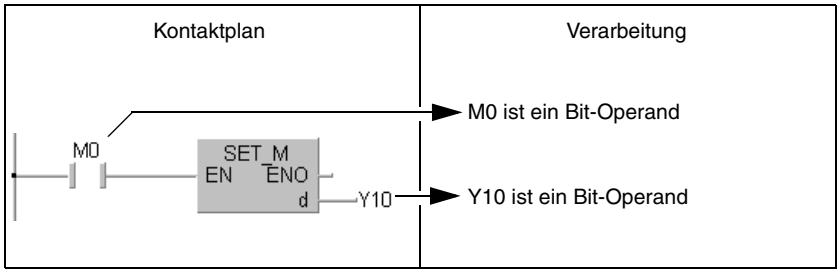
3.5.1 Verarbeitung von Daten

Verarbeitung von Bit-Daten

Ein Bit-Operand (X, Y, M, K, S, B oder F) kann zwei Zustände (EIN=1 oder AUS=0) annehmen. Sein Zustand kann demnach mit einem Bit (1 oder 0) ausgedrückt werden. Bit-Verarbeitung erfolgt immer dann, wenn ein bestimmter Bit-Operand im Programm angesprochen wird. Bei der Verarbeitung von 16- oder 32-Bit-Anweisungen werden mehrere Bit-Operanden in Blöcken von 16 oder 32 Operandenadressen zusammengefasst.

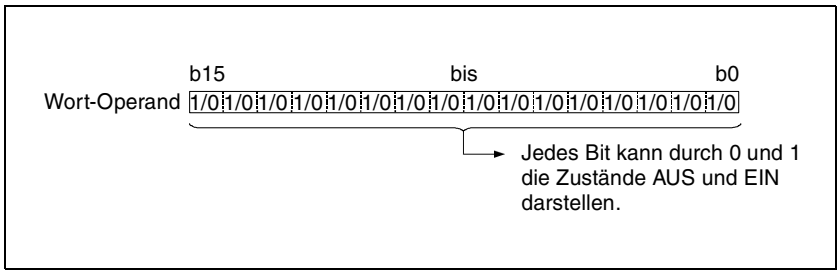
- Verwendung von Bit-Operanden

  Ein Bit-Operand (z.B. Eingänge, Ausgänge, Merker) besteht aus einem Bit.

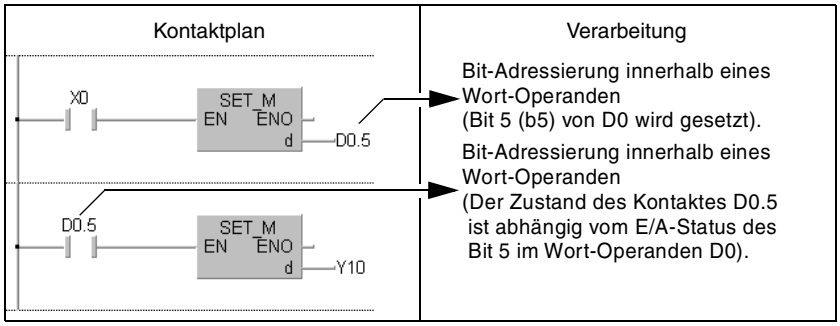


- Verwendung von Wort-Operanden

  Bei den CPUs der MELSEC QnA-Serie und des System Q ist es möglich, jedes einzelne Bit in einem Wort-Operanden zu adressieren.



Die Bit-Adressierung muss im Hexadezimalformat erfolgen. Soll beispielsweise das Bit 5 (b5) von D0 adressiert werden, lautet die Adresse D0.5. Das Bit 10 wird durch D0.A adressiert. Einzelne Bits von Timern, Countern und remanenten Timern können nicht adressiert werden.



- Verwendung von Bit-Blöcken

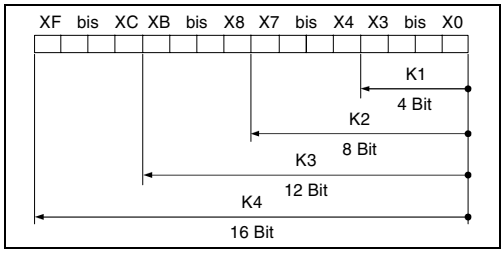
  Einzelne Bits können in 4er-Blöcken zusammengefasst werden und so Wort-Daten verarbeiten. Die genaue Erläuterung finden Sie auf den folgenden Seiten in dem Abs. „Verarbeitung von Wort-Daten (16/ 32 Bit)“.

Verarbeitung von Wort-Daten (16 Bit)

- Verwendung von Bit-Operanden

  Bit-Operanden sind in der Lage, Wort-Daten zu verarbeiten. Dazu muss die Anzahl der Bit-Operanden (Adressen) festgelegt werden. Bis zu 16 Bits können in Blöcken mit jeweils 4 Bits verarbeitet werden. Die Länge eines Blocks wird mit K1 bis K4 festgelegt.

K1X0	4 Adressen von X0 bis X3
K2X0	8 Adressen von X0 bis X7
K3X0	12 Adressen von X0 bis XB
K4X0	16 Adressen von X0 bis XF

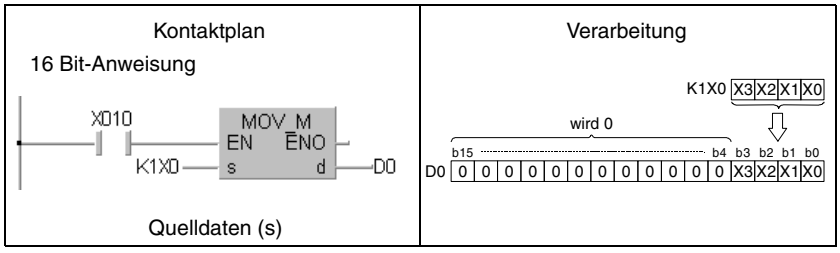


- Angabe der Bit-Blöcke für s

  Bei einer Vorgabe der Blocklänge für Quelldaten s ist der Bereich der als Quelldaten verarbeiteten Werte in der folgenden Tabelle aufgeführt.

Blocklänge	16-Bit-Anweisung
K1 (4 Stellen)	0 bis 15
K2 (8 Stellen)	0 bis 255
K3 (12 Stellen)	0 bis 4095
K4 (16 Stellen)	-32768 bis 32767

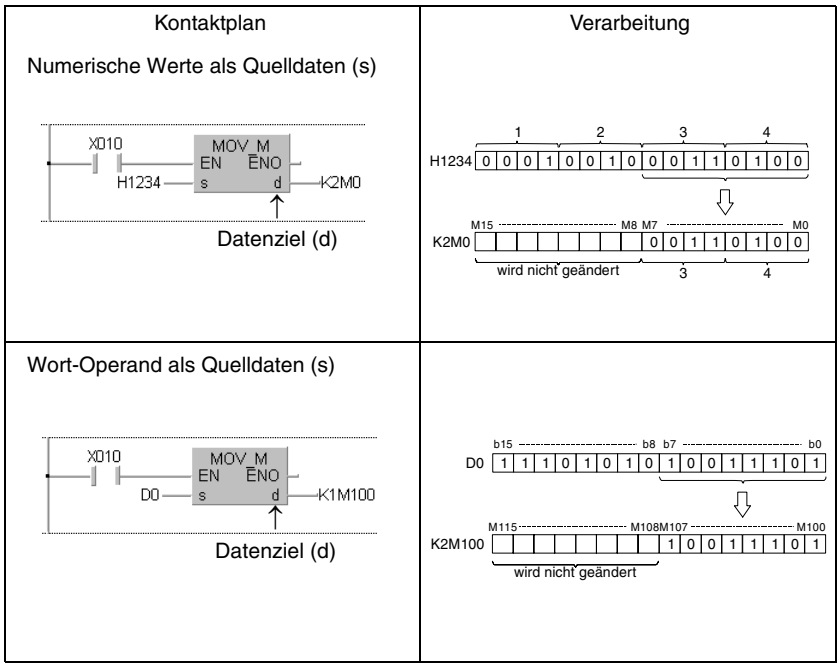
Die nicht benötigten Bit-Adressen werden auf Null gesetzt



HINWEIS *Bei der blockweisen Adressierung von Bit-Operanden kann für die Adresse des ersten Bit-Operanden (Startadresse) ein beliebiger Wert angegeben werden.*

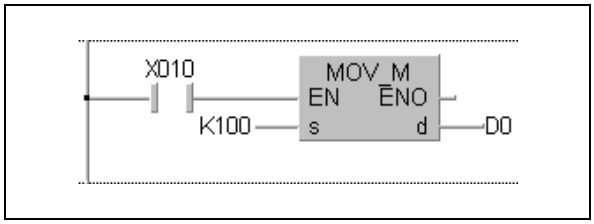
- Angabe der Bit-Blöcke für d

  Die Vorgabe der Blocklänge für Zieldaten d legt den Adressenbereich fest, in den die Daten gesetzt werden sollen. Die Bit-Adressen oberhalb der angegebenen Adressen bleiben unberücksichtigt.



- Verwendung von Wort-Operanden

  Wort-Operanden werden mit einer Adresse festgelegt. Diese Adresse umfasst 16 Bits.

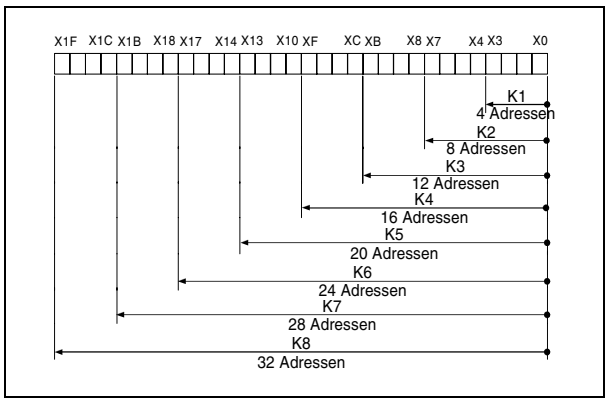


Verarbeitung von Doppelwort-Daten (32 Bit)

- Verwendung von Bit-Operanden

  Bit-Operanden sind in der Lage, Wort-Daten zu verarbeiten. Dazu muss die Anzahl der Bit-Operanden (Adressen) festgelegt werden. Bis zu 32 Bits können in Blöcken mit jeweils 4 Bits verarbeitet werden. Die Länge eines Blocks wird mit K1 bis K8 festgelegt.

K1X0	4 Adressen von X0 bis X3
K2X0	8 Adressen von X0 bis X7
K3X0	12 Adressen von X0 bis XB
K4X0	16 Adressen von X0 bis XF
K5X0	20 Adressen von X0 bis X13
K6X0	24 Adressen von X0 bis X17
K7X0	28 Adressen von X0 bis X1B
K8X0	32 Adressen von X0 bis X1F

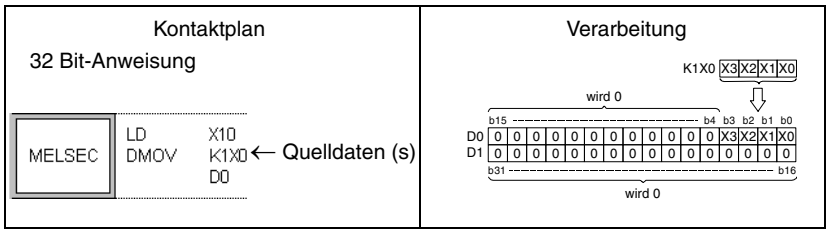


- Angabe der Bit-Blöcke für s

  Bei einer Vorgabe der Blocklänge für Quelldaten s ist der Bereich der als Quelldaten verarbeiteten Werte in der folgenden Tabelle aufgeführt.

Blocklänge	32-Bit-Anweisung
K1 (4 Stellen)	0 bis 15
K2 (8 Stellen)	0 bis 255
K3 (12 Stellen)	0 bis 4095
K4 (16 Stellen)	0 bis 65535
K5 (20 Stellen)	0 bis 1048575
K6 (24 Stellen)	0 bis 16777215
K7 (28 Stellen)	0 bis 268435455
K8 (32 Stellen)	-2147483648 bis 2147483647

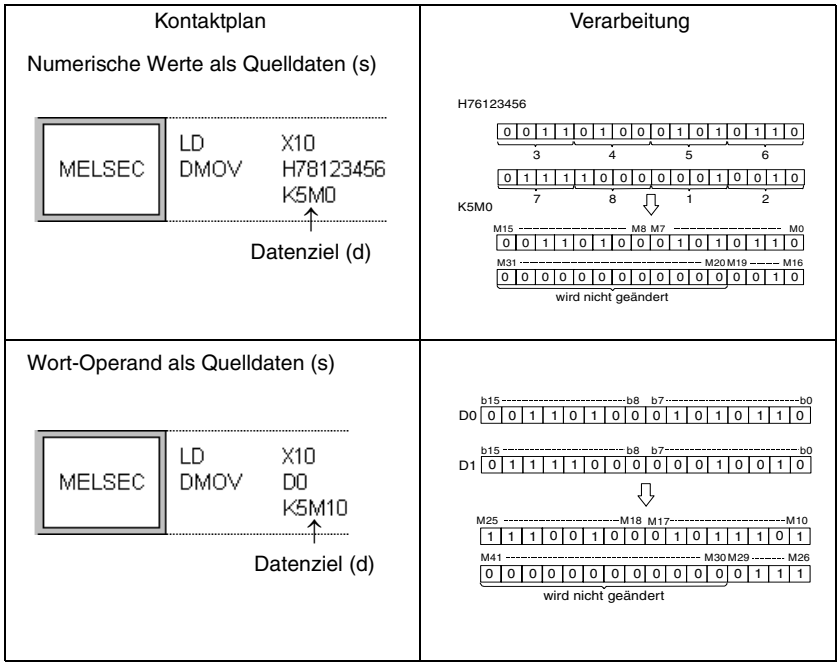
Die nicht benötigten Bit-Adressen werden auf Null gesetzt



HINWEIS *Bei der blockweisen Adressierung von Bit-Operanden kann für die Adresse des ersten Bit-Operanden (Startadresse) ein beliebiger Wert angegeben werden.*

- Angabe der Bit-Blöcke für d

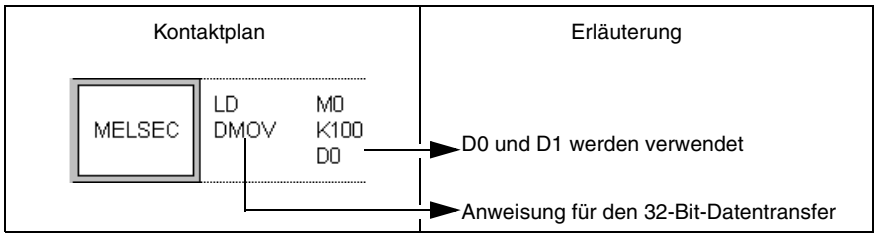
  Die Vorgabe der Blocklänge für Zieldaten (d) legt den Adressenbereich fest, in den die Daten gesetzt werden sollen. Die Bit-Adressen oberhalb der angegebenen Adressen bleiben unberücksichtigt.



- Verwendung von Wort-Operanden

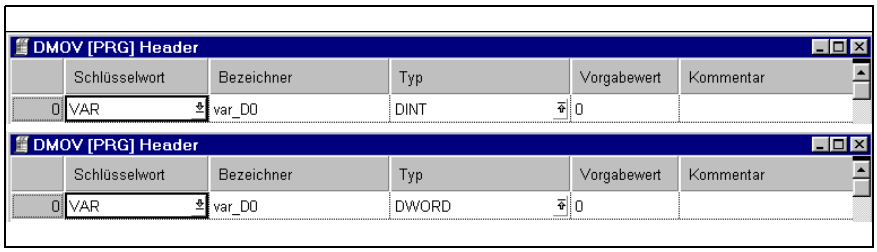
  Doppelwort-Operanden umfassen zwei 16-Bit-Operanden.
  Je nach Programmiersoftware und gewähltem Editor werden Doppelwort-Operanden unterschiedlich programmiert.

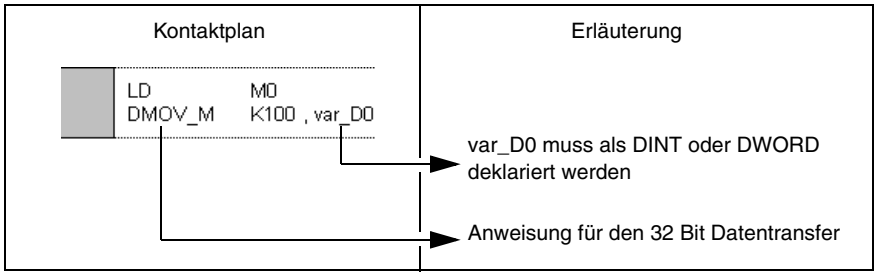
- Im MELSEC-Editor des GX IEC Developers



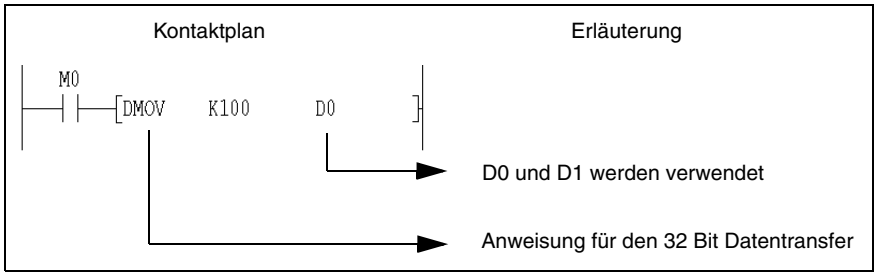
- Im IEC-Editor des GX IEC Developers

  Um einen 32-Bit-Operanden im IEC-Editor des GX IEC Developers verwenden zu können, muss eine Variablendeklaration im Header der Programm-Organisationseinheit (POE) vorgenommen werden.
  Die Datentypen DWORD und DINT sind 32-Bit-Typen.



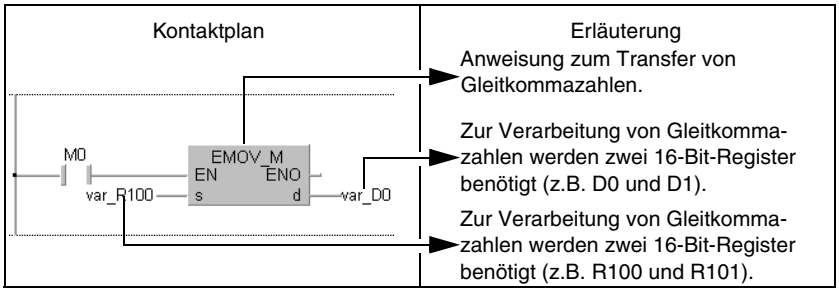


- Im Editor des GX Developers



Verarbeitung von Daten des Typs REAL

Daten des Typs REAL sind 32-Bit-Gleitkommazahlen.
Nur Wort-Operanden sind in der Lage, Gleitkommazahlen zu speichern.
Operanden, die in Anweisungen Gleitkommazahlen verarbeiten, werden mit den unteren 16 Bit adressiert. Die zu speichernde 32-Bit-Gleitkommazahl wird in zwei aufeinanderfolgenden 16-Bit-Registern abgelegt.
Soll mit einer AnA/AnU CPU der Datentyp REAL verarbeitet werden, müssen die entsprechenden erweiterten Anweisungen benutzt werden (siehe Kap. 3.3 „Programmierung der erweiterten Anweisungen“).

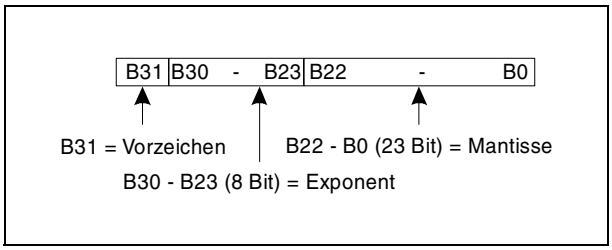


HINWEISE *Beim* GX IEC Developer *wird die Gleitkommazahl mit E□ bezeichnet. Anweisungen zur Verarbeitung von Gleitkommazahlen beginnen mit einem E.*

Zwei Wortoperanden sind nötig, um eine Gleitkommazahl zu speichern. Dazu wird sie in der folgenden Weise zerlegt:

Vorzeichen; $2^{[Exponent]}$; [Mantisse]

Die Bitkonfiguration der Register und deren Bedeutung werden in der folgenden Zeichnung erläutert.



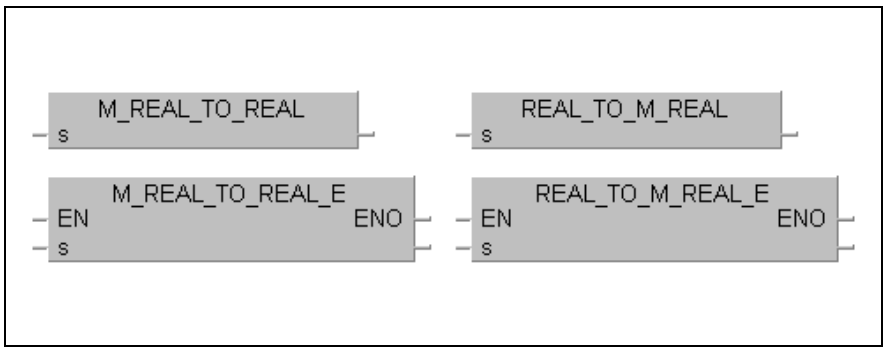
- Vorzeichen der Gleitkommazahl: Das Vorzeichen wird in b 31 gespeichert.
  0 = Positiv
  1 = Negativ
- Exponent: Das n von 2^n wird von b23 bis b30 binär gespeichert. Die Bedeutung des Binärwertes von n wird in der folgenden Zeichnung dargestellt.

b23 bis b30	FFH	FEH	FDH			81H	80H	7FH	7EH			02H	01H	00H
n	Frei	127	126			2	1	0	-1			-125	-126	Frei

Beispiel: Steht in b23 bis b30 binärcodiert 81H, dann ist n=2.

• Mantisse: Mit den 23 Bit von b0 bis b22 lassen sich binär 7 Stellen darstellen (XXXXXX oder 1,XXXXXX).

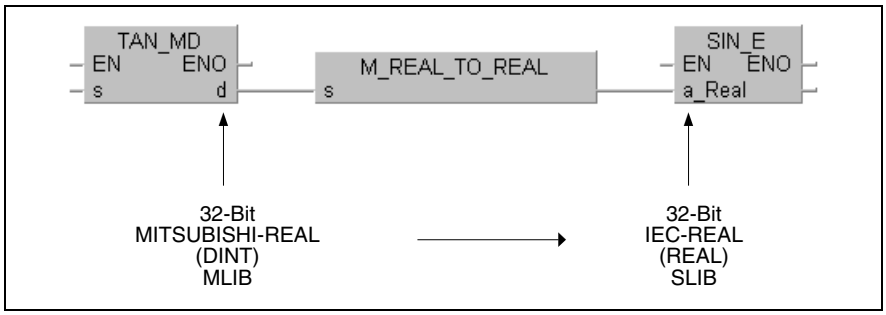
Da die REAL IEC-Funktionen als Ein-/Ausgang den Datentyp REAL, und die MELSEC Befehle den Datentyp DINT verwenden, gibt es folgende Funktionen, um diesen Unterschied auszugleichen .



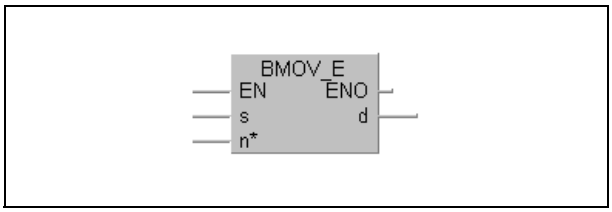
Die Umwandlung des IEC-Datentyps REAL in den MELSEC-Datentyp übernimmt die Anweisung REAL_TO_M_REAL (REAL_TO_M_REAL_E).

Die Umwandlung des MELSEC Datentyps REAL in den IEC-Datentyp übernimmt die Anweisung M_REAL_TO_REAL (M_REAL_TO_REAL_E).

Beispiel: Bei der Verwendung von erweiterten Anweisungen, die den Datentyp REAL verarbeiten, und IEC-Befehlen, ist die REAL-zu-REAL-Umwandlung notwendig.



Im GX IEC Developer dient die BMOV_E-Anweisung zum Ausschalten der Variablen-Prüfung. Sie erzeugt keinen zusätzlichen SPS-Code.



An s kann ein beliebiger Datentyp angegeben werden, auch Arrays sind möglich. Mit n wird festgelegt, wieviele 16-Bit-Informationen kopiert werden.

3.5.2 Array- und Registeradressierung im GX IEC Developer

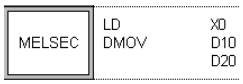
Adressierung von 32-Bit-Registern

Für die Adressierung von 32-Bit-Registern (Datentyp DINT, DWORD) ist eine Variablendeklaration im Header der Programm-Organisationseinheit (POE) erforderlich.

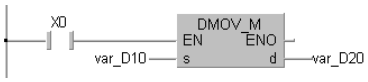
In dem folgenden Beispiel benötigt die DMOV-Anweisung zwei 16-Bit-Register, um ein 32-Bit-Datenwort zu verschieben. Bei der Adressierung im GX Developer und im MELSEC-Editor des GX IEC Developer*s* werden nur die Startregister (im Beispiel D10, D20) angegeben. Das jeweils zweite erforderliche 16-Bit-Register (D11, D21) wird vom Compiler automatisch adressiert.

Im IEC-Editor des GX IEC Developers ist statt der Angabe des Startregisters eine Variable (im Beispiel var_D10, var_D20) mit einem bestimmten Datentyp (im Beispiel DINT(32 Bits)) gemäss dem Header der Anweisung im Header der Programm-Organisationseinheit zu definieren. Für diese Variablen werden vom Compiler intern korrespondierende Adressen vergeben.

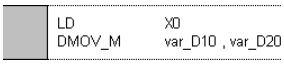
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



Header der DMOV-Anweisung

	Schlüsselwort	Bezeichner	Typ	Vorgabewert	Kommentar
0	VAR_INPUT	s	ANY32	0	
1	VAR_OUTPUT	d	ANY32	0	

Header der Programm-Organisationseinheit (POE)

	Schlüsselwort	Bezeichner	Typ	Vorgabewert	Kommentar
0	VAR	var_D10	DINT	0	
1	VAR	var_D20	DINT	0	

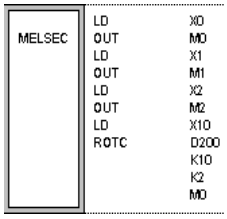
Adressierung von Arrays

Bei der Programmierung von Anweisungen, die ein Array mit Arrayelementen als Ein- oder Ausgangsoperanden (16-Bit-Register) verwenden, sind die Variablen im Header der Programm-Organisationseinheit gemäß dem Header der Anweisung anzugeben.

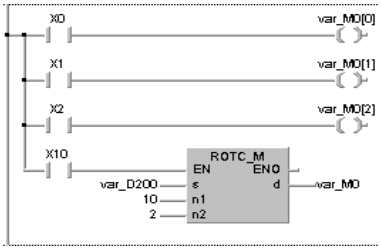
Die Adressierung der einzelnen Arrayelemente erfolgt durch Angabe des Arrays mit der Angabe des Arrayelementes in eckigen Klammern (var_xx[x]).

In der folgenden Abbildung ist die Adressierung mittels Arrays an der Positionieranweisung für Rotationstische (ROTC) dargestellt.

MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste

```
LD      X0
ST      var_M0[0]
LD      X1
ST      var_M0[1]
LD      X2
ST      var_M0[2]
LD      X10
ROTC_M  var_D200 , 10 , 2 , var_M0
```

Header der ROTC-Anweisung

	Schlüsselwort	Bezeichner	Typ	Vorgabewert	Kommentar
0	VAR_INPUT	s	ARRAY [1.. 3] OF ANY16	0,0,0	
1	VAR_INPUT	n1	ANY16	0	
2	VAR_INPUT	n2	ANY16	0	
3	VAR_OUTPUT	d	ARRAY [1.. 8] OF BOOL	8(FALSE)	

Header der Programm-Organisationseinheit (POE)

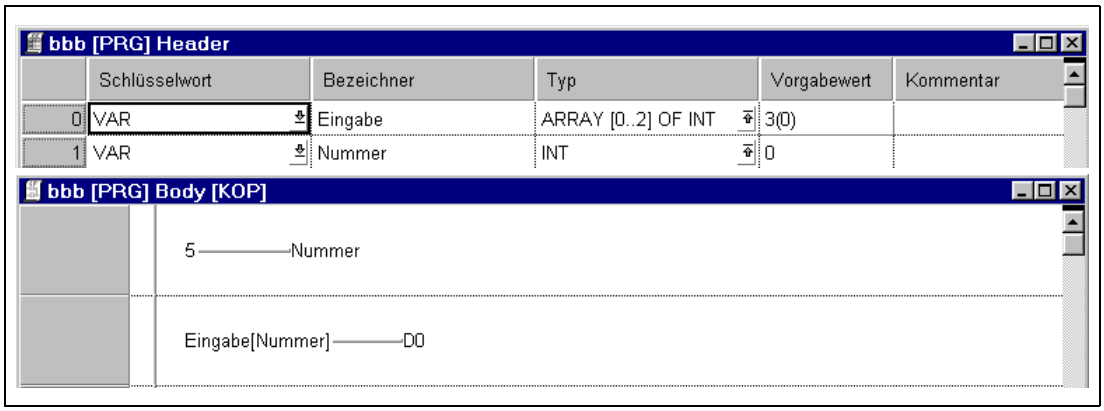
	Schlüsselwort	Bezeichner	Typ	Vorgabewert	Kommentar
0	VAR	var_D200	ARRAY [0..2] OF INT	3(0)	
1	VAR	var_M0	ARRAY [0..7] OF BOOL	8(FALSE)	

Dem Header der ROTC-Anweisung ist zu entnehmen, dass der Eingangsoperandenbereich s aus 3 Arrayelementen des Typs ANY16 und der Ausgangsoperandenbereich aus 8 Arrayelementen des Typs BOOL besteht.

Im GX Developer, im MELSEC-Editor des GX IEC Developers und in MELSEC MEDOC werden für die Ein-/Ausgangsoperandenbereiche s und d jeweils nur die Start-Operanden D200 und M0 angegeben. Der Compiler adressiert für s die Register D200 bis D202 und für d die Merker M0 bis M7.

In den IEC-Editoren müssen für s und d Arrays definiert werden. Das Eingangsarray s ist als var_D200 definiert. Es besteht aus 3 Arrayelementen (var_D200[0] – var_D200[2]) des Typs INT (16-Bit-Integer). Das Ausgangsarray d ist als var_M0 definiert. Es besteht aus 8 Arrayelementen (var_M0[0] – var_M0[7]) des Typs BOOL (Bit). Für diese Variablen werden vom Compiler intern korrespondierende Adressen vergeben.

HINWEIS *Arrays können auch variabel adressiert werden. In diesem Fall wird statt der Angabe des Arrayelementes in eckigen Klammern z.B. [Nummer] angegeben (z.B. Eingabe [Nummer]). „Nummer“ muss im Header der Programm-Organisationseinheit deklariert werden. In das Register „Nummer“ kann dann ein Wert geschoben werden, der mit dem entsprechenden Arrayelement korrespondiert.*



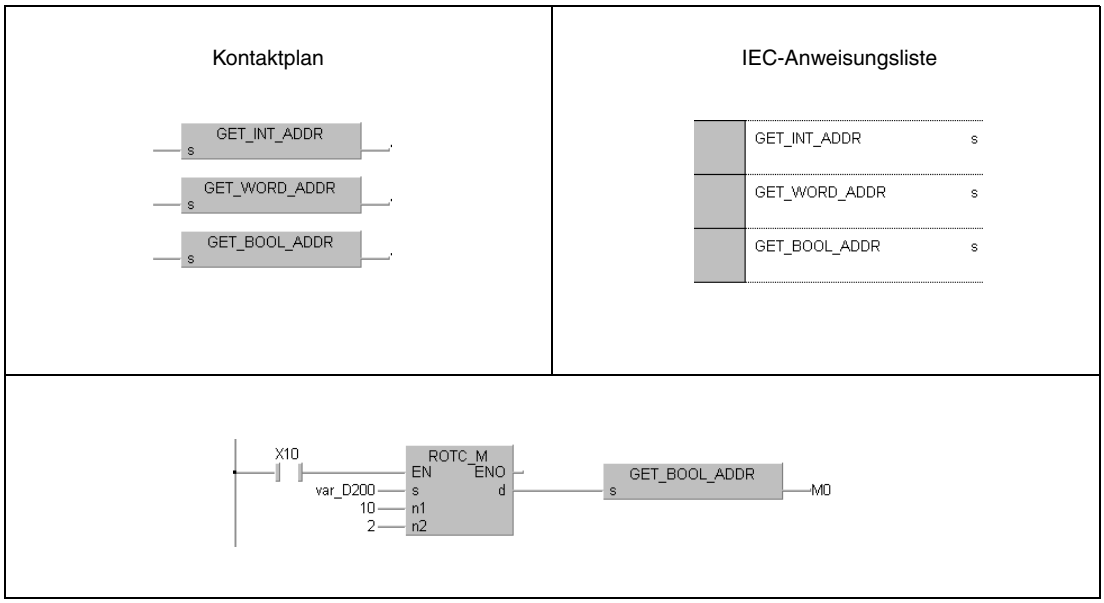
Anweisungen zur Array-/ Startadressenkonvertierung

Der Befehlssatz zur Konvertierung eines Ausgangsarrays in eine Startadresse eines Operandenbereichs besteht aus drei Anweisungen.

Die Anweisung GET_INT_ADDR wandelt ein Ausgangsarray mit Arrayelementen vom Typ INT (16-Bit-Integer) in eine Startadresse eines Operandenbereichs um.

Die Anweisung GET_WORD_ADDR wandelt ein Ausgangsarray mit Arrayelementen vom Typ WORD (16-Bit-Wort) in eine Startadresse eines Operandenbereichs um.

Die Anweisung GET_BOOL_ADDR wandelt ein Ausgangsarray mit Arrayelementen vom Typ BOOL (Bit) in eine Startadresse eines Operandenbereichs um.



Nach der Konvertierung können die Arrayelemente als separate Operanden verarbeitet werden. Dadurch entfällt die Variablendeklaration im Header der Programm-Organisationseinheit.

In dem oben abgebildeten Programm mit der ROTC-Anweisung können durch die Konvertierung statt der Arrayelemente var_M0[0] – var_M0[7] die Merker M0 bis M7 verwendet werden.

Die Adressierung der Operanden der konvertierten Arrayelemente ist beim GX IEC Developer und beim GX Developer identisch.

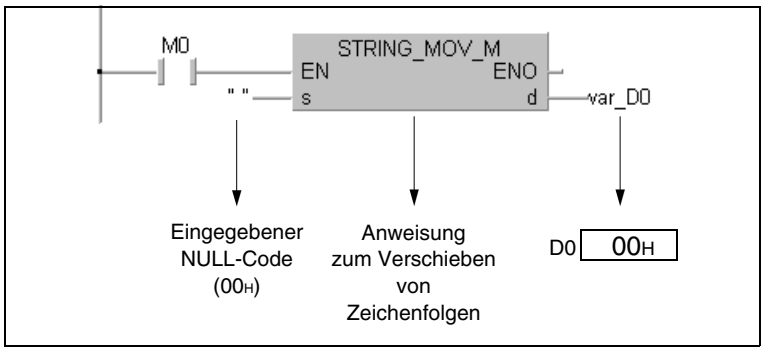
Die Anweisungen konvertieren nur Ausgangsarrays. Eingangsarrays müssen wie vorne beschrieben adressiert und deklariert werden.

3.5.3 Verwendung von Zeichenfolgendaten (STRING)

Der Datentyp STRING ($) verarbeitet Zeichenfolgen.
Zeichenfolgen sind alle eingegebenen Zeichen (max. 50 Zeichen) bis zum NULL-Code (00H).

- Wenn das eingegebene Zeichen der NULL-Code (00H) ist

  Zur Speicherung des NULL-Codes wird ein Datenwort (Register) benötigt.

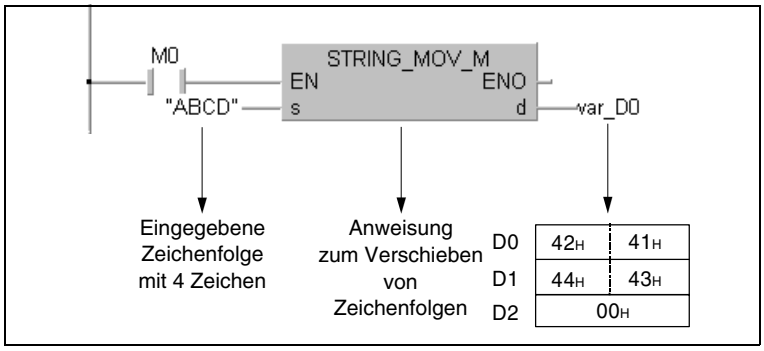


- Wenn die Anzahl der Zeichen in der Zeichenfolge gerade ist

  Um eine Zeichenfolge mit gerader Anzahl von Zeichen zu speichern, wird eine Anzahl von Datenworten (Registern) benötigt, die nach folgender Formel berechnet wird:

  (Anzahl der Zeichen / 2) + 1

  Wenn zum Beispiel die Zeichenfolge „ABCD“ nach D0 verschoben werden soll, werden für die Zeichenfolge die Register D0 bis D1 benötigt, und für den NULL-Code, der das Ende der Zeichenfolge kennzeichnet, wird D2 benötigt.

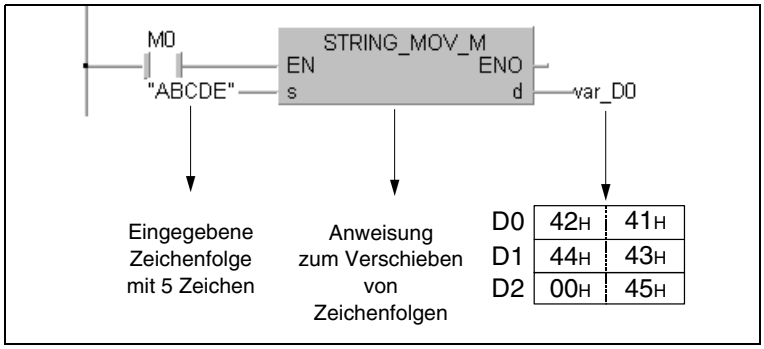


- Wenn die Anzahl der Zeichen in der Zeichenfolge ungerade ist

  Um eine Zeichenfolge mit ungerader Anzahl von Zeichen zu speichern, wird eine Anzahl von Datenworten (Registern) benötigt, die nach folgender Formel berechnet wird:

  (Anzahl der Zeichen / 2)

  Wenn zum Beispiel die Zeichenfolge „ABCDE“ nach D0 verschoben werden soll, werden für die Zeichenfolge die Register D0 bis D2 benötigt. Der NULL-Code, der das Ende der Zeichenfolge kennzeichnet, wird in das höherwertige Byte von D2 geschrieben.



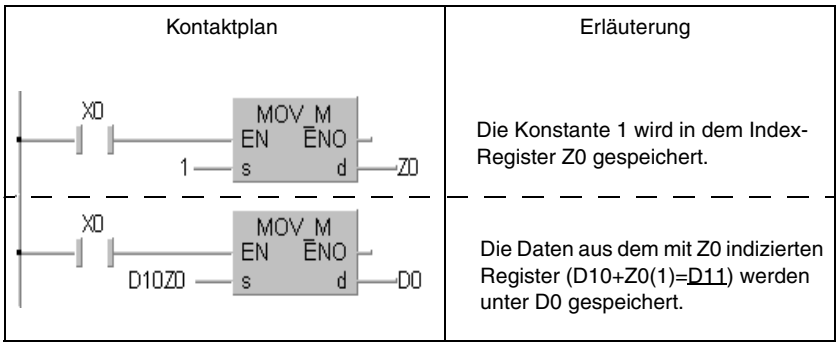
3.6 Index-Vergabe

Da sich die Index-Vergabe bei den CPUs des System Q und der Q-Serie von den CPUs der A-Serie unterscheidet, werden die Besonderheiten der CPU-Typen in den Kapiteln 3.6.1 und 3.6.2 näher erläutert.

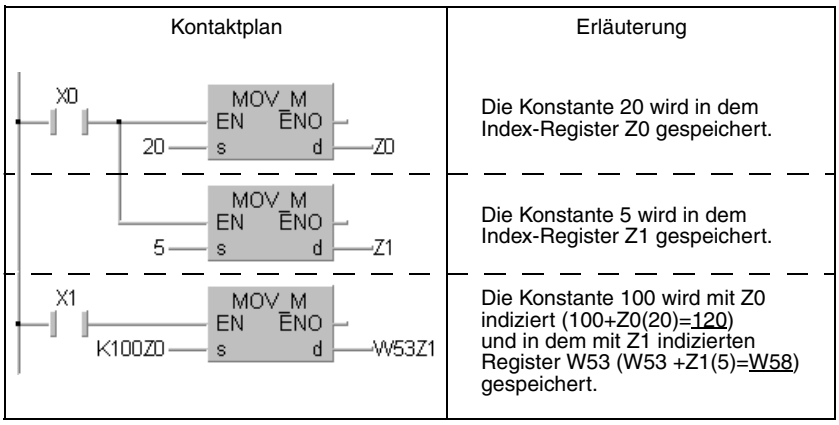
Die Index-Vergabe ist eine indirekte Adressierung eines Operanden durch ein Index-Register. Bei Verwendung der Index-Vergabe in einem Programm erhält der Operand die direkt eingegebene Operandenadresse plus den Inhalt des Index-Registers als Adresse.

Anwendung der Index-Vergabe im Programm

Das Programm in der folgenden Abbildung zeigt ein Beispiel der Index-Vergabe. In der ersten Programmzeile wird dem Index-Register Z0 der Wert 1 zugewiesen. Das Register dient der zweiten Programmzeile als Index für D10. In D0 wird daher der Wert aus D11 (D10Z = D(10+1) = D11) gespeichert.



Die folgende Abbildung enthält ein weiteres Beispiele der Index-Vergabe zur Verdeutlichung der Operandenverarbeitung (Z0=20, Z1=5).



Operanden, die durch die Index-Vergabe adressiert werden können.

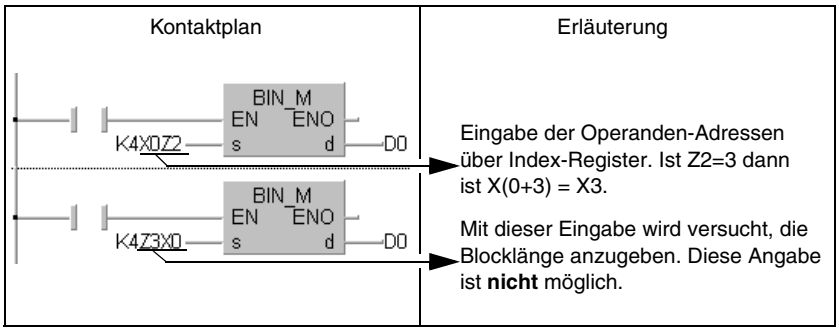
Die Index-Vergabe kann auf Operanden, Kontakte und Spulen angewendet werden. Die Index-Register dienen der indirekten Adressierung eines Operanden und enthalten einen numerischen Wert zwischen -32768 und 32767.

Operanden, die nicht über die Index-Vergabe adressiert werden können.

Operand	Bedeutung
E	Gleitkommazahlen
$	Zeichenfolgen
□.□	Bit-Adressierung von Wortoperanden
FX, FY, FD	Funktionsoperanden
P	Pointer, die als Label verwendet werden
I	Interrupt-Pointer, die als Label verwendet werden
Z	Index-Register
S	Schrittmerker
TV, STV	Sollwerte von Timern
CV	Sollwerte von Countern
N	Nesting-Ebenen
A0	AKKU
A1	AKKU

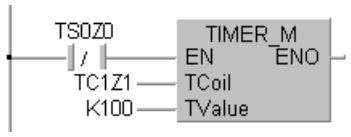
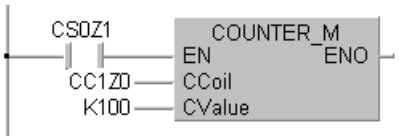
Bit-Daten (außer AnN)

Operanden können auch bei der Blockadressierung indiziert adressiert werden. Die Blocklänge kann nicht über Index-Register beeinflusst werden.

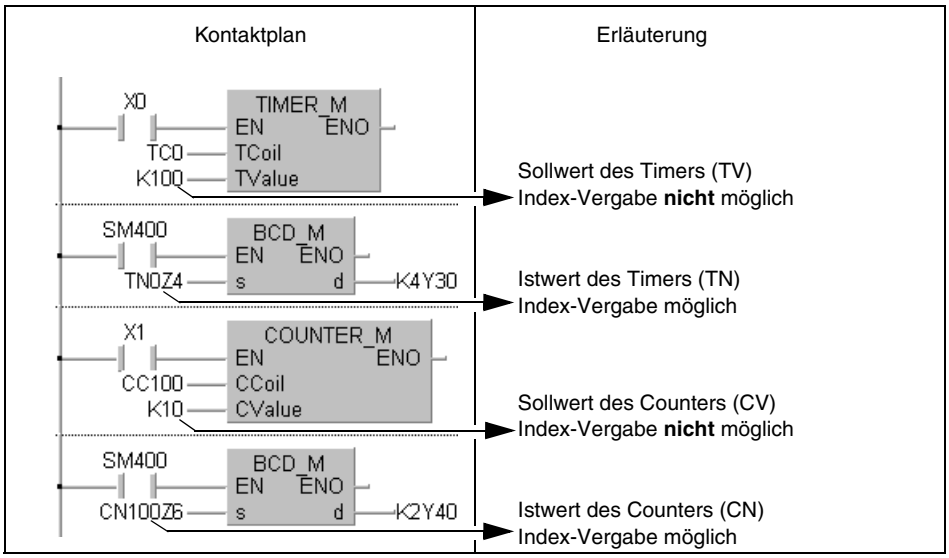


3.6.1 Besonderheiten der Q-CPUs und QnA CPUs

Eine Q-CPU und eine CPU der QnA-Serie besitzt 16 Index-Register (Z0 – Z15).
Die folgende Tabelle gibt die Wertebereiche von Timern und Countern an, die über die Index-Vergabe adressiert werden können.

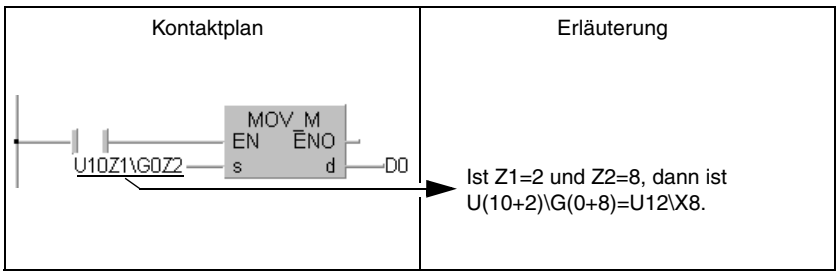
Operand	Bedeutung	Anwendungsbeispiel
TC	Es können nur die Register Z0 und Z1 zur Adressierung von Timer-Kontakten und Spulen verwendet werden.	
CC	Es können nur die Register Z0 und Z1 zur Adressierung von Counter-Kontakten und Spulen verwendet werden.	

HINWEISE *Bei der indizierten Adressierung von Timer- und Counter-Istwerten bestehen keine Beschränkungen.*

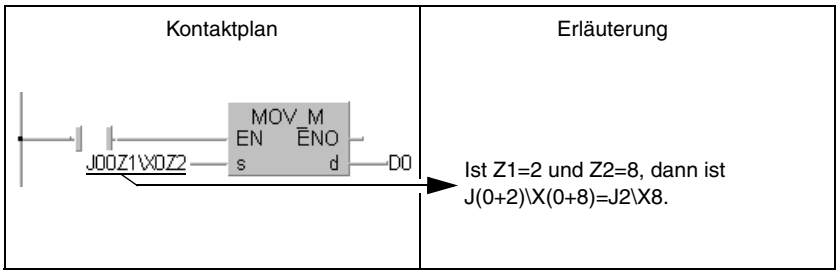


Eine weiterer Unterschied zu den A CPUs besteht in der Möglichkeit, E/A-Adressen und Pufferspeicheradressen, Netzwerknummern und Operandenadressen von Netzwerkmodulen indiziert zu adressieren.

Die folgende Abbildung zeigt die Adressierung von E/A-Adressen und Pufferspeicheradressen in Sondermodulen.



Die folgende Abbildung zeigt die Adressierung von Netzwerknummern und Operandenadressen von Netzwerkmodulen.



HINWEISE *Weiterführende Informationen zum Thema Sonder- und Netzwerkmodule finden Sie im „QnA CPU-Programming Manual (Fundamentals)“, im „QCPU (Q mode) Users Manual (Functions/ programming fundamentals)“ oder in den produktspezifischen Handbüchern.*

3.6.2 Besonderheiten der AnA-, AnAS- und AnU-CPUs

Operandenadressen können im Programm mit einem Index (Z oder V) versehen werden. Zur Kennzeichnung ist der Index mit einem Vorzeichen versehen.

In folgenden Fällen tritt bei der Verarbeitung von Anweisungen ein Verarbeitungsfehler auf.

- Der Adressbereich der Operanden wurde während der Index-Vergabe überschritten. Die Konstanten K und H werden in diesem Fall ausgeschlossen.
- Die Startadresse eines Operandenbereiches überschreitet bei Index-Vergabe den erlaubten Adressbereich.

HINWEISE *Zur Verkürzung der Verarbeitungszeiten überprüfen die CPUs der AnA-, AnAS-, und AnU-Serie nicht die Operandenadressen bei Index-Vergabe. Aus diesem Grund werden Fehler in Verbindung mit der Index-Vergabe nicht als Verarbeitungsfehler erkannt.*

Tritt ein Fehler in Verbindung mit der Index-Vergabe auf, können sich ungewollt Operandendaten ändern.

Programme, die eine Index-Vergabe enthalten, müssen daher mit größter Sorgfalt geschrieben werden!

In Verbindung mit einer AnA, AnAS oder AnU CPU ist die Index-Vergabe auch bei Bit-Operanden möglich, die in einer LD-, OUT- oder ähnlichen Anweisung verwendet werden.

Speichern von 32-Bit-Daten in Index-Registern

32-Bit-Daten können in die erweiterten Index-Register (Z1 bis Z6 und V1 bis V6) einer AnA bzw. AnU CPU gespeichert werden. Folgende Index-Register müssen hierzu paarweise genutzt werden:

Z1 und V1

Z2 und V2

Z3 und V3

Z4 und V4

Z5 und V5

Z6 und V6

Zn enthält die niederwertigen 16 Bit, Vn enthält die höherwertigen 16 Bit. In einer 32-Bit-Anweisung darf nur der Operand Z angegeben werden. Wird der Operand V angegeben, kann das Programm nicht verarbeitet werden.

32-Bit-Anweisungen können ausschließlich in den oben aufgeführten Registerpaarungen gespeichert werden. Andere Kombinationen sind nicht zulässig. Wird ein Operand eines Registerpaars zur Index-Vergabe in einer Anweisung verwendet, werden die Daten in diesem Register als 16-Bit-Daten zur Index-Vergabe verarbeitet.

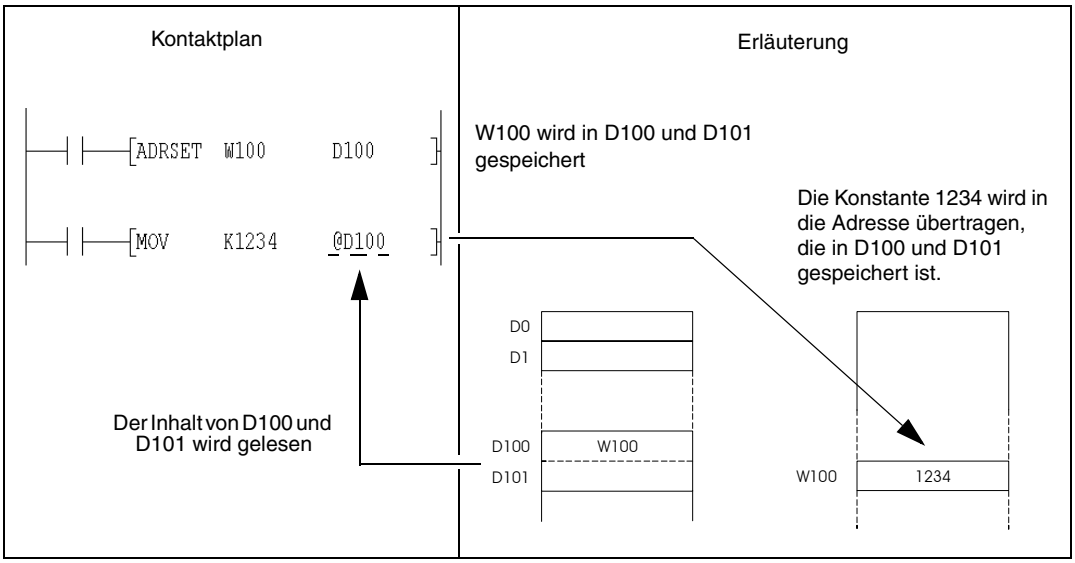
3.7 Indirekte Adressierung (Nur GX Developer)

Bei der indirekten Adressierung wird eine Operandenadresse in einem Wort-Operanden abgelegt. Im Programm wird der Operand, mit dem die Operation ausgeführt werden soll, nicht mehr direkt angesprochen, sondern über den Operanden, der die gespeicherte Adresse enthält. Die indirekte Adressierung kann dann eingesetzt werden, wenn die Index-Vergabe unzureichend ist.

Im Programm wird der Operand, in dem die Adresse des indirekt angespochenen Operanden gespeichert ist, mit dem Zeichen „@“ gekennzeichnet. Zum Beispiel wird durch die Angabe von „@D100“ der Inhalt von D100 und D101 als Adresse verwendet.

Mit der ADRSET-Anweisung wird die Operandenadresse, die indirekt angesprochen werden soll, gespeichert.

HINWEIS *Die ADRSET-Anweisung kann bei der Programmierung mit dem* GX IEC Developer *nicht verwendet werden.*

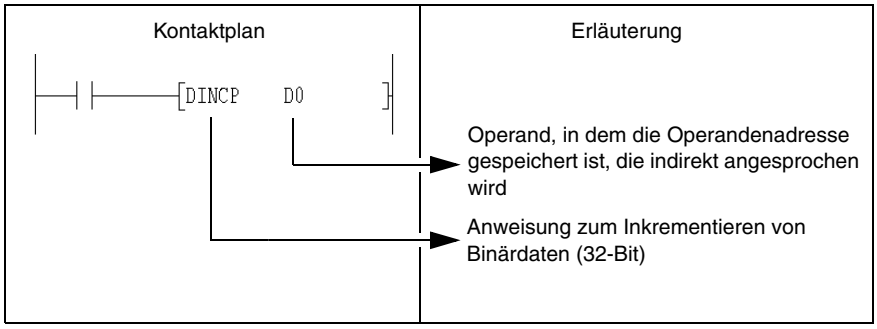


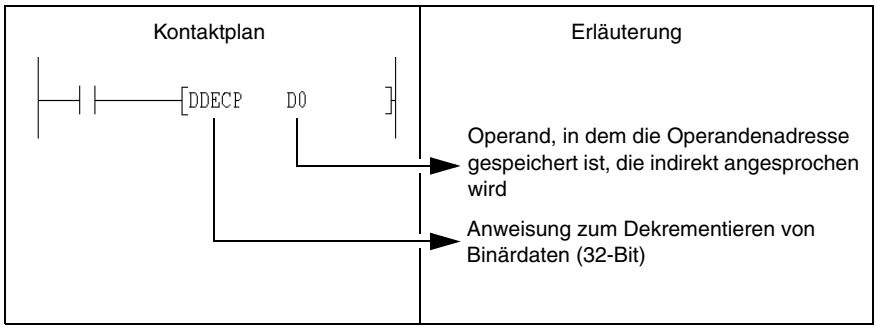
In der folgenden Tabelle sind die Operanden aufgeführt, die indirekt adressiert werden können.

Operanden		Indirekte Adressierung	Beispiel zur indirekten Adressierung
Interne Operanden (System, Anwender)	Bit	nicht möglich	—
	Wort	möglich	• @D100 • @D100Z2 (Index-Vergabe)
MELSECNET/10	Bit	nicht möglich	—
	Wort	möglich (Die Operandenadresse kann nicht mit der ADRSET-Anweisung gespeichert werden.)	• @J1\W10 • @J1Z1\W10Z2 (Index-Vergabe)
Sondermodule			• @U10\G0 • @U10Z1\G0Z2 (Index-Vergabe)
Index-Register Zn		nicht möglich	—
File-Register		möglich	• @R0, @ZR20000 • @R0Z1, @ZR20000Z1 (Index-Vergabe)
Nesting-Ebenen		nicht möglich	—
Pointer			—
Konstanten			—
Andere			—

HINWEIS *Weiterführende Informationen zum Thema Operanden finden Sie im „QnA CPU-Programming Manual (Fundamentals)“ oder „QCPU (Q mode) Users Manual (Functions/programming fundamentals)“*

HINWEIS *Zum Speichern der Operandenadresse zur indirekten Adressierung werden zwei Wörter verwendet. Deshalb müssen, wenn eine gespeicherte Adresse durch Rechenoperationen erhöht oder erniedrigt werden soll, 32-Bit Daten addiert oder subtrahiert werden. Im folgenden Beispiel wird der Operand, der die Adresse des indirekt adressierten Operanden aufnimmt, durch 32-Bit-Anweisungen inkrementiert und dekrementiert und so die Adresse des indirekt adressierten Operanden um 1 erhöht bzw. vermindert.*





3.8 Verarbeitungsfehler

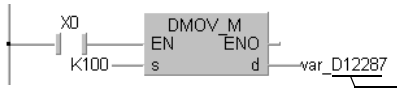
Verarbeitungsfehler treten in den folgenden Fällen auf:

- Bei Zutreffen der Fehlerbedingungen, die unter dem Punkt „Fehlerquellen“ der einzelnen Anweisungen erläutert sind, tritt eine Fehlermeldung auf.
- Bei Verwendung eines Pufferregisters ist kein Sondermodul an der vorbestimmten E/A-Adresse angeschlossen.
- Bei Verwendung eines Link-Operanden existiert das entsprechende Netzwerk nicht.
- Bei Verwendung eines Link-Operanden ist das Netzwerkmodul mit der vorbestimmten E/A-Adresse nicht angeschlossen.

HINWEIS *Wenn in den Parametern ein File-Register definiert wurde, aber keine Speicherkarte (nur Q/QnA-CPUs) installiert ist, erscheint eine Fehlermeldung (2401 = File Set Error). Wenn auf ein File-Register zugegriffen wurde, obwohl in den Parametern keine File-Register definiert wurden, erscheint keine Fehlermeldung. Falls das File-Register ausgelesen wird, erscheint der Code „FFFF$_H$“ .*

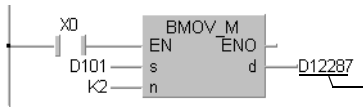
3.8.1 Überprüfung des Operandenbereichs

- Verwenden Anweisungen Operanden mit fester Länge (MOV, DMOV, usw.), wird der Operandenbereich nicht überprüft.
  In den Fällen, in denen der dazu gehörige Adressbereich überschritten wird, werden die zu schreibenden Daten in ein freies Register geschrieben.
  Werden beispielsweise 12k Adressen zugewiesen, erscheint so lange keine Fehlermeldung, bis die Registeradresse D12287 überschritten wurde.

Kontaktplan	Erläuterung
	D12287 und D12288 werden in diesem Beispiel angesprochen, aber D12288 existiert nicht. Es wird ein freies Register mit den Inhalten von D12288 überschrieben.

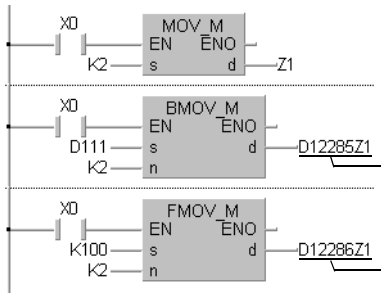
Auch bei Verwendung der indizierten Adressierung wird die Überprüfung des Operandenbereichs nicht durchgeführt.

- Verwenden Anweisungen Operanden mit variabler Länge, wird die Überprüfung des Operandenbereichs durchgeführt (BMOV, FMOV und andere Anweisungen, die Startadressen verwenden). In den Fällen, in denen der dazu gehörige Adressbereich überschritten wird, tritt eine Fehlermeldung auf. Werden beispielsweise 12k Adressen zugewiesen, erscheint die Fehlermeldung erst bei Überschreiten des Registeradresse D12287.

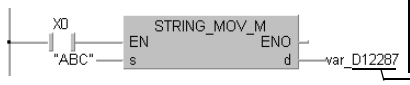
Kontaktplan	Erläuterung
	D12287 und D12288 werden in diesem Beispiel angesprochen, D12288 existiert jedoch nicht. Es wird eine Fehlermeldung ausgegeben.

Die Überprüfung des Operandenbereiches wird auch durchgeführt, wenn eine indizierte Adressierung vorliegt.

Es tritt keine Fehlermeldung auf, wenn die Startadresse des Operanden den Adressbereich überschreitet.

Kontaktplan	Erläuterung
	D12287 und D12288 werden in diesem Beispiel angesprochen, D12288 existiert jedoch nicht. Es wird eine Fehlermeldung ausgegeben. Die Startadresse D12289 überschreitet den zulässigen Bereich. Die Daten werden ab dem Register W0 ohne Ausgabe einer Fehlermeldung gespeichert.

- Da Zeichenfolgen variable Längen aufweisen, wird eine Überprüfung des Operandenbereichs durchgeführt. In den Fällen, in denen der entsprechende Operandenbereich überschritten wird, erfolgt eine Fehlermeldung.
  Werden beispielsweise 12k Adressen zugewiesen, erscheint so lange keine Fehlermeldung, bis die Registeradresse D12287 überschritten wird.

Kontaktplan	Erläuterung
	D12287 und D12288 werden in diesem Beispiel angesprochen, D12288 existiert aber nicht. Es wird eine Fehlermeldung ausgegeben.

- Die Überprüfung des Operandenbereichs wird bei indizierter Adressierung der Direktausgabe (DY) durchgeführt.

3.8.2 Überprüfung der Operandendaten

Bei Verwendung von binären Daten

- Überschreitet das Verarbeitungsergebnis den Wertebereich, tritt keine Fehlermeldung auf. Das Carry-Flag (Überlauf-Flag) wird in diesem Fall nicht gesetzt.

Bei Verwendung von BCD-Daten

- Jede Stelle der BCD-Werte (0 bis 9) wird überprüft.
  Überschreitet eine einzelne Stelle den Bereich von 0 bis 9 (A bis F), tritt eine Fehlermeldung auf.
- Überschreitet das Verarbeitungsergebnis den Wertebereich, tritt keine Fehlermeldung auf. Das Carry-Flag (Überlauf-Flag) wird in diesem Fall nicht gesetzt.

Bei Verwendung von Gleitkommazahlen

Verarbeitungsfehler treten in folgenden Fällen auf:

- Der Wert der Gleitkommazahl nimmt den Wert 0 an
- Der absolute Wert der Gleitkommazahl unterschreitet den Wert $1,0 \times 2^{-127}$
- Der absolute Wert der Gleitkommazahl übererschreitet den Wert $1,0 \times 2^{129}$

Bei Verwendung von Zeichenfolgen

Es wird keine Überprüfung der Operandendaten durchgeführt.

3.9 Ausführungsbedingungen der Anweisungen

3.9.1 Eingangsbedingung

Die folgenden 4 Ausführungsbedingungen existieren für die Ausführung der Anweisungen:

- Ausführung ohne Bedingungen

  Die Anweisungen werden ohne Berücksichtigung des Signalzustandes der Operanden ausgeführt.
  Beispiel: LD X0, OUT Y10

- Ausführung bei gesetzter Bedingung

  Die Anweisungen werden für die Setzdauer der Ausführungsbedingung ausgeführt.
  Beispiel: MOV, FROM

- Ausführung bei ansteigender Flanke

  Die Anweisungen werden bei ansteigender Flanke (Signalwechsel von 0 nach 1) der Ausführungsbedingung ausgeführt.
  Beispiel: PLS, MOVP

- Ausführung bei abfallender Flanke

  Die Anweisungen werden bei abfallender Flanke (Signalwechsel von 1 nach 0) der Ausführungsbedingung ausgeführt.
  Beispiel: PLF

Für die meisten Anweisungen existieren zwei Arten der Ausführung:

- bei gesetzter Ausführungsbedingung
- bei ansteigender Flanke der Ausführungsbedingung

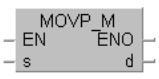
Bei gesetzter Ausführungsbedingung wird die Anweisung so lange ausgeführt, wie die Ausführungsbedingung gesetzt ist. Diese Anweisungen sind nicht besonders gekennzeichnet.

Beispiel: MOV_M/ MOV

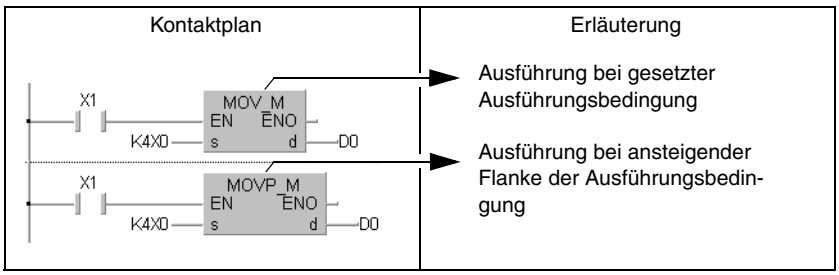
Kontaktplan	MELSEC-Anweisungsliste
	

Bei Auswertung der ansteigenden Flanke der Ausführungsbedingung wird die Anweisung nur dann ausgeführt, wenn ein Signalwechsel von 0 nach 1 stattfindet.

Beispiel: MOVP_M/ MOVP

Kontaktplan	MELSEC-Anweisungsliste
	

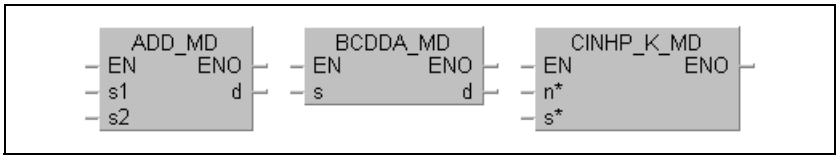
Das folgende Beispiel zeigt die Ausführung der MOV-Anweisung bei gesetzter Ausführungsbedingung und die Ausführung bei ansteigender Flanke der Ausführungsbedingung.



3.9.2 EN-Eingang und ENO-Ausgang

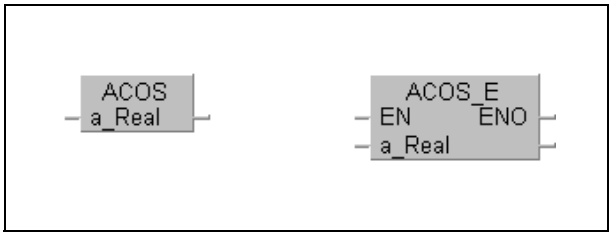
Alle in diesem Handbuch beschriebenen Anweisungen gehören im GX IEC Developer zur Herstellerbibliothek. Diese Anweisungen verfügen zusätzlich zu den Eingangs- und Ausgangsvariablen immer über einen EN-Eingang und einen ENO-Ausgang.

Die folgende Abbildung zeigt einige MELSEC-Anweisungen aus der Herstellerbibliothek des GX IEC Developers.



In der IEC-Standardbibliothek erscheinen fast alle Anweisungen doppelt. Sie unterscheiden sich nur durch die Endung „_E“. Diese Anweisungen haben einen EN-Eingang und einen ENO-Ausgang.

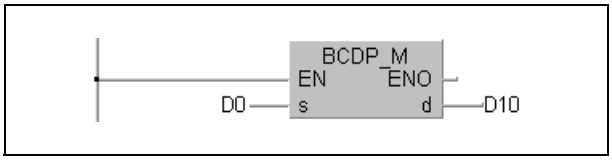
Die folgende Abbildung zeigt zwei IEC-Anweisungen aus der Standardbibliothek des GX IEC Developers.



Die folgenden Beispiele zeigen die unterschiedliche Verarbeitung der Anweisung mit und ohne EN-Ein- und ENO-Ausgänge.

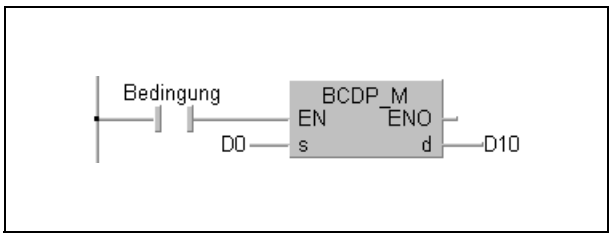
Beispiel 1: Ohne zusätzliche Verküpfung

Ohne zusätzliche Verknüpfung ist die Ausführungsbedingung der Anweisung ständig gesetzt.



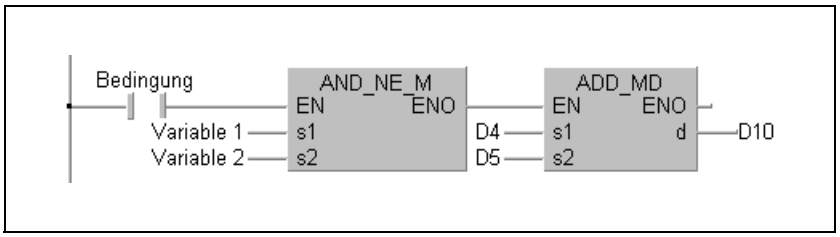
Beispiel 2: Verknüpfung mit einem Kontakt

Wird der EN-Eingang mit einem Kontakt verknüpft, wird die Anweisung bei Zutreffen der Bedingung ausgeführt.



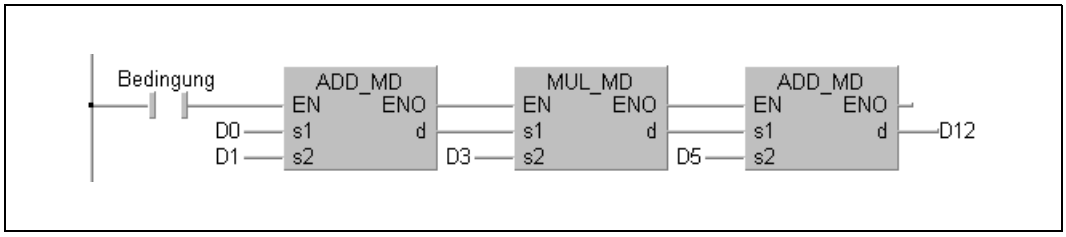
Beispiel 3: Verknüpfung mit einem Operationsergebnis

Wird das boolsche Ergebnis einer Rechenoperation auf den EN-Eingang gelegt, erfolgt die Ausführung der Anweisung nur dann, wenn das Ergebnis der Rechenoperation TRUE lautet.



Beispiel 4: Verknüpfung mit der vorherigen Anweisung

Wird der EN-Eingang an den ENO-Ausgang der vorherigen Anweisung angeschlossen, werden die Anweisungen nur dann ausgeführt, wenn die Bedingung zutrifft.



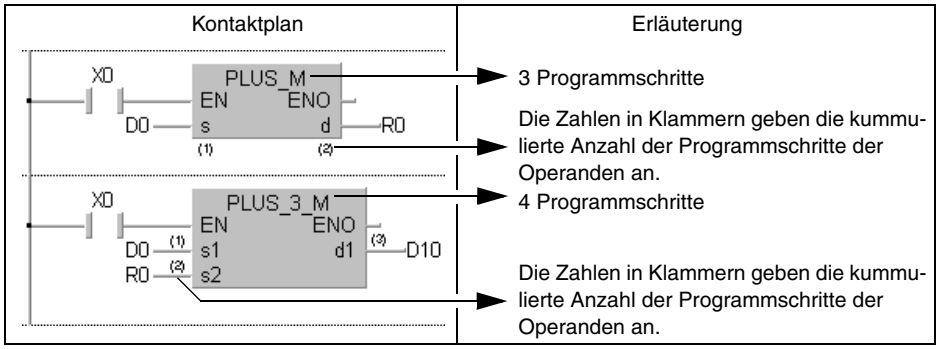
HINWEIS *Der ENO-Ausgang muss nicht zwingend angeschlossen werden. Das Signal am EN-Eingang wird auf den ENO-Ausgang durchgeschleift. Ist der EN-Eingang „TRUE“, ist auch der ENO-Ausgang „TRUE“.*

3.10 Anzahl der Programmschritte

Um die Speicherkapazität für Programmschritte im internen Speicher, ROM oder RAM-Speicher der Speicherkarten und Speicherkassetten nicht zu überschreiten, ist eine Berechnung der Gesamtschritte eines Programms erforderlich. In folgenden Kapiteln wird die Berechnung der Schritte der Anweisungen für die System Q, OnA und A CPUs beschrieben.

3.10.1 Bei einer System Q- oder QnA-CPU

Die Anzahl der Programmschritte einer Anweisung richtet sich nach der Anzahl der Basisschritte. Die meisten Anweisungen benötigen für ihre Ausführung nur eine Anzahl von Basisschritten. Die Anzahl Basisschritte ist abhängig von der Anzahl der verwendeten Operanden plus 1.
Das folgende Beispiel zeigt die Berechnung der Basisschrittzahl der „ PLUS-Anweisung“.



- Die Anzahl der Programmschritte bei Verwendung von Eingangs- und Ausgangsanweisungen:
  Die Anzahl der Programmschritte bei Verwendung der Eingangsanweisungen (LD, LDI, AND, ANI, OR, ORI) ist abhängig von den verwendeten Operanden.
  Werden interne Operanden oder File-Register (R0 bis R32767) verwendet, beträgt die Anzahl 1.
  Werden direkt adressierbare Eingänge (DX) verwendet, beträgt die Anzahl 2.
  Bei der Verwendung anderer Operanden beträgt die Anzahl 3.

  Die Anzahl der Programmschritte bei Verwendung der Ausgangsanweisungen (LDP, LDF, ANDP, ANDF, ORP, ORF) ist abhängig von den verwendeten Operanden.
  Werden interne Operanden oder File-Register (R0 bis R32767) verwendet, beträgt die Anzahl 2.
  Werden direkt adressierbare Eingänge (DX) verwendet, beträgt die Anzahl 3.
  Bei der Verwendung anderer Operanden beträgt die Anzahl 4.

- Die Anzahl der Programmschritte bei Verwendung einiger Transferanweisungen:

Operanden, die die Anzahl der Programmschritte erhöhen	Addierte Schritte	Beispiel
Operanden von Sondermodulen	1	MOV U4\G10 D0
Link-Operanden		MOV J3\B20 D0
Seriell adressierte File-Register		MOV ZR123 D0
32 Bit-Konstanten		DMOV K123 D0
Gleitkommazahlen als Konstanten		EMOV E0.1 D0
Zeichenfolgen	Bei ungerader Anzahl: (Anzahl der Zeichen/2)-1 Bei gerader Anzahl: Anzahl der Zeichen/2	$MOV „123“ D0

In Fällen, in denen mehrere Faktoren zutreffen, addiert sich die Anzahl der Schritte. Wenn beispielsweise MOV U1\G10 ZR123 programmiert ist, wird 1 Schritt für den Pufferspeicher und 1 Schritt für das seriell adressierte File-Register addiert, womit sich eine Gesammtschrittzahl von 2 ergibt.

3.10.2 Bei einer AnA, AnAS und AnU CPU

In Verbindung mit einer AnA-, AnAS- oder AnU-CPU sind eine Reihe von Besonderheiten zu beachten, auf die in diesem Abschnitt näher eingegangen werden soll.

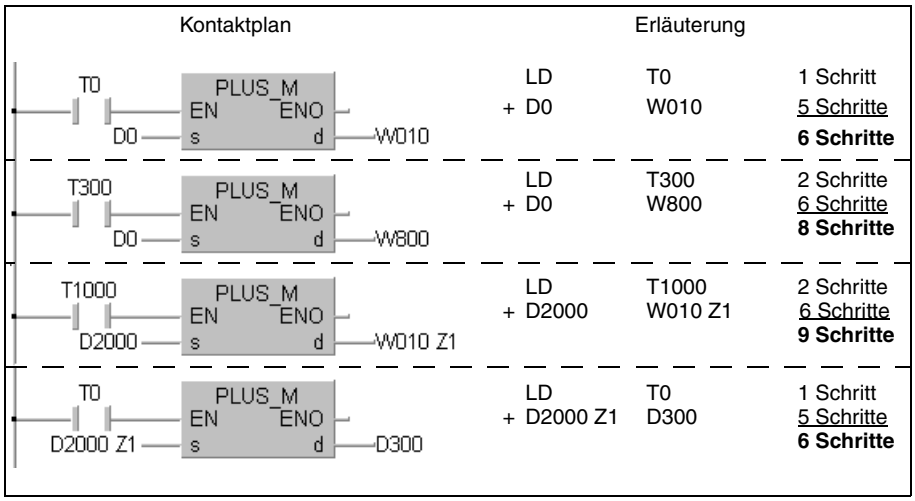
Die Anzahl der Schritte nimmt um 1 zu, wenn eine der in der folgenden Tabelle aufgeführten Operandenadressen (erweiterter Bereich der AnA-Serie) in einer Anweisung angesprochen wird

Operanden	Adressbereich
Merker M, L, S	2048 bis 8191
Timer T	256 bis 2047
Counter C	256 bis 1023
Link-Merker B	400 bis FFF
Datenregister D	1024 bis 6143
Link-Register W	400 bis FFF

Erfolgt für einen Operanden aus dem erweiterten Adressenbereich eine Index-Vergabe mit einem erweiterten Index-Register, nimmt die Anzahl der Schritte ebenfalls um 1 zu.

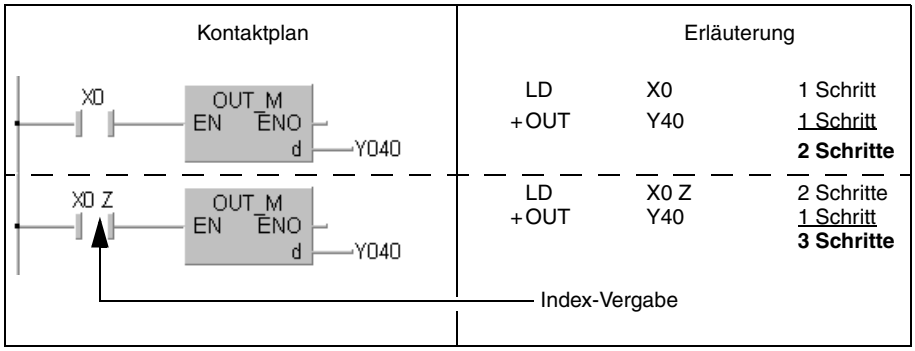
Die folgende Abbildung enthält einige Beispiele zur Berechnung von Programmschritten. Das erste Beispiel zeigt die Zusammensetzung der Schritte bei der Programmierung von Anweisungen aus dem normalen Adressenbereich.

Die darauf folgenden Beispiele zeigen die Zusammensetzung von Programmschritten bei der Verwendung von Operanden aus dem erweiterten Adressenbereich.



Bei einer Index-Vergabe in einer 1-Schritt-Anweisung (wie z.B. LD oder OUT) nimmt die Anzahl der Schritte um 1 zu.

Das folgende Beispiel zeigt die Unterschiede zwischen der Programmierung mit und ohne Index. Die Anzahl der Schritte erhöht sich auch dann nur um 1, wenn die Index-Vergabe mit einem erweiterten Index-Register (Z1 bis Z6, V1 bis V6) erfolgt.



4 Aufbau der Kapitel

Das vorliegende Kapitel enthält eine Einleitung zu den Kapiteln 5 bis 9 und beschreibt die Formate und den Aufbau der Erläuterungen zu den Anweisungen der MELSEC A-/Q-Serie und des System Q.

Wie die folgende Abbildung zeigt, beginnt jedes der oben angegebenen Kapitel mit einer Tabelle, in der die Gliederung der Anweisungen, die in diesem Kapitel behandelt werden, aufgeführt und erläutert wird.

6 Applikationsanweisungen Teil I

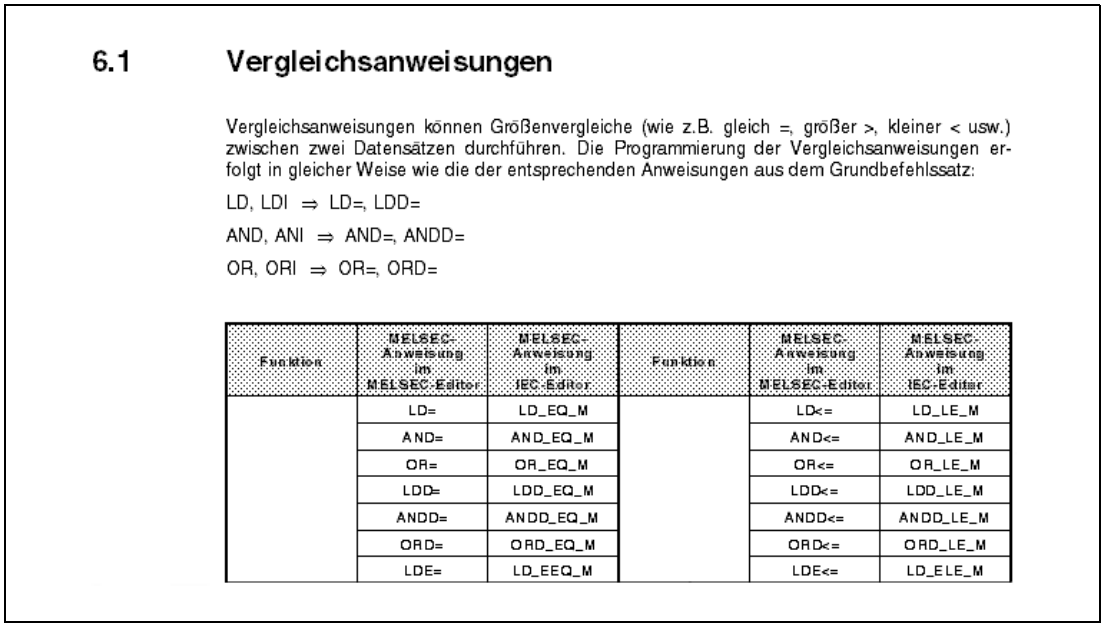
Die Applikationsanweisungen Teil I beinhalten Anweisungen, die numerische 16- und 32-Bit-Daten, Gleitkommazahlen und Zeichenfolgen verarbeiten können. In erster Linie werden mit Hilfe dieser Applikationsanweisungen Vergleichs- und Rechenoperationen durchgeführt.

Einteilung	Bedeutung
Vergleichsanweisungen	Datenvergleich, wie z.B. =, >, ≥ usw.
Arithmetikanweisungen	Addition, Subtraktion, Multiplikation, Division, von BIN- und BCD-Daten, Gleitkommazahlen und BIN-Datenblöcken, Verknüpfung von Zeichenfolgen, Inkrement, Dekrement
Konvertierungsanweisungen	Datenkonvertierung wie z.B. BCD → BIN und BIN → BCD
Transferanweisungen	Übertragung, Austausch und Negation von Daten
Programmverzweigungs-anweisungen	Sprung, Unterprogrammaufruf
Anweisungen zum Interrupt-Programmaufruf	Interrupt-Programmaufruf
Datenaktualisierungs-anweisungen	Link-Refresh und E/A-Schnittstellenauffrischung
Sonstige Anweisungen	Ein-/Zweiphasiger Auf-/Abwärtszähler, programmierbare Timer, Sonderfunktionstimer, Positionieranweisung, Rampensignal, Impulszähler, Impulsausgang, Puls-Weiten-Modulation, Eingabematrix

Jeder Gliederungspunkt wird in dem Kapitel aufgeführt und mit Programmbeispielen erläutert.

4.1 Übersicht über die Anweisungen

Jeder Gliederungspunkt beginnt mit einer Tabelle, in der alle Anweisungen aufgeführt sind, die in diesem Absatz erläutert werden. Wie die folgende Abbildung zeigt, wird die Schreibweise der Anweisungsvarianten im MELSEC- und im IEC-Editor dargestellt.



6.1 Vergleichsanweisungen

Vergleichsanweisungen können Größenvergleiche (wie z.B. gleich =, größer >, kleiner < usw.) zwischen zwei Datensätzen durchführen. Die Programmierung der Vergleichsanweisungen erfolgt in gleicher Weise wie die der entsprechenden Anweisungen aus dem Grundbefehlssatz:

LD, LDI ⇒ LD=, LDD=

AND, ANI ⇒ AND=, ANDD=

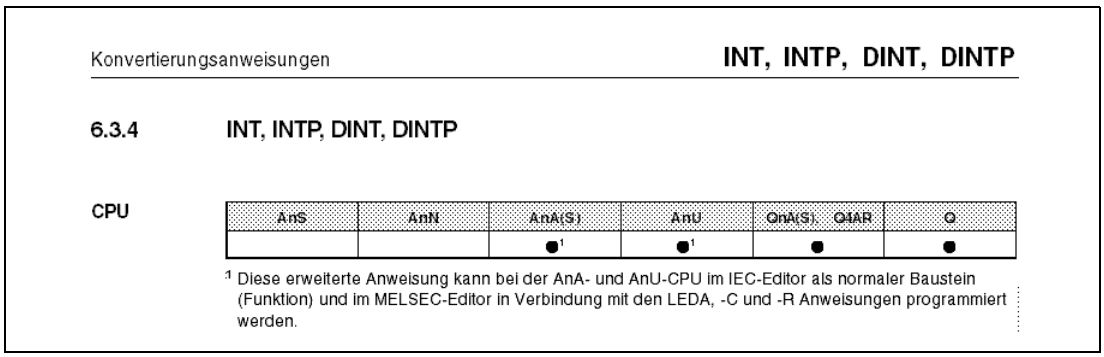
OR, ORI ⇒ OR=, ORD=

Funktion	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor	Funktion	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor
	LD=	LD_EQ_M		LD<=	LD_LE_M
	AND=	AND_EQ_M		AND<=	AND_LE_M
	OR=	OR_EQ_M		OR<=	OR_LE_M
	LDD=	LDD_EQ_M		LDD<=	LDD_LE_M
	ANDD=	ANDD_EQ_M		ANDD<=	ANDD_LE_M
	ORD=	ORD_EQ_M		ORD<=	ORD_LE_M
	LDE=	LD_EEQ_M		LDE<=	LD_ELE_M

Wenn im GX IEC Developer die Auswahlmöglichkeit besteht, sollte immer die IEC-Anweisung verwendet werden.

4.2 Die CPU-Tabelle

Die Absätze, in denen die Anweisungen erläutert werden, beginnen mit einer Tabelle, in der dargestellt wird, auf welcher CPU (AnS, AnN, AnA, AnAS, AnU, QnA, QnAS, Q4AR, Q) die Anweisung ausgeführt werden kann. Die CPU, die diese Anweisung ausführen kann, ist mit einem schwarzem Punkt gekennzeichnet.



Konvertierungsanweisungen INT, INTP, DINT, DINTP

6.3.4 INT, INTP, DINT, DINTP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	Q
		●[1]	●[1]	●	●

[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R Anweisungen programmiert werden.

Bei Besonderheiten der Ausführung auf einer bestimmten CPU ist diese in der Tabelle mit einer Fußnote versehen, die die Besonderheit erläutert (z. B. erweiterte Anweisungen siehe „3.3 Programmierung der erweiterten Anweisungen“).

4.3 Operanden MELSEC A

In der Tabelle „Operanden MELSEC A“ sind alle verfügbaren Operanden aufgelistet, die für die internen Variablen (z.B. s1, s2, d) verwendet werden können.

Operanden MELSEC A

	Operanden																						Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene						
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N					M9012	M9010 M9011
s1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					K1 ↓ K4	5/7 [1]	●		●
s2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●						●			

[1] Die Anzahl der Schritte beträgt 7, wenn die Index-Funktion ausgeführt, die Blocklänge eines Bit-Operanden nicht K4 lautet oder die Kopfadresse eines Bit-Operanden nicht das Mehrfache von 8 (bzw. 16 bei einer A3H, A3M, AnA, AnAS oder AnU CPU) ist. Die Anzahl der Schritte bei der Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.9.2 "Bei einer AnA, AnAS und AnU CPU" dieser Programmieranleitung zu entnehmen.

Die verwendbaren Bit- und Wortoperanden sind einzeln aufgelistet. Ist ein Operand nicht verwendbar, fehlt die Kennzeichnung durch einen schwarzen Punkt.
Ob dezimale (K) oder hexadezimale (H, 16#) Konstanten von der Anweisung verwendet werden können, zeigt die Spalte „Konstante“.
Die Spalte „Pointer“ gibt Aufschluss darüber, ob die Anweisung Pointer (P) und/oder (Interrup-Pointer (I) verwenden kann.
Wenn die Anweisung in Verschachtelungsebenen ausgeführt werden kann, ist dies in der Spalte „Ebene“ gekennzeichnet.
Die für die Anweisung verfügbaren Blocklängen für Bit-Operanden sind in der Spalte „Blocklänge“ aufgeführt. Das Beispiel in der Abbildung zeigt, dass die Anweisung Blocklängen von (K1 bis K4) 4 bis 16 Bit adressieren kann.
Die Anzahl der verwendeten Programmschritte wird in der Spalte „Schritte“ aufgeführt.
Wenn die Anweisung die indizierte Adressierung verwenden kann, ist das in der Spalte „Index“ aufgeführt.
Ob die Anweisung bei einem Ergebnisüberlauf das Carry-Flag setzen kann, zeigt die Spalte „Carry Flag“.
Die Möglichkeit, das Error-Flag zu setzen, wird in der Spalte „Error Flag“ gekennzeichnet.

Bei Besonderheiten sind diese mit einer Fußnote an der Kennzeichung versehen und unter der Tabelle erläutert.

4.4 Operanden MELSEC Q

Der Begriff MELSEC Q umfasst alle CPUs des MELSEC System Q und die QnA-, QnAS- und Q4AR- CPUs.

In der Tabelle „Operanden MELSEC Q“ sind alle verfügbaren Operanden aufgelistet, die für die internen Variablen (z.B. s1, s2, d) verwendet werden können.

Die Operanden werden nicht einzeln aufgeführt; es wird nur unterschieden, ob die Anweisung Bit- und/oder Wort-Operanden ansprechen kann.

Operanden MELSEC Q

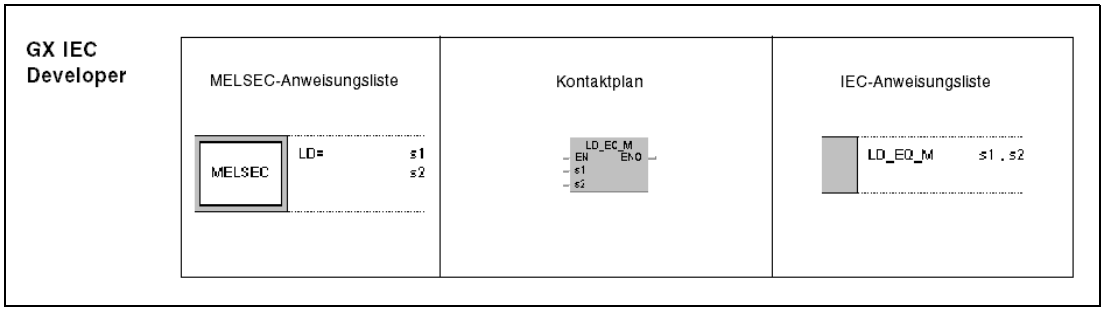
	Operanden										
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere	Error Flag	Schritte
	Bit	Wort		Bit	Wort						
s1	●	●	●	●	●	●	●	●	—	—	3
s2	●	●	●	●	●	●	●	●	—	—	

Verfügt die Anweisung über die Möglichkeit, auf File-Register zu zugreifen, wird das in der Spalte „File-Register“ gekennzeichnet.
In der Spalte „MELSECNET/10 Direkt **J□\□**“ wird angegeben, ob die Anweisung das Lesen/ Schreiben von Bit- und/oder Wort-Daten in/aus im MELSECNET/10 angeschlossenen Stationen vornehmen kann. Mit „**J□**“ wird die Stationsnummer und mit wird „□“ die Operanden-Adresse angegeben.
Die Spalte „Sondermodule **U□\G□**“ gibt Aufschluss darüber, ob die Anweisung das Lesen/ Schreiben von Daten in/aus dem Pufferspeicher eines angeschlossenen Sondermoduls vornehmen kann. Mit „**U□**“ wird die Kopfadresse des Sondermoduls und mit „**G□**“ wird die Pufferspeicheradresse angegeben.
Wenn die Anweisung indizierte Adressierung verwendet, ist das in der Spalte „Index-Register Zn“ aufgeführt.
Ob dezimale (K) oder hexadezimale (H, 16#) Konstanten von der Anweisung verwendet werden können, zeigt die Spalte „Konstanten K, H (16#)“.
In der Spalte „Andere“ wird angegeben, ob die Anwendung sonstige Operanden und Konstanten verwendet.
Die Möglichkeit, das Error-Flag zu setzen, wird in der Spalte „Error Flag“ gekennzeichnet.
Die Anzahl der verwendeten Programmschritte wird in der Spalte „Schritte“ aufgeführt.

4.4.1 Darstellung im GX IEC Developer

Anschließend an die Operanden-Tabellen werden die Darstellungsformate der Anweisung beim GX IEC Developer dargestellt.

Die folgende Abbildung zeigt von links nach rechts die Anweisung LD_EQ_M im MELSEC-Editor (MELSEC-Anweisungsliste) und im IEC-Editor (Kontaktplan und IEC-Anweisungsliste).



4.4.2 Darstellung im GX Developer

An die Darstellungsformate der Anweisung im GX IEC Developer schließt sich die Darstellung im GX Developer an.

GX Developer

4.5 Variablen

Die Variablentabelle enthält alle internen Variablen der Anweisung.

Variablen

Operand	Befehlswert	Datentyp	
		MELSEC	IEC
s	s+0: Messung der Umdrehungen pro Minute (systemintern).	BIN 16 Bit	Array [1..3] of ANY16
	s+1: Nummer der Position.		
	s+2: Nummer des Sektors.		
n1	Teilung (von 2 bis 32767)		ANY16
n2	Anzahl der Abschnitte für niedrige Rotationsgeschwindigkeit (Schleichgang) (von 0 bis n1).		ANY16
d	d+0: Eingangssignal (Phase A)	Bit	Array [1..8] of Bool
	d+1: Eingangssignal (Phase B)		
	d+2: Eingangssignal der Nullpunktsbestimmung.		
	d+3: Ausgangssignal vorwärts, hohe Drehzahl (systemintern).		
	d+4: Ausgangssignal vorwärts, niedrige Drehzahl (systemintern).		
	d+5: Ausgangssignal Stopp (systemintern).		
	d+6: Ausgangssignal rückwärts, hohe Drehzahl (systemintern).		
	d+7: Ausgangssignal rückwärts, niedrige Drehzahl (systemintern).		

Die Spalte „Befehlswert“ beschreibt die Funktion der Operanden und Operandenelemente. In der Spalte „Datentyp“ sind die Datentypen der Operanden aufgeführt. Sofern Datentypenunterschiede zwischen dem MELSEC- und dem IEC-Editor bestehen, sind diese auch aufgeführt. Weitere Informationen zu dem Thema „Variablen“ finden Sie in den Kapiteln „3.4 Programmieren von Variablen“ und „3.5 Datentypen“.

4.6 Funktionsweise

Der Punkt Funktionsweise erläutert detailliert die Arbeitsweise der Anweisung.

Die folgende Abbildung zeigt die Beschreibung der Funktionsweise der LDF/LDP-Anweisung.

Eingangsanweisungen **LDP, LDF, ANDP, ANDF, ORP, ORF**

Funktions-weise

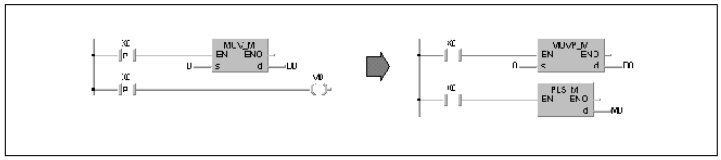
Beginn einer Verknüpfung, flankengesteuert

LDP **ansteigende Flanke**

LDF **abfallende Flanke**

Diese Anweisungen beschreiben analog zu den LD- und LDI-Anweisungen einen Kontakt, der durch einen Bit- oder Wortoperanden gebildet wird. Das Ergebnis der LDP-Anweisung ist 1, wenn das adressierte Bit des Operanden von 0 auf 1 wechselt (ansteigende Flanke, positive Flanke). Das Ergebnis der LDF-Anweisung ist 1, wenn das adressierte Bit des Operanden von 1 auf 0 wechselt (abfallende Flanke, negative Flanke). Die LDP-Anweisung hat bei alleiniger Verwendung die Funktion der PLS-Anweisung und erzeugt bei ansteigender Flanke der Eingangsbedingung einen Ausgangsimpuls.

Das linke Beispiel der unteren Abbildung zeigt einen Kontaktplan bei Verwendung einer LDP-Anweisung, im rechten Beispiel wird keine LDP-Anweisung verwendet.



4.7 Hinweise

In den Hinweisen wird auf Besonderheiten, Fehler und Risiken in der Programmierung der Anweisung hingewiesen.

HINWEISE

Die MEP- und MEF-Anweisungen können unter Umständen nicht richtig ausgeführt werden, wenn auf die vorgeschalteten Kontakte von einem Unterprogramm oder einer FOR-/NEXT-Anweisung zugegriffen wird. In diesem Fall ist die EGP-/EGF-Anweisung zu verwenden.

Aufgrund der Tatsache, daß die MEP-/MEF-Anweisungen mit den unmittelbar vor den MEP-/MEF-Anweisungen anstehenden Signalen arbeiten, sollte vor den Anweisungen eine AND-Anweisung programmiert werden. Die MEP-/MEF-Anweisungen können nicht in Verbindung mit einer LD- oder OR-Anweisung programmiert werden.

4.8 Fehlerquellen

Die Beschreibung der Fehlerquellen bezieht sich hauptsächlich auf die Fehlercodes der Q-Serie und des System Q (siehe „11.1 Liste der Fehlercodes (Q00J-, Q00- und Q01CPU) und „11.2 Liste der Fehlercodes (QnA-Serie und System Q)“). Um Informationen über die Fehlercodes der A-Serie zu erhalten, sehen Sie in den Kapiteln „11.3 Liste der Fehlercodes A-Serie (außer AnA und AnAS)“ und „11.4 Fehlercodeliste der AnA- und AnAS-CPUs“ nach.

Die folgende Abbildung zeigt die Fehlerquellen der DELTA-/DELTAP-Anweisung.

Fehler-quellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Der in d angegebene Ausgang liegt außerhalb des Adreßbereichs für Ausgänge (Fehlercode 4101).

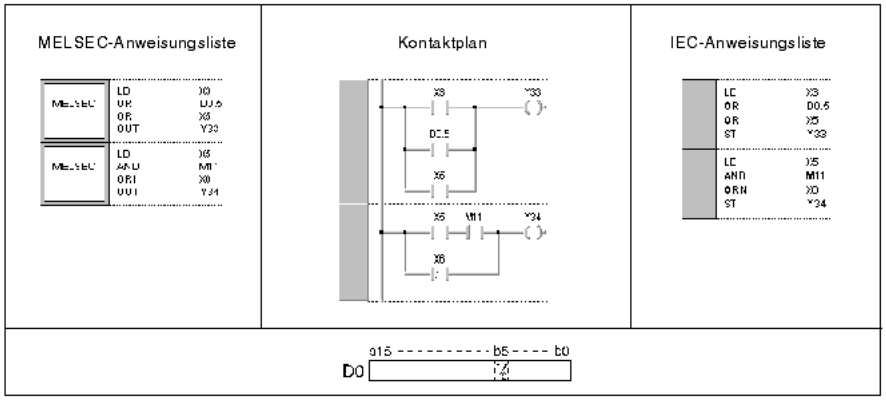
4.9 Beispiele

Die in den Absätzen angeführten Beispiele sind primär Programmbeispiele der Q-Serie.
Die Programmbeispiele wurden in den Darstellungsweisen der MELSEC-Anweisungsliste, dem Kontaktplan und der IEC-Anweisungsliste programmiert. Zum besseren Verständnis wurden in vielen Fällen noch Grafiken hinzugefügt.

Die folgende Abbildung zeigt ein Programmbeispiel der Anweisungen LD, AND, OR und ORI.

Beispiel 1 LD, AND, OR, ORI

Im folgenden Programm werden Reihen- und Parallelschaltungen von Kontakten gezeigt. Das Bit 5 (b5) in D0 wird ebenfalls als Kontakt abgefragt.

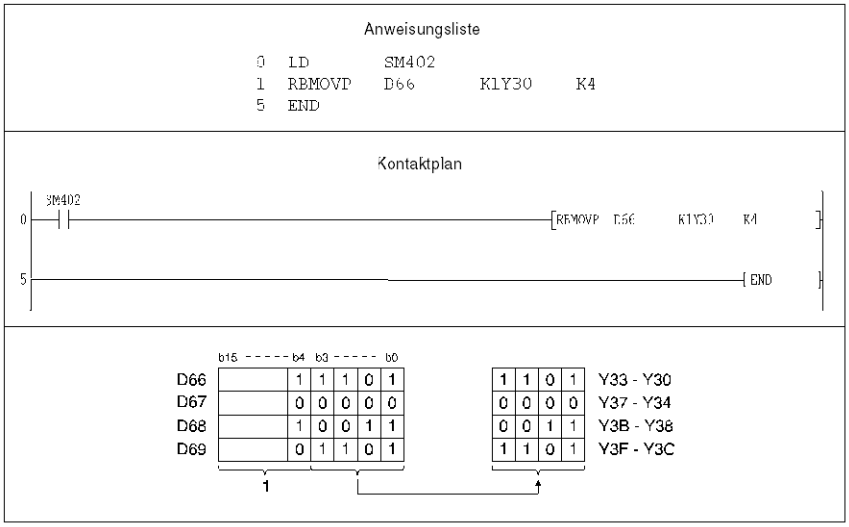


Die folgende Abbildung zeigt ein Programmbeispiel der Anweisung RBMOVP, das mit dem GX Developer programmiert wurde.

Beispiel 1 RBMOVP

Das folgende Programm überträgt mit positiver Flanke von SM402 die Daten der untersten 4 Bits (von b0 bis b3) von D66 bis D69 an die Ausgänge Y30 bis Y3F. Die Anzahl der zu übertragenden Blöcke (4) ist durch die Konstante K4 angegeben.

Die Bit-Schemen zeigen die Bit-Struktur jeweils vor und nach der Übertragung.



[1] Diese Bits bleiben bei der Operation unberücksichtigt.

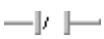
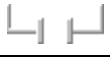
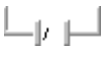
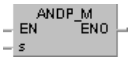
5 Grundbefehlssatz

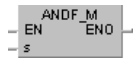
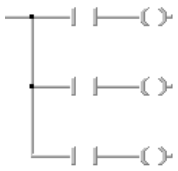
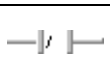
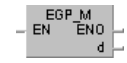
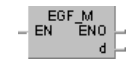
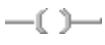
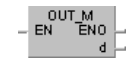
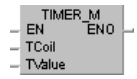
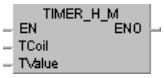
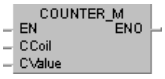
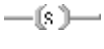
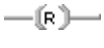
Der Grundbefehlssatz umfasst neben herkömmlichen Anweisungen zur Programmierung von Ein- und Ausgangskontakten auch Sprungbefehle, Blockverknüpfungen und Schieberegisterfunktionen, Master-Control-, Programmende- und weitere Anweisungen und stellt das Grundgerüst zur Programmierung der MELSEC Serien dar.

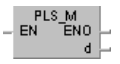
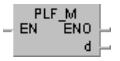
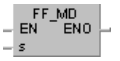
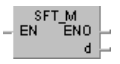
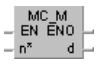
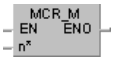
In den folgenden Tabellen ist die Aufteilung des Grundbefehlssatzes dargestellt.

Einteilung	Bedeutung
Eingangs-anweisungen	Beginn einer Verknüpfung, Reihen- und Serienschaltung von Kontakten
Verknüpfungs-anweisungen	Serielle und parallele Blockverknüpfung, Speichern und Verarbeiten eines Verknüpfungsergebnisses, Signalumkehr von Operationsergebnissen, Umwandlung von Operationsergebnissen in gepulste Ergebnisse, Setzen von Flankenmerkern
Ausgangs-anweisungen	Bitoperanden, Zähler- und Zeitkontakte, Ausgabe, Setzen und Rücksetzen von Fehlermerkern, Setzen und Rücksetzen von Operanden, flankengesteuerte Differenzausgabe, Umkehr des Schaltzustandes eines Operanden, Erzeugung von Schaltimpulsen
Verschiebe-anweisungen	Verschieben von Bit-Operanden
Master-Control-Anweisungen	Aktivieren und Deaktivieren einzelner Programmbereiche
Programmende-anweisungen	Ende eines Programmbereiches, Ende von Haupt- und Unterprogrammen
Weitere Anweisungen	Unterbrechung der Verarbeitung, Leerschritt im Programm.

HINWEIS *In der folgenden Tabelle sind neben den MELSEC-Anweisungen in den verschiedenen Editoren auch die entsprechenden IEC-Anweisungen aufgeführt.*

MELSEC-Anweisung				IEC-Anweisung im IEC-Editor
im MELSEC-Editor	im IEC-Editor			
	Anweisungsliste	Kontaktplan		
LD	—		—	LD
LDI	—		—	LDN
AND	—			AND
ANI	—			ANDN
OR	—		—	OR
ORI	—		—	ORN
LDP	LDP_M	—	—	—
LDF	LDF_M	—	—	—
ANDP	ANDP_M	—		—

MELSEC-Anweisung				IEC-Anweisung im IEC-Editor
im MELSEC-Editor	im IEC-Editor			
	Anweisungsliste	Kontaktplan		
ANDF	ANDF_M	—		—
ORP	ORP_M	—		—
ORF	ORF_M	—		—
ANB	—		—	AND (...)
ORB	—		—	OR (...)
MPS	MPS_M			—
MRD	MRD_M			—
MPP	MPP_M			—
INV	INV_M			NOT
MEP	MEP_M	—		—
MEF	MEF_M	—		—
EGP	EGP_M	—		—
EGF	EGF_M	—		—
OUT	OUT_M			ST
OUT (T)	TIMER_M	—		—
OUT H (T)	TIMER_H_M	—		—
OUT (C)	COUNTER_M	—		—
SET	SET_M			S
RST	RST_M			R

MELSEC-Anweisung				IEC-Anweisung im IEC-Editor
im MELSEC-Editor	im IEC-Editor			
	Anweisungsliste	Kontaktplan		
PLS	PLS_M	—		R_TRIG ●[1]
PLF	PLF_M	—		F_TRIG ●[1]
FF	FF_M	—		—
CHK	CHK_M	—		—
DELTA	DELTA_M	—		—
SFT	SFT_M	—		SHL/SHR
MC	MC_M	—		—
MCR	MCR_M	—		—
FEND	FEND_M	—		●[2]
END	END_M	—		●[2]
STOP	STOP_M	—		—
NOP	—	—	—	—

[1] Hierbei handelt es sich um IEC-Funktionsbausteine.

[2] FEND und END werden vom GX Developer und GX IEC Developer automatisch erzeugt.

5.1 Eingangsanweisungen

5.1.1 LD, LDI, AND, ANI, OR, ORI

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
	●	●	●	●	●	●	●	●	●														1 ●[1]			

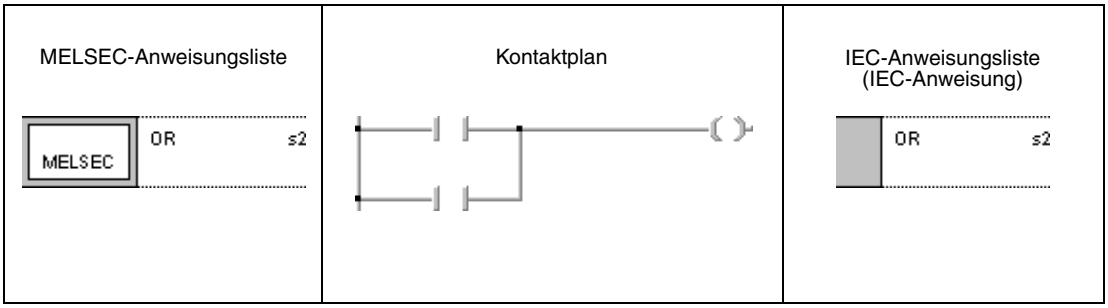
[1] Die Anzahl der Schritte bei Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

Operanden MELSEC Q

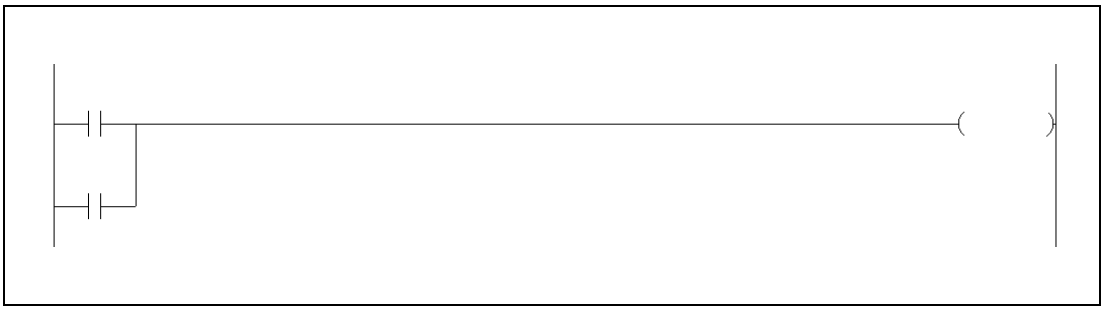
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort				DX, BL		
s	●	●	●	●	●	●	—	—	●	—	1 ●[1]

[1] Die Anzahl der Schritte kann variieren:
- Bei Verwendung eines internen Operanden oder der Fileregister R0 bis R32767: 1 Schritt
- Bei Verwendung direkt adressierbarer Eingänge (DX): 2 Schritte
- Bei Verwendung anderer Operanden: 3 Schritte

GX IEC Developer



GX Developer



Variablen

Operand	Befehlswert	Datentyp
s	Die Operanden werden als Kontakte gesetzt.	Bit

Funktions-weise

Beginn einer Verknüpfung

LD Lade (Schließerkontakt)

LDI Lade invers (Öffnerkontakt)

Der Beginn einer Verknüpfung erfolgt immer mit einer LD- **(L**oa**D**; lade) oder einer LDI- (**L**oa**D** **I**nverse; lade invers) Anweisung. Die LD-Anweisung beschreibt einen Schließerkontakt und die LDI-Anweisung einen Öffnerkontakt. Die in der Anweisung enthaltene Operandenadresse ist die Eingangsbedingung der folgenden Anweisung.

Reihenschaltung

AND mit Schließern

ANI mit Öffnern

Eine Reihenschaltung von Kontakten erfolgt mit einer AND-Anweisung als Schließer oder einer ANI-Anweisung als Öffner. Der in der Anweisung enthaltene Operand stellt die Weiterschaltbedingung zur nachgeschalteten Anweisung.

Beide Befehle stellen logische Verknüpfungen dar und dürfen nicht an den Anfang einer Verknüpfung programmiert werden.

Parallelschaltung

OR von Schließern

ORI von Öffnern

Eine Parallelschaltung von Kontakten erfolgt mit einer OR-Anweisung als Schließer oder einer ORI-Anweisung als Öffner. Der in den Anweisungen enthaltene Operand stellt die Weiterschaltbedingung der vorangestellten Anweisung zur nachgeschalteten Anweisung.

Beide Befehle stellen logische Verknüpfungen dar und dürfen nicht an den Anfang einer Verknüpfung programmiert werden.

HINWEISE

Die in den Anweisungen angegebenen Operanden können auch Wortoperanden sein. In diesen Fällen wird der Zustand eines ausgewählten Bits als Kontakt eingelesen (nur Q-Serie und System Q).

Die Angabe des abzufragenden Bits wird bei Wortoperanden hexadezimal vorgenommen. Bit b11 in D0 wird beispielsweise hexadezimal mit D0.0B angegeben (nur Q-Serie und System Q).

Weitere Informationen über Adressierung von Bits in Wortoperanden enthält der Abschnitt "Zusammensetzung von Anweisungen" der Programmieranleitung (nur Q-Serie und System Q).

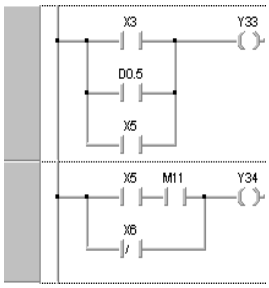
Beispiel 1 LD, AND, OR, ORI

Im folgenden Programm werden Reihen- und Parallelschaltungen von Kontakten gezeigt. Das Bit 5 (b5) in D0 wird ebenfalls als Kontakt abgefragt.

MELSEC-Anweisungsliste

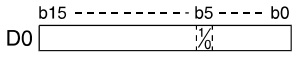
```
MELSEC  LD   X3
        OR   D0.5
        OR   X5
        OUT  Y33
MELSEC  LD   X5
        AND  M11
        ORI  X6
        OUT  Y34
```

Kontaktplan



IEC-Anweisungsliste

```
LD   X3
OR   D0.5
OR   X5
ST   Y33
LD   X5
AND  M11
ORN  X6
ST   Y34
```



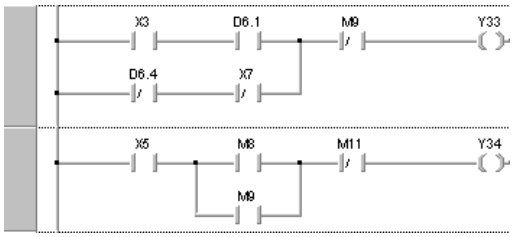
Beispiel 2 LD, LDI, AND, ANI, OR

Im folgenden Programm werden gemischte Schaltungen gezeigt. Einige Kontaktpunkte der Schaltung werden durch ORB und ANB-Anweisungen verknüpft. Die adressierten Bits (b1 und b4) aus D6 werden als Kontakte eingelesen.

MELSEC-Anweisungsliste

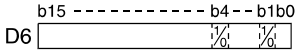
```
MELSEC  LD   X3
        AND  D6.1
        LDI  D6.4
        ANI  X7
        ORB
        ANI  M9
        OUT  Y33
MELSEC  LD   X5
        LD   M8
        OR   M9
        ANB
        ANI  M11
        OUT  Y34
```

Kontaktplan



IEC-Anweisungsliste

```
LD   X3
AND  D6.1
OR(
     D6.4
NOT
ANDN X7
)
ANDN M9
ST   Y33
LD   X5
AND(
     M8
OR   M9
)
ANDN M11
ST   Y34
```



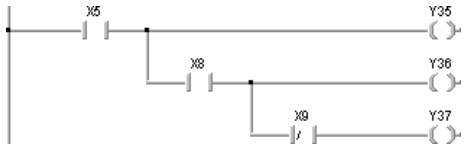
Beispiel 3 LD, AND, ANI

Im folgenden Programm werden die Operandenergebnisse der Anweisungen an Y35 bis Y37 ausgegeben.

MELSEC-Anweisungsliste

```
MELSEC
        LD    X5
        OUT   Y35
        AND   X8
        OUT   Y36
        ANI   X9
        OUT   Y37
```

Kontaktplan



IEC-Anweisungsliste

```
        LD    X5
        ST    Y35
        AND   X8
        ST    Y36
        ANDN  X9
        ST    Y37
```

5.1.2 LDP, LDF, ANDP, ANDF, ORP, ORF

CPU

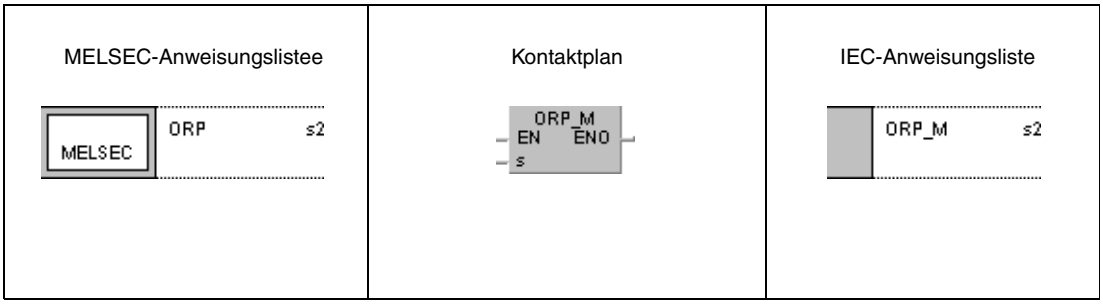
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Operanden MELSEC Q

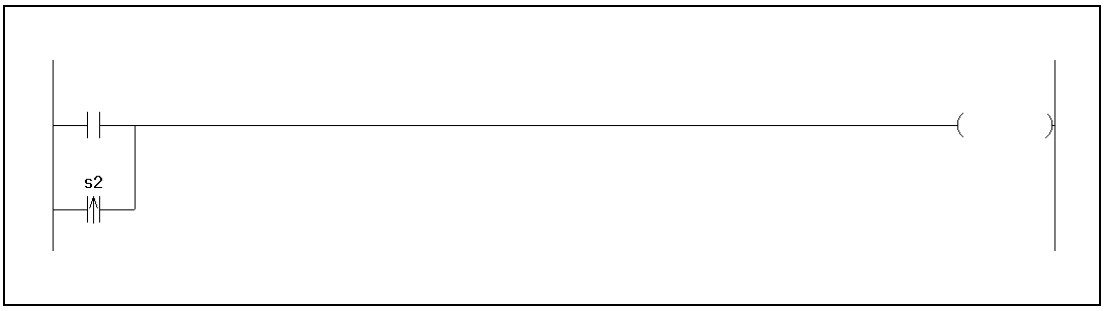
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort				DX		
s	●	●	●	●	●	●	—	—	●	—	2 ●[1]

[1] Die Anzahl der Schritte kann variieren:
- Bei Verwendung eines internen Operanden oder der Fileregister R0 bis R32767: 2 Schritte
- Bei Verwendung direkt adressierbarer Eingänge (DX): 3 Schritte
- Bei Verwendung anderer Operanden: 4 Schritte

GX IEC Developer



GX Developer



Variablen

Operand	Befehlswert	Datentyp
s	Die Operanden werden als Kontakte gesetzt.	Bit

Funktions-weise

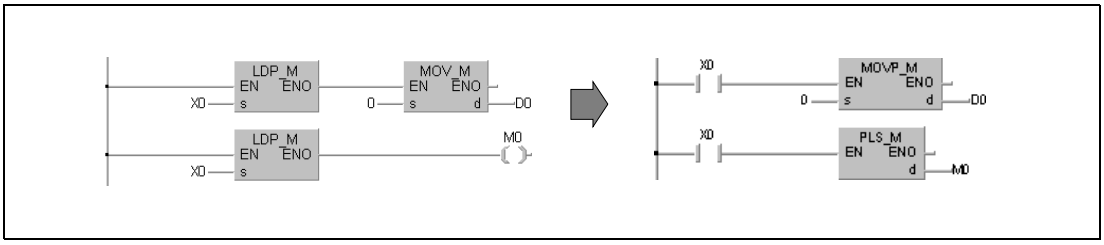
Beginn einer Verknüpfung, flankengesteuert

LDP ansteigende Flanke

LDF abfallende Flanke

Diese Anweisungen beschreiben analog zu den LD- und LDI-Anweisungen einen Kontakt, der durch einen Bit- oder Wortoperanden gebildet wird. Das Ergebnis der LDP-Anweisung ist 1, wenn das adressierte Bit des Operanden von 0 auf 1 wechselt (ansteigende Flanke, positive Flanke). Das Ergebnis der LDF-Anweisung ist 1, wenn das adressierte Bit des Operanden von 1 auf 0 wechselt (abfallende Flanke, negative Flanke). Die LDP-Anweisung hat bei alleiniger Verwendung die Funktion der PLS-Anweisung und erzeugt bei ansteigender Flanke der Eingangsbedingung einen Ausgangsimpuls.

Das linke Beispiel der unteren Abbildung zeigt einen Kontaktplan bei Verwendung einer LDP-Anweisung, im rechten Beispiel wird keine LDP-Anweisung verwendet.



Reihenschaltung, flankengesteuert

ANDP ansteigende Flanke

ANDF abfallende Flanke

Die ANDP-Anweisung schaltet einen Kontakt mit einem durch einen Bit- oder Wortoperanden gebildeten Kontakt in Reihe. Dieser Kontakt hat den Zustand 1, wenn das adressierte Bit des Operanden von 0 auf 1 wechselt. Bei der ANDF-Anweisung hat der gebildete Kontakt den Zustand 1, wenn das adressierte Bit des Operanden von 1 auf 0 wechselt.

Parallelschaltung, flankengesteuert

ORP ansteigende Flanke

ORF abfallende Flanke

Die ORP-Anweisung schaltet einen Kontakt mit einem durch einen Bit- oder Wortoperanden gebildeten Kontakt parallel. Dieser Kontakt hat den Zustand 1, wenn das adressierte Bit des Operanden von 0 auf 1 wechselt. Bei der ORF-Anweisung hat der gebildete Kontakt den Zustand 1, wenn das adressierte Bit des Operanden von 1 auf 0 wechselt.

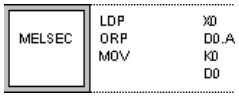
Durch ANDP/ORP-Anweisung spezifizierter Operand	Ergebnis der ANDP/ORP-Anweisung	Durch ANDF/ORF-Anweisung spezifizierter Operand	Ergebnis der ANDF/ORF-Anweisung
Bit-Operand/ Wortoperand		Bit-Operand/ Wortoperand	
0 → 1	1	0 → 1	0
0	0	0	
1		1	
1 → 0		1 → 0	1

HINWEIS *Die Adressierung der abzufragenden Bits in Wort-Operanden erfolgt hexadezimal. Das Bit 11 in D0 wird beispielsweise hexadezimal mit D0.0B adressiert.*

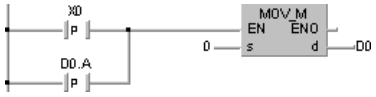
Beispiel ORP

Im folgenden Programm wird mit positiver Flanke von X0 oder mit Setzen (positiver Flanke) des Bits 10 (b10) des Datenregisters D0 eine MOV-Anweisung ausgeführt.

MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste

```
LD      X0
PLS_M   M0
ORP_M   D0.A
MOV_M   0, D0
```

5.2 Verknüpfungsanweisungen

5.2.1 ANB, ORB

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
																							1			

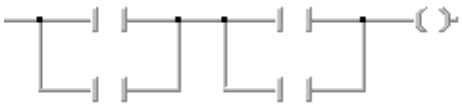
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
—	—	—	—	—	—	—	—	—	—	—	1

GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan



IEC-Anweisungsliste (IEC-Anweisung)

GX Developer

Variablen

Operand	Befehlswert	Datentyp
—	—	—

Funktions-weise

Reihenverknüpfung von Parallelschaltungen

ANB Serielle Blockverknüpfung

Die ANB-Anweisung (UND-Block) schaltet 2 oder mehr Parallelschaltungen in Reihe und stellt die Weiterschaltbedingung. Sollen mehrere Blöcke in Reihe geschaltet werden, ist nach jedem Parallelblock eine ANB-Anweisung zu programmieren.

Die ANB-Verknüpfung ist eine unabhängige Anweisung und erfordert keinen Operanden.

Innerhalb eines Programms kann die ANB-Anweisung beliebig oft programmiert werden.

Sollen mehrere Blöcke direkt hintereinander verknüpft werden, ist die Anzahl der ANB-Anweisungen bei QnA, AnA, AnAS und AnU CPUs auf 15 (entsprechend 16 Blöcken) und bei allen anderen CPUs auf 7 (entsprechend 8 Blöcken) begrenzt. Bei Überschreitung der entsprechenden Grenzen ist ein einwandfreier Betrieb nicht mehr gewährleistet.

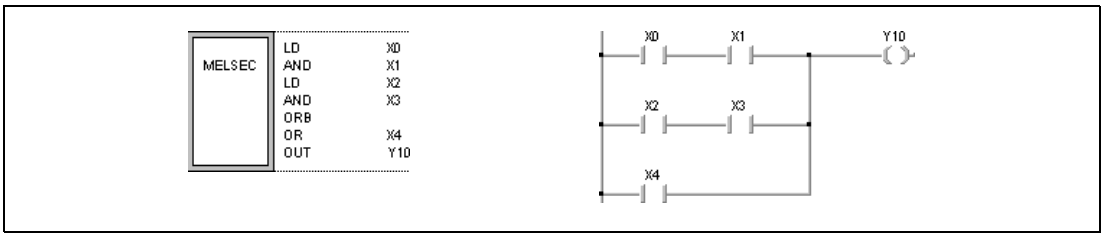
Parallelverknüpfungen von Reihenschaltungen

ORB Parallele Blockverknüpfung

Die ORB-Anweisung (ODER-Block) schaltet 2 oder mehr Reihenschaltungen parallel und stellt die Weiterschaltbedingung.

Sollen mehrere Blöcke parallel geschaltet werden, ist nach jedem einzelnen Block eine ORB-Anweisung zu programmieren.

Bei parallelen Blockverknüpfungen, in denen nur ein Kontakt vorkommt, ist die OR- oder ORI-Anweisung anstelle der ORB-Anweisung zu setzen.



Die ORB-Verknüpfung ist eine unabhängige Anweisung und erfordert keinen Operanden.

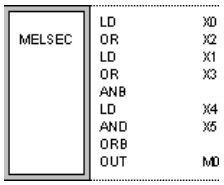
Innerhalb eines Programms kann die ORB-Anweisung beliebig oft programmiert werden.

Sollen mehrere Blöcke direkt nebeneinander verknüpft werden, ist die Anzahl der ORB-Anweisungen bei QnA, AnA, AnAS und AnU CPUs auf 15 (entsprechend 16 Blöcken) und bei allen anderen CPUs auf 7 (entsprechend 8 Blöcken) begrenzt. Bei Überschreitung der entsprechenden Grenzen ist ein einwandfreier Betrieb nicht mehr gewährleistet.

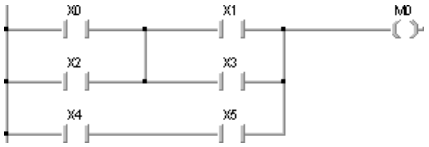
Beispiel ANB, ORB

Im folgenden Programm wird die Parallelschaltung von X0 und X2 in Reihe mit der Parallelschaltung von X1 und X3 verknüpft. Dieses Ergebnis wird mit der Reihenschaltung von X4 und X5 parallel verknüpft.

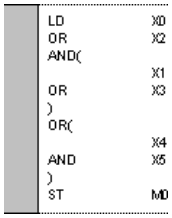
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



5.2.2 MPS, MRD, MPP

HINWEIS I*n den IEC-Editoren sollten diese Anweisungen nicht verwendet werden.*

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
																							1			

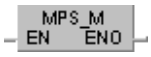
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sonder-module U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
—	—	—	—	—	—	—	—	—	—	—	1

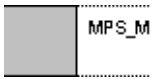
GX IEC Developer

MELSEC-Anweisungsliste

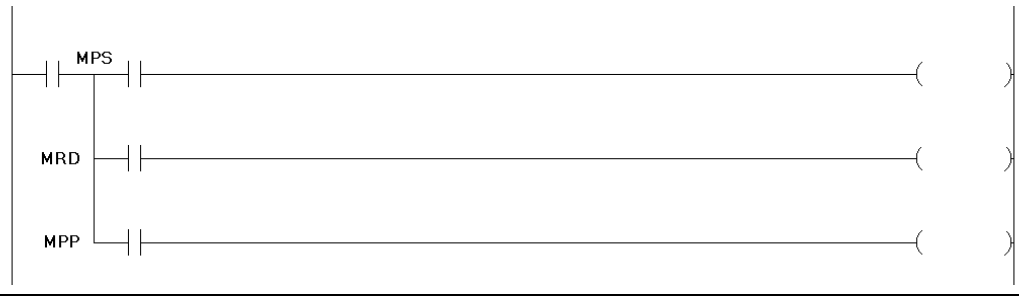
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

Operand	Befehlswert	Datentyp
—	—	—

Funktions-weise

Ergebnisverarbeitung

MPS Ergebnis speichern

Mit Hilfe der MPS-Anweisung wird das Ergebnis der Verknüpfung, die der MPS-Anweisung vorangestellt ist, gespeichert.

In Verbindung mit der Q, QnA, AnA, AnAS und AnU CPU können bis zu 16 und mit allen anderen CPUs bis zu 12 aufeinanderfolgende MPS-Anweisungen pro Netzwerk programmiert werden. Befindet sich zwischen den in Folge programmierten MPS-Anweisungen eine MPP-Anweisung, verringert sich die Maximalzahl um 1.

MRD Ergebnis lesen

Die MRD-Anweisung liest das mittels MPS-Anweisung gespeicherte Verknüpfungsergebnis aus. Der nächste Schritt wird in Abhängigkeit des eingelesenen Ergebnisses ausgeführt.

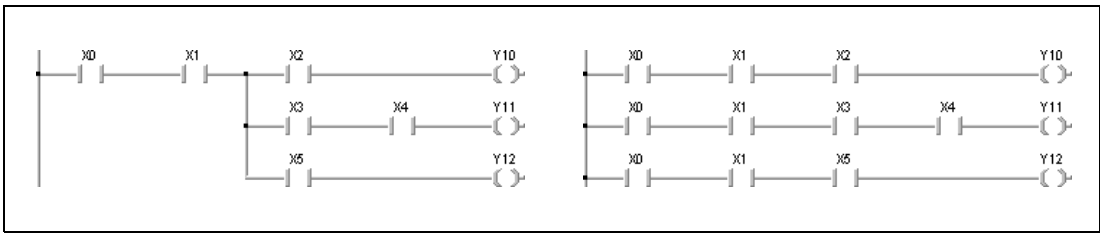
MPP Ergebnis lesen und löschen

Die MPP-Anweisung liest das mittels MPS-Anweisung gespeicherte Verknüpfungsergebnis, führt den nächsten Schritt in Abhängigkeit des eingelesenen Verknüpfungsergebnisses aus und löscht das Verknüpfungsergebnis.

Die MPS-, MRP- oder MPP-Anweisungen sind unabhängige Anweisungen und erfordern deshalb keine Operanden.

In der Kontaktplanprogrammierung werden die MPS-, MRD- und MPP-Anweisungen nicht gesondert dargestellt. Inwieweit es sich um MPS-, MRD- oder MPP-Verknüpfungen handelt, ist vom Aufbau der Kontaktplanverknüpfung abhängig.

In der linken Abbildung ist ein Kontaktplan mit Verwendung der MPS- und MRD-Anweisung abgebildet. Der rechte Kontaktplan ist ohne Verwendung der MPS-, MRD- und MPP-Anweisung programmiert.



Die Anzahl der MPS-Anweisungen im Programm muss mit der Anzahl der MPP-Anweisungen übereinstimmen.

Ist die Anzahl der MPS-Anweisungen größer als die Anzahl der MPP-Anweisungen, wird anstelle einer MPP-Anweisung eine NOP-Anweisung gesetzt und der Programmverlauf entsprechend geändert.

Ist die Anzahl der MPP-Anweisungen größer als die Anzahl der MPS-Anweisungen, bricht die logische Programmfolge ab. In diesem Fall wird die Programmverarbeitung nicht weiter fortgesetzt, und die SPS gibt eine Fehlermeldung aus.

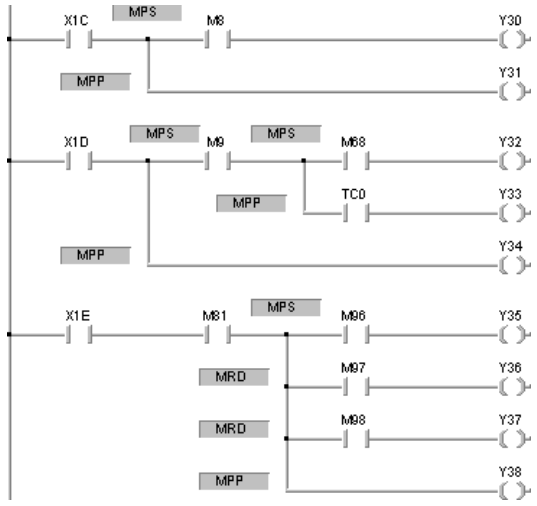
Beispiel 1 MPS, MRD, MPP

Im folgenden Programm wird die Verwendung der Anweisungen bei der Programmierung von gemischten Schaltungen erläutert.

MELSEC-Anweisungsliste

```
MELSEC
LD    X1C
MPS
AND   M8
OUT   Y30
MPP
OUT   Y31
LD    X1D
MPS
AND   M9
MPS
AND   M68
OUT   Y32
MPP
AND   T0
OUT   Y33
MPP
OUT   Y34
LD    X1E
AND   M81
MPS
AND   M96
OUT   Y35
MRD
AND   M97
OUT   Y36
MRD
AND   M98
OUT   Y37
MPP
OUT   Y38
```

Kontaktplan



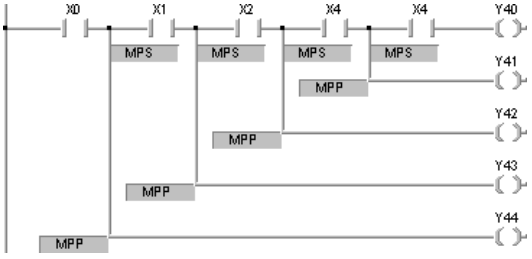
Beispiel 2 MPS, MRD, MPP

Im folgenden Programm wird die Programmierung der Anweisungen zur Ausgabe der Zwischenergebnisse einer Reihenschaltung erläutert.

MELSEC-Anweisungsliste

```
MELSEC
LD    X0
MPS
AND   X1
MPS
AND   X2
MPS
AND   X3
MPS
AND   X4
OUT   Y40
MPP
OUT   Y41
MPP
OUT   Y42
MPP
OUT   Y43
MPP
OUT   Y44
```

Kontaktplan



5.2.3 INV

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

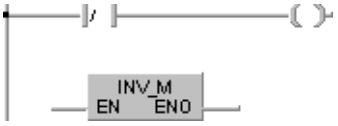
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort				U		
—	—	—	—	—	—	—	—	—	—	—	1

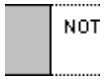
GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan



IEC-Anweisungsliste (IEC-Anweisung)



GX Developer

Variablen

Operand	Befehlswert	Datentyp
—	—	—

Funktions-weise

Signalumkehr von Operationsergebnissen

INV Inversionsanweisung

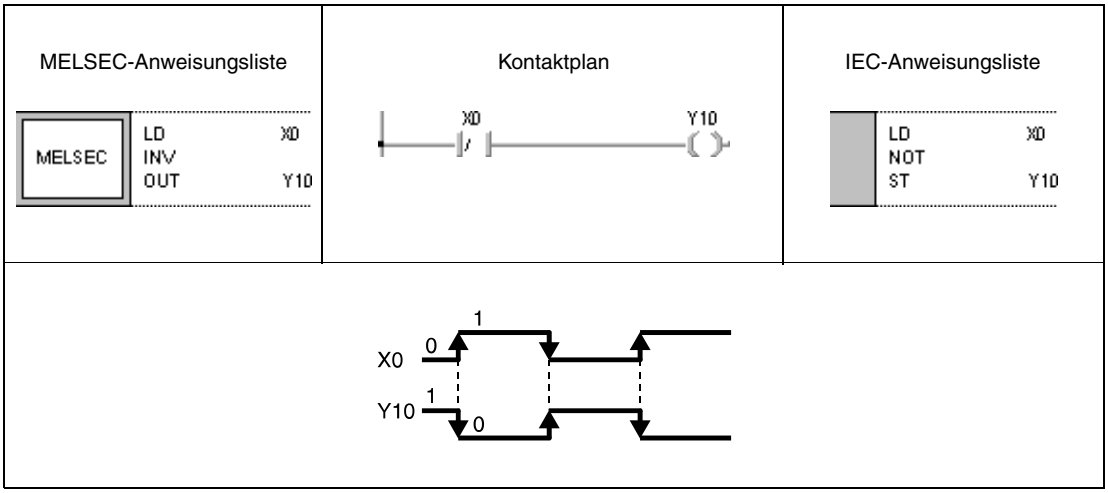
Die INV-Anweisung kehrt das vor der Anweisung anstehende Operationsergebnis um.

War das Ergebnis vor der Anweisungsausführung 1, ist es nach der Ausführung 0.

War das Ergebnis vor der Anweisungsausführung 0, ist es nach der Ausführung 1.

Beispiel

Im folgenden Programm wird das Operationsergebnis von X0 umgekehrt und das invertierte Signal an Y10 ausgegeben.



5.2.4 MEP, MEF

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Operanden MELSEC Q

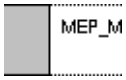
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort				U		
—	—	—	—	—	—	—	—	—	—	—	1

GX IEC Developer

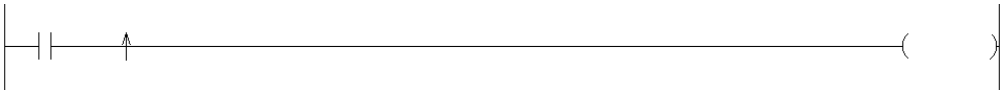
MELSEC-Anweisungsliste

Kontaktplan

IEC-Anweisungsliste



GX Developer



Variablen

Operand	Befehlswert	Datentyp
—	—	—

Funktionsweise

Umwandlung von Operationsergebnissen in gepulste Ergebnisse

MEP Pulserzeugung bei ansteigender Flanke des Operationsergebnisses

Die MEP-Anweisung wird überall dort eingesetzt, wo die verwendeten Anweisungen ihr Operationsergebnis nicht in Form eines definierten Ausgangsimpulses ausgeben können. Die MEP-Anweisung wird hinter die entsprechende Anweisung geschaltet und erzeugt bei einem Wechsel des Eingangssignals von 0 auf 1 (ansteigende Flanke) am Eingang einmalig einen Ausgangsimpuls. Der nächste Impuls wird erst bei einer erneuten ansteigenden Flanke am Eingang erzeugt.

MEF Pulserzeugung bei abfallender Flanke des Operationsergebnisses

Die MEF-Anweisung arbeitet vom Prinzip her wie die MEP-Anweisung, mit dem Unterschied, dass hier ein einmaliger Ausgangsimpuls erzeugt wird, wenn das Eingangssignal von 1 auf 0 wechselt (abfallende Flanke). Der nächste Impuls wird erst bei einer erneuten abfallenden Flanke am Eingang erzeugt.

Diese beiden Anweisungen eignen sich besonders bei der Verwendung von mehreren zusammengeschalteten Kontakten. Mehrere in Reihe geschaltete Schließer würden beispielsweise in geschlossenem Zustand ständig eine 1 als Operationsergebnis haben. Wird darüber ein Merker gesetzt, könnte dieser nicht zurückgesetzt werden. Durch die Reihenschaltung mit einer MEP-Anweisung ist das Rücksetzen möglich, da nur ein Ausgangsimpuls gesetzt wird, wenn der Ausgangszustand der Reihenschaltung von 0 auf 1 wechselt.

HINWEISE

Die MEP- und MEF-Anweisungen können unter Umständen nicht richtig ausgeführt werden, wenn auf die vorgeschalteten Kontakte von einem Unterprogramm oder einer FOR-/NEXT-Anweisung zugegriffen wird. In diesem Fall ist die EGP-/EGF-Anweisung zu verwenden.

Aufgrund der Tatsache, dass die MEP-/MEF-Anweisungen mit den unmittelbar vor den MEP-/MEF-Anweisungen anstehenden Signalen arbeiten, sollte vor den Anweisungen eine AND-Anweisung programmiert werden. Die MEP-/MEF-Anweisungen können nicht in Verbindung mit einer LD- oder OR-Anweisung programmiert werden.

Beispiel

MEP

Im folgenden Programm wird mit positiver Flanke des Reihenschaltungsergebnisses von X0 und X1 der Merker M0 durch den Ausgangsimpuls der MEP-Anweisung gesetzt.

MELSEC-Anweisungsliste

```
MELSEC   LD    X0
         AND   X1
         MEP
         SET   M0
```

IEC-Anweisungsliste

```
         LD    X0
         AND   X1
         MEP_M
         S     M0
```

Kontaktplan

5.2.5 EGP, EGF

CPU

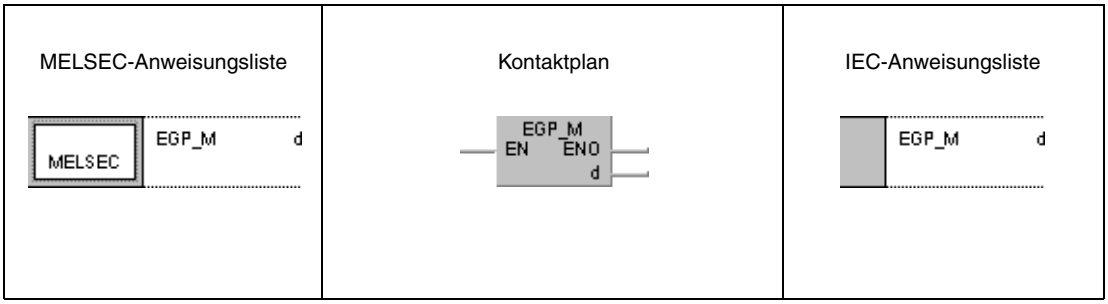
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort				U		
d	●[1]	—	—	—	—	—	—	—	—	—	1

[1] Nur V

GX IEC Developer



GX Developer

Variablen

Operand	Befehlswert	Datentyp
d	Flankenmerker, in dem das Ergebnis gespeichert wird.	Bit (nur V)

Funktionsweise

Setzen eines Flankenmerkers

EGP Setzen des Flankenmerkers bei ansteigender Flanke des Operationsergebnisses

Die EGP-Anweisung setzt den Flankenmerker (V) in Abhängigkeit des Operationsergebnisses der vorgeschalteten Anweisung. Wechselt das Ergebnis von 0 auf 1, wird der Flankenmerker gesetzt. Bei allen anderen Eingangszuständen der EGP-Anweisung, z.B. Wechsel von 1 auf 0 oder Halten der Zustände 1 oder 0 wird der Flankenmerker nicht gesetzt.

EGF Setzen des Flankenmerkers bei abfallender Flanke des Operationsergebnisses

Die EGF-Anweisung arbeitet im Prinzip wie die EGP-Anweisung. Bei dieser Anweisung wird der Flankenmerker (V) gesetzt, wenn das Ergebnis der vorgeschalteten Anweisung von 1 auf 0 wechselt. Bei Wechsel des Ergebnisses von 0 auf 1 oder Halten der Zustände 1 oder 0 wird der Flankenmerker nicht gesetzt.

Die EGP-/EGF-Anweisungen werden in Unterprogrammen oder Programmen, die mit Adressierungen über Index-Register (indizierte Adressierung) arbeiten und sich zwischen FOR- und NEXT-Anweisungen befinden, verwendet.

Die EGP-/EGF-Anweisung kann wie eine AND-Anweisung verwendet werden.

Beispiel EGP

Im folgenden Programm wird das Indexregister Z0 zunächst auf 0 gesetzt und dann das Unterprogramm UP1 (1) aufgerufen. Dort wird mit positiver Flanke X0Z0 auf X0 und V0Z0 auf V0 gesetzt. Außerdem wird D0Z0 auf D0 gesetzt und um 1 inkrementiert.

Nach dem Rücksprung wird eine 1 in das Indexregister Z0 geschrieben, und das Unterprogramm wird erneut aufgerufen (2). Mit positiver Flanke von X1 wird V1 gesetzt und D1 inkrementiert.

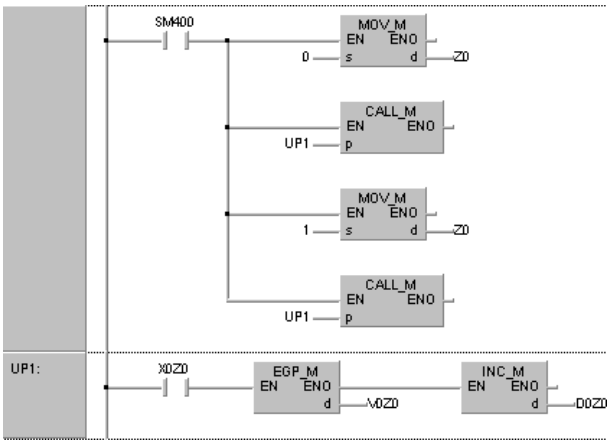
MELSEC-Anweisungsliste

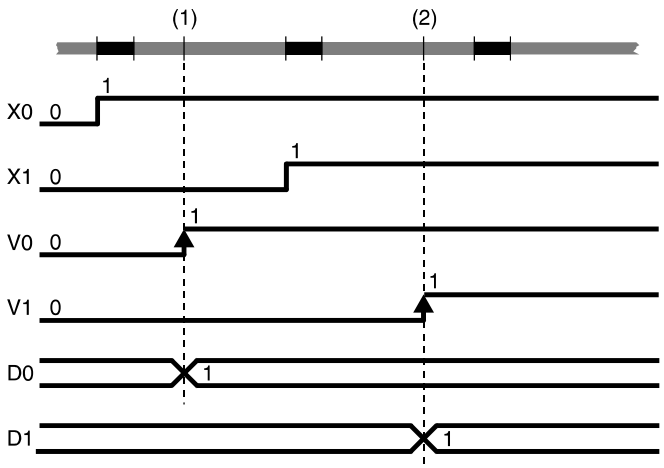
```
MELSEC   LD     SM400
         MOV    K0
                Z0
         CALL   UP1
         MOV    K1
                Z0
         CALL   UP1
         FEND
UP1:     LD     X0Z0
MELSEC   EGP    V0Z0
         INC    D0Z0
         RET
```

IEC-Anweisungsliste

```
         LD     SM400
         MOVE_E 0, Z0
         JMP    UP1
         LD     SM400
         MOVE_E 1, Z0
         JMP    UP1
UP1:     LD     X0Z0
         EGP_M  V0Z0
         INC_M  D0Z0
```

Kontaktplan





5.3 Anweisungen für Ausgangskontakte

5.3.1 OUT

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
d		●	●	●	●	●																	●[1]	●[2]		

[1] Generell 1 Schritt, aber 3 Schritte bei Programmierung von Sonder- oder Fehlermerkern als Operand in der OUT-Anweisung. Die Anzahl der Schritte bei der Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

[2] Die Index-Funktion ist nur bei AnA, AnAS oder AnU CPUs verfügbar.

Operanden MELSEC Q

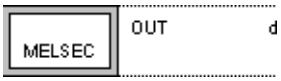
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort				DY		
d	●[1]	●	●	●	●	●	—	—	●	—	1 ●[2]

[1] Außer T,C,F

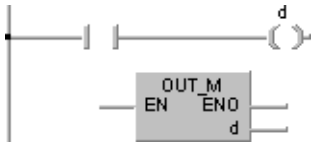
[2] 1 Schritt bei der Verwendung interner Operanden, 2 Schritte bei Verwendung direkt adressierbarer Ausgänge DY und 3 Schritte bei der Verwendung aller anderen Operanden (inkl. der File-Register mit seriellem Zugriff).

GX IEC Developer

MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste (IEC-Anweisung)

GX Developer

Variablen

Operand	Befehlswert	Datentyp
d	Ein- und auszuschaltender Ausgangskontakt.	Bit

Funktions-weise

Ausgangsanweisungen

OUT Setzanweisung für Ausgänge

Ein Ausgang wird in Abhängigkeit der vorangestellten Eingangsbedingung geschaltet.

Mehrere OUT-Anweisungen können parallel nach einer Eingangsbedingung programmiert werden.

Der Schaltzustand eines OUT-Kontaktes kann in den nachfolgenden Programmschritten als Eingangsbedingung in Form eines herkömmlichen Schließer- oder Öffnerkontaktes benutzt werden.

Eingangs-bedingung	OUT-Anweisung			Bei Bit-Bestimmung für einen Wortoperanden
	Ausgangskontakt	Kontaktform		Bestimmtes Bit
		Schließer	Öffner	
0	AUS	kein Durchgang	Durchgang	0
1	EIN	Durchgang	kein Durchgang	1

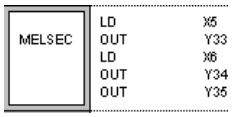
Fehler-quellen

Siehe Teil I der Programmieranleitung.

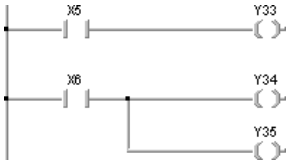
Beispiel 1 OUT

Im folgenden Programm wird die Programmierung der OUT-Anweisung bei Verwendung von Bit-Operanden als Ausgänge (Y33 bis Y35) gezeigt.

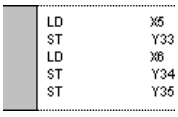
MELSEC-Anweisungsliste



Kontaktplan



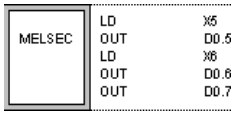
IEC-Anweisungsliste



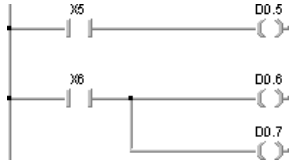
Beispiel 2 OUT

Im folgenden Programm wird die Programmierung der OUT-Anweisung bei Verwendung von Bits des Wortoperanden D0 als Ausgänge (Bits b5 bis b7) gezeigt.

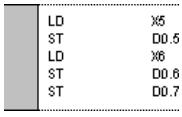
MELSEC-Anweisungsliste

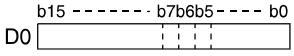


Kontaktplan



IEC-Anweisungsliste





5.3.2 OUT T, OUTH T

CPU

AnS	AnN	An(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)								Konstante		Pointer		Ebene				
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
d								●									●	●					1			
●[1]										●[2]							●[2]									

[1] Einstellwert

[2] Zur Verwendung der erweiterten Timer und Counter einer AnA, AnAS oder AnU CPU sind die Hinweise in Abs. A.3.1 „Timer-Vergleich“ dieser und Abs. A.3.2 „Counter-Vergleich“ dieser Programmieranleitung zu beachten.

Operanden MELSEC Q

	Operanden								Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sonder-module U□\G□	Index-Register Zn	Konstanten K	Andere	
	Bit	Wort		Bit	Wort					
d	●[1]	—	—	—	—	—	—	—	—	—	4
●[2]	—	●[3]	●	—	●	●	—	●[4]	—	—	4

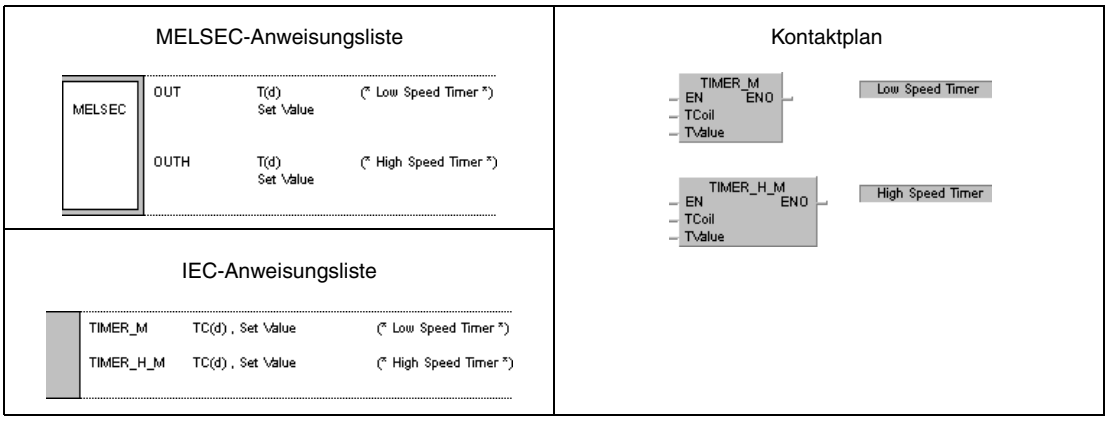
[1] Nur T

[2] Einstellwert

[3] Außer T und C

[4] Die Einstellwerte für Timer müssen als dezimale Konstanten (K) angegeben werden. Hexadezimale Konstanten können nicht verwendet werden.

GX IEC Developer



GX Developer

Variablen

Operand	Befehlswert	Datentyp
d	Timernummer	Bit
Einstellwert	Einstellwerte (Set Value) für Timer	BIN-16-Bit

Funktionsweise

Setzanweisung für Timerkontakte

OUT T Langsamer Timer (100 ms)

OUTH T Schneller Timer (10 ms)

Ist die Eingangsbedingung einer OUT(H) T-Anweisung gegeben, schaltet der Zeitkontakt ein und bleibt für einen vorgegebenen Zeitraum gesetzt. Dieser Zeitraum wird direkt durch eine Konstante oder variabel durch den Wert eines Datenregisters vorgegeben.

Der Schaltzustand des OUT(H) T-Kontaktes wird in einem (oder mehreren) nachfolgenden Programmschritt(en) als Eingangsbedingung in Form eines herkömmlichen Schließer- oder Öffnerkontaktes programmiert.

Nach Ablauf der vorgegebenen Zeit (Istwert = Sollwert) wird der nachgeschaltete Eingangskontakt gesetzt.

Es können mehrere OUT(H) T-Anweisungen hinter ein und derselben Eingangsbedingung programmiert werden.

Timer als Ausgangskontakt			Timer als Eingangsbedingung			
Art	Kontaktzustand	Istwert	Kontaktzustand vor Ablauf		Kontaktzustand nach Ablauf	
			Schließer	Öffner	Schließer	Öffner
100 ms	AUS	0	kein Durchgang	Durchgang	kein Durchgang	Durchgang
10 ms						
100 ms remanent	AUS	Istwert bleibt erhalten	kein Durchgang	Durchgang	Durchgang	kein Durchgang
10 ms remanent						

Nach Ablauf eines Timers bleibt der Kontaktzustand eines remanenten Timers so lange erhalten, bis dieser durch eine RST-Anweisung zurückgesetzt wird.

Es können keine negativen Werte (-32768 bis -1) als Sollwerte für Timer programmiert werden. Wurde als Sollwert 0 eingegeben, wird der Timer so abgearbeitet, als sei der Wert 1 vorgegeben.

Die Ausführung der OUT(H) T-Anweisung hat folgende Auswirkungen:
Die Timerspule des in d angegebenen Timers wird gesetzt oder zurückgesetzt.
Der entsprechende Timerkontakt wird gesetzt oder zurückgesetzt.
Die Einstellwerte der Timer werden aktualisiert.

Wird während der Ausführung einer OUT(H) T-Anweisung zu der Anweisung gesprungen, werden die Kontaktzustände und Timerwerte nicht verändert.

Bei mehrmaliger Verwendung derselben Anweisung in einem Zyklus, wird der Wert der Wiederholungen aktualisiert.

Die Adressierung über Index-Register (indizierte Adressierung) der Counter-Spulen und Kontakte kann nur über die Index-Register Z0 und Z1 erfolgen.

HINWEISE

Das Register für den Timer-Sollwert darf nicht indirekt adressiert werden!

Weitere Einzelheiten über die Programmierung und Arbeitsweise von Timern sind dem Abs. A.3.1 „Timer-Vergleich“ dieser Programmieranleitung zu entnehmen.

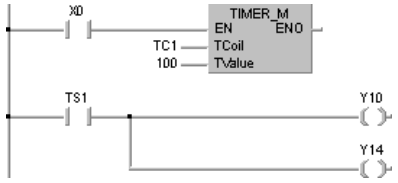
Beispiel 1 OUT T

Im folgendem Programm werden 10 Sekunden nach dem Einschalten von X0 die Ausgänge Y10 und Y14 gesetzt. Hier wird ein langsamer Timer (100 ms) verwendet.

MELSEC-Anweisungsliste

```
MELSEC
LD    X0
OUT   T1
      K100
LD    T1
OUT   Y10
OUT   Y14
```

Kontaktplan



IEC-Anweisungsliste

```
LD      X0
TIMER_M TC1 , 100
LD      TS1
ST      Y10
ST      Y14
```

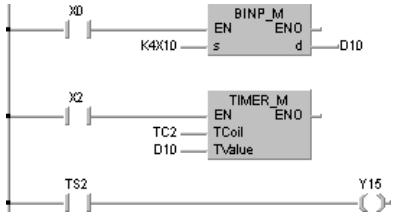
Beispiel 2 OUT T

Im folgenden Programm wird der Einstellwert des Timers über die Eingänge X10 bis X1F in Form von BCD-Daten eingelesen. Dazu werden die BCD-Daten zunächst mit der positiven Flanke von X0 in Binärdaten umgewandelt und in D10 gespeichert. Nach dem Einschalten von X2 wird der Einstellwert des Timers übernommen. Nach Ablauf der Zeit wird Y15 gesetzt. Hier wird ebenfalls ein langsamer Timer (100 ms) verwendet.

MELSEC-Anweisungsliste

```
MELSEC
LD    X0
BINP  K4X10
      D10
LD    X2
OUT   T2
      D10
LD    T2
OUT   Y15
```

Kontaktplan



IEC-Anweisungsliste

```
LD      X0
BINP_M  K4X10,D10
LD      X2
TIMER_M TC2,D10
LD      TS2
ST      Y15
```

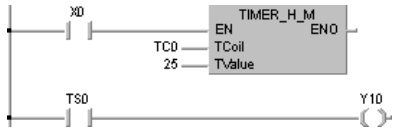
Beispiel 3 OUTH T

Im folgenden Programm wird der Ausgang Y10 250 ms nach Einschalten von X10 gesetzt. Hier wird ein schneller Timer (10 ms) verwendet.

MELSEC-Anweisungsliste

```
MELSEC
LD    X0
OUTH  T0
      K25
LD    T0
OUT   Y10
```

Kontaktplan



IEC-Anweisungsliste

```
LD      X0
TIMER_H_M TC0 , 25
LD      TS0
ST      Y10
```

5.3.3 OUT C

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
d									●								●	●					1			
●[1]										●[2]							●[2]	●								

[1] Einstellwert

[2] Zur Verwendung der erweiterten Timer und Counter einer AnA, AnAS oder AnU CPU sind die Hinweise in Abs. A.3.1 „Timer-Vergleich“ und Abs. A.3.2 „Counter-Vergleich“ dieser Programmieranleitung zu beachten.

Operanden MELSEC Q

	Operanden								Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K	Andere		
	Bit	Wort		Bit	Wort				U		
d	●[1]	—	—	—	—	—	—	—	—	—	4
●[2]	—	●[3]	●	—	●	●	—	●[4]	—	—	4

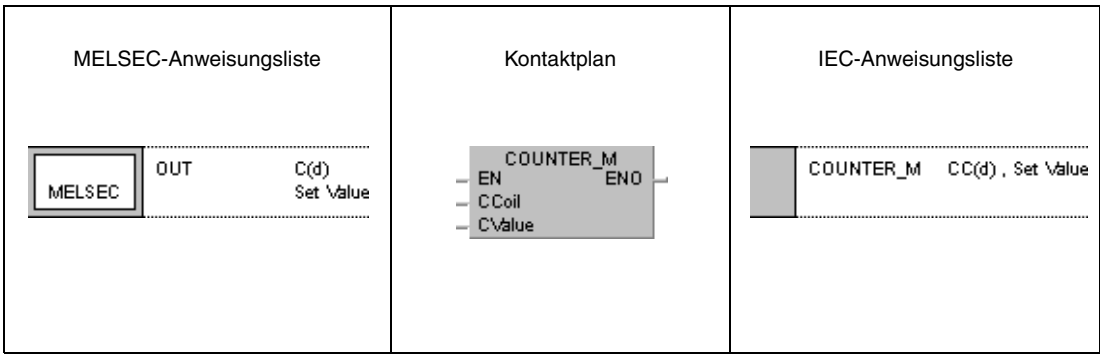
[1] Nur C

[2] Einstellwert

[3] Außer T und C

[4] Die Einstellwerte für Zähler können nur in Form einer dezimalen Konstanten (K) angegeben werden. Hexadezimale Konstanten (H) oder reelle Zahlen können nicht verwendet werden.

GX IEC Developer



GX Developer

Variablen

Operand	Befehlswert	Datentyp
d	Counternummer	Bit
Einstellwert	Einstellwerte (Set Value) für Counter	BIN-16-Bit

Funktionsweise

Setzen eines Counterkontaktes

OUT C Counter

Ist die Eingangsbedingung einer OUT C-Anweisung gegeben, wird der Istwert (Zählwert) des Counters um 1 erhöht.

Der Schaltzustand des OUT C-Kontaktes wird in einem (oder mehreren) nachfolgenden Programmschritt(en) als Eingangsbedingung in Form eines herkömmlichen Schließer- oder Öffnerkontaktes programmiert.

Hat der Counter seinen Sollwert erreicht, wird der nachgeschaltete Eingangskontakt gesetzt.

Bleibt die Eingangsbedingung der OUT C-Anweisung eingeschaltet, wird der Zählvorgang nicht fortgesetzt. Es ist daher nicht notwendig, den Zähleingang als Puls-Eingang auszuführen.

Nach Ablauf des Counters können der Zählwert und der Kontaktzustand erst durch Ausführung einer RST-Anweisung zurückgesetzt werden.

Werden die erweiterten Counter C256 bis C1023 in Verbindung mit einer AnA, AnAS oder AnU CPU eingesetzt, sind die Sollwerte entsprechend den Hinweisen in Abs. "Sollwerte von Erweiterten Timern und Countern" der Programmieranleitung zu programmieren.

Es können keine negativen Werte (-32768 bis -1) als Sollwerte für Counter programmiert werden. Wurde als Sollwert eine 0 eingegeben, wird der Counter so abgearbeitet, als sei der Wert 1 vorgegeben.

Die Adressierung über Index-Register (indizierte Adressierung) der Counter-Spulen und -Kontakte kann nur über die Index-Register Z0 und Z1 erfolgen.

HINWEISE *Die Einstellbereiche können nicht in direkt adressierten Datenregistern gespeichert werden.*

Weitere Einzelheiten über die Programmierung und Arbeitsweise von Countern sind dem Abs. A.3.2 „Counter-Vergleich“ dieser Programmieranleitung zu entnehmen.

Beispiel 1 OUT C

Im folgenden Programm wird Y30 eingeschaltet, nachdem X0 zehn mal gesetzt wurde, und wieder ausgeschaltet, wenn X1 gesetzt wird.

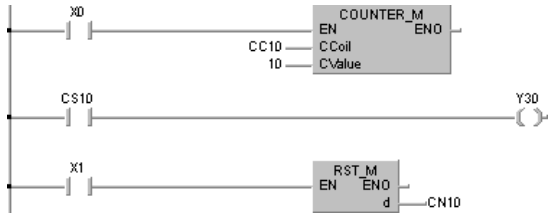
MELSEC-Anweisungsliste

```
MELSEC   LD      X0
         OUT     C10
                 K10
         LD      C10
         OUT     Y30
         LD      X1
         RST     C10
```

IEC-Anweisungsliste

```
         LD      X0
         COUNTER_M  CC10 , 10
         LD      CS10
         ST      Y30
         LD      X1
         RST_M   CN10
```

Kontaktplan



Beispiel 2 OUT C

Im folgenden Programm wird der Sollwert von C10 mit positiver Flanke von X0 auf 10 gesetzt (D0 =10) und mit positiver Flanke von X1 auf 20 gesetzt (D0 = 20). Der Counter beginnt entsprechend dem Sollwert in D0 zu zählen, wenn X3 eingeschaltet wird, und schaltet Y30 ein, sobald der Zähler seinen Sollwert erreicht hat.

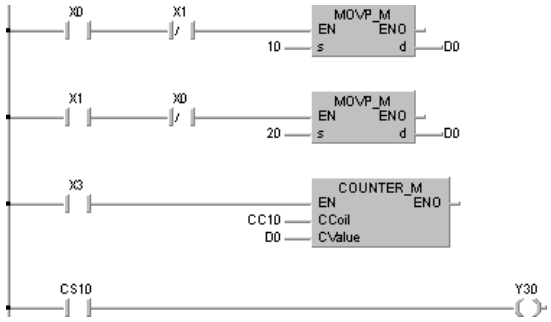
MELSEC-Anweisungsliste

```
MELSEC
        LD      X0
        ANI     X1
        MOVP    K10
                D0
        LD      X1
        ANI     X0
        MOVP    K20
                D0
        LD      X3
        OUT     C10
                D0
        LD      C10
        OUT     Y30
```

IEC-Anweisungsliste

```
        LD          X0
        ANDN        X1
        MOVP_M      10 , D0
        LD          X1
        ANDN        X0
        MOVP_M      20 , D0
        LD          X3
        COUNTER_M   CC10 , D0
        LD          CS10
        ST          Y30
```

Kontaktplan



5.3.4 OUT F

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
d							●																●[1]	●[2]		

[1] Generell 1 Schritt, aber 3 Schritte bei Programmierung von Sonder- oder Fehlermerkern als Operand in der OUT-Anweisung. Die Anzahl der Schritte bei der Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

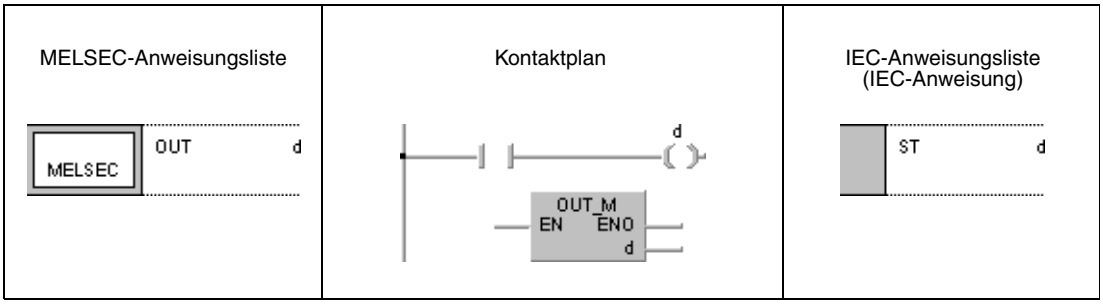
[2] Die Index-Funktion ist nur bei AnA, AnAS oder AnU CPUs verfügbar.

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
d	●[1]	—	—	—	—	—	—	—	—	—	4

[1] Nur F

GX IEC Developer



GX Developer

Variablen

Operand	Befehlswert	Datentyp
d	Einzuschaltender Fehlermerker	Bit (nur F-Merker)

Funktionsweise

Ausgabe von Fehlermerkern

OUT F Fehlermerker (Q-Serie und System Q)

Ist die Eingangsbedingung einer OUT F-Anweisung gegeben, schaltet der Fehlermerker ein und folgender Ablauf erfolgt:

Die Adresse des Fehlermerkers wird an der LED-Anzeige der CPU (Q3A und Q4AR) ausgegeben, und die "USER"-LED leuchtet auf.

Die Adressen der eingeschalteten Fehlermerker werden in den Diagnoseregistern SD64 bis SD79 gespeichert.

Der Wert in SD63 wird um 1 erhöht.

Ist in Diagnoseregister SD63 der Wert 16 erreicht, das heißt, es sind 16 Adressen von eingeschalteten Fehlermerkern gespeichert, wird in dem Bereich von SD64 bis SD79 keine weitere Adresse gespeichert.

Wird ein Fehlermerker über eine OUT-Anweisung ausgeschaltet, ändert dies nichts an der LED-Anzeige, am Zustand der "USER"-LED oder am Inhalt der Diagnoseregister SD63 bis SD79.

Das Löschen von Fehlermerkern, Registern und Anzeigen erfolgt mit der RST F-Anweisung.

OUT F Fehlermerker (A-Serie)

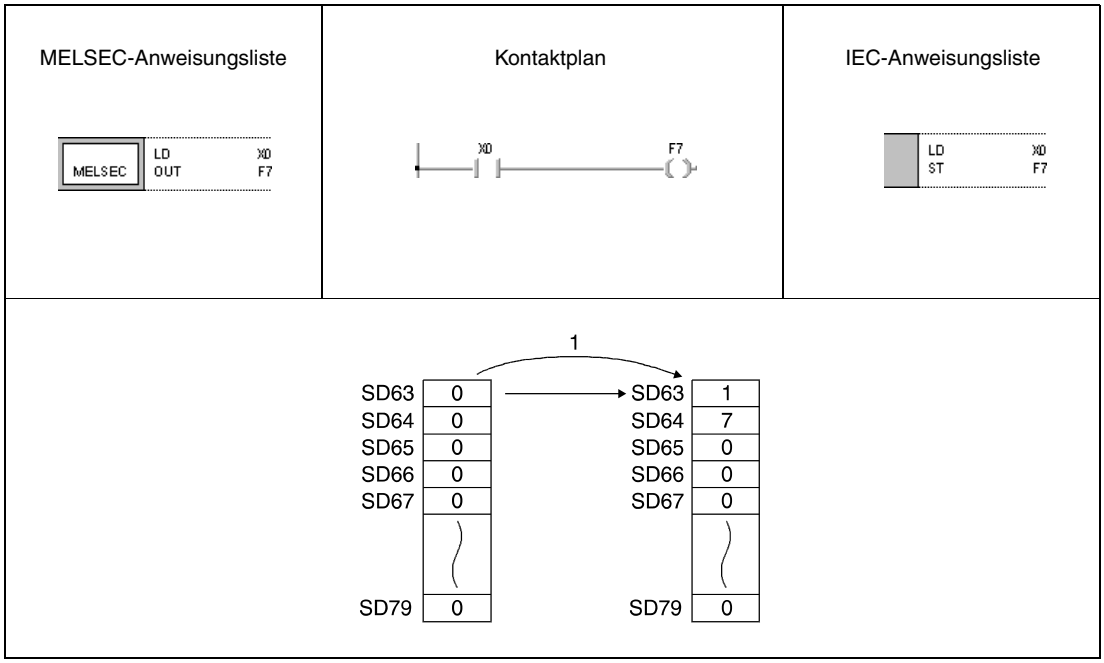
Wird im Programm ein Fehlermerker (F) gesetzt, leuchten die ERROR-LED am CPU-Modul sowie entsprechende LED-Anzeigen an den Steuerungen auf. Die Anzahl der eingeschalteten Fehlermerker wird in einem Sonderregister gespeichert.

Fehlermerker dürfen nicht über eine OUT-Anweisung gesetzt werden, da die LED-Fehleranzeige nicht mit dem Kontaktzustand der Ausgangsanweisung übereinstimmt. Um dieses zu vermeiden, sollte ein Fehlermerker mit Hilfe einer SET-Anweisung gesetzt werden. Das Einschalten eines Fehlermerkers über eine OUT-Anweisung hat auch zur Folge, dass bei einem Wegfall der Eingangsbedingung auch der Fehlermerker ausgeschaltet wird. Der Anzeigezustand der LED-Anzeigen und der ERROR-LED am CPU-Modul sowie der Inhalt der Sonderregister ändern sich hierdurch nicht.

Beispiel (Q-Serie)

OUT F

Im folgenden Programm wird mit Einschalten von X0 der Fehlermerker F7 eingeschaltet. Der Wert 7 wird in den Registern SD64 bis SD79 gespeichert. Der Wert in Register SD63 wird um 1 erhöht (eine Adresse eines Fehlermerkers gespeichert).



[1] X0 wird eingeschaltet.

5.3.5 SET

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
d		●	●	●	●	●																	1 ●[1]	●[2]		

[1] Die Anzahl der Schritte ist 3 Schritte, wenn Sonder-, Link- oder Fehlermerker (M, B oder F) mit einer SET-Anweisung gesetzt oder ein Sondermerker oder ein beliebiger Wortoperand zurückgesetzt werden.

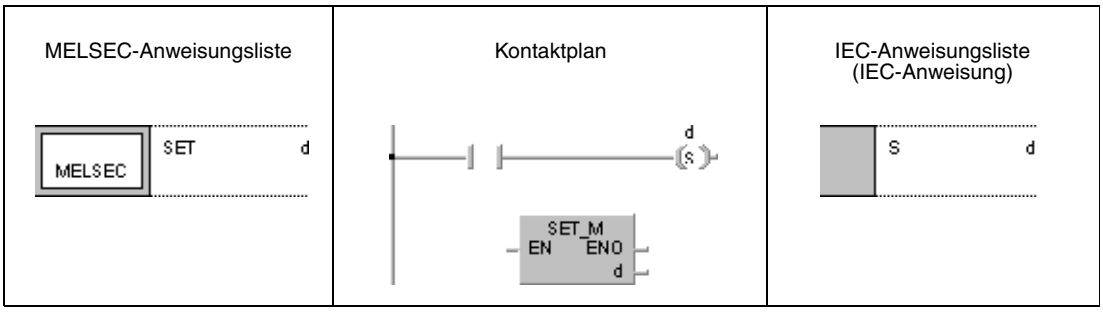
[2] Die Index-Funktion ist nur bei AnA, AnAS oder AnU CPUs verfügbar.

Operanden MELSEC Q

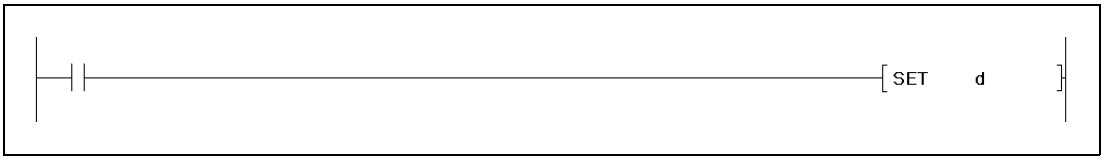
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sonder-module U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort				BL	DY	
d	●	●	●	●	●	●	—	—	●	●	—	1 ●[1]

[1] 1 Schritt bei der Verwendung interner Operanden, 2 Schritte bei Verwendung direkt adressierbarer Ausgänge DY oder der SFC-Programmoperanden (BL), 3 Schritte bei der Verwendung aller anderen Operanden (inkl. der Fileregister mit seriellem Zugriff) und 4 Schritte bei der Verwendung von Timern (T) oder Countern (C).

GX IEC Developer



GX Developer



Variablen

Operand	Befehlswert	Datentyp
d	Einzuschaltender Ausgangskontakt / Bit-Bestimmung eines Wortoperanden	Bit

Funktionsweise

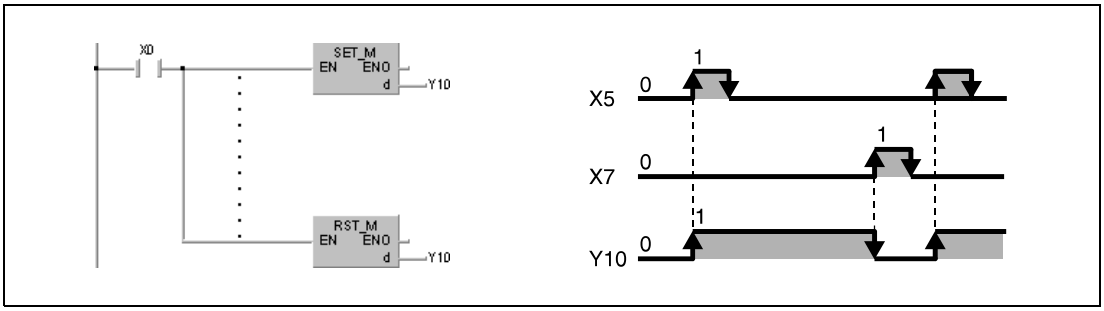
Setzen eines Operanden

SET Setzanweisung

Die SET-Anweisung besteht aus dem SET-Befehl (Setzen) gefolgt von der Operandenadresse d, die gesetzt werden soll.

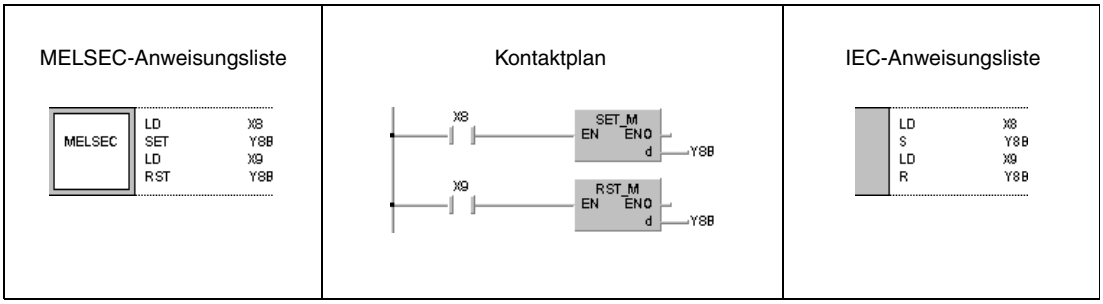
Nach Ausführung der Eingangsbedingung werden die SET-Anweisung und die angegebene Operandenadresse d gesetzt oder das bestimmte Bit des Wortoperanden auf 1 gesetzt.

Wird die Eingangsbedingung wieder ausgeschaltet, bleibt der gesetzte Operand auch weiterhin gesetzt. Der Operand kann mit Hilfe einer RST-Anweisung zurückgesetzt werden.



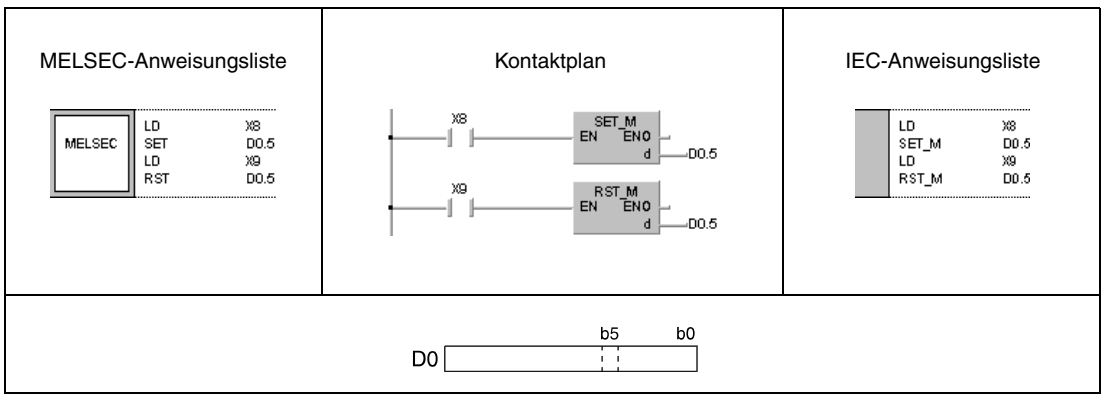
Beispiel 1 SET

Im folgenden Programm wird bei Einschalten von X8 der Ausgang Y8B gesetzt. Bei Einschalten von X9 wird Y8B zurückgesetzt.



Beispiel 2 SET

Im folgenden Programm wird bei Einschalten von X8 das Bit 5 (b5) in D0 von 0 auf 1 gesetzt. Bei Einschalten von X9 wird dieses Bit zurückgesetzt.



5.3.6 RST

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)								Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
d		●	●	●	●	●		●	●	●	●	●	●	●	●	●							1 ●[1]	●[2]		

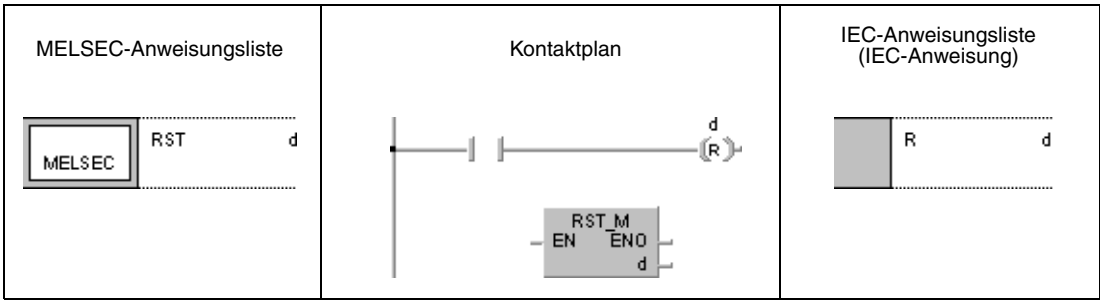
[1] Die Anzahl der Schritte ist 3 Schritte, wenn Sonder-, Link- oder Fehlermerker (M, B oder F) mit einer SET-Anweisung gesetzt oder ein Sondermerker oder ein beliebiger Wortoperand zurückgesetzt werden.

[2] Die Index-Funktion ist nur bei AnA, AnAS oder AnU CPUs verfügbar.

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort				BL	DY	
d	●	●	●	●	●	●	●	—	—	●	—	2

GX IEC Developer



GX Developer

Variablen

Operand	Befehlswert	Datentyp
d	Zurückzusetzender Ausgangskontakt / Bit-Bestimmung eines Wortoperanden	Bit ●[1]

[1] Eine Besonderheit der RST_M-Anweisung besteht in der Möglichkeit, ganze Datenwörter zu löschen. Dabei werden weniger Schritte benötigt als bei Verwendung der MOV-Anweisung in Verbindung mit der Konstanten K0.

Funktionsweise

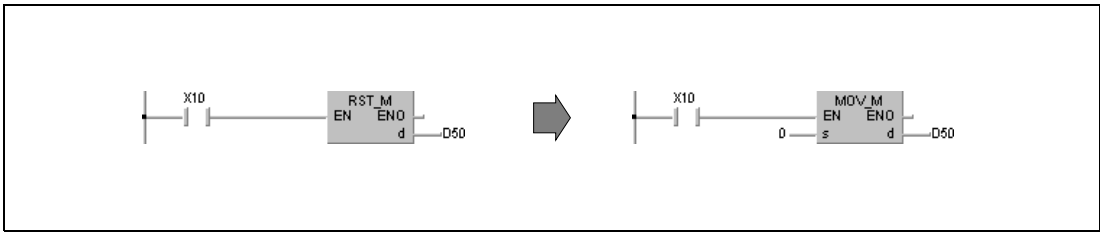
Zurücksetzen eines Operanden

RST Rücksetzanweisung

Die RST-Anweisung besteht aus dem RST-Befehl (Rücksetzen) gefolgt von der Operandenadresse, die zurückgesetzt werden soll.

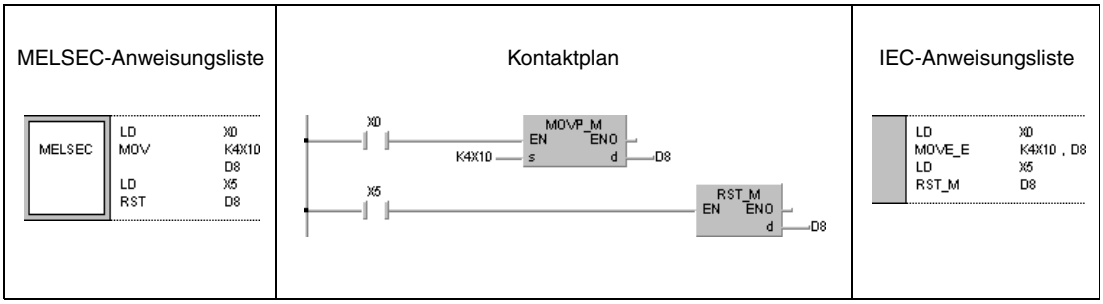
Nach Ausführung der RST-Anweisung werden Ein- und Ausgangskontakte von Bit-Operanden ausgeschaltet, Istwerte von Timern und Countern (T, C) auf 0 gesetzt sowie die zugehörigen Kontakte ausgeschaltet, das bestimmte Bit des Wortoperanden auf 0 gesetzt und der Inhalt von Wortoperanden auf 0 gesetzt.

Die Funktion der RST-Anweisung in der folgenden Abbildung ist mit der Funktion der rechts angegebenen MOV-Anweisung identisch. X10 fungiert als RST-Eingang.



Beispiel 1 RST

Im folgenden Programm wird der Inhalt von X10 bis X1F mit positiver Flanke von X0 in das Datenregister D8 geschrieben. Nach Einschalten von X5 wird der Inhalt von D8 auf 0 zurückgesetzt.



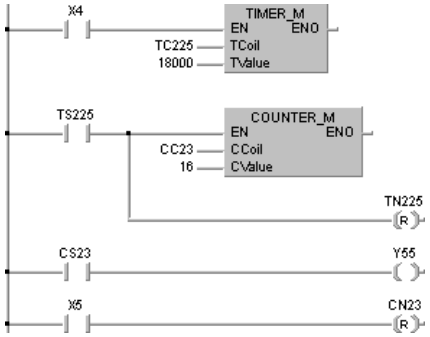
Beispiel 2 RST T, C

Im folgenden Programm wird ein Beispiel zum Rücksetzen von remanenten Timern und Countern gezeigt. In der ersten Programmzeile wird T225 gesetzt, wenn X4 für 30 Minuten eingeschaltet ist. Im zweiten Schritt werden die Einschaltmomente von T225 über C23 gezählt. Gleichzeitig wird T225 zurückgesetzt. Sobald der Timer 16 mal eingeschaltet wurde (Istwert von C23 = 16), wird Ausgang Y55 eingeschaltet. Der Zähler wird nach dem Einschalten von X5 auf 0 gesetzt.

MELSEC-Anweisungsliste

```
MELSEC
        LD      X4
        OUT     T225
                K18000
        LD      T225
        OUT     C23
                K16
        RST     T225
        LD      C23
        OUT     Y55
        LD      X5
        RST     C23
```

Kontaktplan



IEC-Anweisungsliste

```
        LD      X4
        TIMER_M TC225 , 18000
        LD      TS225
        COUNTER_M CC23 , 16
        R       TN225
        LD      CS23
        ST      Y55
        LD      X5
        R       CN23
```

5.3.7 SET F, RST F

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
d							●																3	●[1]		

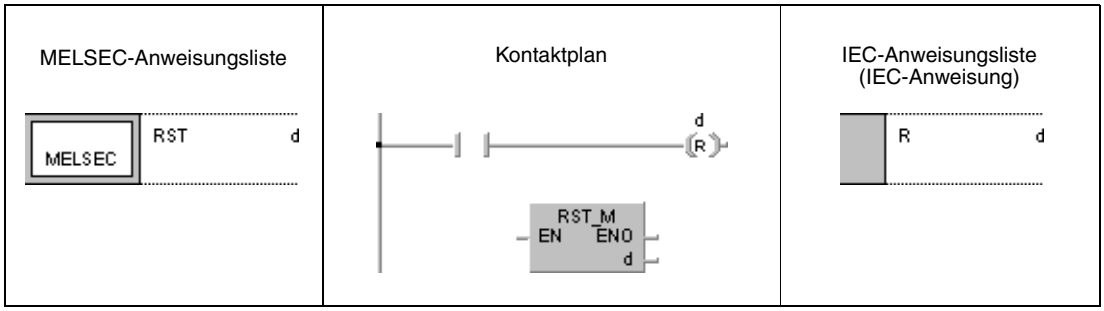
[1] Die Index-Funktion ist nur bei AnA, AnAS oder AnU CPUs verfügbar.

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sonder-module U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort				U		
d	●[1]	—	—	—	—	—	—	—	—	—	3

[1] Nur F

GX IEC Developer



GX Developer

Variablen

Operand	Befehlswert	Datentyp
d (SET)	Zu setzender Fehlermerker.	Bit (nur F)
d (RST)	Zurückzusetzender Fehlermerker.	Bit (nur F)

Funktionsweise

Setzen/Rücksetzen von Fehlermerkern (Q-Serie und System Q)

SET F Setzanweisung

Die SET F-Anweisung besteht aus dem SET-Befehl (Setzen) gefolgt von der Operandenadresse d, die gesetzt werden soll. Nach Ausführung der Eingangsbedingung werden die SET-Anweisung und die angegebene Operandenadresse d gesetzt. Das Ausgangsziel der SET-Anweisung zum Setzen des Fehlermerkers ist ein Impuls.

Folgender Ablauf erfolgt:

Die Adresse des Fehlermerkers wird an der LED-Anzeige der CPU (Q3A und Q4AR) ausgegeben, und die "USER"-LED leuchtet auf.
Die Adressen der eingeschalteten Fehlermerker werden in den Registern SD64 bis SD79 gespeichert.
Der Wert in SD63 wird um 1 erhöht.

Ist in Register SD63 der Wert 16 erreicht, das heißt, es sind 16 Adressen von eingeschalteten Fehlermerkern gespeichert, wird in dem Bereich von SD64 bis SD79 keine weitere Adresse gespeichert.

RST F Rücksetzanweisung

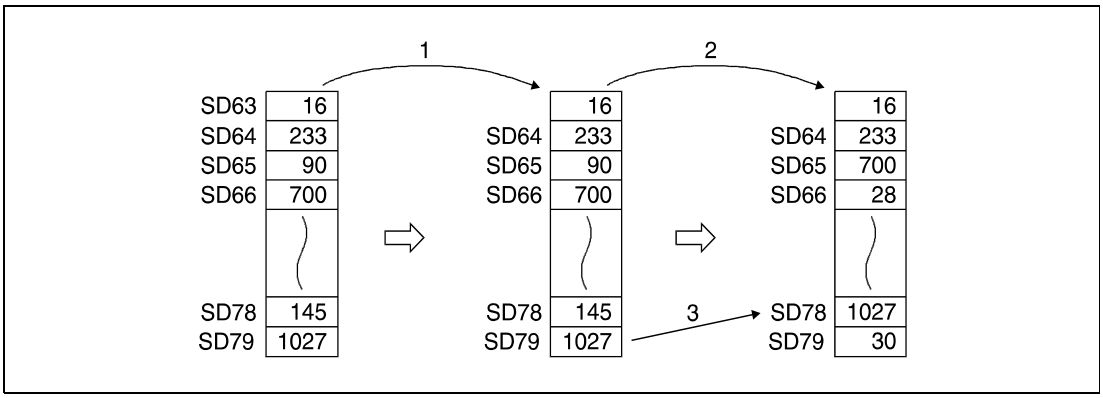
Die RST F-Anweisung besteht aus dem RST-Befehl (Rücksetzen) gefolgt von der Operandenadresse d, die gesetzt werden soll.

Nach Ausführung der Eingangsbedingung wird die RST-Anweisung gesetzt und die angegebene Operandenadresse d zurückgesetzt. Das Ausgangssignal zum Rücksetzen eines Fehlermerkers ist ein Impuls.

Die Adresse eines ausgeschalteten Fehlermerkers wird aus den Registern SD64 bis SD79 gelöscht und der Wert im Register SD63 wird um 1 vermindert. War der Wert im Register SD63 gleich 16 und werden über die RST-Anweisung Fehlermerker aus den Registern SD64 bis SD79 gelöscht, werden nun die Fehlermerker eingeschaltet, die zuvor nicht mehr gespeichert werden konnten. Die Adressen der Fehlermerker werden dann in den freigewordenen Registern zwischen SD64 und SD79 gespeichert.

Ist nach Ausführung der RST F-Anweisung der Wert im Diagnoseregister SD63 auf 0 gesunken und sind alle Fehlermerker ausgeschaltet, erlöschen die LED-Anzeige und die "USER"-LED.

In der unteren Abbildung wird F30 im ersten Schritt gesetzt (1), kann aber nicht eingetragen werden, da bereits 16 Adressen von SD64 bis SD79 gespeichert sind. Im zweiten Schritt (2) wird F90 zurückgesetzt. Dadurch ist die Speicherung von F30 in SD79 möglich, da die gespeicherten Fehlermerker um das frei gewordene Register (SD65) nach oben gesetzt werden (3).



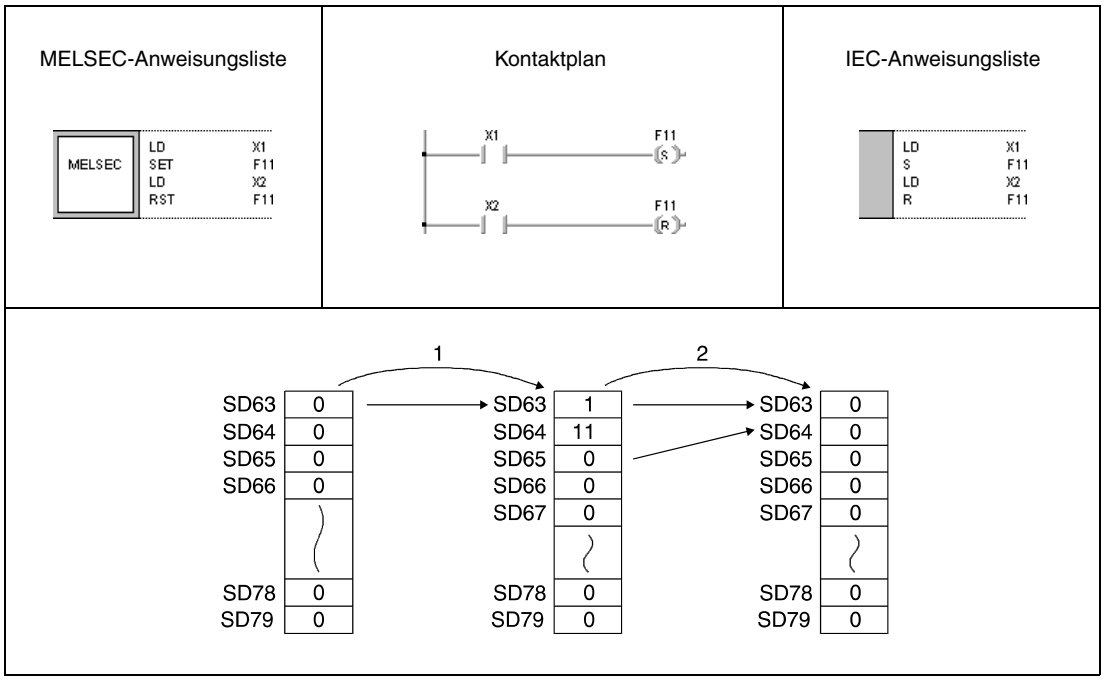
Setzen/Rücksetzen von Fehlermerkern (A-Serie)

SET F/ RST F Setz-/Rücksetzanweisung

Wird mit Hilfe einer SET-RST-Anweisung ein Fehlermerker F gesetzt/zurückgesetzt, ändern sich die entsprechenden LED-Anzeigen und der Zustand der Error-LED an der CPU sowie der Inhalt der zugehörigen Sonderregister. Fehlermerker werden über Impulse gesetzt/zurückgesetzt.

Beispiel SET F/ RST F (Q-Serie und System Q)

Im folgenden Programm wird der Fehlermerker F11 eingeschaltet, wenn X1 gesetzt wird (1). Der Wert 11 wird in einem Register von SD64 bis SD79 gespeichert, und der Wert in SD63 wird um 1 erhöht. Danach wird der Fehlermerker F11 ausgeschaltet, wenn X2 gesetzt wird (2). Der Wert 11 wird aus dem Diagnoseregister zwischen SD64 und SD79 gelöscht, und der Wert in SD63 wird um 1 reduziert.



5.3.8 PLS, PLF

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)								Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
d		●	●	●	●	●	●																3 ●[1]	●[2]		

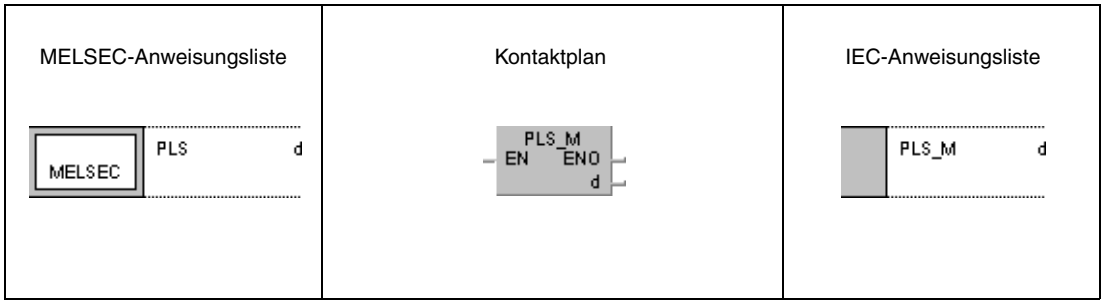
[1] Die Anzahl der Schritte bei Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

[2] Die Index-Funktion ist nur bei AnA, AnAS oder AnU CPUs verfügbar.

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort				DY		
d	●	●	●	●	●	●	—	—	●	—	2

GX IEC Developer



GX Developer

Variablen

Operand	Befehlswert	Datentyp
d	Operand, dessen Ausgangssignal in einen Impuls umgewandelt wird.	Bit

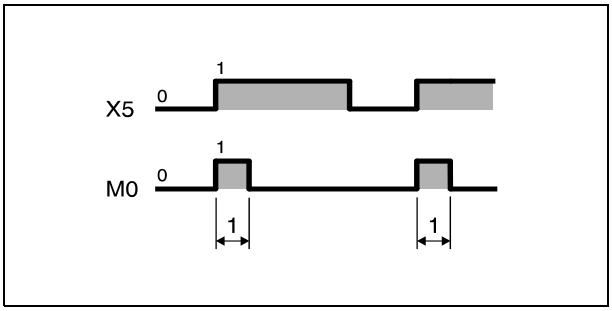
Funktions-weise

Flankengesteuerte Differenzausgabe

PLS Ausgabe bei steigender Signalflanke

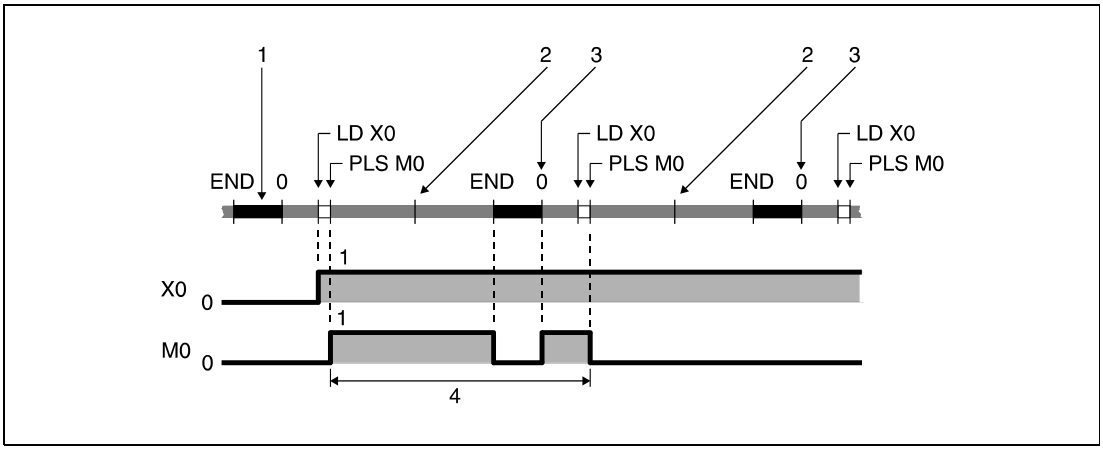
Die PLS-Anweisung besteht aus dem PLS-Befehl gefolgt von der Operandenadresse d, die gesetzt werden soll.

Die PLS-Anweisung (Puls; Impuls) schaltet den Operanden bei ansteigender Signalflanke (positive Flanke) der Eingangsbedingung für die Dauer eines Programmzyklus ein. Ist der vorgegebene Operand bereits gesetzt, wird dieser Operand für die Dauer eines Programmzyklus ausgeschaltet.



[1] Ein Zyklus

Ist die Anweisung, die den Impuls erzeugt, gesetzt und wird dann der Betriebsartenschalter der CPU von RUN auf STOP und anschließend wieder auf RUN gesschaltett, wird die PLS-Anweisung nicht ausgeführt.



[1] END-Verarbeitung

[2] Betriebsartenschalter RUN → STOP

[3] Betriebsartenschalter STOP → RUN

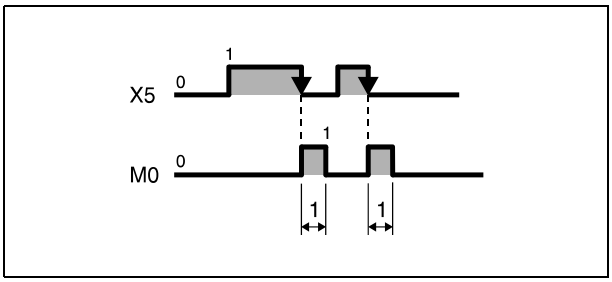
[4] Ein Zyklus von PLS M0

Wird ein Latch-Merker L in Verbindung mit einer PLS-Anweisung programmiert, wird nach dem Wiedereinschalten der CPU im ersten Zyklus durch die PLS-Anweisung ein Impuls erzeugt, wenn der Latch-Merker vor dem Ausschalten der CPU gesetzt war. Im folgendem Zyklus wird der durch die PLS-Anweisung angesteuerte Operand zurückgesetzt.

PLF Ausgabe bei fallender Signalflanke

Die PLF-Anweisung besteht aus dem PLF-Befehl gefolgt von der Operandenadresse d, die gesetzt werden soll.

Die PLF-Anweisung schaltet den Operanden bei fallender Signalflanke (negative Flanke) der Eingangsbedingung für die Dauer eines Programmzyklus ein. Ist der vorgegebene Operand bereits gesetzt, wird dieser Operand für die Dauer eines Programmzyklus ausgeschaltet.



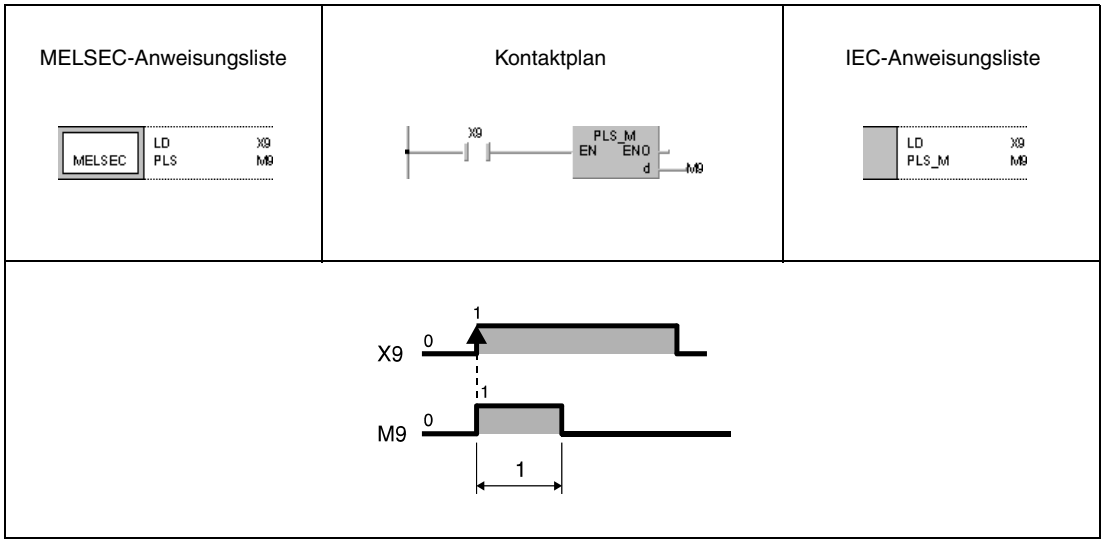
[1] Ein Zyklus

Ist die Anweisung, die den Impuls erzeugt, gesetzt und wird der Betriebsartenschalter der CPU von RUN auf STOP und anschließend wieder auf RUN gesetzt, wird die PLF-Anweisung nicht ausgeführt.

HINWEIS *Wenn die PLS- oder PLF-Anweisung innerhalb eines Sprungbefehls (mit CJ-Anweisung) programmiert wird, kann der in der Anweisung angegebene Operand bei Nicht-Ausführung des Programmteils für mehr als einen Programmzyklus gesetzt sein.*

Beispiel 1 PLS

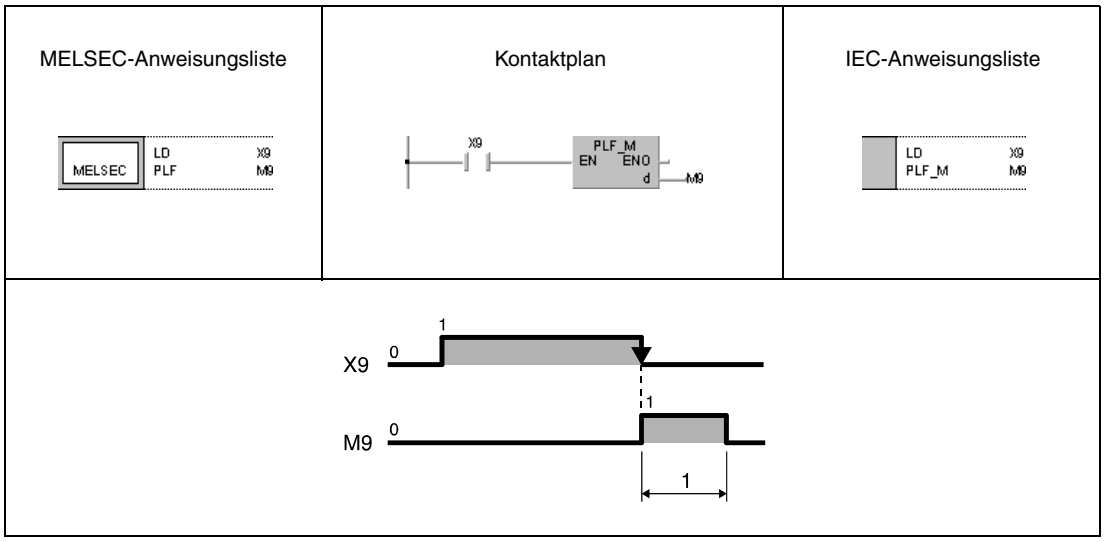
Im folgenden Programm wird mit positiver Flanke von X9 der Merker M9 für einen Programmzyklus gesetzt.



[1] Ein Zyklus

Beispiel2 PLF

Im folgendem Programm wird mit negativer Flanke von X9 der Merker M9 für einen Programmzyklus gesetzt.



[1] Ein Zyklus

5.3.9 FF

CPU

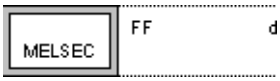
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Operanden MELSEC Q

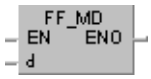
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort				DY		
d	●	●	●	●	●	●	—	—	●	—	2

GX IEC Developer

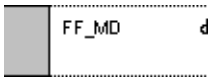
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

Operand	Befehlswert	Datentyp
d	Bit-Operand oder adressiertes Bit eines Wortoperanden, dessen Zustand invertiert wird.	Bit

Funktionsweise

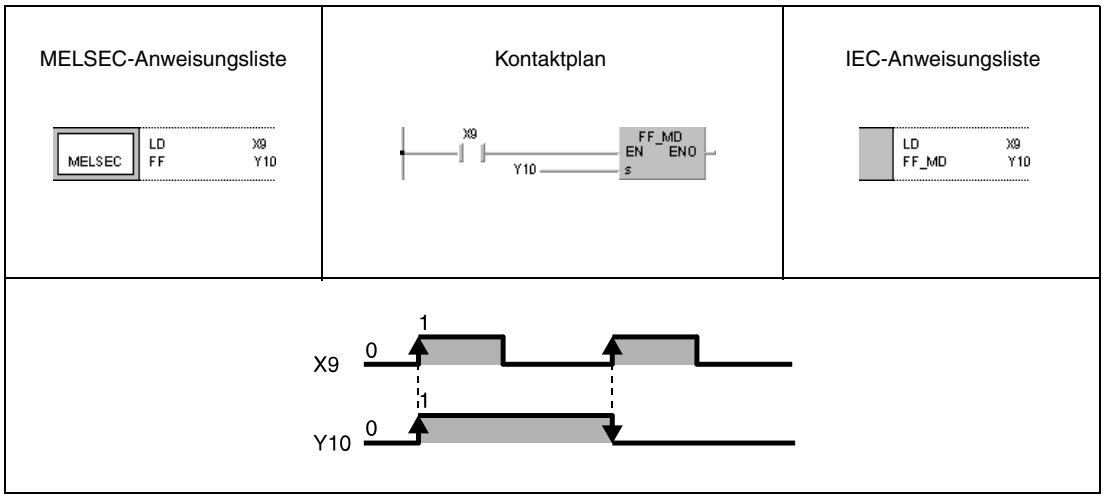
Umkehr des Schaltzustandes eines Bit-Ausgangsoperanden

FF Invertierung eines Bit-Ausgangsoperanden

Die FF-Anweisung kehrt den Signalzustand des in d angegebenen Operanden bei ansteigender Flanke am Eingang der FF-Anweisung um. Der Operand kann ein Bit-Operand oder ein bestimmtes Bit eines Wortoperanden sein. War der Zustand des Ausgangsoperanden vorher 1, ist er nach der FF-Anweisung 0. War der Zustand des Ausgangsoperanden vor der Ausführung der Anweisung 0, ist der Ausgangszustand hinterher 1.

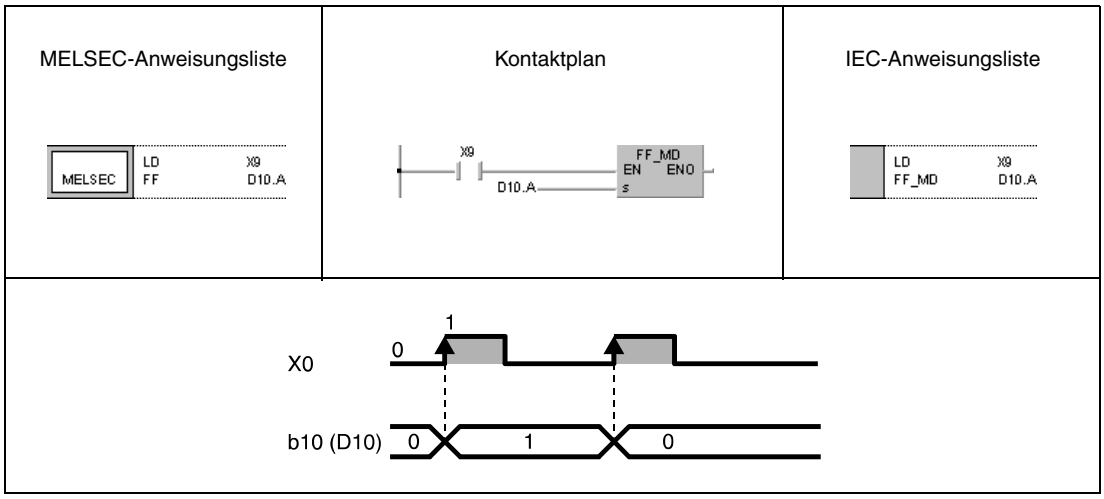
Beispiel 1 FF

Im folgenden Programm wird mit positiver Flanke von X9 der Ausgangszustand von Y10 invertiert.



Beispiel 2 FF

Im folgenden Programm wird mit positiver Flanke von X9 das Bit 10 (b10) von D10 invertiert.



5.3.10 CHK

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●				

Operanden MELSEC A

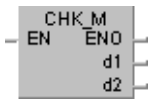
	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
d1		●	●	●	●	●	●																			
d2 ●[1]		●	●	●	●	●	●	●	●	●	●	●	●	●	●							K1 ↓ K4	5			

[1] Der Operand d2 hat keinen Einfluß auf die Programmverarbeitung.

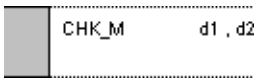
GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan



IEC-Anweisungsliste



Variablen

Operand	Befehlswert	Datentyp
d1	Bit-Operand, dessen Zustand invertiert wird.	Bit
d2	Scheinoperand (Dummy)	Bit

Funktions-weise

Umkehr des Schaltzustandes eines Bit-Ausgangsoperanden (A-Serie)

Allgemeine Hinweise

Die Funktion der CHK-Anweisung ist von der gewählten Verarbeitungsart abhängig. Bei Direktverarbeitung der E-/A-Zustände (außer AnA und A2C CPUs) dient die Anweisung zur Fehlerkontrolle. Bei AnS und AnN CPUs ermöglicht die CHK-Anweisung in der Verarbeitung nach dem Prozessabbild die Umkehr des Schaltzustandes eines Ausgangsoperanden (Flip-Flop).

CHK Invertierung eines Bit-Ausgangsoperanden

Eine vollständige CHK-Anweisung besteht aus einem CHK-Befehl, dem Operanden d1, dessen Schaltzustand umgekehrt werden soll, und einem Scheinoperand d2.

Sobald die Eingangsbedingung der CHK-Anweisung gegeben ist, ändert sich der Schaltzustand des in der CHK-Anweisung bestimmten Operanden. Nach Ausschalten und wiederholtem Einschalten der Eingangsbedingung wird der angegebene Operand wieder in seinen Ausgangszustand gesetzt.

Obwohl d2 nur ein sogenannter Scheinoperand ist, muss hier ein Operand gesetzt werden (siehe Operandentabelle). Wird für d2 ein Bit-Operand gesetzt, ist die Blocklänge mit K1 bis K4 festzulegen. Die Angabe des Wertes ist dabei belanglos, da die Blocklänge nur zum Schein gesetzt wird. Der in d2 programmierte Operand kann frei für andere Zwecke programmiert werden.

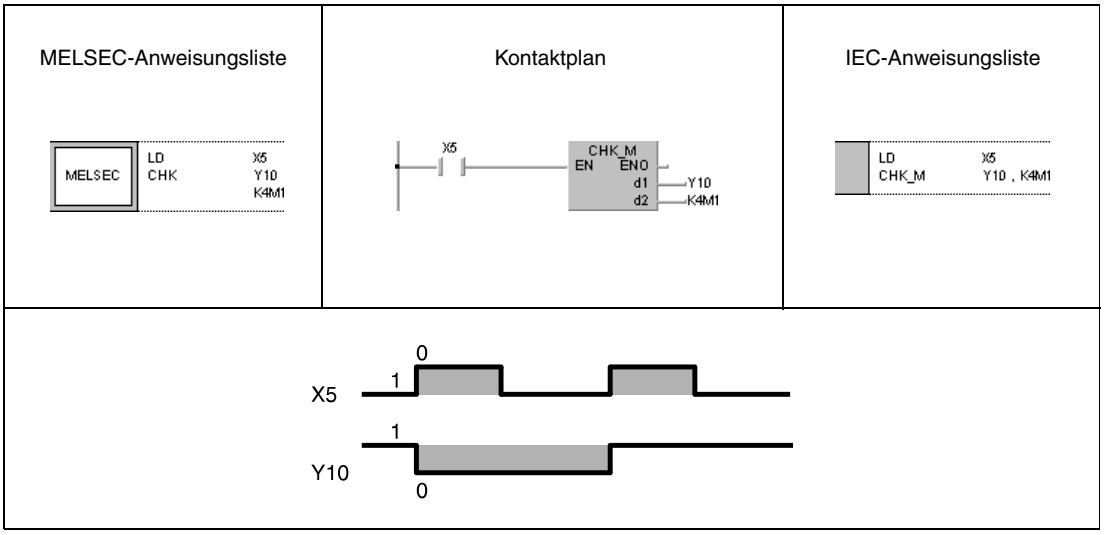
Die hier beschriebene CHK-Anweisung kann nur in der Verarbeitung nach dem Prozessabbild ausgeführt werden.

Die Umkehr des Schaltzustandes eines Ausgangsoperanden muss mindestens einen Programmzyklus lang andauern.

Beispiel

CHK

Im folgenden Programm wird mit positiver Flanke von X5 der Ausgangszustand von Y10 umgekehrt.



5.3.11 DELTA, DELTAP

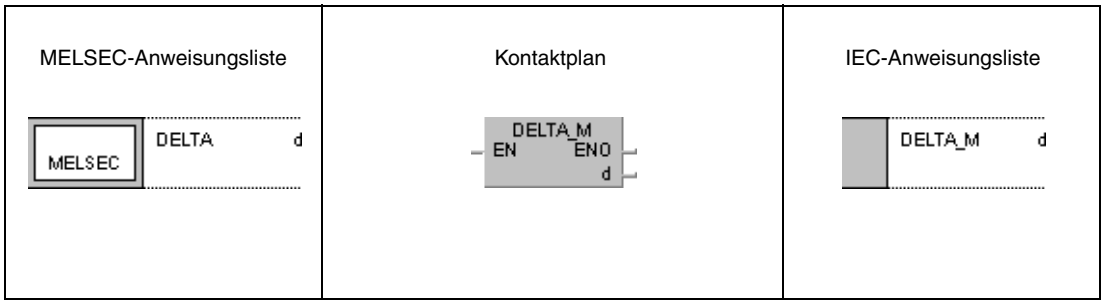
CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	Q
				●	●

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort				DY		
d	—	—	—	—	—	—	—	—	●	SM0	2

GX IEC Developer



GX Developer

Variablen

Operand	Befehlswert	Datentyp
d	Direkt adressierbarer Ausgang, auf den die Anweisung angewendet wird.	Bit ●[1]

[1] nur direkte Ausgänge

Funktionsweise

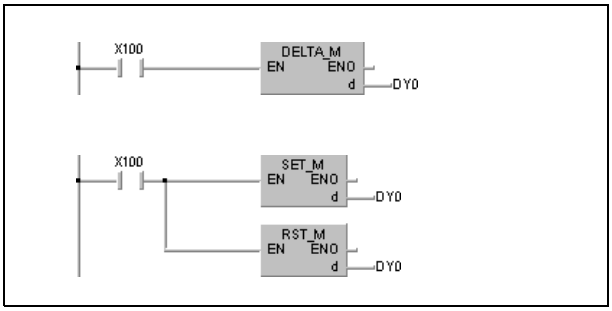
Erzeugung eines Schaltimpulses an einem direkt adressierbaren Ausgang

DELTA Anweisung für Schaltimpulse

Die DELTA-Anweisungen erzeugen an einem in d angegebenen direkt adressierbaren Ausgang (DY) einen Schaltimpuls, das bedeutet, der Ausgang wird für eine bestimmte Zeit gesetzt.

Wenn der von der DELTA-Anweisung angesprochnene Ausgang DY0 lautet, so ist die ausgeführte Funktion mit der durch die SET-/RST-Anweisung programmierten Funktion identisch (siehe Abbildung).

Die DELTA-Anweisung wird bei Sonderfunktionsbausteinen in Verbindung mit flankengesteuerten Ausführungsbedingungen angewendet.



Fehlerquellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

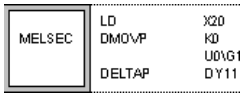
- Der in d angegebene Ausgang liegt außerhalb des Adressbereichs für Ausgänge (Fehlercode 4101).

Beispiel

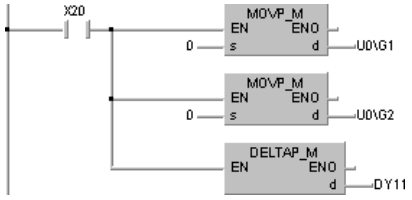
DELTAP

Im folgenden Programm wird mit positiver Flanke von X20 eine Voreinstellung von CH1 des Ausgabemoduls AD61 im Steckplatz 0 des Hauptbaugruppenträgers vorgenommen. Die Adressen 1 und 2 im Pufferspeicher des Ausgabemoduls werden mit 0 voreingestellt. Mit der Ausführung der DELTAP-Anweisung wird die Voreinstellung wirksam.

MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste

```
LD      X20
MOVP_M  0 , U0\G1
MOVP_M  0 , U0\G2
DELTAP_M DY11
```

5.4 Verschiebeanweisungen

5.4.1 SFT, SFTP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)								Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
d		●	●	●	●	●	●																3 ●[1]	●[2]		

[1] Die Anzahl der Schritte bei Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

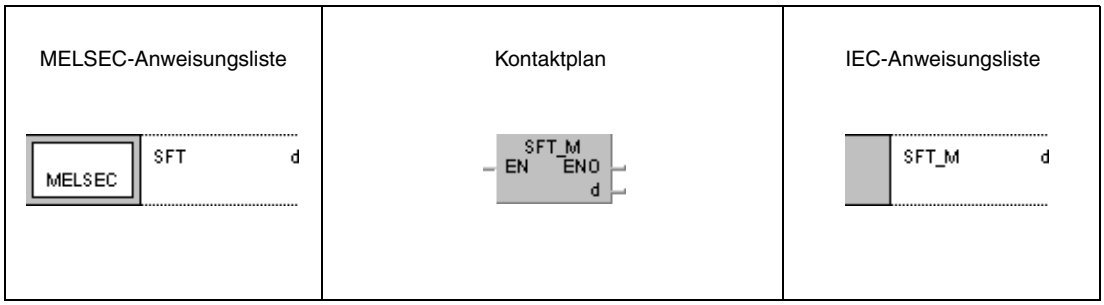
[2] Die Index-Funktion ist nur bei AnA, AnAS oder AnU CPUs verfügbar.

Operanden MELSEC Q

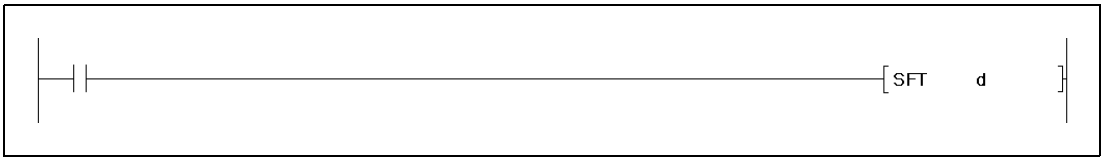
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sonder-module U□\G□	Index-Register Zn	Konstanten DY	Andere		
	Bit	Wort		Bit	Wort				U		
d	●[1]	●[1]	●[1]	●[1]	●[1]	●[1]	—	—	●	—	2

[1] Außer T und C

GX IEC Developer



GX Developer



Variablen

Operand	Befehlswert	Datentyp
d	Operand, der verschoben wird.	Bit

Funktions-weise

Verschiebeanweisung

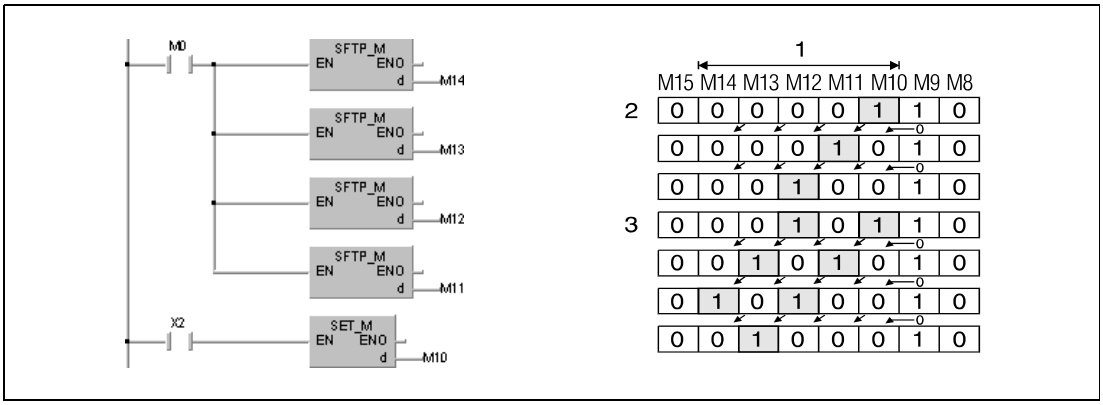
SFT Verschiebung von Bit-Operanden

Die SFT-Anweisung verschiebt Operanden um ein Bit. Bei Programmierung der SFT-Anweisung erfolgt das Verschieben nur dann, wenn die Eingangsbedingung gegeben ist (ansteigende Flanke).

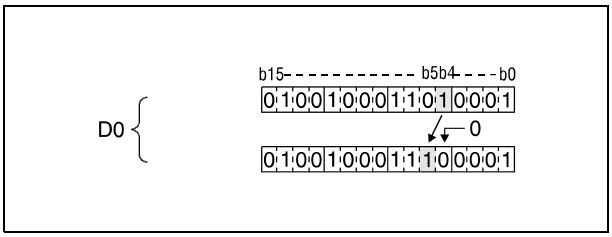
Bei Ausführung der Anweisung wird der Zustand einer Operandenadresse (festgelegt mit d-1) zur Zieladresse d verschoben. Der Zustand des Operanden mit der niedrigeren Adresse d-1 wird zurückgesetzt. Das anschließende Setzen der verschobenen Operandenadresse sollte mit einer SET-Anweisung erfolgen.

Bei Programmierung mehrerer SFT-Anweisungen in Folge ist mit der höheren Operandenadresse zu beginnen.

In der untenstehenden Abbildung wird beim Setzen von X2 (2,3) jeweils der Merker M10 gesetzt. Der Zustand von M10 (1) wird gemäß der Ausführung der SFT P-Anweisung im Verschiebebereich (1) verschoben.



Bei der Verschiebung von Bits in Wortoperanden wird der Zustand (0/1) des Bits d-1 nach d verschoben. Das Bit d-1 wird nach der SFT-Anweisung mit 0 beschrieben. In folgender Abbildung wird Bit 5 (b5) in D0 verschoben. Bit 4 (b4) wird nach Ausführung der Anweisung mit 0 beschrieben.



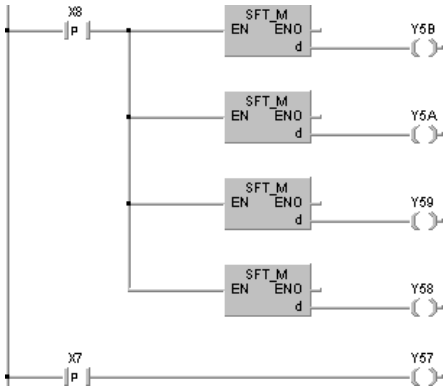
Beispiel SFT

Im folgenden Programm wird mit positiver Flanke von X8 der Zustand von Y57 nach Y5B verschoben. Mit der positiven Flanke von X7 wird Y57 gesetzt.

MELSEC-Anweisungsliste

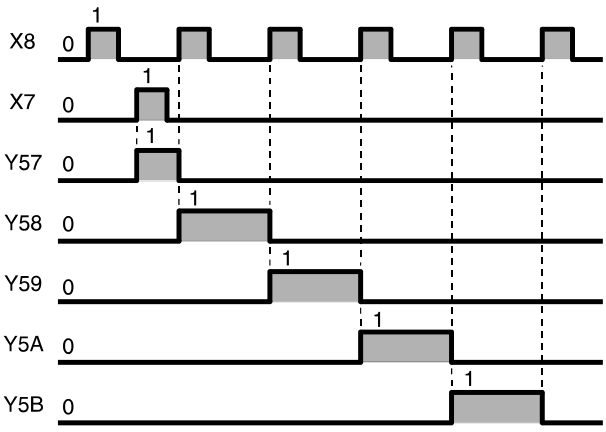
```
MELSEC
LDP   X8
SFT   Y5B
SFT   Y5A
SFT   Y59
SFT   Y58
LDP   X7
SET   Y57
```

Kontaktplan



IEC-Anweisungsliste

```
LD      X8
PLS_M   M8
LD      M8
SFT_M   Y5B
SFT_M   Y5A
SFT_M   Y59
SFT_M   Y58
LD      X7
PLS_M   M7
LD      M7
S       Y57
```



5.5 Master-Control-Anweisungen

5.5.1 MC, MCR

HINWEIS In den IEC-Editoren sollten diese Anweisungen nicht verwendet werden.

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
n																					●		3/5	●[1]		
d		●	●	●	●	●	●																●[2]			

[1] Die Index-Funktion ist nur bei AnA, AnAS oder AnU CPUs verfügbar.

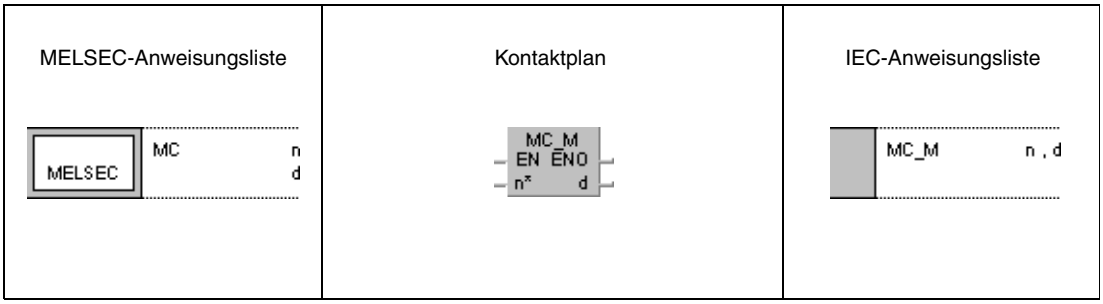
[2] Die Anzahl der Schritte beträgt 5 für die MC-Anweisung und 3 für die MCR-Anweisung. Näheres zur Anzahl der Schritte bei Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

Operanden MELSEC Q

	Operanden										Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□N□		Sondermodule U□G□	Index-Register Zn	Konstanten K, H (16#)	Andere			
	Bit	Wort		Bit	Wort				N	DY		
n	—	—	—	—	—	—	—	—	●		—	1/2
d	●	●	●	●	●	●	—	—		●	—	●[1]

[1] Die Anzahl der Schritte beträgt 2 für die MC-Anweisung und 1 für die MCR-Anweisung.

GX IEC Developer



GX Developer

Variablen

Operand	Befehlswert	Datentyp
n	Ebene für das Nesting-Programm (A-Serie = N0 - N7, Q-Serie/System Q = N0 - N14)	Ebene
d	Operand, der die Nesting-Ebene einschaltet.	Bit

Funktions-weise

Aktivierung/ Deaktivierung einzelner Programmbereiche

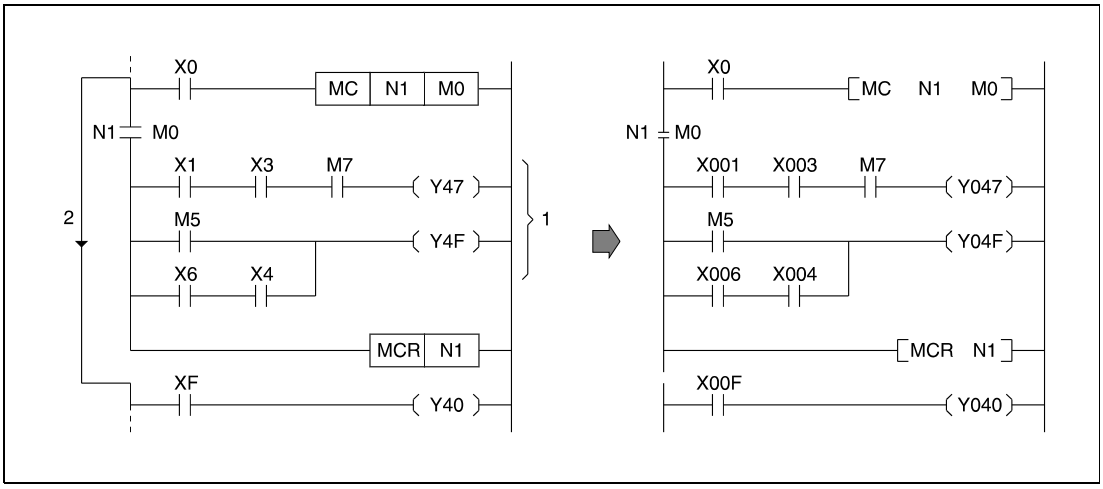
Allgemeines

Die MC-Anweisung wird eingesetzt, um effizient zwischen einzelnen Programmteilen zur Ausführung separater Prozesse umzuschalten. Nach dem Einschalten der Eingangsbedingung wird der Programmteil zwischen der Zieladresse d und der MCR-Anweisung ausgeführt. Der Bereich der MC-Anweisung wird durch die Angabe der Ebene (N0 bis N7 für die A-Serie und N0 bis N14 für die Q-Serie/System Q) festgelegt.

Da mit dem GX IEC Developer eine anschauliche Programmierung der MC-/MCR-Anweisung nicht möglich ist, werden hier die Kontaktpläne der GX Developer Software zur Veranschaulichung eingesetzt.

Der Kontaktplan verdeutlicht die Arbeitsweise der MC-Anweisung. Bei ausgeschaltetem Eingang X0 wird der in Ebene 1 (gekennzeichnet durch N1) befindliche Programmteil übersprungen (1). Nach dem Einschalten von X0 wird der an N1 befindliche Programmteil bis zur MCR-Anweisung ausgeführt (2).

Bei Programmierung einer MC-Anweisung im Kontaktplanmodus muss die Anweisungform nicht in gesonderter Weise eingegeben werden. Nach Konvertierung der Eingaben werden MC-Kontakte automatisch angezeigt.



MC Aktivierung einzelner Programmbereiche

Die MC-Anweisung ist die Startanweisung für den Master Control zum Aufruf eines definierten Programmteils. Ist die Eingangsbedingung der MC-Anweisung eingeschaltet, werden die zwischen MC- und MCR-Anweisung liegenden Operandenadressen normal verarbeitet.

Die Weiterverarbeitung der zwischen MC- und MCR-Anweisung liegenden Operandenadressen erfolgt auch nach Ausschalten der Eingangsbedingung zur MC-Anweisung. Die Programmzykluszeit verkürzt sich hierdurch nicht. Sobald die Eingangsbedingung abfällt, werden die zwischen MC- und MCR-Anweisung liegenden Operanden wie folgt verarbeitet:

Operanden	Verarbeitung
10-ms-Timer 100-ms-Timer	Zählwert wird auf 0 gesetzt. Ein- und Ausgangskontakt werden ausgeschaltet
remanente 10-ms-Timer (nur Q-Serie und System Q) remanente 100-ms-Timer Counter	Zählwert und Zustand der Eingangskontakte werden beibehalten. Ausgangskontakt wird ausgeschaltet.
Operanden in einer Out-Anweisung	Alle Ausgänge werden ausgeschaltet.
Operanden in einer SET-, RST-, und SFT-Anweisung	Istzustand wird beibehalten.

HINWEIS *Befindet sich in dem Programmteil zwischen MC- und MCR-Anweisung eine Anweisung, für die keine unmittelbare Eingangsbedingung programmiert werden muss (FOR bis NEXT, EI, DI usw.), führt die SPS diese Anweisung ohne Berücksichtigung der Eingangsbedingung zur MC-Anweisung aus.*

Innerhalb der MC-Anweisung sind identische Ebenennummern n möglich, wenn unterschiedliche Operandenadressen d gesetzt werden.

Nach dem Setzen der MC-Anweisung wird der Kontakt des in d angegebenen Operanden eingeschaltet. Ist der Operand auch an anderer Stelle als Eingangsbedingung im Programm vorhanden, werden die Kontakte als Doppelkontakt verarbeitet und parallel ein- und ausgeschaltet. Der in d angegebene Operand sollte daher nicht ein weiteres Mal im Programm vorhanden sein.

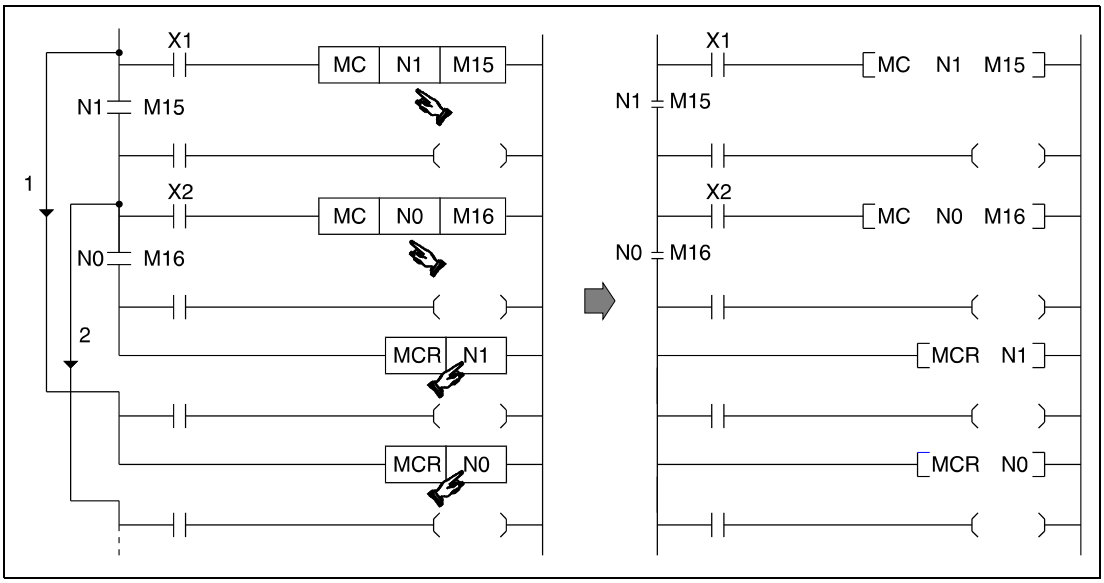
MCR Deaktivierung einzelner Programmbereiche

Die MCR-Anweisung setzt die MC-Anweisung zurück und kennzeichnet das Ende des Programmteils für den Master Control.

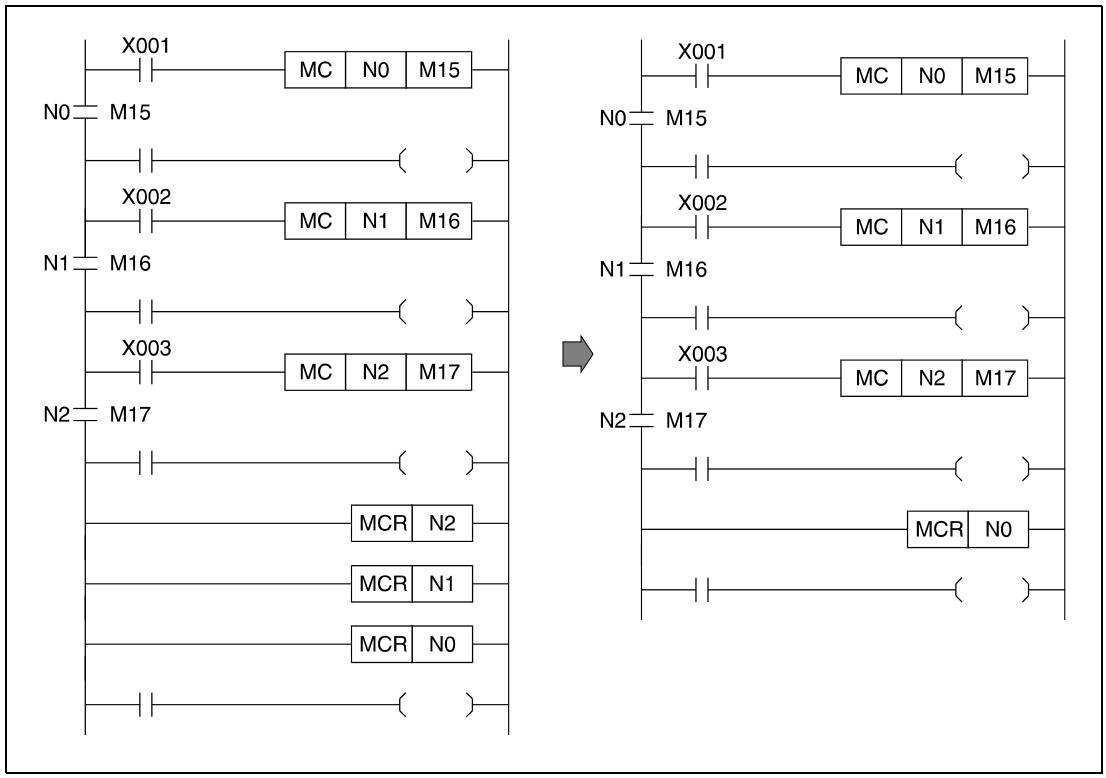
Eine MCR-Anweisung darf nicht über einen Eingangskontakt angesprochen werden.

Folgende Punkte sind zur Programmierung der Nesting-Adressen zu beachten:

Das Nesting ist in der Q-Serie bzw. dem System Q in 15 Ebenen von N0 bis N14 und in der A-Serie in 8 Ebenen von N0 bis N7 möglich. Der erste Programmbereich, der über die MC-Anweisung aufgerufen wird, muss mit der niedrigsten Nesting-Adresse und die erste MCR-Anweisung mit der höchsten Nesting-Adresse beginnen. Werden Nesting-Adressen in anderer Reihenfolge vergeben, erfolgt eine falsche Zuordnung der Ausführungsebenen (1, 2), und die SPS kann das Programm nicht ordnungsgemäß abarbeiten. Die folgende Abbildung verdeutlicht diesen Sachverhalt.



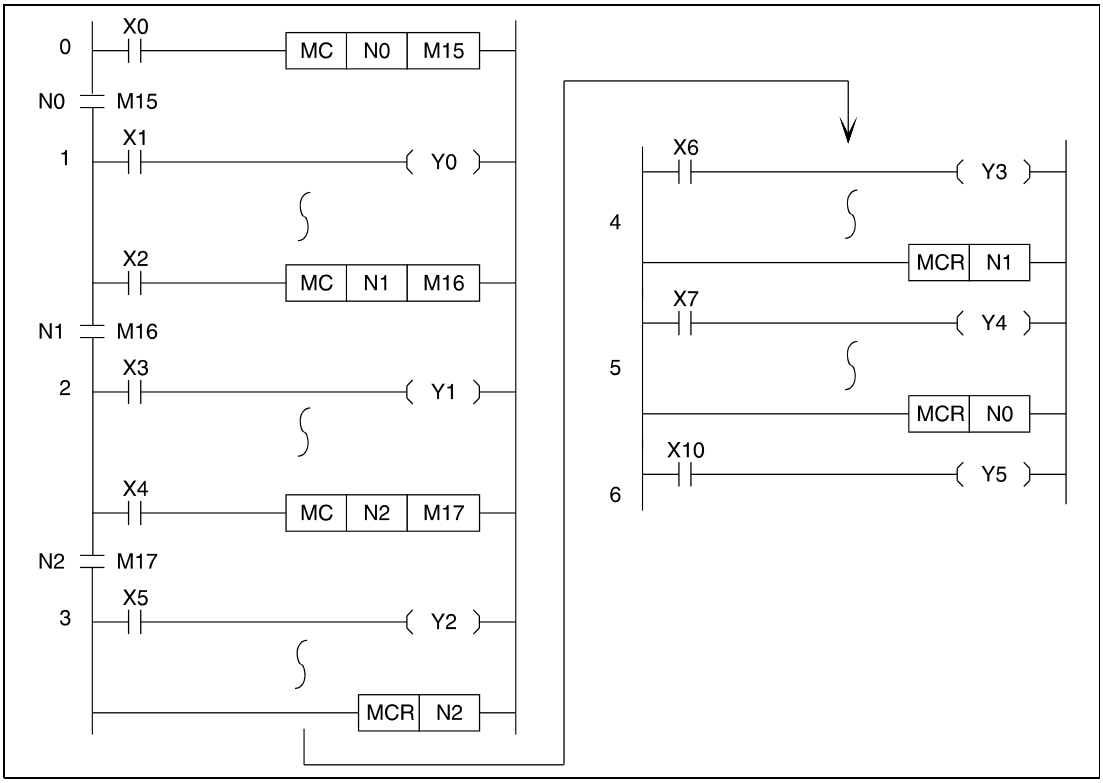
Liegen mehrere MCR-Anweisungen in Folge vor, kann das Programm dahingehend gekürzt werden, dass nur einmalig die niedrigste Nesting-Adresse zur Beendigung sämtlicher MC-Teilprogramme programmiert wird.



Beispiel MC, MCR

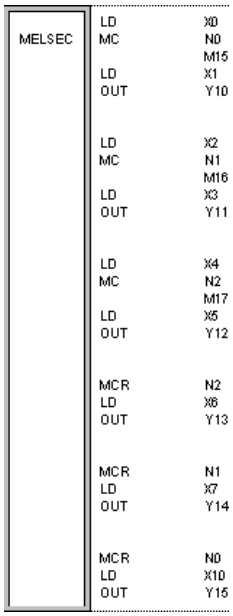
Eine MC-Anweisung wird in Verbindung mit einer Nesting-Adresse N zur Bestimmung der Ausführungsebene programmiert. Nesting-Adressen können bei der Q-Serie und dem System Q im Bereich zwischen N0 und N14 und bei der A-Serie zwischen N0 bis N7 vergeben werden.

Mit Hilfe der Nesting-Adressen kann die Ausführungsfolge der MC-Programmbereiche festgelegt werden. Das Beispiel zeigt ein Programmbeispiel mit mehreren Ausführungsebenen unter Verwendung der Nesting-Adressen. Zum besseren Verständnis ist auch hier der GX Developer-Kontaktplan abgebildet.

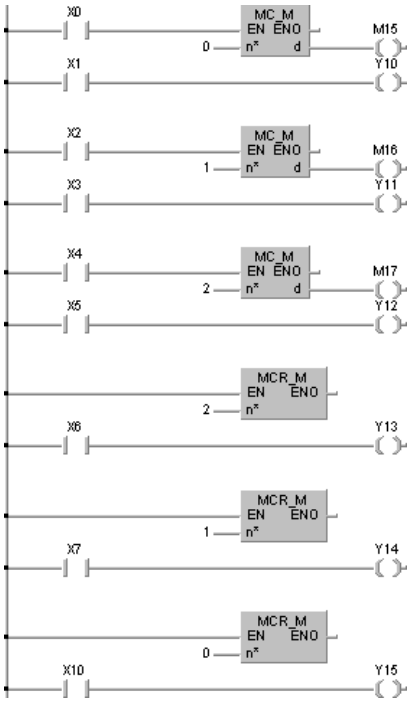


Ergänzend ist im folgenden die Darstellung im GX Developer zu diesem Beispiel angegeben.

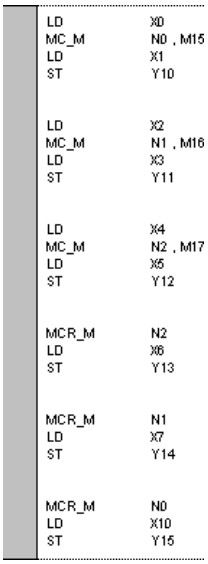
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



5.6 Definition des Programmendes

5.6.1 FEND

HINWEIS In den IEC-Editoren sollte diese Anweisung nicht verwendet werden.

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
																							1			●

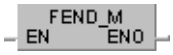
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten	Andere		
	Bit	Wort		Bit	Wort						
—	—	—	—	—	—	—	—	—	—	SM0	1

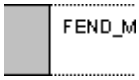
GX IEC Developer

MELSEC-Anweisungsliste

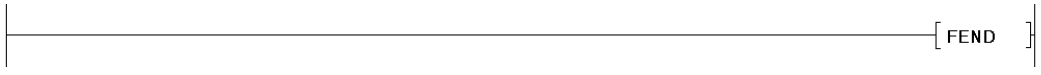
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

Operand	Befehlswert	Datentyp
—	—	—

Funktions-weise

Ende eines Programmbereichs

FEND Anweisung zum Beenden eines Programmbereichs

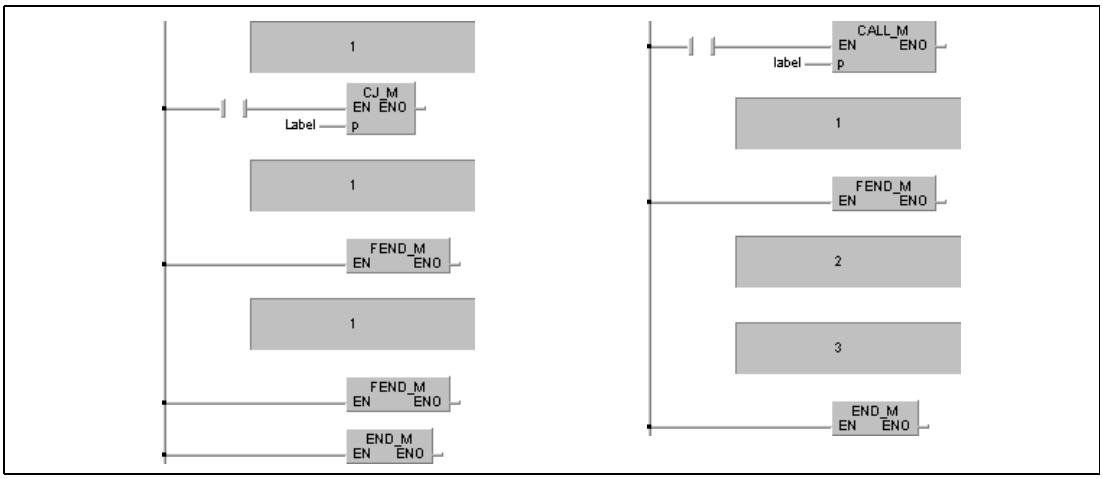
Die FEND-Anweisung kennzeichnet das Ende eines Programmbereiches. Dieser Bereich kann sowohl ein Haupt- als auch ein Unterprogrammbereich sein.

Nach Ausführung der FEND-Anweisung springt das Programm zur END-Anweisung. Die Ausführung der internen Prozesse wie Timer-/Counter-Verarbeitung oder die CPU-Selbstdiagnose beginnt erneut bei Programmschritt 0.

Das Beispiel links zeigt den Abschluss von Programmbereichen bei Verzweigung in ein Unterprogramm mit Hilfe einer CJ-Anweisung.

Bei Ausführung der CJ-Anweisung wird das Programm bis zur nächsten FEND-Anweisung hinter dem Programmteil ausgeführt, in den gesprungen wurde. Ohne Ausführung der CJ-Anweisung springt das Programm nach Erreichen der ersten FEND-Anweisung im Programm zurück nach Programmschritt 0.

Das rechte Programm zeigt die Verwendung der FEND-Anweisung zur Trennung eines Hauptprogrammbereiches von einem Unterprogramm- bzw. Interruptprogrammbereich.



[1] Hauptprogramm-Routine

[2] Unterprogramm-Routine

[3] Interrupt-Routine

HINWEISE

Beim GX Developer muss die FEND-Anweisung vom Anwender programmiert werden. Nachdem diese Programm-Organisationseinheit abgearbeitet wurde, wird keine weitere mehr ausgeführt, da sie sich nach der FEND-Anweisung befindet.

Eine Alternative zu dieser Programmierung ist die Programmierung im IEC-Editor. In diesem Fall wird die FEND-Anweisung vom GX IEC Developer automatisch gesetzt.

Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error-Flag wird gesetzt:

- Nach Ausführung einer CALL-, FCALL-, ECALL oder EFCALL-Anweisung wird die FEND-Anweisung vor der Verarbeitung einer RET-Anweisung ausgeführt (Q-Serie/System Q = Fehlercode 4211).
- Nach Ausführung einer FOR-Anweisung wird die FEND-Anweisung vor der Verarbeitung einer NEXT-Anweisung ausgeführt (Q-Serie = Fehlercode 4200).
- Die FEND-Anweisung wird während eines Interrupt-Programms und vor einer IRET-Anweisung ausgeführt (Q-Serie/System Q = Fehlercode 4221).
- Die FEND-Anweisung wird zwischen einer CHKCIR- und einer CHKEND-Anweisung ausgeführt (Q-Serie/System Q = Fehlercode 4230).
- Die FEND-Anweisung wird zwischen einer IX- und einer IXEND-Anweisung ausgeführt (Q-Serie/System Q = Fehlercode 4231).

5.6.2 END

HINWEIS In den IEC-Editoren sollte diese Anweisung nicht verwendet werden.

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
																							1			●

Operanden MELSEC Q

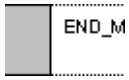
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort				U		
—	—	—	—	—	—	—	—	—	—	SM0	1

GX IEC Developer

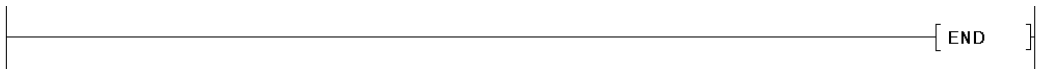
MELSEC-Anweisungsliste

Kontaktplan

IEC-Anweisungsliste



GX Developer



Variablen

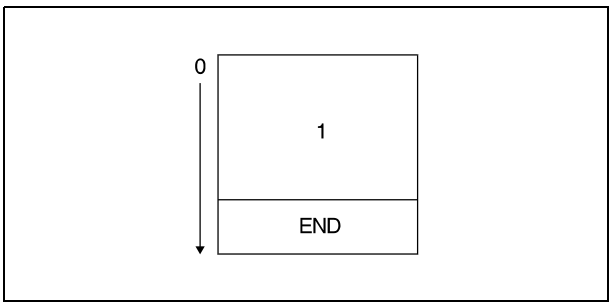
Operand	Befehlswert	Datentyp
—	—	—

Funktionsweise

Ende eines Haupt- oder Unterprogramms

END Anweisung zum Beenden eines Programms

Die END-Anweisung legt das Ende eines Programms fest. An diesem Schritt beginnt der Programmzyklus wieder mit Programmschritt 0.

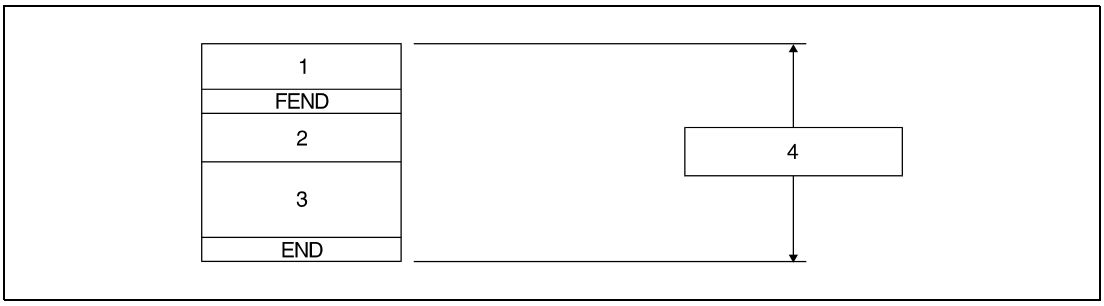


[1] Unterprogramm

Eine END-Anweisung kann nicht innerhalb einer Programmroutine programmiert werden. Zum Beenden einer Programmroutine ist die FEND-Anweisung zu programmieren.

Wird die END-Anweisung im Programm nicht gesetzt, erfolgt eine Fehlermeldung bei Programmstart, und die Verarbeitung durch die SPS wird abgebrochen. Ist der Umfang des Unterprogramms über Parameter festgelegt, erfolgt ebenfalls eine Fehlermeldung, wenn keine END-Anweisung programmiert wurde.

Die richtige Programmierung der END- und FEND-Anweisung in den einzelnen Programmroutinen wird anhand der folgenden Abbildung verdeutlicht.



[1] Hauptprogrammroutine
[2] Unterprogrammroutine
[3] Interruptroutine
[4] Hauptprogramm

HINWEISE *Der GX Developer und der GX IEC Developer erzeugen die END-Anweisung automatisch.*

Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error-Flag wird gesetzt:

- Das Sprungziel einer CJ-, SCJ- oder JMP-Anweisung liegt unterhalb der END-Anweisung.
- Eine Unterprogramm- oder Interrupt-Routine unterhalb der END-Anweisung wurde aufgerufen.
- Nach Ausführung einer CALL-, FCALL-, ECALL oder EFCALL-Anweisung wird die END-Anweisung vor Verarbeitung einer RET-Anweisung ausgeführt (Q-Serie/System Q = Fehlercode 4211).
- Nach Ausführung einer FOR-Anweisung wird die END-Anweisung vor der Verarbeitung einer NEXT-Anweisung ausgeführt (Q-Serie/System Q = Fehlercode 4200).
- Die END-Anweisung wird während eines Interrupt-Programms und vor einer IRET-Anweisung ausgeführt (Q-Serie/System Q = Fehlercode 4221).
- Die END-Anweisung wird zwischen einer CHKCIR- und einer CHKEND-Anweisung ausgeführt (Q-Serie/System Q = Fehlercode 4230).
- Die END-Anweisung wird zwischen einer IX- und einer IXEND-Anweisung ausgeführt (Q-Serie/System Q = Fehlercode 4231).

5.7 Sonstige Anweisungen

5.7.1 STOP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)								Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
																							1			

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort				U		
—	—	—	—	—	—	—	—	—	—	SM0	1

GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan

IEC-Anweisungsliste

GX Developer

Variablen

Operand	Befehlswert	Datentyp
—	—	—

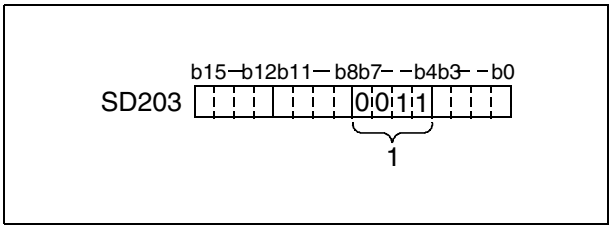
Funktions-weise

Unterbrechung der Verarbeitung

STOP Unterbrechungsanweisung

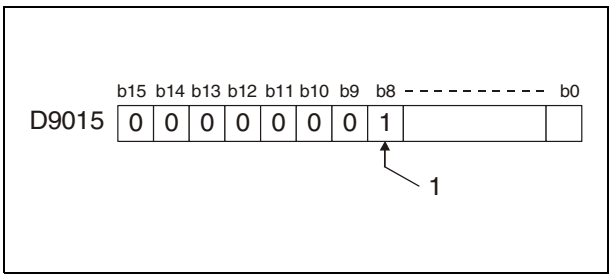
Nach dem Einschalten der Eingangsbedingung werden alle Ausgänge (Y) ausgeschaltet und die Programmverarbeitung der SPS gestoppt. Die Funktion entspricht dem Umschalten des RUN-STOP-Schalters (CPU-Modul) in die STOP-Position.

Bei Ausführung der STOP-Anweisung wird bei den CPUs der Q-Serie und des System Q das 5. bis 8. Bit (b4 bis b7) von Register SD203 auf 3 gesetzt.



[1] Zahl 3 im Binär-Format

Bei Ausführung der STOP-Anweisung wird bei CPUS der MELSEC A-Serie das 9. Bit (b8) von Sonderregister D9015 auf 1 gesetzt.



[1] Wird auf 1 gesetzt.

Um die Verarbeitung der SPS nach Ausführung der STOP-Anweisung wieder aufzunehmen, ist der RUN-STOP-Schalter kurz von RUN auf STOP und anschließend wieder auf RUN zu stellen.

Wenn der RESET-Schalter nach Ausführung der STOP-Anweisung auf LATCH CLEAR geschaltet wird, hat das keine Auswirkung auf den Inhalt des Zwischenspeichers. Zum Löschen des Zwischenspeicherinhalts muss der RUN/STOP-Schalter zunächst auf STOP gestellt und anschließend der RESET-Schalter auf L.CL. (LATCH CLEAR) gestellt werden.

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error-Flag wird gesetzt:

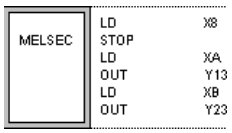
- Nach Ausführung einer CALL-, FCALL-, ECALL oder EFCALL-Anweisung wird die END-Anweisung vor der Verarbeitung einer RET-Anweisung ausgeführt.
  (Q-Serie/System Q = Fehlercode 4211)
- Nach Ausführung einer FOR-Anweisung wird die END-Anweisung vor der Verarbeitung einer NEXT-Anweisung ausgeführt (Q-Serie/System Q = Fehlercode 4200).
- Die END-Anweisung wird während eines Interrupt-Programms und vor einer IRET-Anweisung ausgeführt (Q-Serie/System Q = Fehlercode 4221).
- Die END-Anweisung wird zwischen einer CHKCIR- und einer CHKEND-Anweisung ausgeführt (Q-Serie/System Q = Fehlercode 4230).
- Die END-Anweisung wird zwischen einer IX- und einer IXEND-Anweisung ausgeführt (Q-Serie/System Q = Fehlercode 4231).

Beispiel

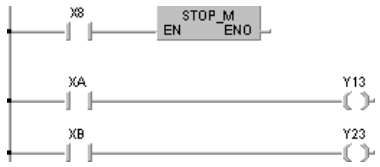
STOP

Im folgenden Programm wird mit Einschalten von X8 die Verarbeitung unterbrochen. Alle nachfolgenden Programmschritte werden nach dem Stellen des RUN-STOP-Schalters von RUN auf STOP und wieder auf RUN abgearbeitet.

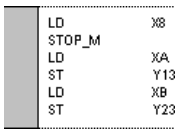
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



5.7.2 NOP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
																							1			

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
—	—	—	—	—	—	—	—	—	—	—	1

GX IEC Developer

MELSEC-Anweisungsliste	Kontaktplan	IEC-Anweisungsliste
		

GX Developer

Variablen

Operand	Befehlswert	Datentyp
—	—	—

Funktionsweise

Leerschritte

NOP Leerschritt im Programm

Die NOP-Anweisung (**N**o **OP**eration) beschreibt einen Leerschritt im Programm und hat keine Auswirkung auf die Verarbeitung bestehender Programmteile. Es wird ein logischer Programmleerschritt erzeugt, der später durch andere Anweisungen in einem noch nicht fertiggestellten Programm aufgefüllt werden kann.

Die Programmierung einer NOP-Anweisung ist in folgenden Fällen sinnvoll:

Leerstellen zur Fehlerbereinigung des Ablaufprogramms;

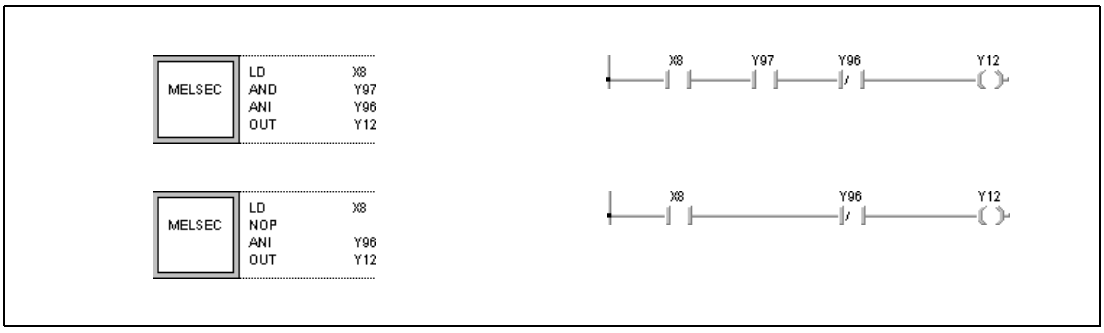
Löschen einer Anweisung (überschreiben mit NOP), ohne die Programmschrittnummern zu ändern;

Vorübergehendes Löschen einer Anweisung für spätere Editierung.

HINWEIS *Nach Abschluss der Programmierfolge sollten NOP-Anweisungen so weit möglich gelöscht werden, um die Programmzykluszeit zu verkürzen.*

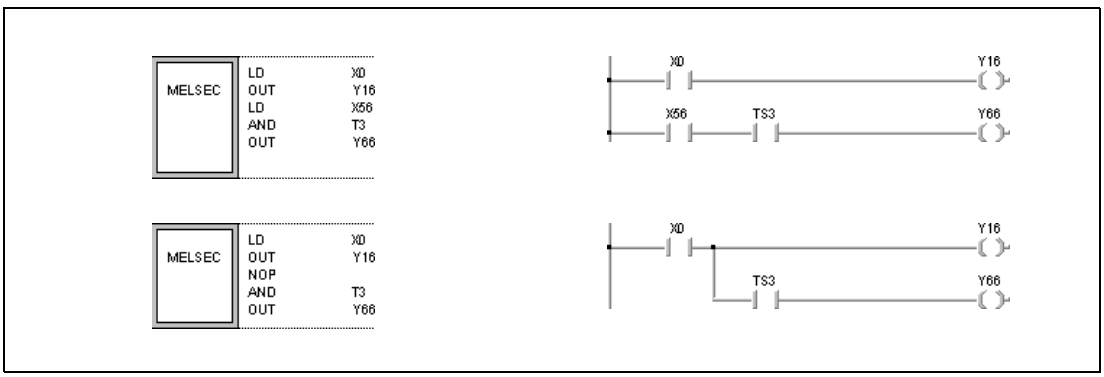
Beispiel 1 NOP

Im folgenden Programm werden zur Fehlerbereinigung eines Programms Verknüpfungskontakte (AND) durch Leerstellen ersetzt.



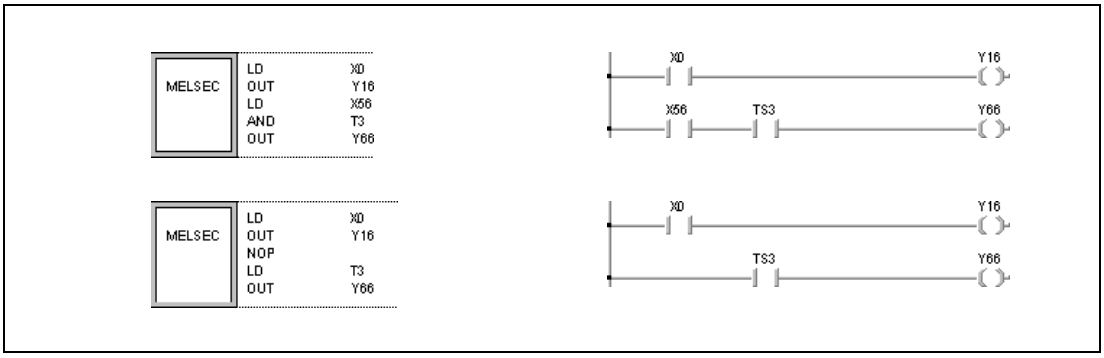
Beispiel 2 NOP

Im folgenden Programm wird eine LD-Anweisung durch eine NOP-Anweisung ersetzt.



Beispiel 3 NOP

Im folgenden Programm wird eine LD-Anweisung durch eine NOP-Anweisung ersetzt.



HINWEIS *Der Ersatz eines Eingangskontaktes (LD, LDI) durch eine NOP-Anweisung ist mit besonderer Vorsicht vorzunehmen, da die Programmlogik hierdurch stark verändert wird.*

6 Applikationsanweisungen Teil I

Die Applikationsanweisungen Teil I beinhalten Anweisungen, die numerische 16- und 32-Bit-Daten, Gleitkommazahlen und Zeichenfolgen verarbeiten können. In erster Linie werden mit Hilfe dieser Applikationsanweisungen Vergleichs- und Rechenoperationen durchgeführt.

Einteilung	Bedeutung
Vergleichsanweisungen	Datenvergleich, wie z.B. =, >, ≥ usw.
Arithmetikanweisungen	Addition, Subtraktion, Multiplikation, Division, von BIN- und BCD-Daten, Gleitkommazahlen und BIN-Datenblöcken, Verknüpfung von Zeichenfolgen, Inkrement, Dekrement
Konvertierungsanweisungen	Datenkonvertierung wie z.B. BCD → BIN und BIN → BCD
Transferanweisungen	Übertragung, Austausch und Negation von Daten
Programmverzweigungs-anweisungen	Sprung, Unterprogrammaufruf
Anweisungen zum Interrupt-Programmaufruf	Interrupt-Programmaufruf
Datenaktualisierungs-anweisungen	Link-Refresh und E/A-Schnittstellenauffrischung
Sonstige Anweisungen	Ein-/Zweiphasiger Auf-/Abwärtszähler, programmierbare Timer, Sonderfunktionstimer, Positionieranweisung, Rampensignal, Impulszähler, Impulsausgang, Puls-Weiten-Modulation, Eingabematrix

6.1 Vergleichsanweisungen

Vergleichsanweisungen können Größenvergleiche (wie z.B. gleich =, größer >, kleiner < usw.) zwischen zwei Datensätzen durchführen. Die Programmierung der Vergleichsanweisungen erfolgt in gleicher Weise wie die der entsprechenden Anweisungen aus dem Grundbefehlssatz:

LD, LDI $\Rightarrow$ LD=, LDD=

AND, ANI $\Rightarrow$ AND=, ANDD=

OR, ORI $\Rightarrow$ OR=, ORD=

Funktion	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor	Funktion	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor
= gleich	LD=	LD_EQ_M	≤ kleiner gleich	LD<=	LD_LE_M
	AND=	AND_EQ_M		AND<=	AND_LE_M
	OR=	OR_EQ_M		OR<=	OR_LE_M
	LDD=	LDD_EQ_M		LDD<=	LDD_LE_M
	ANDD=	ANDD_EQ_M		ANDD<=	ANDD_LE_M
	ORD=	ORD_EQ_M		ORD<=	ORD_LE_M
	LDE=	LD_EEQ_M		LDE<=	LD_ELE_M
	ANDE=	AND_EEQ_M		ANDE<=	AND_ELE_M
	ORE=	OR_EEQ_M		ORE<=	OR_ELE_M
	LD$=	LD_STRING _EQ_M		LD$<=	LD_STRING _LE_M
	AND$=	AND_STRING _EQ_M		AND$<=	AND_STRING _LE_M
	OR$=	OR_STRING _EQ_M		OR$<=	OR_STRING _LE_M
	BKCMP=	BKCMP_EQ_M		BKCMP<=	BKCMP_LE_M
	BKCMP=P	BKCMP_EQP_M		BKCMP<=P	BKCMP_LEP_M
≠ ungleich	LD<>	LD_NE_M	< kleiner	LD<	LD_LT_M
	AND<>	AND_NE_M		AND<	AND_LT_M
	OR<>	OR_NE_M		OR<	OR_LT_M
	LDD<>	LDD_NE_M		LDD<	LDD_LT_M
	ANDD<>	ANDD_NE_M		ANDD<	ANDD_LT_M
	ORD<>	ORD_NE_M		ORD<	ORD_LT_M
	LDE<>	LD_ENE_M		LDE<	LD_ELT_M
	ANDE<>	AND_ENE_M		ANDE<	AND_ELT_M
	ORE<>	OR_ENE_M		ORE<	OR_ELT_M
	LD$<>	LD_STRING _NE_M		LD$<	LD_STRING _LT_M
	AND$<>	AND_STRING _NE_M		AND$<	AND_STRING _LT_M
	OR$<>	OR_STRING _NE_M		OR$<	OR_STRING _LT_M
	BKCMP<>	BKCMP_NE_M		BKCMP<	BKCMP_LT_M
	BKCMP<>P	BKCMP_NEP_M		BKCMP<P	BKCMP_LTP_M

Funktion	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor	Funktion	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor
> größer	LD>	LD_GT_M	≥ größer gleich	LD>=	LD_GE_M
	AND>	AND_GT_M		AND>=	AND_GE_M
	OR>	OR_GT_M		OR>=	OR_GE_M
	LDD>	LDD_GT_M		LDD>=	LDD_GE_M
	ANDD>	ANDD_GT_M		ANDD>=	ANDD_GE_M
	ORD>	ORD_GT_M		ORD>=	ORD_GE_M
	LDE>	LD_EGT_M		LDE>=	LD_EGE_M
	ANDE>	AND_EGT_M		ANDE>=	AND_EGE_M
	ORE>	OR_EGT_M		ORE>=	OR_EGE_M
	LD$>	LD_STRING _GT_M		LD$>=	LD_STRING _GE_M
	AND$>	AND_STRING _GT_M		AND$>=	AND_STRING _GE_M
	OR$>	OR_STRING _GT_M		OR$>=	OR_STRING _GE_M
	BKCMP>	BKCMP_GT_M		BKCMP>=	BKCMP_GE_M
	BKCMP>P	BKCMP_GTP_M		BKCMP>=P	BKCMP_GEP_M

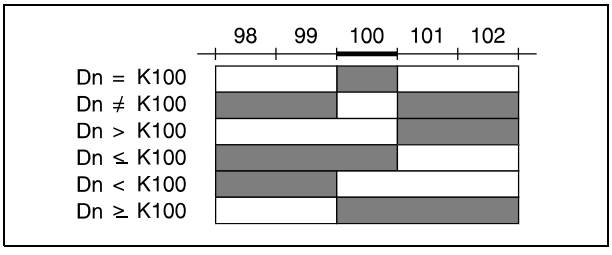
HINWEIS *Sie sollten in den IEC-Editoren die IEC-Befehle nutzen.*

IEC-Befehle

Funktion	IEC-BEFEHL	Bedeutung
=	EQ	Equal
<>	NE	Not Equal
<=	LE	Less Equal
<	LT	Less Than
>=	GE	Greater Equal
>	GT	Greater Than

Ausführungsbedingungen

Die folgende Abbildung zeigt die Ausführungsbedingungen der verschiedenen Vergleichsanweisungen.



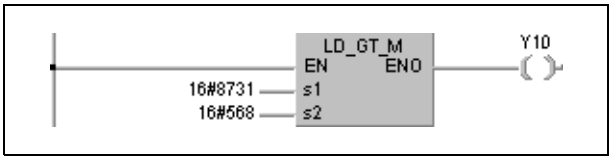
■ = 1 = AN

□ = 0 = AUS

HINWEISE *Bei Vergleichsanweisungen werden alle angesprochenen Datenwerte als Binärwerte verarbeitet.*

Bei dem Vergleich 16#8000 > 16#7999 wird das Vergleichsergebnis auf FALSCH gesetzt, obwohl eigentlich WAHR zu erwarten ist. Die Werte werden binär gewandelt, und somit wird auch das Bit 15 (b15) gesetzt. Ist das Bit 15 gesetzt, wird die Zahl automatisch negativ.

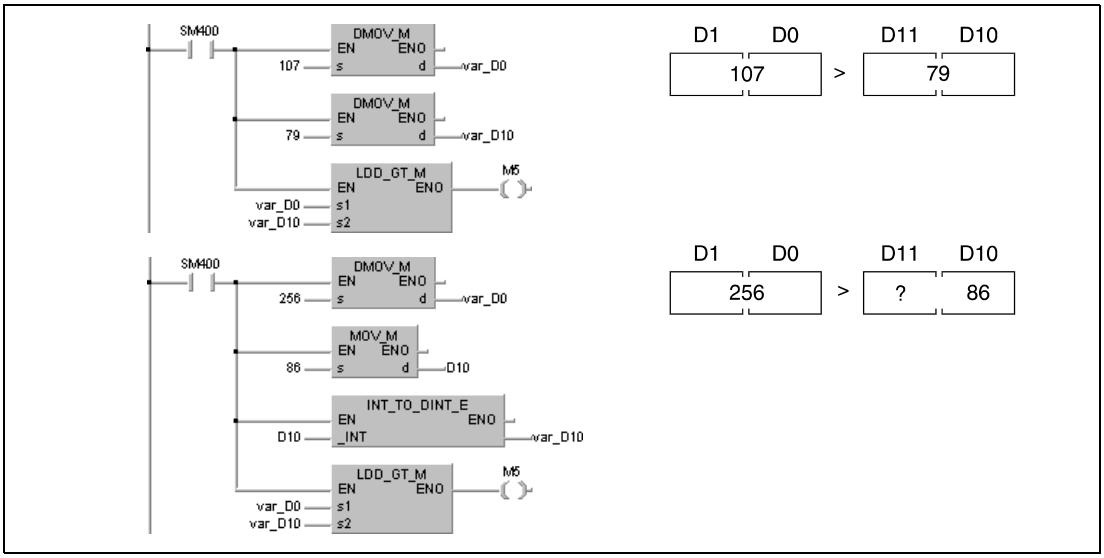
Beispiel 1 Vergleich von 2 vierstelligen BCD-Werten



8731_H wird binär als -30927 und 568_H als 1384 verarbeitet. Da das Vergleichsergebnis hier -30927 > 1384 lautet, wird Y10 nicht gesetzt.

Bei Vergleichsfunktionen mit 32-Bit-Daten ist der numerische Eingangswert über eine 32-Bit-Anweisung wie z.B. DMOV zu bestimmen. Erfolgt die Bestimmung über eine 16-Bit-Anweisung wie z.B. MOV, kann die Funktion nicht richtig ausgeführt werden, da bei 32-Bit-Vergleichen immer der n und (n+1) Datenwert benutzt wird.

Beispiel 2 Vergleichsfunktion mit 32-Bit-Daten



Das Beispiel zeigt zwei Vergleichsoperationen mit 32-Bit-Daten. Bei dem oberen Programm schaltet M5 ein, da hier beide Werte über die 32-Bit-Anweisung DMOV bestimmt wurden.

Im unteren Programm tritt ein nicht eindeutiges Ergebnis auf, da der Wert der oberen Bytes nicht eindeutig definiert ist.

HINWEIS *Diese Programmbeispiele sind ohne Variablendefinition im Header der Programm-Organisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

6.1.1 =, < >, >, < =, <, > =

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
s1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				K1 ↓ K4	5/7 ●[1]	●		●
s2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●								

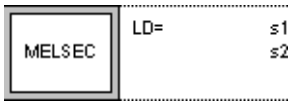
[1] Die Anzahl der Schritte beträgt 7, wenn die Index-Funktion ausgeführt, die Blocklänge eines Bit-Operanden nicht K4 lautet oder die Kopfadresse eines Bit-Operanden nicht das Mehrfache von 8 (bzw. 16 bei einer A3H, A3M, AnA, AnAS oder AnU CPU) ist. Die Anzahl der Schritte bei der Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

Operanden MELSEC Q

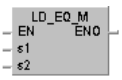
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sonder-module U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	●	●	●	●	●	●	●	●	—	—	3
s2	●	●	●	●	●	●	●	●	—	—	

GX IEC Developer

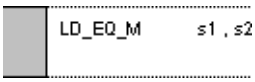
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

Operand	Befehlswert	Datentyp
s1	Vergleichsdaten oder Operand, in dem die Vergleichsdaten gespeichert sind.	BIN-16-Bit
s2		

Funktionsweise

16-Bit-Datenvergleich

=, <>, >, <=, <, >= Vergleichsanweisungen

Eine 16-Bit-Vergleichsanweisung besteht aus der Anweisung selbst und den Daten s1 und s2, die miteinander verglichen werden sollen.

Die aufgezeichneten Vergleichsanweisungen werden als Schließerkontakt behandelt. Der Datenvergleich erfolgt mit 16-Bit-Daten.

Der nachgeschaltete Ausgang schaltet in Abhängigkeit des Ergebnisses der Vergleichsfunktion. Die entsprechenden Schaltbedingungen sind in der folgenden Tabelle aufgeführt.

Symbol der Anweisung	Zustand des Ausgangs	
	1 wenn:	0 wenn:
=	s1 = s2	s1 ≠ s2
<>	s1 ≠ s2	s1 = s2
>	s1 > s2	s1 ≤ s2
<=	s1 ≤ s2	s1 > s2
<	s1 < s2	s1 ≥ s2
>=	s1 ≥ s2	s1 < s2

HINWEISE

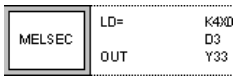
Bei Vergleichsanweisungen werden alle angesprochenen Datenwerte als Binärwerte verarbeitet.

Bei dem Vergleich 16#8000 > 16#7999 wird das Vergleichsergebnis auf FALSCH gesetzt, obwohl eigentlich WAHR zu erwarten ist. Die Werte werden binär gewandelt, und somit wird auch das Bit 15 (b15) gesetzt. Ist das Bit 15 gesetzt, wird die Zahl automatisch negativ.

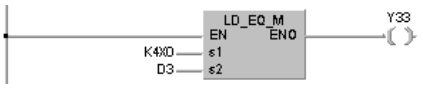
Beispiel 1 Vergleichsanweisung =

Das folgende Programm vergleicht die Daten von X0 bis XF mit dem Datenwert in D3. Sind beide Werte gleich, wird Y33 gesetzt.

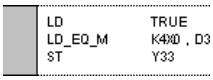
MELSEC-Anweisungsliste



Kontaktplan



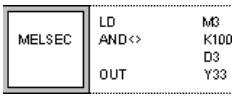
IEC-Anweisungsliste



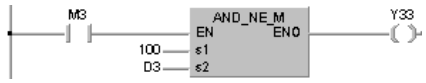
Beispiel 2 Vergleichsanweisung <>

Das folgende Programm vergleicht den Binärwert 100 mit dem Datenwert in D3. Ist der Wert in D3 ungleich 100, und ist M3 gesetzt, wird Y33 gesetzt.

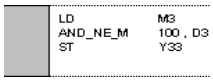
MELSEC-Anweisungsliste



Kontaktplan



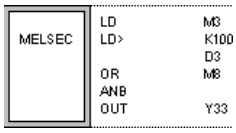
IEC-Anweisungsliste



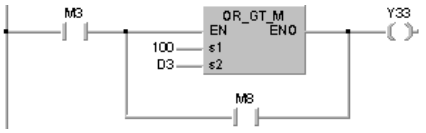
Beispiel 3 Vergleichsanweisung >

Das folgende Programm vergleicht den Binärwert 100 mit dem Datenwert in D3. Ist der Wert in D3 kleiner als 100, und ist M3 gesetzt, wird Y33 gesetzt. Wenn M8 und M3 gesetzt sind, wird Y33 ebenfalls gesetzt.

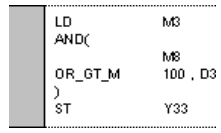
MELSEC-Anweisungsliste



Kontaktplan



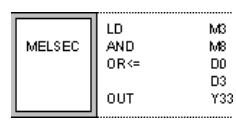
IEC-Anweisungsliste



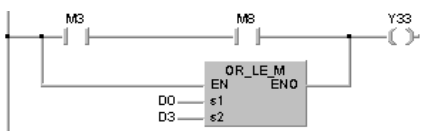
Beispiel 4 Vergleichsanweisung <=

Das folgende Programm vergleicht den Datenwert in D0 mit dem Datenwert in D3. Ist der Wert in D0 kleiner oder gleich D3, wird Y33 gesetzt. Wenn M8 und M3 gesetzt sind, wird Y33 ebenfalls gesetzt.

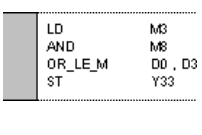
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



6.1.2 D=, D<>, D>, D<=, D<, D>=

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
s1	●	●	●	●	●	●	●	●	●	●	●	●	●		●		●	●				K1 ↓ K8	11 ● 1	●		●
s2	●	●	●	●	●	●	●	●	●	●	●	●	●		●		●	●								

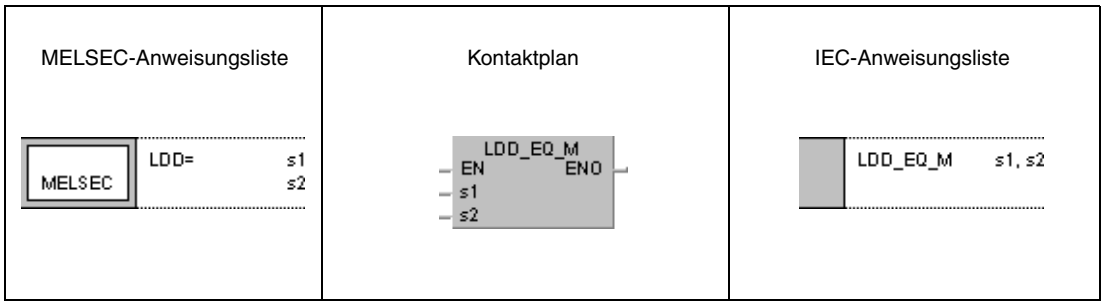
[1] Die Anzahl der Schritte bei der Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	●	●	●	●	●	●	●	●	—	—	3[1)]
s2	●	●	●	●	●	●	●	●	—	—	

[1] Die Anzahl der Programmschritte ist abhängig vom Typ der CPU und von den verwendeten Operanden.
Bei Verwendung einer QnA-CPU oder einer Single-Prozessor-Q-CPU: 3
Bei Verwendung einer Multi-Prozessor-Q-CPU und
interner Wortoperanden (außer File-Register ZR): 5
Konstanten: 5
Bit-Operanden, deren Adresse ein Vielfaches von 16 ist, die die Bit-Blockbezeichnung K8 haben und die nicht durch Index-Vergabe bearbeitet werden: 5
Bei Verwendung einer Multi-Prozessor-Q-CPU und anderer Operanden als oben aufgeführt: 3

GX IEC Developer



GX Developer

Variablen

Operand	Befehlswert	Datentyp
s1	Vergleichsdaten oder Operand, in dem die Vergleichsdaten gespeichert sind.	BIN-32-Bit
s2		

Funktionsweise

32-Bit-Datenvergleich

D=, D<>, D>, D<=, D<, D>= Vergleichsanweisungen

Eine 32-Bit-Vergleichsanweisung besteht aus der Anweisung selbst und den Daten s1 und s2, die miteinander verglichen werden sollen.

Die aufgezeichneten Vergleichsanweisungen werden als Schließerkontakt behandelt. Der Datenvergleich erfolgt mit 32-Bit-Daten.

Der nachgeschaltete Ausgang schaltet in Abhängigkeit des Ergebnisses der Vergleichsfunktion. Die entsprechenden Schaltbedingungen sind in der folgenden Tabelle aufgeführt.

Symbol der Anweisung	Zustand des Ausgangs	
	1 wenn:	0 wenn:
D=	s1 = s2	s1 ≠ s2
D<>	s1 ≠ s2	s1 = s2
D>	s1 > s2	s1 ≤ s2
D<=	s1 ≤ s2	s1 > s2
D<	s1 < s2	s1 ≥ s2
D>=	s1 ≥ s2	s1 < s2

HINWEISE

Bei Vergleichsanweisungen werden alle angesprochenen Datenwerte als Binärwerte verarbeitet.

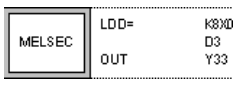
Bei dem Vergleich 16#8000 > 16#7999 wird das Vergleichsergebnis auf FALSCH gesetzt, obwohl eigentlich WAHR zu erwarten ist. Die Werte werden binär gewandelt, und somit wird auch das Bit 15 (b15) gesetzt. Ist das Bit 15 gesetzt, wird die Zahl automatisch negativ.

Beispiel 1

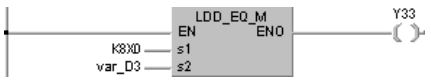
Vergleichsanweisung D=

Das folgende Programm vergleicht die Daten von X0 bis X1F mit dem Datenwert in D3 und D4. Sind beide Werte gleich, wird Y33 gesetzt.

MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste

```
LD      TRUE
LDD_EQ_M K8X0 , var_D3
ST      Y33
```

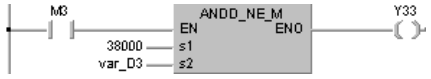
Beispiel 2 Vergleichsanweisung D<>

Das folgende Programm vergleicht den Binärwert 38000 mit dem Datenwert in D3 und D4. Ist der Wert in D3 und D4 ungleich 38000, wird Y33 gesetzt, wenn M3 gesetzt ist.

MELSEC-Anweisungsliste

```
MELSEC  LD      M3
        ANDD<>  K38000
                D3
        OUT     Y33
```

Kontaktplan



IEC-Anweisungsliste

```
        LD      M3
        ANDD_NE_M  38000 , var_D3
        ST      Y33
```

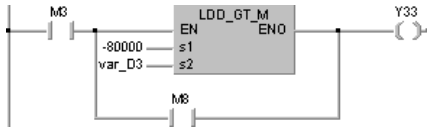
Beispiel 3 Vergleichsanweisung D>

Das folgende Programm vergleicht den Binärwert -80000 mit dem Datenwert in D3 und D4. Ist der Wert in D3 und D4 kleiner als -80000, wird Y33 gesetzt, wenn M3 gesetzt ist. Der Ausgang Y33 wird ebenfalls gesetzt, wenn M3 und M8 gesetzt sind.

MELSEC-Anweisungsliste

```
MELSEC  LD      M3
        LDD>    K-80000
                D3
        OR      M8
        ANB
        OUT     Y33
```

Kontaktplan



IEC-Anweisungsliste

```
        LD      M3
        AND(
                M8
        ORD_GT_M  K-80000 , var_D3
        )
        ST      Y33
```

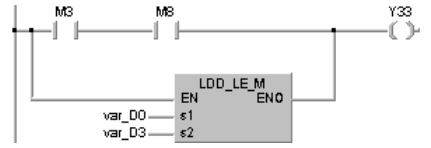
Beispiel 4 Vergleichsanweisung D<=

Das folgende Programm vergleicht den Datenwert in D0 und D1 mit dem Datenwert in D3 und D4. Ist der Wert in D3 und D4 größer oder gleich D0 und D1, wird Y33 gesetzt. Der Ausgang Y33 wird ebenfalls gesetzt, wenn M3 und M8 gesetzt sind.

MELSEC-Anweisungsliste

```
MELSEC  LD      M3
        AND     M8
        ORD<=   D0
                D3
        OUT     Y33
```

Kontaktplan



IEC-Anweisungsliste

```
        LD      M3
        AND     M8
        ORD_LE_M  var_D0 , var_D3
        ST      Y33
```

HINWEIS *Diese Programmbeispiele sind ohne Variablendefinition im Header der Programm-Organisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

6.1.3 E=, E<>, E>, E< =, E<, E>=

CPU

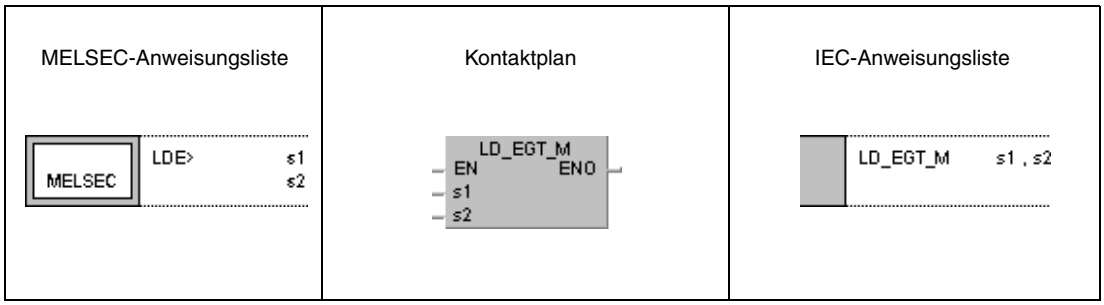
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●[1]

[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□N□		Sondermodule U□\G□	Index-Register Zn	Konstanten E	Andere		
	Bit	Wort		Bit	Wort						
s1	—	●	●	—	●	●	—	●	—	—	3
s2	—	●	●	—	●	●	—	●	—	—	

GX IEC Developer



GX Developer

Variablen

Operand	Befehlswert	Datentyp
s1	Vergleichsdaten oder Operand, in dem die Vergleichsdaten gespeichert sind.	Reelle Zahlen
s2		

Funktionsweise

Vergleich von Gleitkommazahlen

E=, E<>, E>, E<=, E<, E>= Vergleichsanweisungen

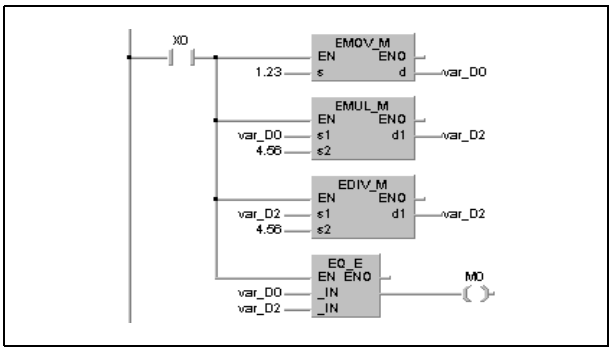
Eine Vergleichsanweisung für Gleitkommazahlen besteht aus der Anweisung selbst und den Daten s1 und s2, die miteinander verglichen werden sollen.

Die aufgezeichneten Vergleichsanweisungen werden als Schließerkontakt behandelt. Der Datenvergleich erfolgt mit Gleitkommazahlen.

Der nachgeschaltete Ausgang schaltet in Abhängigkeit des Ergebnisses der Vergleichsfunktion. Die entsprechenden Schaltbedingungen sind in der folgenden Tabelle aufgeführt.

Symbol der Anweisung	Zustand des Ausgangs	
	1 wenn:	0 wenn:
E=	s1 = s2	s1 ≠ s2
E<>	s1 ≠ s2	s1 = s2
E>	s1 > s2	s1 ≤ s2
E<=	s1 ≤ s2	s1 > s2
E<	s1 < s2	s1 ≥ s2
E>=	s1 ≥ s2	s1 < s2

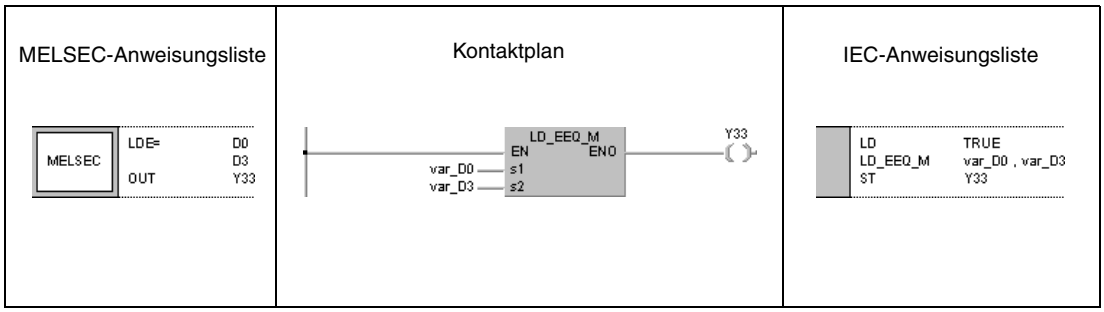
HINWEIS *Beim Vergleich von Gleitkommazahlen ist zu beachten, dass unter Umständen zwei vor einer Operation identischen Werte nach dieser Operation durch Rundungsfehler nicht mehr identisch sind. In diesem Fall wird der Merker M0 im unteren Beispiel nicht gesetzt.*



HINWEIS *Diese Programmbeispiele sind ohne Variablendefinition im Header der Programm-Organisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.*

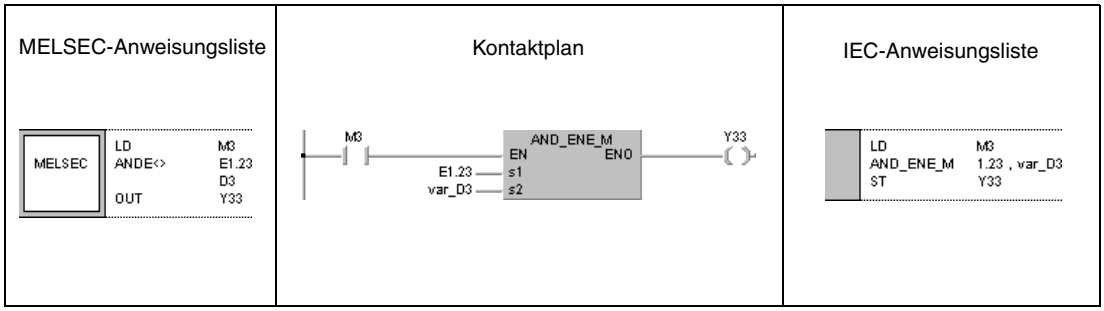
Beispiel 1 Vergleichsanweisung E=

Das folgende Programm vergleicht die in D0 und D1 abgelegte Gleitkommazahl mit der in D3 und D4 abgelegten Gleitkommazahl. Sind beide Werte gleich, wird Y33 gesetzt.



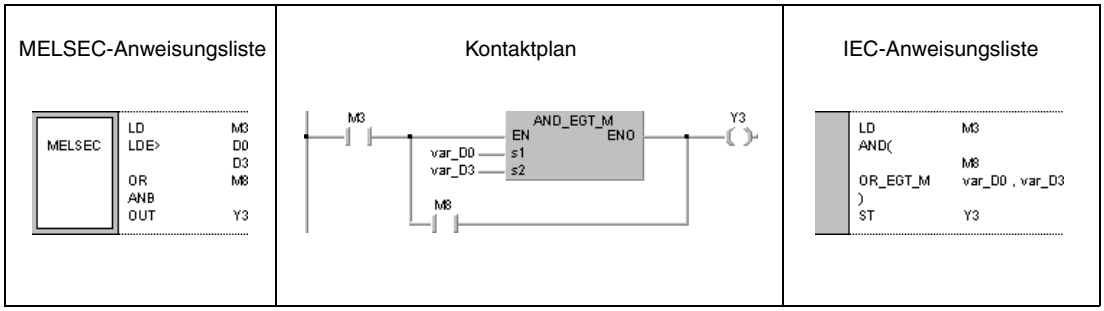
Beispiel 2 Vergleichsanweisung E<>

Das folgende Programm vergleicht die Gleitkommazahl 1.23 mit der in D3 und D4 abgelegten Gleitkommazahl. Ist der Wert in D3 und D4 ungleich 1.23, wird Y33 gesetzt, wenn M3 gesetzt ist.



Beispiel 3 Vergleichsanweisung E>

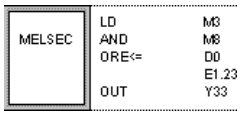
Das folgende Programm vergleicht die in D0 und D1 abgelegte Gleitkommazahl mit der in D3 und D4 abgelegten Gleitkommazahl. Ist der Wert in D3 und D4 kleiner als der Wert in D0 und D1, wird Y3 gesetzt, wenn M3 gesetzt ist. Der Ausgang Y3 wird ebenfalls gesetzt, wenn M3 und M8 gesetzt sind.



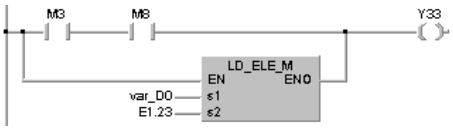
Beispiel 4 Vergleichsanweisung E<=

Das folgende Programm vergleicht die in D0 und D1 abgelegte Gleitkommazahl mit der Gleitkommazahl 1.23. Ist der Wert in D0 und D1 kleiner oder gleich 1.23, wird Y33 gesetzt. Der Ausgang Y33 wird ebenfalls gesetzt, wenn M3 und M8 gesetzt sind.

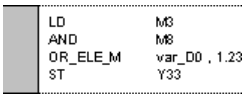
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



HINWEIS *Diese Programmbeispiele sind ohne Variablendefinition im Header der Programm-Organisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradresierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

6.1.4 $ =, $ < >, $ >, $ < =, $ <, $ > =

CPU

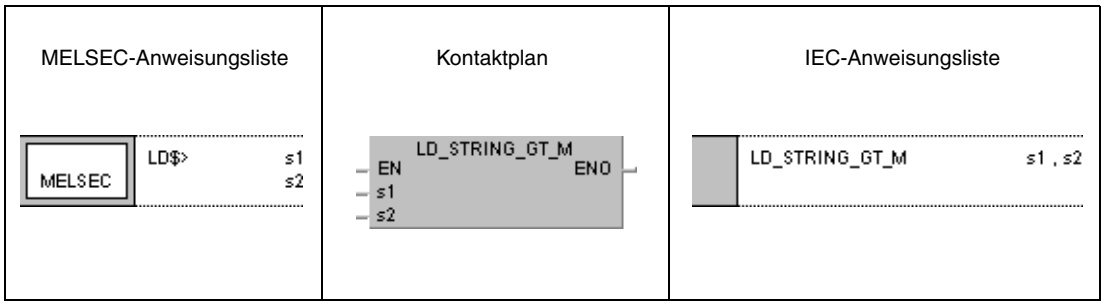
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●[1]

[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□N□		Sondermodule U□NG□	Index-Register Zn	Konstanten $	Andere		
	Bit	Wort		Bit	Wort						
s1	—	●	●	—	—	—	—	●	—	SM0	3
s2	—	●	●	—	—	—	—	●	—		

GX IEC Developer



GX Developer

Variablen

Operand	Befehlswert	Datentyp
s1	Erste Adresse der Vergleichsdaten oder Operand, in dem die Vergleichsdaten gespeichert sind.	Zeichenfolge
s2		

Funktionsweise

Vergleich von Zeichenfolgen

$=, $<>, $>, $<=, $<, $>= Vergleichsanweisungen

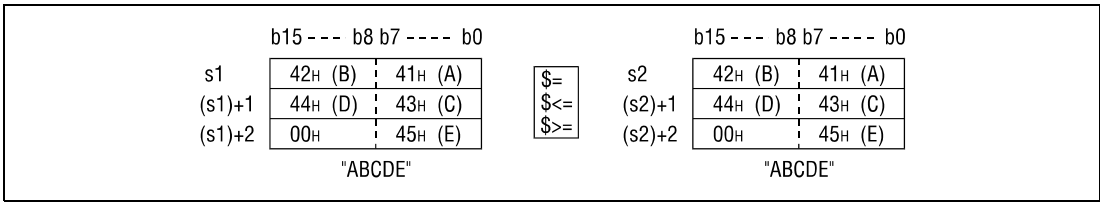
Eine Vergleichsanweisung für Zeichenfolgen besteht aus der Anweisung selbst und den Daten s1 und s2, die miteinander verglichen werden sollen.

Die aufgezeichneten Vergleichsanweisungen werden als Schließerkontakt behandelt.

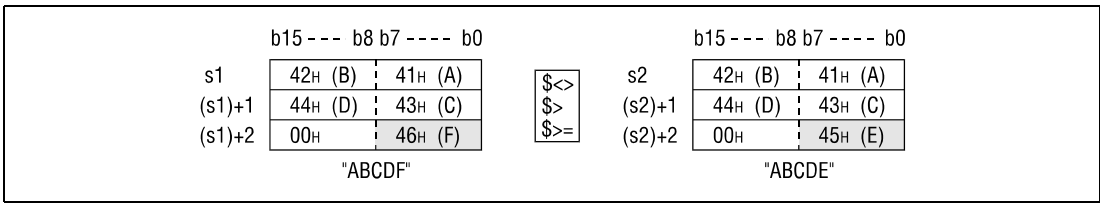
Der Datenvergleich der Zeichenfolgen im ASCII-Code erfolgt zeichenweise beginnend mit dem ersten Zeichen der Zeichenfolge.

Die in s1 und s2 abgelegten Zeichenfolgen beginnen mit dem niedrigstwertigen Byte (1. Zeichen) und enden mit dem Code "00H".

Für den Fall, dass beide Zeichenfolgen den gleichen Inhalt und die gleiche Größe haben, ist das Ergebnis der Anweisungen $=, $<= und $>= gleich 1.

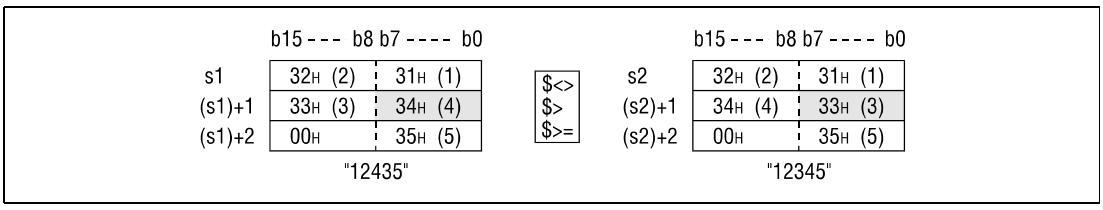


Bei unterschiedlichen Zeichenfolgen wird die Zeichenfolge mit dem höchsten Zeichencode als größer erkannt.



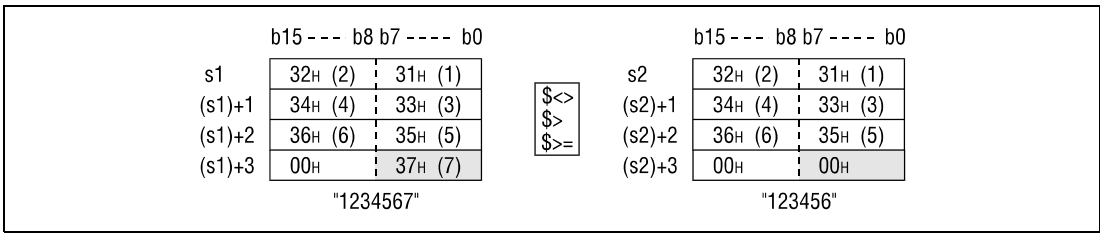
Das Ergebnis der Anweisungen $<>, $> und $>= ist im oben aufgeführten Fall 1.

Bei unterschiedlichen Zeichenfolgen entscheidet das erste von der anderen Zeichenfolge abweichende Zeichen durch die Höhe des Zeichencodes, welche Zeichenfolge die größere ist.



Das Ergebnis der Anweisungen $<>, $> und $>= ist im oben aufgeführten Fall 1.

Bei Zeichenfolgen unterschiedlicher Länge wird die längere Zeichenfolge als größer erkannt.



Das Ergebnis der Anweisungen $<>, $> und $>= ist im oben aufgeführten Fall 1.

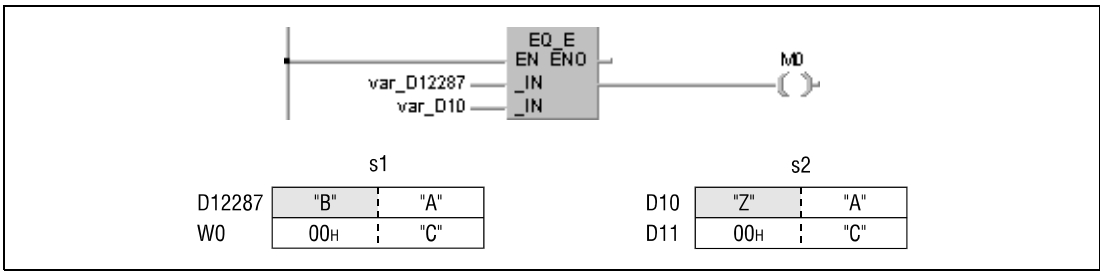
Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Der Code "00H" existiert nicht innerhalb des Bereichs der Daten s1 und s2 (Fehlercode 4101).

HINWEISE

Die Vergleichsanweisung für Zeichenfolgen überprüft gleichzeitig den für die Speicherung vorgesehenen Datenbereich. Aus diesem Grund wird in Fällen, in denen die abgelegte Zeichenfolge diesen Bereich verlässt , jedoch innerhalb der Zeichenfolge eine Abweichung erkannt wird, das Ergebnis der Anweisung ohne Fehlermeldung ausgegeben.



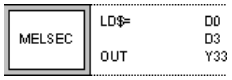
In dem oben angegebenen Beispiel liegen die höchstwertigen 16 Bit (D12288) der in s1 abgelegten Zeichenfolge außerhalb des für die Speicherung vorgesehenen Datenbereichs und werden vom Programm in W0 umbenannt. Da jedoch das zweite Zeichen der in s1 abgelegten Zeichenfolge von dem Zeichen in s2 abweicht, ist das Ergebnis der Vergleichsoperation 0. Da die Zeichenfolge außerhalb dieses Datenbereichs liegt und der Vergleich der Inhalte eine Abweichung ergibt, wird das Ergebnis (0) ohne Fehlermeldung ausgegeben.

Diese Programme sind ohne Programm-Organisationseinheits (POE)-Header im GX IEC Developer nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.

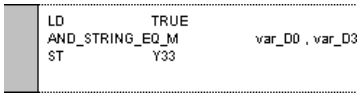
Beispiel 1 Vergleichsanweisung $=

Das folgende Programm vergleicht die Zeichenfolgen in D0 mit den Zeichenfolgen in D3. Sind beide Werte gleich, wird Y33 gesetzt.

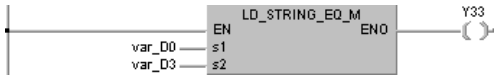
MELSEC-Anweisungsliste



IEC-Anweisungsliste



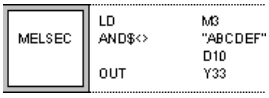
Kontaktplan



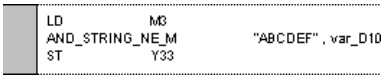
Beispiel 2 Vergleichsanweisung $<>

Das folgende Programm vergleicht die Zeichenfolge "ABCDEF" mit der in D10 abgelegten Zeichenfolge. Ist der Inhalt in D10 von der Zeichenfolge "ABCDEF" verschieden, wird Y33 gesetzt.

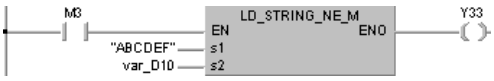
MELSEC-Anweisungsliste



IEC-Anweisungsliste



Kontaktplan



Beispiel 3 Vergleichsanweisung $>

Das folgende Programm vergleicht die Zeichenfolge in D10 mit der Zeichenfolge in D100. Wird die Zeichenfolge in D10 als größer erkannt, wird Y33 gesetzt.

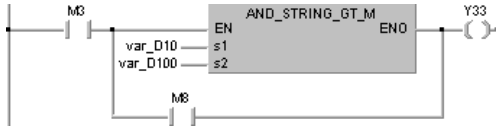
MELSEC-Anweisungsliste

```
MELSEC   LD      M3
         LD$>    D10
                 D100
         OR      M8
         ANB
         OUT     Y33
```

IEC-Anweisungsliste

```
LD              M3
AND(
                M8
OR_STRING_GT_M          var_D10 , var_D100
)
ST              Y33
```

Kontaktplan



Beispiel 4 Vergleichsanweisung $<=

Das folgende Programm vergleicht die Zeichenfolgen in D0 mit der Zeichenfolge "12345". Wird die Zeichenfolge in D0 als kleiner erkannt, wird Y33 gesetzt.

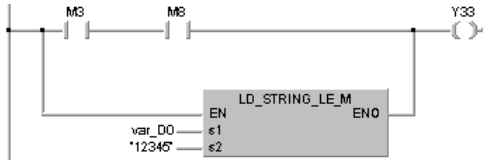
MELSEC-Anweisungsliste

```
MELSEC   LD      M3
         AND     M8
         OR$<=   D0
                 "12345"
         OUT     Y33
```

IEC-Anweisungsliste

```
LD              M3
AND             M8
OR_STRING_LE_M          var_D0 , "12345"
ST              Y33
```

Kontaktplan



HINWEIS *Diese Programme sind ohne Programm-Organisationseinheits (POE)-Header im GX IEC Developer nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

6.1.5 BKCMP, BKCMPP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

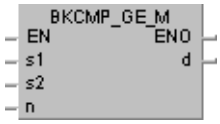
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	—	●	●	—	—	—	—	●	—	SM0	5
s2	—	●	●	—	—	—	—	—	—		
d	●	●	●	—	—	—	—	—	—		
n	●	●	●	●	●	●	●	●	—		

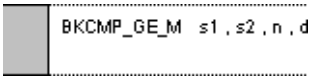
GX IEC Developer

MELSEC-Anweisungsliste

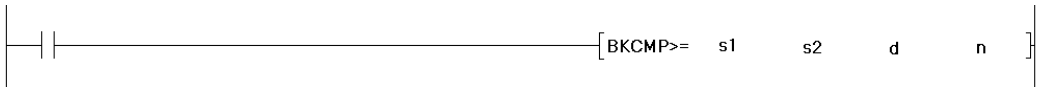
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

Operand	Befehlswert	Datentyp
s1	Vergleichsdaten oder erste Adresse, ab der die Vergleichsdaten gespeichert sind.	BIN-16-Bit
s2	Erste Adresse, ab der die Vergleichsdaten gespeichert sind.	BIN-16-Bit
d	Erste Adresse, ab der die Ergebnisdaten des Vergleichs gespeichert werden sollen.	Bit
n	Anzahl der zu vergleichenden Adressen (Länge des Datenblocks).	BIN-16-Bit

Funktionsweise

Blockweiser Vergleich von Binärdaten

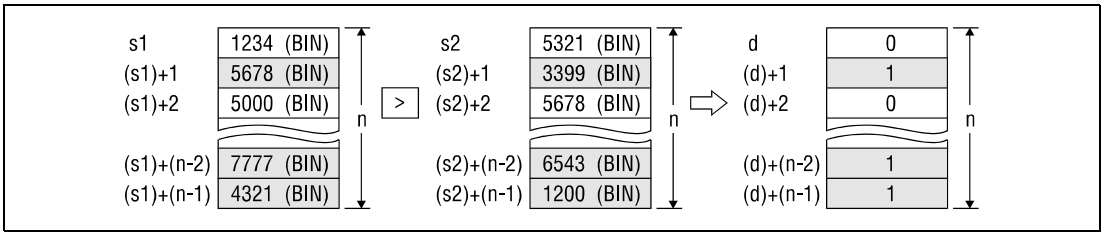
BKCMP Vergleichsanweisungen

Eine Vergleichsanweisung für Binärdatenblöcke besteht aus der Anweisung selbst, den Daten s1 und s2, die miteinander verglichen werden sollen, der Zielbezeichnung d, in dem die Ergebnisse abgelegt werden, und der Anzahl n der zu vergleichenden Datenblöcke.

Verglichen wird jeweils der n-te 16-Bit-Block von s1 mit dem n-ten 16-Bit-Block von s2, beginnend bei dem niedrigstwertigen 16-Bit-Block. Das Ergebnis jedes Blockvergleichs wird in d gespeichert.

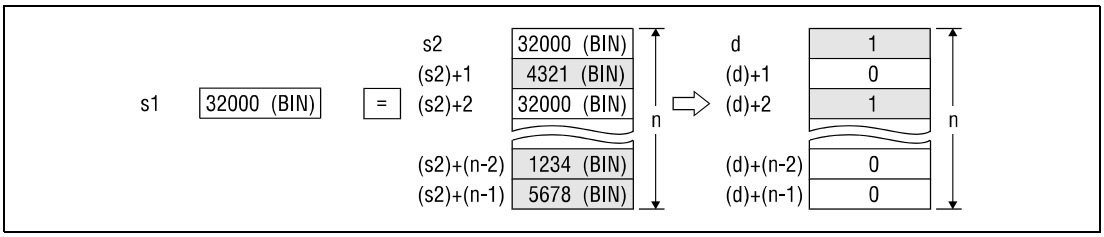
Ist das Vergleichsergebnis eines Blockes 1, lautet der entsprechende Eintrag in d gleich 1.

Ist das Vergleichsergebnis eines Blockes 0, lautet der entsprechende Eintrag in d gleich 0.



Die Vergleichsoperationen werden in Einheiten zu 16 Bit durchgeführt.

Eine in s1 abgelegte Konstante muss eine 16-Bit-Binärzahl zwischen -32768 und 32767 sein.



Die Vergleichsergebnisse der jeweiligen Anweisung sind in nachfolgender Tabelle aufgeführt.

Symbol der Anweisung	Zustand des Ausgangs für n-ten 16-Bit-Block	
	1 wenn:	0 wenn:
BKCMP=	s1 = s2	s1 ≠ s2
BKCMP<>	s1 ≠ s2	s1 = s2
BKCMP>	s1 > s2	s1 ≤ s2
BKCMP<=	s1 ≤ s2	s1 > s2
BKCMP<	s1 < s2	s1 ≥ s2
BKCMP>=	s1 ≥ s2	s1 < s2

Wenn alle in d gespeicherten Blockvergleichsergebnisse den Wert 1 haben, wird das Blockvergleichssignal SM704 gesetzt.

Die Speicherplätze in d, die vor dem Eintrag des Vergleichsergebnisses den Wert 1 haben, behalten diesen Wert. Aus diesem Grund sollten die Speicherplätze in d bei Verwendung der BKCMP_P-Anweisung vor dem erneuten Eintrag von Vergleichsergebnissen, z.B. nach einer Veränderung der Daten s1 und s2, mit dem Wert 0 beschrieben werden.

Fehler-quellen

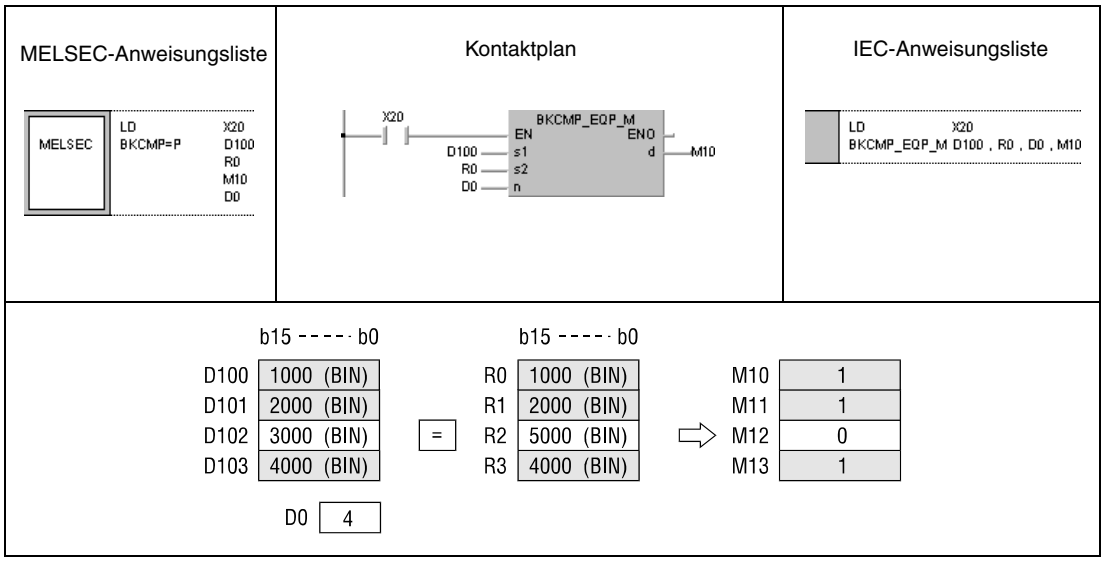
In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die in n angegebene Anzahl von Datenblöcken in s1, s2 und d liegt außerhalb des für die Speicherung vorgesehenen Bereiches der Operanden (MELSEC Q: Fehlercode 4101).
- Der Bereich [s1 bis (s1)+(n-1)] überlappt den Bereich [d bis (d)+(n-1)] (Q-Serie/System Q: Fehlercode 4101).
- Der Bereich [s2 bis (s2)+(n-1)] überlappt den Bereich [d bis (d)+(n-1)] (Q-Serie/System Q: Fehlercode 4101).
- Der Bereich [s1 bis (s1)+(n-1)] überlappt den Bereich [s2 bis (s2)+(n-1)] (Q-Serie/System Q: Fehlercode 4101).

Beispiel 1

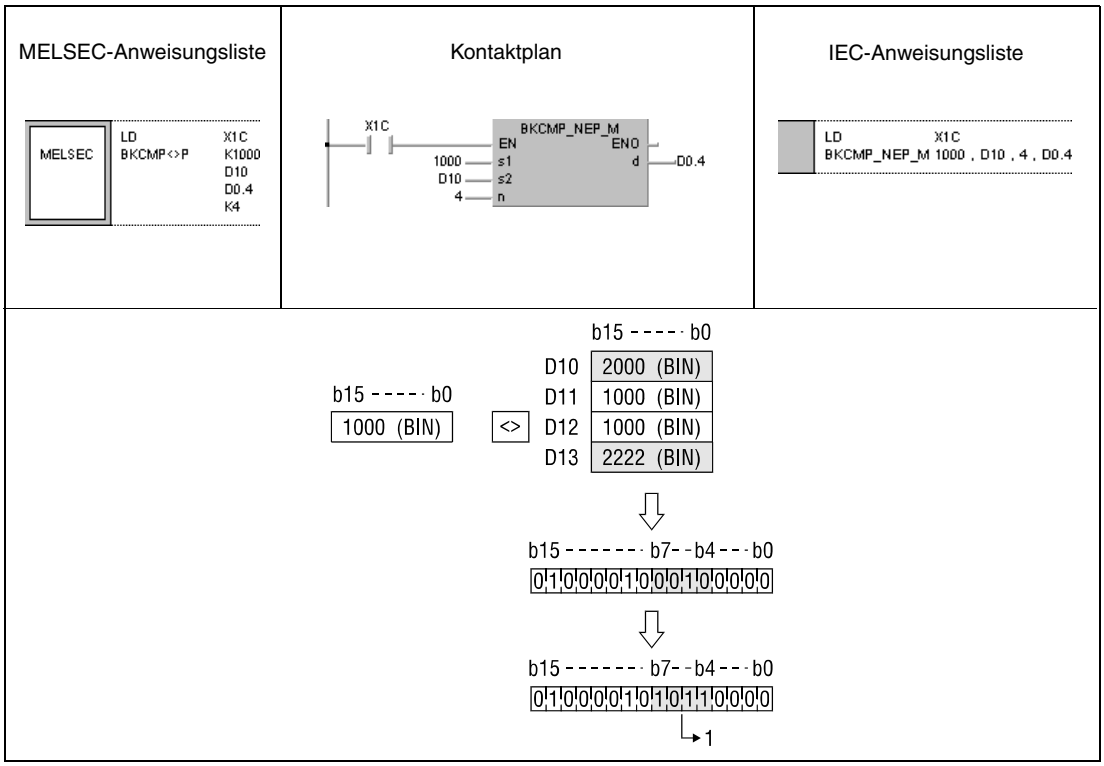
Vergleichsanweisung BKCMP=P

Das folgende Programm führt mit positiver Flanke (ansteigender Flanke) von X20 eine Vergleichsoperation zwischen den Datenblöcken beginnend bei D100 und denen beginnend bei R0 aus. Die Ergebnisse des Vergleichs werden bei M10 beginnend gespeichert. Die Anzahl der zu vergleichenden Blöcke (4) ist in D0 hinterlegt.



Beispiel 2 Vergleichsanweisung BKCMP<>P

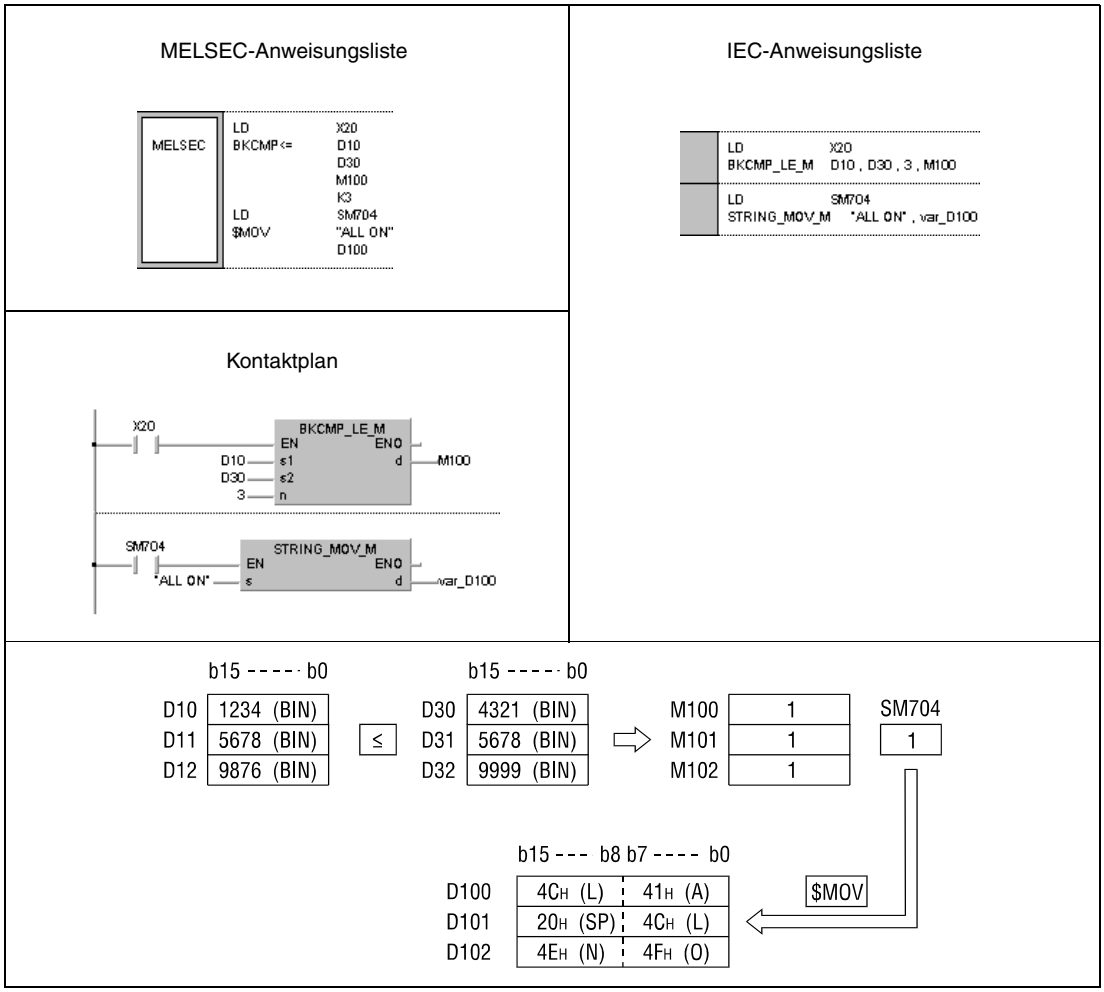
Das folgende Programm führt mit positiver Flanke von X1C eine Vergleichsoperation zwischen der Konstanten K1000 und den Datenblöcken beginnend bei D10 aus. Die Anzahl der zu vergleichenden Blöcke (4) gibt die Konstante K4 an. Die Vergleichsergebnisse werden in D0 von Bit 4 (b4) bis Bit 7 (b7) gespeichert.



[1] Diese beiden Bits verändern ihren Zustand nach der Operation nicht (siehe Funktionsweise).

Beispiel 3 Vergleichsanweisung BKCMP<=

Das folgende Programm führt für die Einschaltdauer von X20 eine Vergleichsoperation zwischen den Datenblöcken beginnend bei D10 und den Datenblöcken beginnend bei D30 aus. Die Anzahl der zu vergleichenden Blöcke (3) gibt die Konstante K3 an. Die Vergleichsergebnisse werden bei M100 beginnend gespeichert. Haben alle Ergebnisse in M100 den Wert 1, wird das Blockvergleichssignal SM704 gesetzt und die Zeichenfolge "ALL ON" an D100 übertragen.



HINWEIS *Dieses Programm ist ohne Programm-Organisationseinheits (POE)-Header im GX IEC Developer nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

6.2 Arithmetikanweisungen

Arithmetikanweisungen sind Anweisungen zur Ausführung einfacher mathematischer Rechenfunktionen wie Addition, Subtraktion, Multiplikation und Division von zwei Datensätzen.

Insgesamt stehen 54 (Q-Serie und System Q), bzw. 40 (A-Serie) verschiedene Arithmetikfunktionen zur Verfügung.

Funktion	BIN		BCD	
	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor
+ Addition	+	PLUS_M, PLUS_3_M	B+	BPLUS_M, BPLUS_3_M
	+P	PLUSP_M, PLUSP_3_M	B+P	BPLUSP_M, BPLUSP_3_M
	D+	DPLUS_M, DPLUS_3_M	DB+	DBPLUS_M, DBPLUS_3_M
	D+P	DPLUSP_M, DPLUSP_3_M	DB+P	DBPLUSP_M, DBPLUSP_3_M
– Subtraktion	-	MINUS_M, MINUS_3_M	B-	BMINUS_M, BMINUS_3_M
	-P	MINUSP_M, MINUSP_3_M	B-P	BMINUSP_M, BMINUSP_3_M
	D-	DMINUS_M, DMINUS_3_M	DB-	DBMINUS_M, DBMINUS_3_M
	D-P	DMINUSP_M, DMINUSP_3_M	DB-P	DBMINUSP_M, DBMINUSP_3_M
× Multiplikation	×	MULTI_3_M	B×	BMULTI_M
	×P	MULTIP_3_M	B×P	BMULTIP_M
	D×	DMULTI_3_M	DB×	DBMULTI_M
	D×P	DMULTIP_3_M	DB×P	DBMULTIP_M
/ Division	/	DIVID_3_M	B/	BDIVID_M
	/P	DIVIDP_3_M	B/P	BDIVIDP_M
	D/	DDIVID_3_M	DB/	DBDIVID_M
	D/P	DDIVIDP_3_M	DB/P	DBDIVIDP_M
+1 Inkrement	INC	INC_M		
	INCP	INCP_M		
	DINC	DINC_M		
	DINCP	DINCP_M		
–1 Dekrement	DEC	DEC_M		
	DECP	DECP_M		
	DDEC	DDEC_M		
	DDECP	DDECP_M		

HINWEIS *Sie sollten in den IEC-Editoren die IEC-Befehle nutzen.*

Funktion	Gleitkommazahlen		BIN-Block	
	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor
+ Addition	E+	EPLUS_M, EPLUS_3_M	BK+	BKPLUS_M
	E+P	EPLUSP_M, EPLUSP_3_M	BK+P	BKPLUSP_M
– Subtraktion	E-	EMINUS_M, EMINUS_3_M	BK-	BKMINUS_M
	E-P	EMINUSP_M, EMINUSP_3_M	BK-P	BKMINUSP_M
× Multiplikation	E×	EMUL_M		
	E×P	EMULP_M		
/ Division	E/	EDIV_M		
	E/P	EDIVP_M		

Funktion	Zeichenfolgen	
	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor
+ Addition	$+	STRING_PLUS_M, STRING_PLUS_3_M
	$+P	STRING_PLUSP_M, STRING_PLUSP_3_M

Die Arithmetikanweisungen für Gleitkommazahlen, BIN-Datenblöcke und Zeichenfolgen gelten nur für die Q-Serie.

Die Arithmetikfunktion mit Binärdaten

Übersteigt das Ergebnis einer Addition den Wert 32767 (2147483647 bei einer 32-Bit-Anweisung), wird das Resultat negativ.

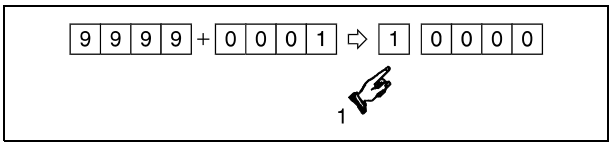
Unterschreitet das Ergebnis einer Subtraktion den Wert -32768 (-2147483648 bei einer 32-Bit-Anweisung) wird das Resultat positiv.

Die Berechnung positiver und negativer Werte stellt sich wie folgt dar:

5 + 8 = 13
5 - 8 = -3
5 × 3 = 15
-5 × 3 = -15
-5 × (-3) = 15
5 / 3 = 1 und Rest 2
-5 / 3 = -1 und Rest -2
5 / (-3) = -1 und Rest 2
-5 / (-3) = 1 und Rest -2

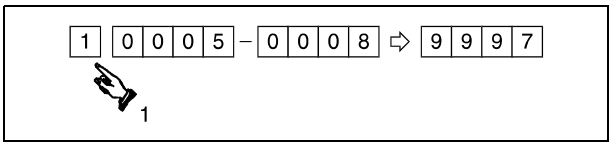
Arithmetikfunktion mit BCD-Daten

Übersteigt das Ergebnis einer Addition den Wert 9999 (99999999 bei einer 32-Bit-Anweisung), bleibt der Übertrag (Carry) unberücksichtigt.



[1] Übertrag fällt weg

Ist der Minuend einer Subtraktion kleiner als der Subtrahend, wird der Übertrag (Carry) wie folgt verarbeitet:



[1] Übertrag

6.2.1 +, +P, -, -P

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

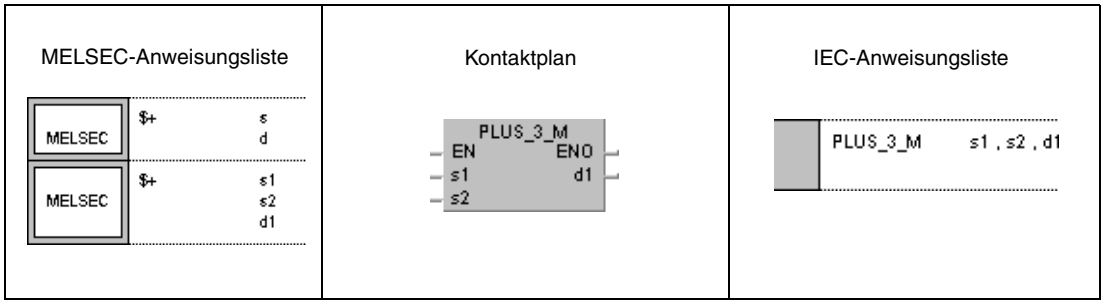
	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				K1 ↓ K4	5 ●[1]	●		●
d		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●										
s1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					7 ●[1]			
s2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●								
d1		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●										

[1] Die Anzahl der Schritte bei Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	●	●	●	●	●	●	●	●	—	—	3
d	●	●	●	●	●	●	●	—	—	—	
s1	●	●	●	●	●	●	●	●	—	—	4
s2	●	●	●	●	●	●	●	●	—	—	
d1	●	●	●	●	●	●	●	—	—	—	

GX IEC Developer



GX Developer

Variablen

Operand	Befehlswert	Datentyp
s	Summanden- oder Subtrahendendaten oder erste Adresse, ab der die Summanden- oder Subtrahendendaten gespeichert sind.	BIN-16-Bit
d	Summanden- oder Minuendendaten oder erste Adresse, ab der die Summe oder Differenz gespeichert ist.	
s1	Summanden- oder Minuendendaten oder erste Adresse, ab der die Summanden- oder Minuendendaten gespeichert sind.	
s2	Summanden- oder Subtrahendendaten oder erste Adresse, ab der die Summanden- oder Subtrahendendaten gespeichert sind.	
d1	Erste Adresse, ab der die Summe oder Differenz gespeichert ist.	

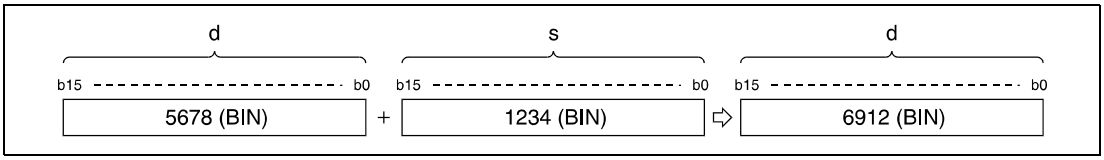
Funktions-weise

Addition und Subtraktion von Binärdaten (16 Bit)

+ BIN-Addition (16 Bit)

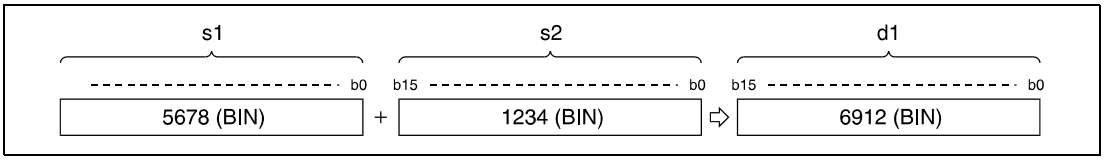
- 1.Variante:

Die in d angegebenen Binärdaten werden zu den Binärdaten in s addiert. Das Additionsergebnis wird in d gespeichert.



- 2.Variante:

Die in s1 angegebenen Binärdaten werden zu den Binärdaten in s2 addiert. Das Additionsergebnis wird an den in d1 angegebenen Operanden ausgegeben.



Der in s, d, s1, s2 und d1 angegebene Datenwert muss eine 16-Bit-Binärzahl zwischen -32768 und 32767 sein.

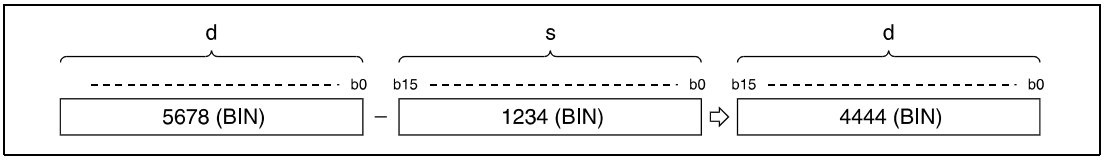
Das jeweils höchstwertige Bit (b15) legt fest, ob die Datenwerte in s, d, s1, s2 oder d1 positiv (Bit = 0) oder negativ (Bit = 1) sind.

Wenn das niedrigstwertige Bit (b0) unterschritten oder das höchstwertige Bit (b15) überschritten wird, wird das Carry Flag nicht gesetzt.

- BIN-Subtraktion (16 Bit)

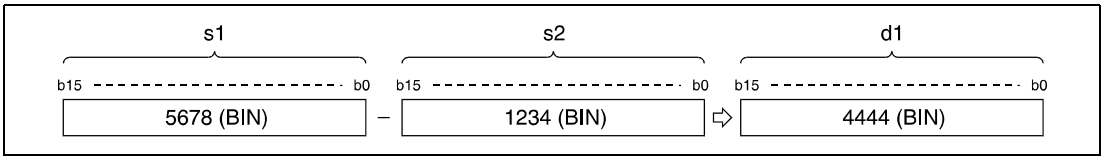
- 1.Variante:

Die in s angegebenen Binärdaten werden von den Binärdaten in d subtrahiert. Das Subtraktionsergebnis wird in d gespeichert.



- 2.Variante:

Die in s2 angegebenen Binärdaten werden von den Binärdaten in s1 subtrahiert. Das Ergebnis wird an den in d1 angegebenen Operanden ausgegeben.



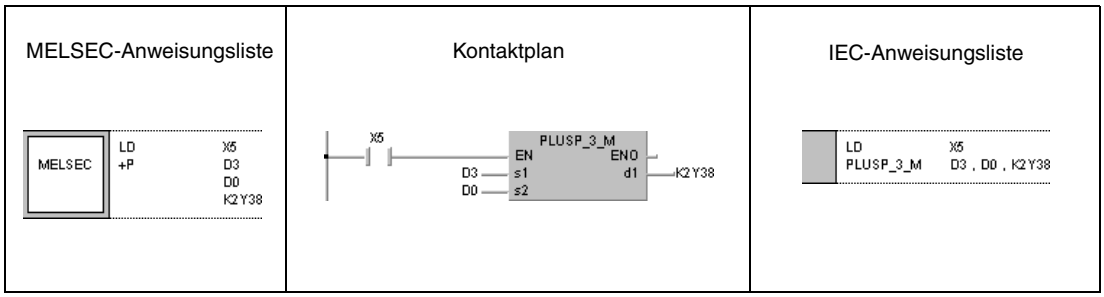
Der in s, d, s1, s2 und d1 angegebene Datenwert muss eine 16-Bit-Binärzahl zwischen -32768 und 32767 sein.

Das jeweils höchstwertige Bit (b15) legt fest, ob die Datenwerte in s, d, s1, s2 oder d1 positiv (Bit = 0) oder negativ (Bit = 1) sind.

Wenn das niedrigstwertige Bit (b0) unterschritten oder das höchstwertige Bit (b15) überschritten wird, wird das Carry Flag nicht gesetzt.

Beispiel 1 +P

Das folgende Programm addiert den Inhalt von D3 mit positiver Flanke von X5 zu dem Inhalt von D0 und gibt das Ergebnis an Y38 bis Y3F aus.



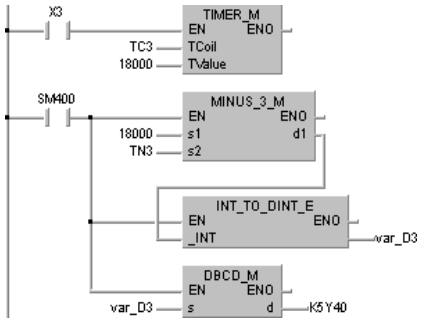
Beispiel 2 -

Das folgende Programm gibt die Differenz zwischen Soll- und Istwert von Timer T3 an Y40 bis Y53 in BCD aus.

MELSEC-Anweisungsliste

```
MELSEC
        LD      X3
        OUT     T3
                K18000
        LD      SM400
        -       K18000
                T3
                D3
        DBCD    D3
                K5Y40
```

Kontaktplan



IEC-Anweisungsliste

```
LD              X3
TIMER_M         TC3 , K18000
LD              SM400
MINUS_3_M       K18000 , TN3 , D3
INT_TO_DINT_E   D3 , var_D3
DBCD_M          var_D3 , K5Y40
```

HINWEIS Diese Programmbeispiele sind ohne Variablendefinition im Header der Programm-Organisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.

6.2.2 D+, D+P, D-, D-P

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
s	●	●	●	●	●	●	●	●	●	●	●	●	●		●		●	●				K1 ↓ K8	9 ●[1]	●		●
d		●	●	●	●	●	●	●	●	●	●	●	●		●											
s1	●	●	●	●	●	●	●	●	●	●	●	●	●		●		●	●					11 ●[1]			
s2	●	●	●	●	●	●	●	●	●	●	●	●	●		●		●	●								
d1		●	●	●	●	●	●	●	●	●	●	●	●		●											

[1] Die Anzahl der Schritte bei Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

Operanden MELSEC Q

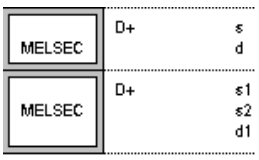
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	●	●	●	●	●	●	●	●	—	—	3[1)]
d	●	●	●	●	●	●	●	—	—	—	
s1	●	●	●	●	●	●	●	●	—	—	4[2)]
s2	●	●	●	●	●	●	●	●	—	—	
d1	●	●	●	●	●	●	●	—	—	—	

[1] Die Anzahl der Programmschritte ist abhängig vom Typ der CPU und von den verwendeten Operanden.
Bei Verwendung einer QnA-CPU oder Single-Prozessor-Q-CPU: 3
Bei Verwendung einer Multi-Processor-Q-CPU und interner Wortoperanden (außer File-Register ZR): 5
Konstanten: 5
Bit-Operanden, deren Adresse ein Vielfaches von 16 ist, die die Bit-Blockbezeichnung K8 haben und die nicht durch Index-Vergabe bearbeitet werden: 5
Bei Verwendung einer Multi-Processor-Q-CPU und anderer Operanden als oben aufgeführt: 3

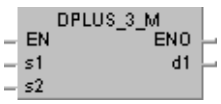
[2] Die Anzahl der Programmschritte ist abhängig vom Typ der CPU und von den verwendeten Operanden.
Bei Verwendung einer QnA-CPU oder Single-Prozessor-Q-CPU: 4
Bei Verwendung einer Multi-Processor-Q-CPU und interner Wortoperanden (außer File-Register ZR): 6
Konstanten: 6
Bit-Operanden, deren Adresse ein Vielfaches von 16 ist, die die Bit-Blockbezeichnung K8 haben und die nicht durch Index-Vergabe bearbeitet werden: 6
Bei Verwendung einer Multi-Processor-Q--CPU und anderer Operanden als oben aufgeführt: 4

GX IEC Developer

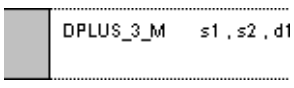
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

BIN-64-Bit

Operand	Befehlswert	Datentyp
s	Summanden- oder Subtrahendendaten oder erste Adresse, ab der die Summanden- oder Subtrahendendaten gespeichert sind.	BIN-32-Bit
d	Summanden- oder Minuendendaten oder erste Adresse, ab der die Summe oder Differenz gespeichert ist.	
s1	Summanden- oder Minuendendaten oder erste Adresse, ab der die Summanden- oder Minuendendaten gespeichert sind.	
s2	Summanden- oder Subtrahendendaten oder erste Adresse, ab der die Summanden- oder Subtrahendendaten gespeichert sind.	
d1	Erste Adresse, ab der die Summe oder Differenz gespeichert ist.	

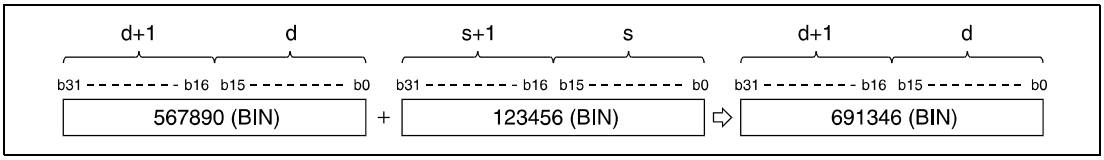
Funktionsweise

Addition und Subtraktion von Binärdaten (32-Bit)

D+ BIN-Addition (32 Bit)

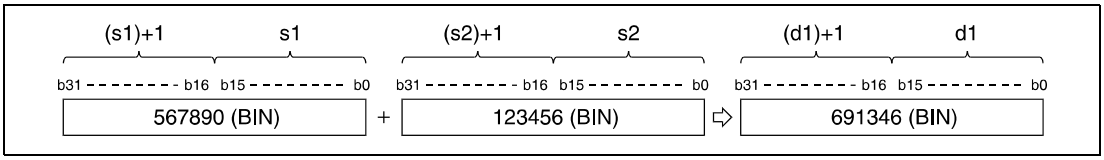
● 1.Variante:

Die in d angegebenen Binärdaten werden zu den Binärdaten in s addiert. Das Additionsergebnis wird in d gespeichert.



● 2.Variante:

Die in s1 angegebenen Binärdaten werden zu den Binärdaten in s2 addiert. Das Additionsergebnis wird an den in d1 angegebenen Operanden ausgegeben.



Der in s, d, s1, s2 und d1 angegebene Datenwert muss eine 32-Bit-Binärzahl zwischen -2147483648 und 2147483647 sein.

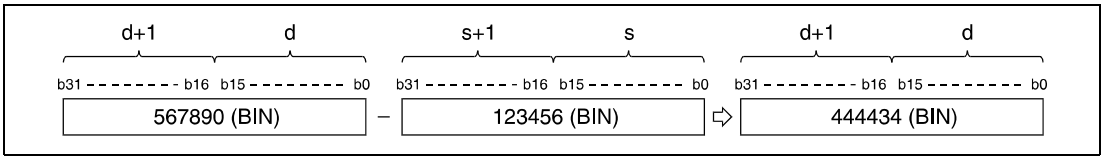
Das jeweils letzte Bit (b31) legt fest, ob die Datenwerte in s, d, s1, s2 oder d1 positiv (Bit = 0) oder negativ (Bit = 1) sind.

Wenn das niedrigstwertige Bit (b0) unterschritten oder das höchstwertige Bit (b31) überschritten wird, wird das Carry Flag nicht gesetzt.

D- BIN-Subtraktion (32 Bit)

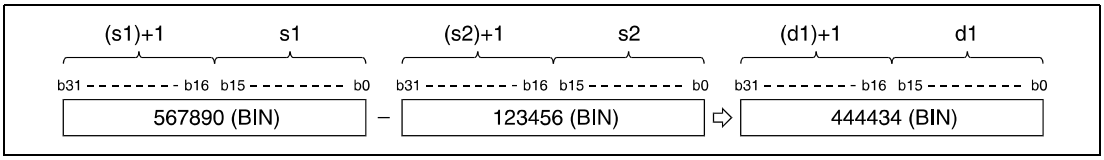
● 1.Variante:

Die in s angegebenen Binärdaten werden von den Binärdaten in d subtrahiert. Das Subtraktionsergebnis wird in d gespeichert.



- 2.Variante:

Die in s2 angegebenen Binärdaten werden von den Binärdaten in s1 subtrahiert. Das Subtraktionsergebnis wird an den in d1 angegebenen Operanden ausgegeben.



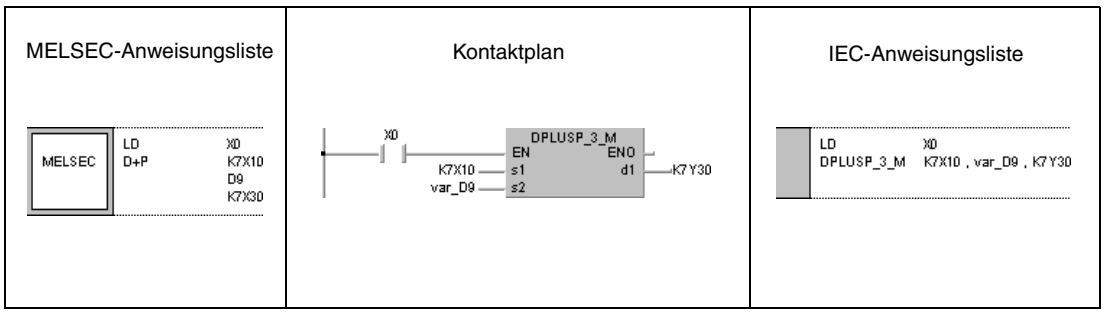
Der in s, d, s1, s2 und d1 angegebene Datenwert muss eine 32-Bit-Binärzahl zwischen -2147483648 und 2147483647 sein.

Das jeweils letzte Bit (b31) legt fest, ob die Datenwerte in s, d, s1, s2 oder d1 positiv (Bit = 0) oder negativ (Bit = 1) sind.

Wenn das niedrigstwertige Bit (b0) unterschritten oder das höchstwertige Bit (b31) überschritten wird, wird das Carry Flag nicht gesetzt.

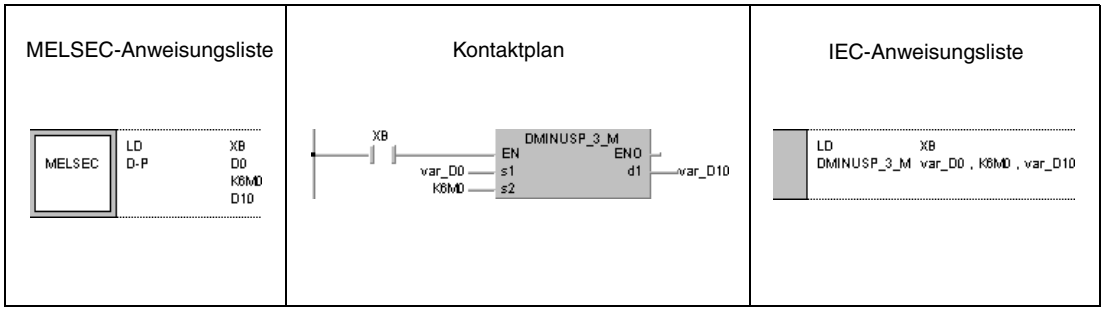
Beispiel 1 D+P

Das folgende Programm addiert mit positiver Flanke von X0 der Inhalt von X10 bis X2B zu dem Inhalt von D9 und D10 und gibt das Ergebnis an Y30 bis Y4B aus.



Beispiel 2 D-P

Das folgende Programm subtrahiert mit positiver Flanke von XB die Daten von M0 bis M23 von D0 und D1 und gibt das Ergebnis an D10 und D11 aus.



HINWEIS *Diese Programmbeispiele sind ohne Variablendefinition im Header der Programm-Organisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

6.2.3 x, xP, /, /P

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)								Konstante		Pointer		Ebene				
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
s1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				K1 ↓ K4	7 ●[1]	●		●
s2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●								
d1		●	●	●	●	●	●	●	●	●	●	●	●		●											

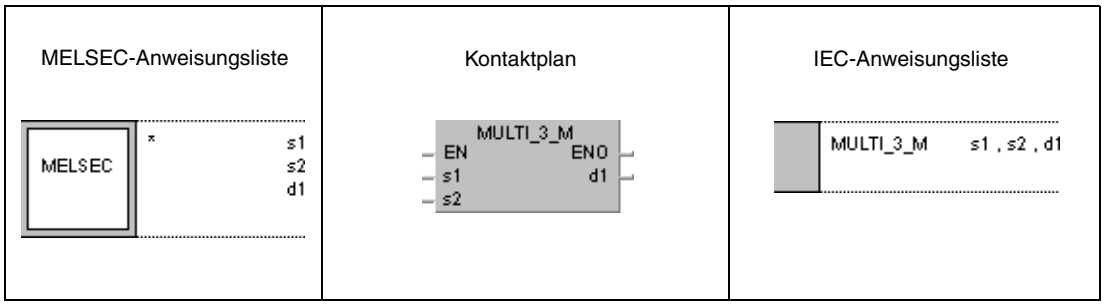
[1] Die Anzahl der Schritte bei Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.

Operanden MELSEC Q

	Operanden								Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	●	●	●	●	●	●	●	●	—	SM0	4[1)]
s2	●	●	●	●	●	●	●	●	—		
d1	●	●	●	●	●	●	●	—	—		

[1] Die Anzahl der Programmschritte ist abhängig vom Typ der CPU und von den verwendeten Operanden.
Bei Verwendung einer QnA-CPU: 4
Bei Verwendung einer Q-CPU und
interner Wortoperanden (außer File-Register ZR): 3
Konstanten: 3
Bit-Operanden, deren Adresse ein Vielfaches von 16 ist, die die Bit-Blockbezeichnung K4 haben und die nicht durch Index-Vergabe bearbeitet werden: 3
Bei Verwendung einer Q-CPU und anderer Operanden als oben aufgeführt: 4

GX IEC Developer



GX Developer

Variablen

Operand	Befehlswert	Datentyp
s1	Multiplikatoren- oder Dividendendaten oder erste Adresse, ab der die Multiplikatoren- oder Dividendendaten gespeichert sind.	BIN-16-Bit
s2	Multiplikatoren- oder Divisorendaten oder erste Adresse, ab der die Multiplikatoren- oder Divisorendaten gespeichert sind.	BIN-16-Bit
d1	Erste Adresse, ab der das Produkt oder der Quotient gespeichert ist.	BIN-32-Bit

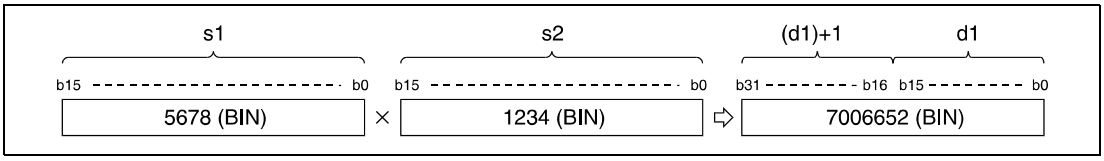
BIN-32-Bit

Funktionsweise

Multiplikation und Division von Binärdaten (16 Bit)

x BIN-Multiplikation (16 Bit)

Die in s1 angegebenen Binärdaten werden mit den Binärdaten in s2 multipliziert, und das Multiplikationsergebnis wird in d1 gespeichert.



Handelt es sich bei dem Ergebnis in d1 um einen Bit-Operanden, ist in der Bit-Struktur mit den niedrigwertigeren Bits zu beginnen.

Beispiel:

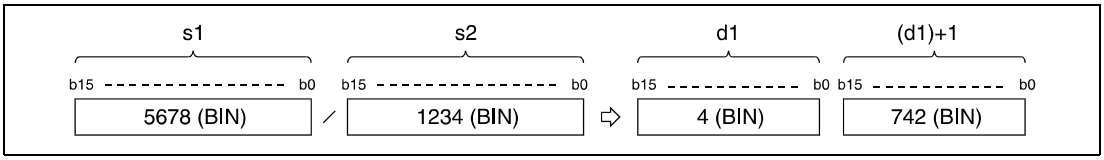
K1: niedrigsten 4 Bit (b0 bis b3)
K4: niedrigsten 16 Bit (b0 bis b15)
K8: 32 Bit (b0 bis b31)

Der in s1 und s2 angegebene Datenwert muss eine 16-Bit-Binärzahl zwischen -32768 und 32767 sein.

Das jeweils höchstwertige Bit (b15 bzw. b31 bei d1) legt fest, ob die Datenwerte in s1, s2 oder d1 positiv (Bit = 0) oder negativ (Bit = 1) sind.

/ BIN-Division (16 Bit)

Die in s1 angegebenen Binärdaten werden durch die Binärdaten in s2 dividiert, und das Divisionsergebnis wird in d1 gespeichert.



Bei Wortoperanden wird das Divisionsergebnis als 32-Bit-Datenwert, aufgeteilt in Quotient und Restbetrag, abgelegt. Bit-Operanden erlauben ausschließlich die Speicherung des Quotienten.

Der Quotient wird in den niedrigstwertigen 16 Bit gespeichert. Der Restwert wird in den höchstwertigen 16 Bit gespeichert (nur bei Wortoperanden).

Der in s1 und s2 angegebene Datenwert muss eine 16-Bit-Binärzahl zwischen -32768 und 32767 sein.

Das jeweils höchstwertige Bit (b15) legt fest, ob die Datenwerte in s1, s2, d1 oder (d1)+1 positiv oder negativ sind.

Fehler-quellen

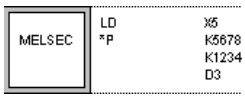
In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- A1 oder V wurde in d1 bestimmt (A-Serie).
- Der Divisor s2 ist gleich 0 (Q-Serie/System Q = Fehlercode 4100).

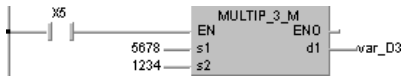
Beispiel 1 xP

Das folgende Programm speichert das Ergebnis der Multiplikation von 5678 und 1234 mit positiver Flanke von X5 als Binärwert in D3 und D4.

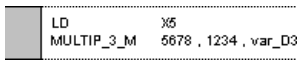
MELSEC-Anweisungsliste



Kontaktplan



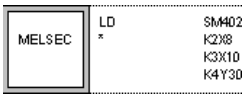
IEC-Anweisungsliste



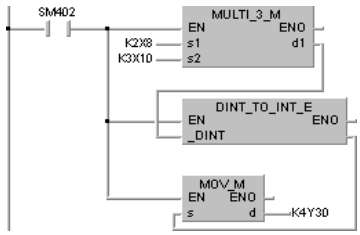
Beispiel 2 x

Das folgende Programm gibt das Ergebnis der Multiplikation der Binärdaten X8 bis XF mit X10 bis X1B an die Ausgänge Y30 bis Y3F aus.

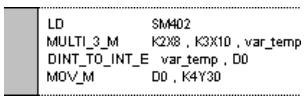
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



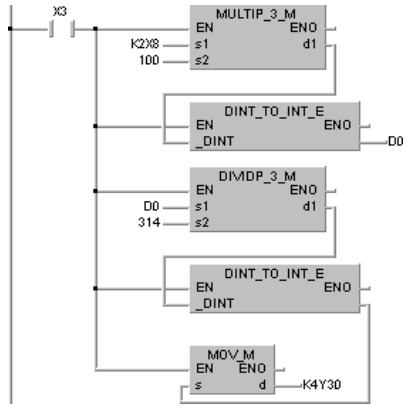
Beispiel 3 /P

Das folgende Programm dividiert mit positiver Flanke von X3 die Daten aus X8 bis XF durch 3,14. Das Ergebnis wird an Y30 bis Y3F ausgegeben.

MELSEC-Anweisungsliste

```
MELSEC   LD    X3
         *P    K2X8
               K100
               D0
         /P    D0
               K314
               K4Y30
```

Kontaktplan



IEC-Anweisungsliste

```
LD              X3
MULTIP_3_M      K2X8 , K100 , var_D0
DINT_TO_INT_E   var_D0 , D0
DIVDP_3_M       D0 , K314 , var_temp
DINT_TO_INT_E   var_temp , D0
MOV_M           D0 , K4Y30
```

HINWEIS *Diese Programmbeispiele sind ohne Variablendefinition im Header der Programm-Organisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

6.2.4 Dx, DxP, D/, D/P

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
s1	●	●	●	●	●	●	●	●	●	●	●	●	●		●		●	●				K1 ↓ K8	11 ● 1	●		●
s2	●	●	●	●	●	●	●	●	●	●	●	●	●		●		●	●								
d1		●	●	●	●	●	●	●	●	●	●	●														

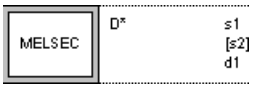
[1] Die Anzahl der Schritte bei Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

Operanden MELSEC Q

	Operanden								Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sonder-module U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	●	●	●	●	●	●	●	●	—	SM0	4
s2	●	●	●	●	●	●	●	●	—		
d1	●	●	●	—	—	—	—	—	—		

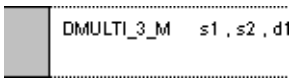
GX IEC Developer

MELSEC-Anweisungsliste



Kontaktplan

IEC-Anweisungsliste



GX Developer

Variablen

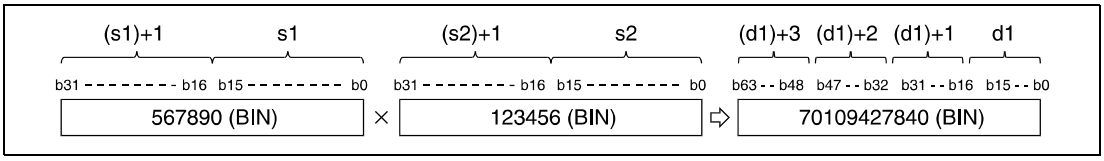
Operand	Befehlswert	Datentyp	
		MELSEC	IEC
s1	Multiplikatoren- oder Dividendendaten oder erste Adresse, ab der die Multiplikatoren- oder Dividendendaten gespeichert sind.	BIN-32-Bit	ANY32
s2	Multiplikatoren- oder Divisorendaten oder erste Adresse, ab der die Multiplikatoren- oder Divisorendaten gespeichert sind.	BIN-32-Bit	ANY32
d1	Erste Adresse, ab der das Produkt oder der Quotient gespeichert ist.	BIN-64-Bit	Array [1..2] of ANY32

Funktionsweise

Multiplikation und Division von Binärdaten (32-Bit)

Dx BIN-Multiplikation (32 Bit)

Die in s1 angegebenen Binärdaten werden mit den Binärdaten in s2 multipliziert, und das Multiplikationsergebnis wird in d1 gespeichert.



Handelt es sich bei dem Ergebnis in d1 um einen Bit-Operanden, können nur die niedrigstwertigen 32 Bit festgelegt werden.

Beispiel:

K1: niedrigsten 4 Bit (b0 bis b3)
K4: niedrigsten 16 Bit (b0 bis b15)
K8: 32 Bit (b0 bis b31)

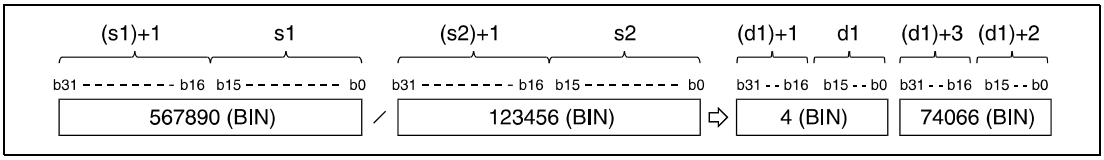
Werden die höchstwertigen 32 Bit eines Multiplikationsergebnisses für einen Bit-Operanden benötigt, müssen die Daten zunächst in einem Wortoperanden gespeichert werden. Anschließend werden die Daten (d1)+2 und (d1)+3 des Wortoperanden in den angegebenen Bit-Operanden übertragen.

Der in s1 und s2 angegebene Datenwert muss eine 32-Bit-Binärzahl zwischen -2147483648 und 2147483647 sein.

Das jeweils höchstwertige Bit (b31 bzw. b63 bei d1) legt fest, ob die Datenwerte in s1, s2 oder d1 positiv (Bit = 0) oder negativ (Bit = 1) sind.

D/ BIN-Division (32 Bit)

Die in s1 angegebenen Binärdaten werden durch die Binärdaten in s2 dividiert, und das Divisionsergebnis wird in d1 gespeichert.



Bei Wortoperanden wird das Divisionsergebnis als Array of 2 DINT (64-Bit), aufgeteilt in Quotient und Restbetrag, abgelegt. Bit-Operanden erlauben ausschließlich die Speicherung des Quotienten.

Der Quotient wird in den niedrigstwertigen Array-Elementen (32-Bit) gespeichert. Der Restwert wird in den höchstwertigen Array-Elementen (32-Bit) gespeichert (nur bei Wortoperanden).

Der in s1 und s2 angegebene Datenwert muss eine 32-Bit-Binärzahl zwischen -2147483648 und 2147483647 sein.

Das jeweils höchstwertigste Bit (b31) legt fest, ob die Datenwerte in s1, s2, d1 oder (d1)+2 positiv (Bit = 0) oder negativ (Bit = 1) sind.

Fehler-quellen

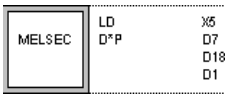
In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- A1 oder V wurde in s1 oder s2 und A0, A1, Z, V in d1 bestimmt (A-Serie).
- Der Divisor s2 ist gleich 0 (Q-Serie/System Q = Fehlercode 4100).

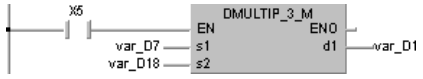
Beispiel 1 DxP

Das folgende Programm speichert das Multiplikationsergebnis der Binärdaten aus D7 und D8 und der Binärdaten aus D18 und D19 mit positiver Flanke von X5 in D1 bis D4.

MELSEC-Anweisungsliste



Kontaktplan



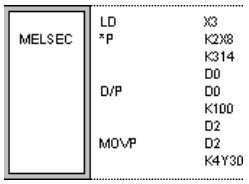
IEC-Anweisungsliste

```
LD              X5
DMULTIP_3_M     var_D7 , var_D18 , var_D1
```

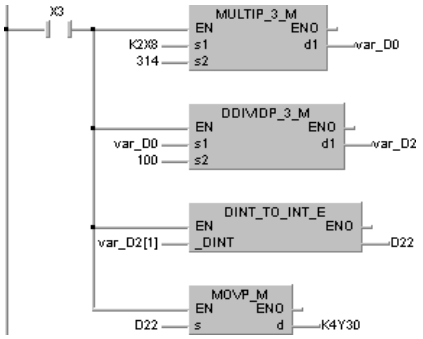
Beispiel 2 xP

Das folgende Programm gibt mit positiver Flanke von X3 das Ergebnis der Multiplikation der Daten in X8 bis XF mit 3,14 an Y30 bis Y3F aus.

MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste

```
LD              X3
MULTIP_3_M      K2X8 , K314 , var_D0
DDIVDP_3_M      var_D0 , K100 , var_D2
DINT_TO_INT_E   var_D2[1] , D2
MOVP_M          D2 , K4Y30
```

HINWEIS *Diese Programmbeispiele sind ohne Variablendefinition im Header der Programm-Organisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

6.2.5 B+, B+P, B-, B-P

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				K1 ↓ K4	7 ●[1]	●		●
d		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●										
s1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					9 ●[1]			
s2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●								
d1		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●										

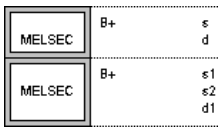
[1] Die Anzahl der Schritte bei Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

Operanden MELSEC Q

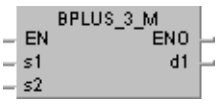
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sonder-module U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	●	●	●	●	●	●	●	●	—	SM0	3
d	●	●	●	●	●	●	●	—	—		
s1	●	●	●	●	●	●	●	●	—	SM0	4
s2	●	●	●	●	●	●	●	●	—		
d1	●	●	●	●	●	●	●	—	—		

GX IEC Developer

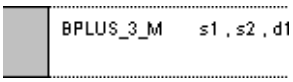
MELSEC-Anweisungsliste



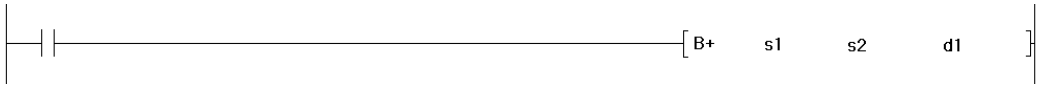
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

Operand	Befehlswert	Datentyp
s	Summanden- oder Subtrahendendaten oder erste Adresse, ab der die Summanden- oder Subtrahendendaten gespeichert sind.	BCD 4-stellig
d	Summanden- oder Minuendendaten oder erste Adresse, ab der die Summe oder Differenz gespeichert ist.	
s1	Summanden- oder Minuendendaten oder erste Adresse, ab der die Summanden- oder Minuendendaten gespeichert sind.	
s2	Summanden- oder Subtrahendendaten oder erste Adresse, ab der die Summanden- oder Subtrahendendaten gespeichert sind.	
d1	Erste Adresse, ab der die Summe oder Differenz gespeichert ist.	

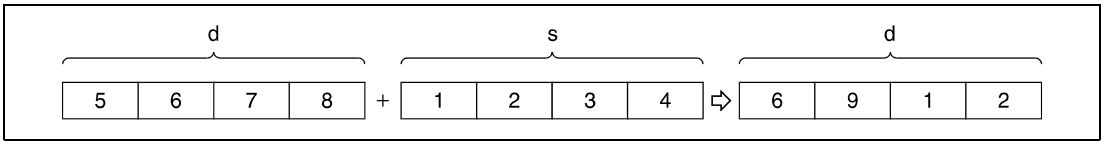
Funktionsweise

Addition und Subtraktion von BCD-Daten (4-stellig)

B+ BCD-Addition (4-stellig)

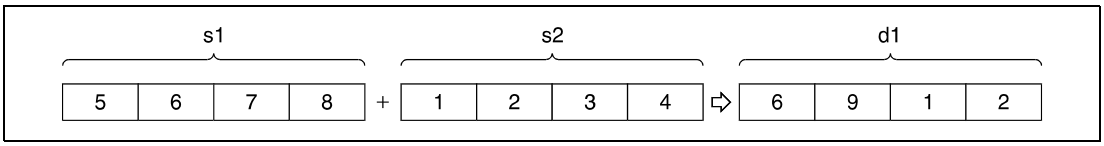
- 1. Variante:

Die in s angegebenen BCD-Daten werden zu den BCD-Daten in d addiert. Das Additionsergebnis wird in d gespeichert.



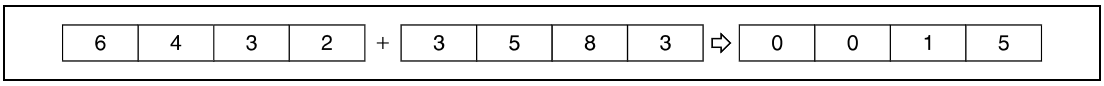
- 2. Variante:

Die in s1 angegebenen BCD-Daten werden zu den BCD-Daten in s2 addiert. Das Additionsergebnis wird an den in d1 angegebenen Operanden ausgegeben.



Der in s, d, s1, s2 und d1 angegebene Datenwert muss eine 4-stellige Dezimalzahl zwischen 0 und 9999 sein. Bei Zahlen mit weniger als 4 Stellen werden die ersten Stellen mit einer 0 aufgefüllt (z.B.: 12 = 0012).

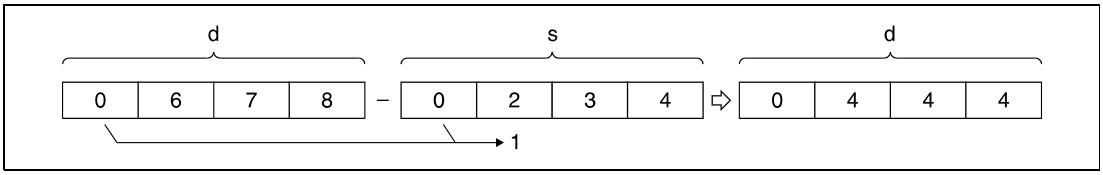
Das Carry Flag wird auch dann nicht gesetzt, wenn das Additionsergebnis den Wert 9999 übersteigt. Die Übertragstelle wird ignoriert.



B- BCD-Subtraktion (4-stellig)

- 1. Variante:

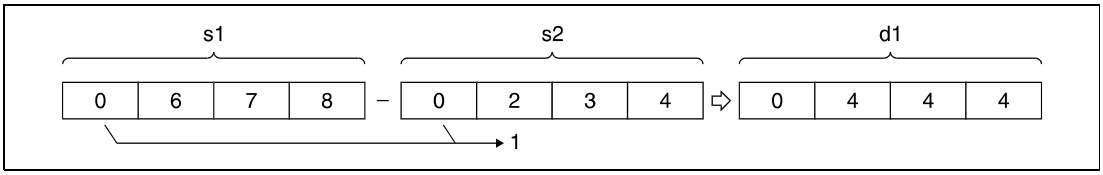
Die in s angegebenen BCD-Daten werden von den BCD-Daten in d subtrahiert. Das Subtraktionsergebnis wird in d gespeichert.



[1] Leerstellen werden mit einer 0 aufgefüllt.

- 2. Variante:

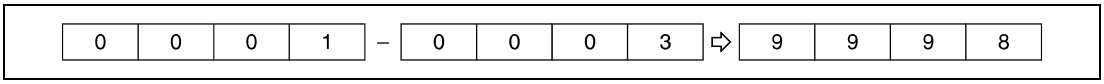
Die in s2 angegebenen BCD-Daten werden von den BCD-Daten in s1 subtrahiert. Das Ergebnis wird an den in d1 angegebenen Operanden ausgegeben.



[1] Leerstellen werden mit einer 0 aufgefüllt

Der in s, d, s1, s2 und d1 angegebene Datenwert muss eine 4-stellige Dezimalzahl zwischen 0 und 9999 sein.

Wird das Subtraktionsergebnis negativ, so wird der Minuend um die Anzahl der im Subtrahenden angegebenen Schritte reduziert, und das Carry Flag wird nicht gesetzt.



Das Programm muss so ausgelegt werden, dass der weitere Verlauf geregelt ist, wenn das Ergebnis entweder positiv oder negativ ist.

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- In s, d, s1, s2 oder d1 wurde ein von 0 bis 9999 verschiedener Datenwert eingetragen (Q-Serie/System Q = Fehlercode 4100).

Beispiel 1 B+P (s, d)

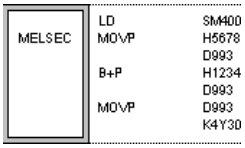
Das folgende Programm addiert die BCD-Daten 5678 und 1234, speichert das Ergebnis in D993 und gibt es an die Ausgänge Y30 bis Y3F aus.

In der ersten Programmzeile wird hierzu mit positiver Flanke von SM400 der Wert 5678 in D993 gespeichert.

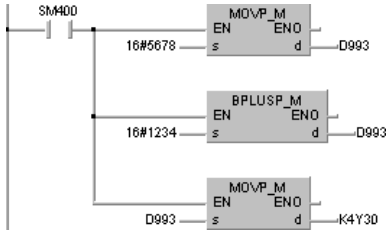
Im nachfolgenden Programmschritt wird der BCD-Wert 1234 zu dem BCD-Wert in D993 hinzuaddiert.

Die MOV-Anweisung im letzten Programmschritt gibt das Ergebnis aus D993 an Y30 bis Y3F aus.

MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste

```
LD      SM400
MOVP_M  16#5678 , D993
BPLUSP_M 16#1234 , D993
MOVP_M  D993 , K4Y30
```

Beispiel 2 B-P (s, d)

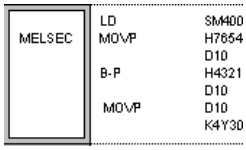
Das folgende Programm subtrahiert den BCD-Datenwert 4321 von 7654, überträgt das Ergebnis nach D10 und gibt das Ergebnis an die Ausgänge Y30 bis Y3F aus.

In der ersten Programmzeile wird hierzu mit positiver Flanke von SM400 der Wert 7654 in D10 gespeichert.

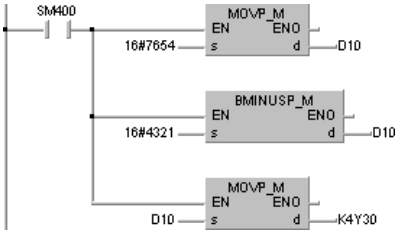
Im nachfolgenden Programmschritt wird der BCD-Wert 4321 von dem BCD-Wert in D10 subtrahiert.

Die MOV-Anweisung im letzten Programmschritt gibt das Ergebnis aus D10 an Y30 bis Y3F aus.

MELSEC-Anweisungsliste



Kontaktplan

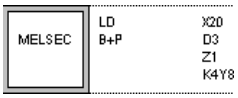
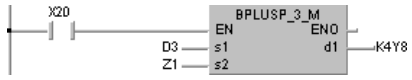
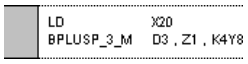


IEC-Anweisungsliste

```
LD      SM400
MOVP_M  16#7654 , D10
BMINUSP_M 16#4321 , D10
MOVP_M  D10 , K4Y30
```

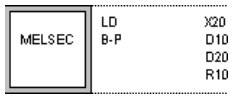
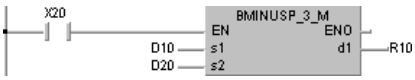
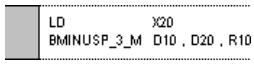
Beispiel 3 B+P (s1, s2, d1)

Das folgende Programm addiert mit positiver Flanke von X20 die BCD-Daten in D3 und die BCD-Daten in Z1. Das Ergebnis wird an Y8 bis Y17 ausgegeben.

MELSEC-Anweisungsliste	Kontaktplan	IEC-Anweisungsliste
		

Beispiel 4 B-P (s1, s2, d1)

Das folgende Programm subtrahiert mit positiver Flanke von X20 die BCD-Daten in D20 von den BCD-Daten in D10. Das Ergebnis wird in R10 gespeichert.

MELSEC-Anweisungsliste	Kontaktplan	IEC-Anweisungsliste
		

6.2.6 DB+, DB+P, DB-, DB-P

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)								Konstante		Pointer		Ebene				
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
s	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●[2]	●	●				K1 ↓ K8	9 ●[1]	●		●
d		●	●	●	●	●	●	●	●	●	●	●	●		●	●[2]										
s1	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●[2]	●	●					11 ●[1]			
s2	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●[2]	●	●								
d1		●	●	●	●	●	●	●	●	●	●	●	●		●	●[2]										

[1] Die Anzahl der Schritte bei Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

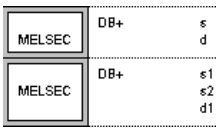
[2] Nur AnA, AnAS und AnU CPUs

Operanden MELSEC Q

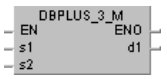
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sonder-module U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	●	●	●	●	●	●	●	●	—	SM0	3
d	●	●	●	●	●	●	●	—	—		
s1	●	●	●	●	●	●	●	●	—	SM0	4
s2	●	●	●	●	●	●	●	●	—		
d1	●	●	●	●	●	●	●	—	—		

GX IEC Developer

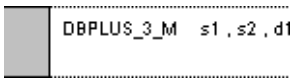
MELSEC-Anweisungsliste



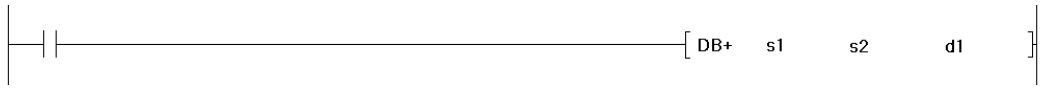
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

Operand	Befehlswert	Datentyp
s	Summanden- oder Subtrahendendaten oder erste Adresse, ab der die Summanden- oder Subtrahendendaten gespeichert sind.	BCD 8-stellig
d	Summanden- oder Minuendendaten oder erste Adresse, ab der die Summe oder Differenz gespeichert ist.	
s1	Summanden- oder Minuendendaten oder erste Adresse, ab der die Summanden- oder Minuendendaten gespeichert sind.	
s2	Summanden- oder Subtrahendendaten oder erste Adresse, ab der die Summanden- oder Subtrahendendaten gespeichert sind.	
d1	Erste Adresse, ab der die Summe oder Differenz gespeichert ist.	

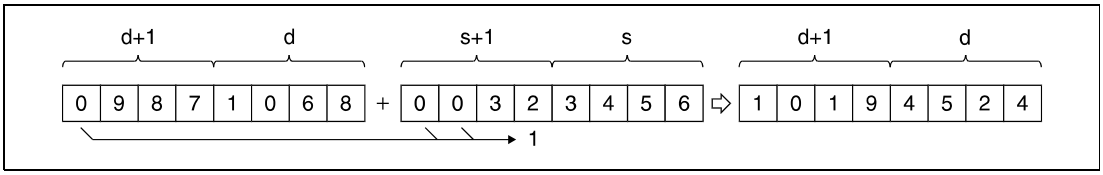
Funktionsweise

Addition und Subtraktion von BCD-Daten (8-stellig)

DB+ BCD-Addition (8-stellig)

- 1.Variante:

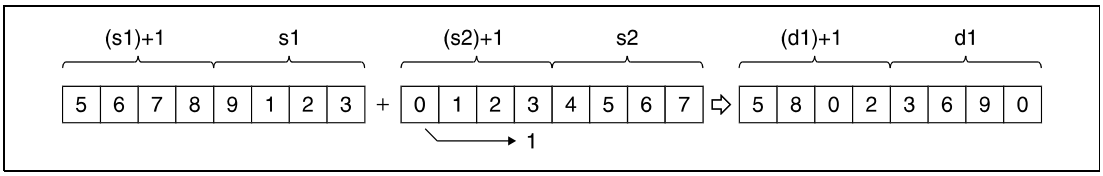
Die in s angegebenen BCD-Daten werden zu den BCD-Daten in d addiert. Das Additionsergebnis wird in d gespeichert.



[1] Leerstellen werden mit einer 0 aufgefüllt

- 2.Variante:

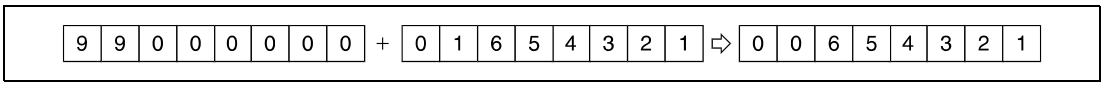
Die in s1 angegebenen BCD-Daten werden zu den BCD-Daten in s2 addiert. Das Additionsergebnis wird an den in d1 angegebenen Operanden ausgegeben.



[1] Leerstellen werden mit einer 0 aufgefüllt

Der in s, d, s1, s2 und d1 angegebene Datenwert muss eine 8-stellige Dezimalzahl zwischen 0 und 99999999 sein. Bei Zahlen mit weniger als 8 Stellen werden die ersten Stellen mit einer 0 aufgefüllt (z.B. 12345 = 00012345).

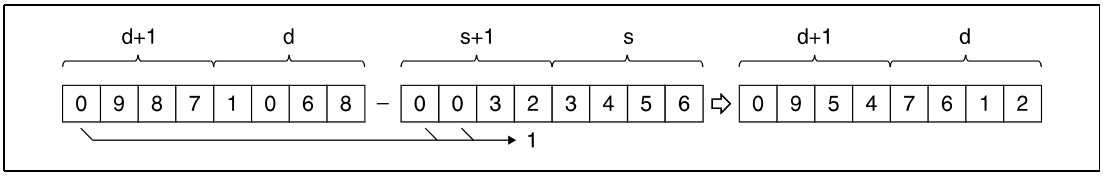
Das Carry Flag wird auch dann nicht gesetzt, wenn das Additionsergebnis den Wert 99999999 übersteigt. Die Übertragstelle wird ignoriert.



DB- BCD-Subtraktion (8-stellig)

- 1. Variante:

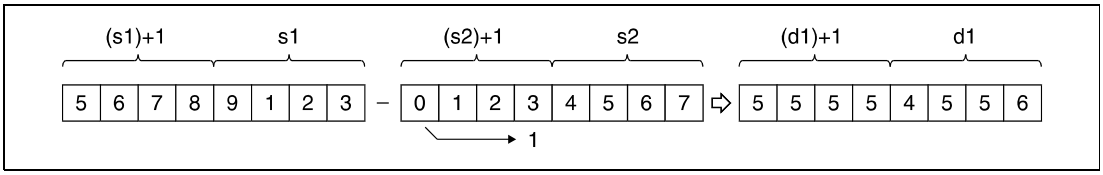
Die in s angegebenen BCD-Daten werden von den BCD-Daten in d subtrahiert. Das Subtraktionsergebnis wird in d gespeichert.



[1] Leerstellen werden mit einer 0 aufgefüllt

- 2. Variante:

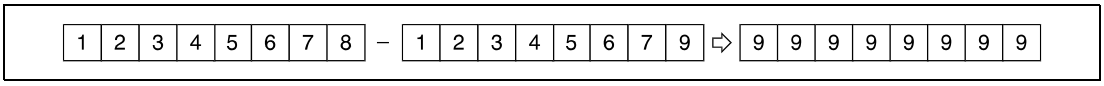
Die in s2 angegebenen BCD-Daten werden von den BCD-Daten in s1 subtrahiert. Das Ergebnis wird an den in d1 angegebenen Operanden ausgegeben.



[1] Leerstellen werden mit einer 0 aufgefüllt

Der in s, d, s1, s2 und d1 angegebene Datenwert muss eine 8-stellige Dezimalzahl zwischen 0 und 99999999 sein.

Wird das Subtraktionsergebnis negativ, so wird der Minuend um die Anzahl der im Subtrahenden angegebenen Schritte reduziert, und das Carry Flag wird nicht gesetzt.



Das Programm muss so ausgelegt werden, dass der weitere Verlauf geregelt ist, wenn das Ergebnis entweder positiv oder negativ ist.

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- In s, d, s1, s2 oder d1 wurde ein von 0 bis 99999999 verschiedener Datenwert eingetragen (Q-Serie/System Q = Fehlercode 4100).

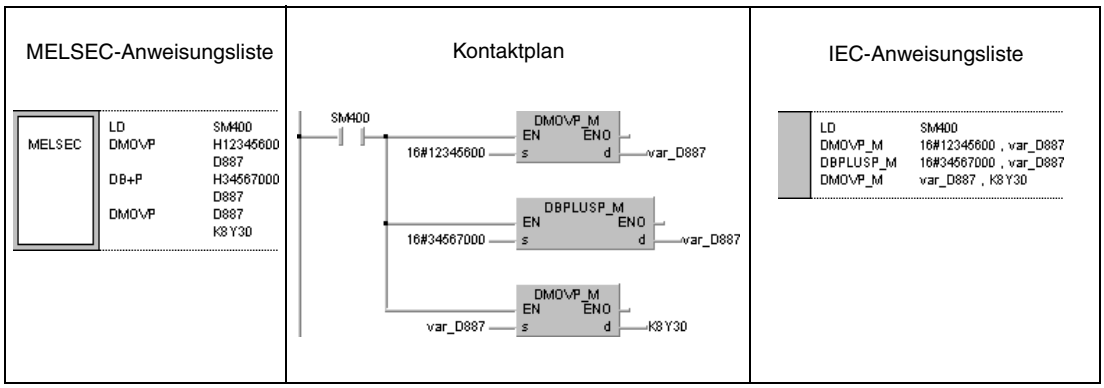
Beispiel 1 DB+P (s, d)

Das folgende Programm addiert die BCD-Daten 12345600 und 34567000, speichert das Ergebnis in D887 und D888 und gibt es an die Ausgänge Y30 bis Y4F aus.

In der ersten Programmzeile wird mit positiver Flanke von SM400 der Wert 12345600 in D887 und D888 gespeichert.

Im nachfolgenden Programmschritt wird der BCD-Wert 34567000 zu dem BCD-Wert in D887 und D888 hinzuaddiert.

Die DMOVP-Anweisung im letzten Programmschritt gibt das Ergebnis aus D887 und D888 an Y30 bis Y4F aus.



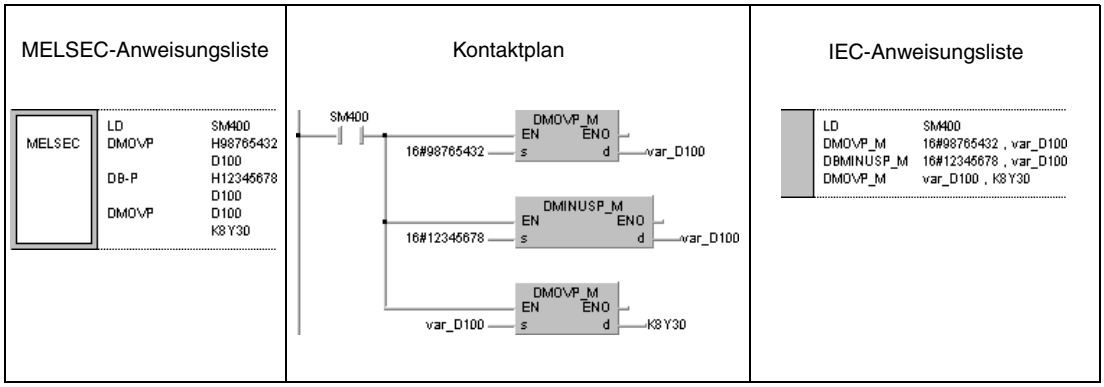
Beispiel 2 DB-P (s, d)

Das folgende Programm subtrahiert die BCD-Daten 12345678 von 98765432, speichert das Ergebnis in D100 und D101 und gibt es an die Ausgänge Y30 bis Y4F aus.

In der ersten Programmzeile wird hierzu mit positiver Flanke von SM400 der Wert 98765432 in D100 und D101 gespeichert.

Im nachfolgenden Programmschritt wird der BCD-Wert 12345678 von dem BCD-Wert in D100 und D101 subtrahiert.

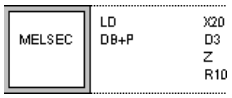
Die DMOVP-Anweisung im letzten Programmschritt gibt das Ergebnis aus D100 und D101 an Y30 bis Y4F aus.



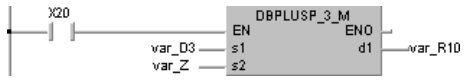
Beispiel 3 DB+P (s1, s2, d1)

Das folgende Programm addiert mit positiver Flanke von X20 die BCD-Daten in D3 und D4 zu den BCD-Daten in Z und V. Das Ergebnis wird in R10 und R11 gespeichert.

MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste

```
LD              X20
DBPLUSP_3_M var_D3 , var_Z  , var_R10
```

HINWEIS *Diese Programmbeispiele sind ohne Variablendefinition im Header der Programm-Organisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

6.2.7 Bx, BxP, B/, B/P

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)								Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
s1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				K1 ↓ K4	9	●		●
s2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				K1 ↓ K4				
d1		●	●	●	●	●	●	●	●	●	●	●	●		●							K1 ↓ K8	●[1]			

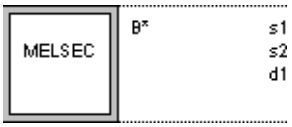
[1] Die Anzahl der Schritte bei Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□N□		Sonder-module U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	●	●	●	●	●	●	●	●	—	SM0	4
s2	●	●	●	●	●	●	●	●	—		
d1	●	●	●	●	●	●	●	—	—		

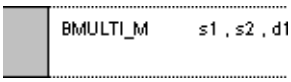
GX IEC Developer

MELSEC-Anweisungsliste

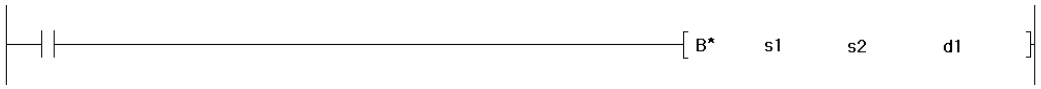


Kontaktplan

IEC-Anweisungsliste



GX Developer



Variablen

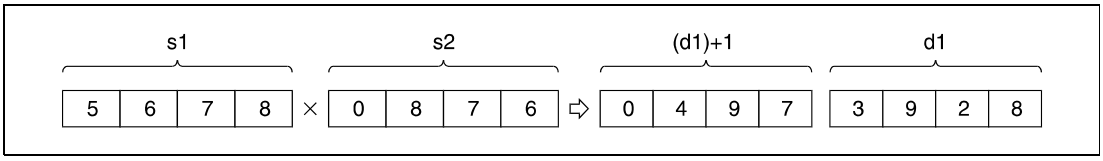
Operand	Befehlswert	Datentyp	
		MELSEC	IEC
s1	Multiplikatoren- oder Dividendendaten oder erste Adresse, ab der die Multiplikatoren- oder Dividendendaten gespeichert sind.	BCD 4-stellig	WORD
s2	Multiplikatoren- oder Divisorendaten oder erste Adresse, ab der die Multiplikatoren- oder Divisorendaten gespeichert sind.	BCD 4-stellig	WORD
d1	Erste Adresse, ab der das Produkt oder der Quotient gespeichert ist.	BCD 8-stellig	2 Arrays of WORD

Funktionsweise

Multiplikation und Division von BCD-Daten (4-Stellig)

Bx BCD-Multiplikation (4-stellig)

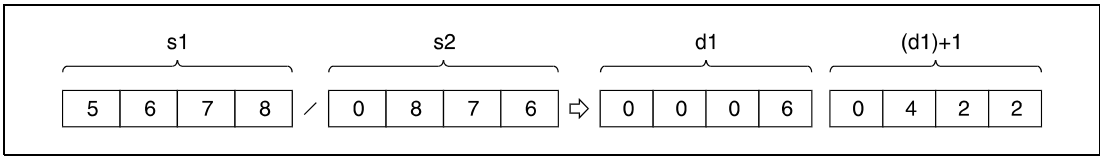
Die in s1 angegebenen BCD-Daten werden mit den BCD-Daten in s2 multipliziert, und das Multiplikationsergebnis wird in d1 gespeichert.



Der in s1 und s2 angegebene Datenwert muss eine 4-stellige Dezimalzahl zwischen 0 und 9999 sein.

B/ BCD-Division (4-stellig)

Die in s1 angegebenen BCD-Daten werden durch die BCD-Daten in s2 dividiert, und das Divisionsergebnis wird in d1 gespeichert.



Das Divisionsergebnis befindet sich in zwei WORD-Arrays (BCD 8-stellig), aufgeteilt in Quotient und Restbetrag, abgelegt.

Der Quotient (4-stellig BCD) wird in den niedrigstwertigen Array-Elementen gespeichert. Der Restwert (4-stellig BCD) wird in den höchstwertigen Array-Elementen gespeichert. Bei Bit-Operanden wird der Restwert des Divisionsergebnisses nicht gespeichert.

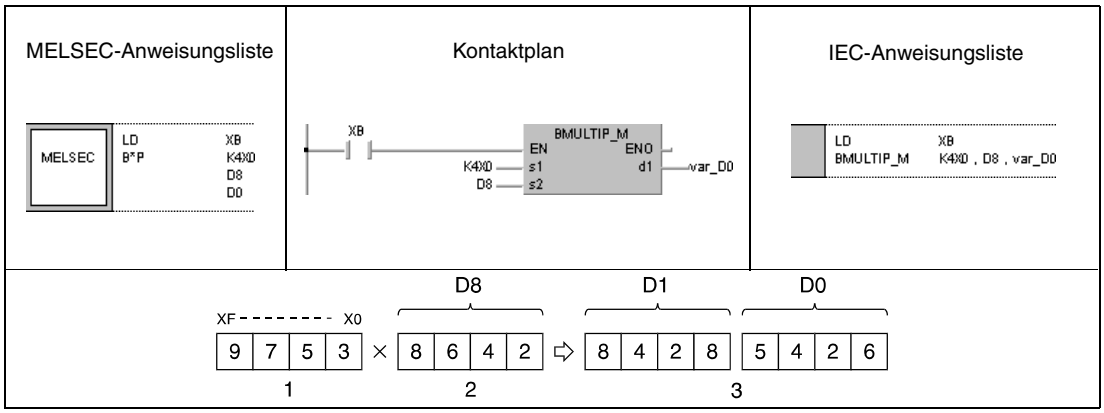
Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Als Datenwert in s1 oder s2 wurde ein von 0 bis 9999 verschiedener Datenwert gesetzt.
- Der Divisor s2 ist gleich 0 (Q-Serie/System Q = Fehlercode 4100).

Beispiel 1 BxP

Das folgende Programm multipliziert mit positiver Flanke von XB die BCD-Daten aus X0 bis XF mit den BCD-Daten aus D8. Das Ergebnis wird in D0 und D1 gespeichert.



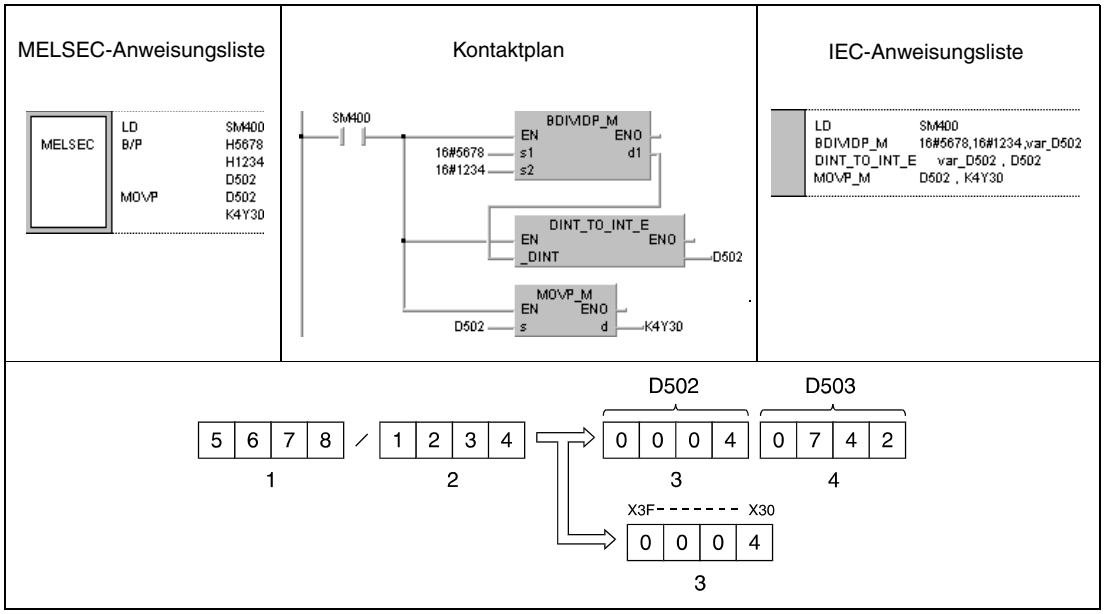
[1] Multiplikand

[2] Multiplikator

[3] Multiplikationsergebnis

Beispiel 2 B/P

Das folgende Beispiel führt mit positiver Flanke von SM400 eine Division der BCD-Daten 5678 und 1234 durch. Das Ergebnis wird in D502 und der Restwert in D503 gespeichert. Im letzten Schritt wird der Quotient (D502) an Y30 bis Y3F ausgegeben.



[1] Dividend

[2] Divisor

[3] Quotient

[4] Rest

HINWEIS Diese Programmbeispiele sind ohne Variablendefinition im Header der Programm-Organisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.

6.2.8 DBx, DBxP, DB/, DB/P

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

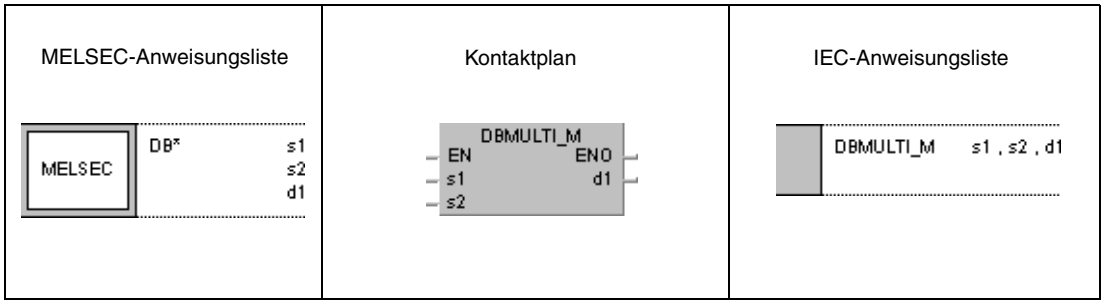
	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
s1	●	●	●	●	●	●	●	●	●	●	●	●	●		●		●	●				K1 ↓ K8	11 ●[1]	●		●
s2	●	●	●	●	●	●	●	●	●	●	●	●	●		●		●	●								
d1		●	●	●	●	●	●	●	●	●	●	●	●													

[1] Die Anzahl der Schritte bei Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

Operanden MELSEC Q

	Operanden								Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sonder-module U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere	
	Bit	Wort		Bit	Wort					
s1	●	●	●	●	●	●	●	●	—	SM0	4
s2	●	●	●	●	●	●	●	●	—		
d1	●	●	●	—	—	—	—	—	—		

GX IEC Developer



GX Developer

Variablen

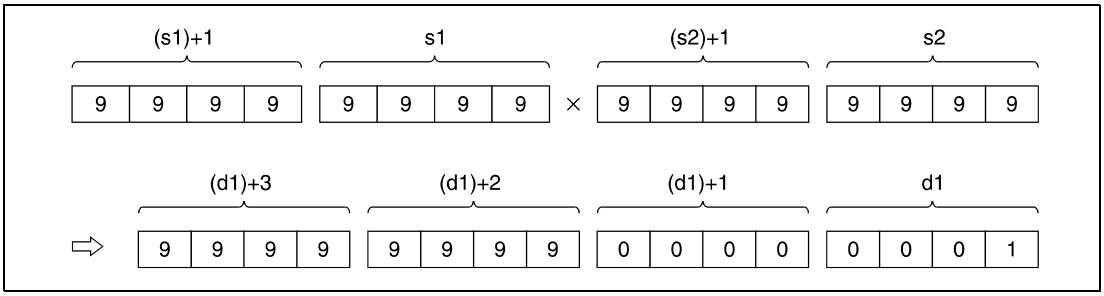
Operand	Befehlswert	Datentyp
s1	Multiplikatoren- oder Dividendendaten oder erste Adresse, ab der die Multiplikatoren- oder Dividendendaten gespeichert sind.	BCD 8-stellig
s2	Multiplikatoren- oder Divisorendaten oder erste Adresse, ab der die Multiplikatoren- oder Divisorendaten gespeichert sind.	BCD 8-stellig
d1	Erste Adresse, ab der das Produkt oder der Quotient gespeichert ist.	BCD 16-stellig

Funktions-weise

Multiplikation und Division von BCD-Daten (8-stellig)

DBx BCD-Multiplikation (8-stellig)

Die in s1 angegebenen BCD-Daten werden mit den BCD-Daten in s2 multipliziert, und das Multiplikationsergebnis wird in d1 gespeichert.



Handelt es sich bei d1 um einen Bit-Operanden, können nur die letzten 8 Stellen (niedrigstwertigen 32 Bit) verarbeitet werden.

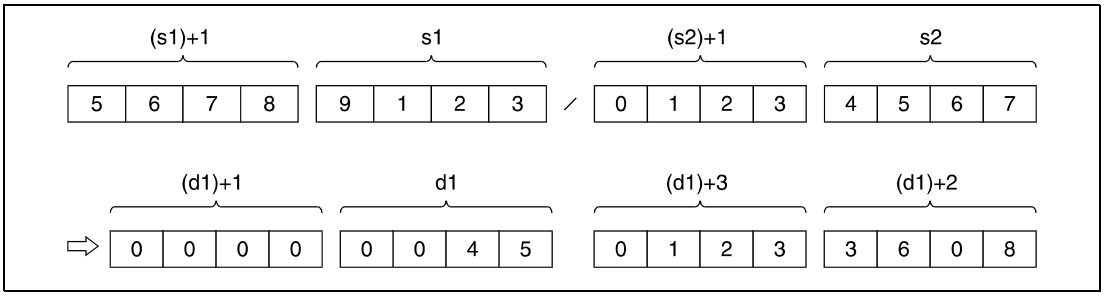
Beispiel:

K1: niedrigsten 4 Bit (b0 bis b3)
K4: niedrigsten 16 Bit (b0 bis b15)
K8: 32 Bit (b0 bis b31)

Der in s1 und s2 angegebene Datenwert muss eine 8-stellige Dezimalzahl zwischen 0 und 99999999 sein. Bei Zahlen mit weniger als 8 Stellen werden die ersten Stellen mit einer 0 aufgefüllt (z.B.: 12345 = 00012345).

DB/ BCD-Division (8-stellig)

Die in s1 angegebenen BCD-Daten werden durch die BCD-Daten in s2 dividiert, und das Divisionsergebnis wird in d1 gespeichert.



Das Divisionsergebnis wird als Array von zwei 32-Bit-Daten-Wörtern, aufgeteilt in Quotient und Restbetrag, abgelegt.

Der Quotient (8-stellig BCD) wird in den niedrigstwertigen 32 Bit gespeichert. Der Restwert (8-stellig BCD) wird in den höchstwertigen 32 Bit gespeichert. Bei Bit-Operanden wird der Restwert des Divisionsergebnisses nicht gespeichert.

Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Als Datenwert in s1 oder s2 wurde ein von 0 bis 99999999 verschiedener Datenwert gesetzt.
- Der Divisor s2 ist gleich 0 (Q-Serie = Fehlercode 4100).

Beispiel 1

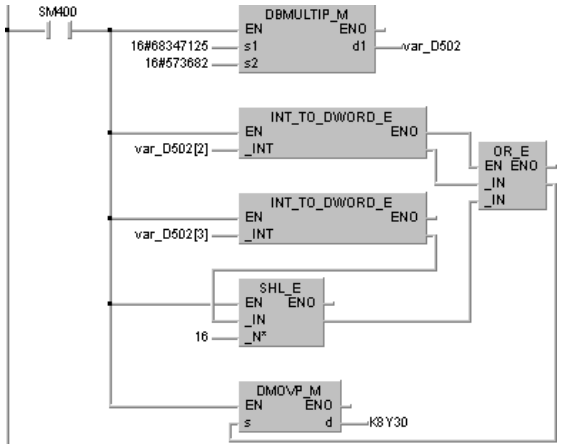
DBxP

Das folgende Programm multipliziert mit positiver Flanke von SM400 die BCD-Daten 68347125 und 573682 miteinander, und speichert das Ergebnis in D502 bis D505. Im folgenden Programmschritt werden die zweiten 8 Stellen des Ergebnisses (D504, D505) an Y30 bis Y4F ausgegeben.

MELSEC-Anweisungsliste

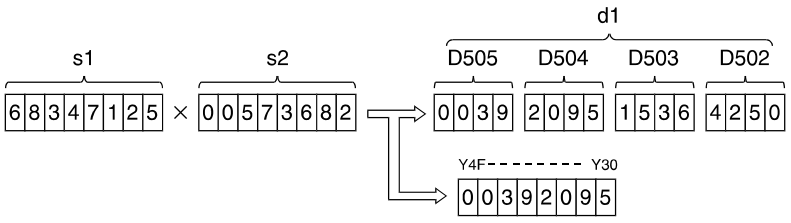
```
MELSEC  LD      SM400
        DB*P    H68347125
                H573682
                D502
        DMOV    D504
                K8Y30
```

Kontaktplan



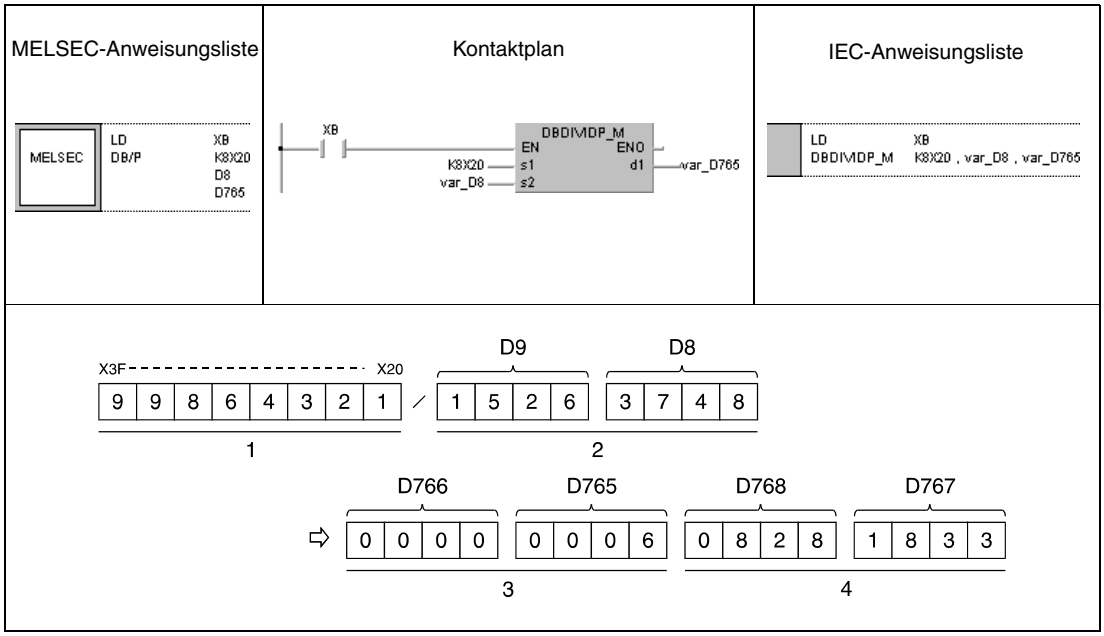
IEC-Anweisungsliste

```
LD              SM400
DBMULTIP_M      16#68347125,16#573682,var_D502
INT_TO_DWORD_E  var_D502[3] , var_temp
INT_TO_DWORD_E  var_D502[4] , var_D504
SHL_E           var_D504 , K16 , var_D504
LD              var_temp
OR              var_D504
ST              var_D504
LD              SM400
DMOV_M          var_D504 , K8Y30
```



Beispiel 2 DB/P

Das folgende Programm führt mit positiver Flanke von XB eine Division zwischen den BCD-Daten aus X20 bis X3F und dem Inhalt von D8 und D9 durch. Das Ergebnis wird in D765 bis D768 gespeichert.



[1] Dividend

[2] Divisor

[3] Quotient

[4] Rest

HINWEIS *Diese Programmbeispiele sind ohne Variablendefinition im Header der Programm-Organisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

6.2.9 E+, E+P, E-, E-P

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●[1]

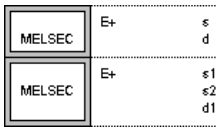
[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

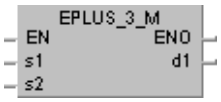
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□N□		Sondermodule U□NG□	Index-Register Zn	Konstanten E	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	●	●	—	●	—	SM0	3
d	—	●	●	—	●	●	—	—	—		
s1	—	●	●	—	●	●	—	●	—	SM0	4
s2	—	●	●	—	●	●	—	●	—		
d1	—	●	●	—	●	●	—	—	—		

GX IEC Developer

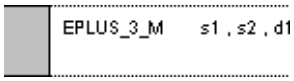
MELSEC-Anweisungsliste



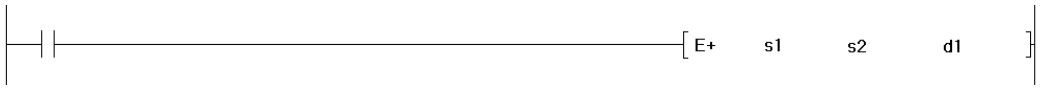
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

Operand	Befehlswert	Datentyp
s	Summanden- oder Subtrahendendaten oder erste Adresse, ab der die Summanden- oder Subtrahendendaten gespeichert sind.	reelle Zahl
d	Summanden- oder Minuendendaten oder erste Adresse, ab der die Summe oder Differenz gespeichert ist.	
s1	Summanden- oder Minuendendaten oder erste Adresse, ab der die Summanden- oder Minuendendaten gespeichert sind.	
s2	Summanden- oder Subtrahendendaten oder erste Adresse, ab der die Summanden- oder Subtrahendendaten gespeichert sind.	
d1	Erste Adresse, ab der die Summe oder Differenz gespeichert ist.	

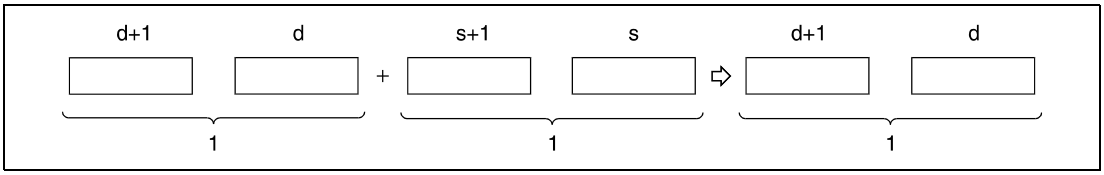
Funktionsweise

Addition und Subtraktion von Gleitkommazahlen

E+ Additionsanweisung für Gleitkommazahlen

- 1. Variante:

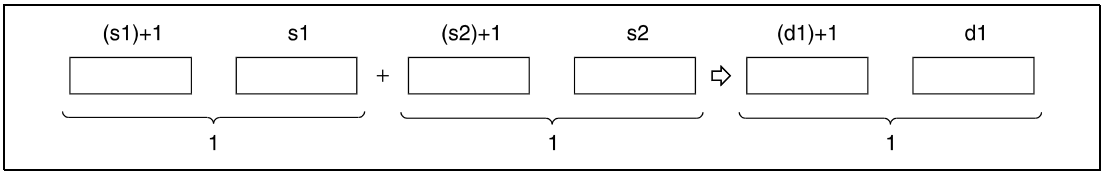
Die in s angegebene Gleitkommazahl wird zu der Gleitkommazahl in d addiert. Das Additionsergebnis wird in d gespeichert.



[1] Gleitkommazahl, Datentyp reelle Zahl

- 2. Variante:

Die in s1 angegebene Gleitkommazahl wird zu der Gleitkommazahl in s2 addiert. Das Additionsergebnis wird an den in d1 angegebenen Operanden ausgegeben.



[1] Gleitkommazahl, Datentyp reelle Zahl

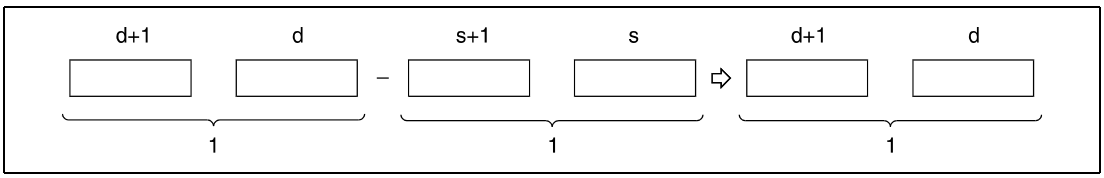
Der in s, d, s1, s2 und d1 angegebene bzw. zu speichernde Datenwert muss Null oder eine Gleitkommazahl in folgenden Grenzen sein:

$\pm 2^{-127} \leq$ Gleitkommazahl (s, d, s1, s2, d1) $< \pm 2^{129}$

E- Subtraktionsanweisung für Gleitkommazahlen

- 1. Variante:

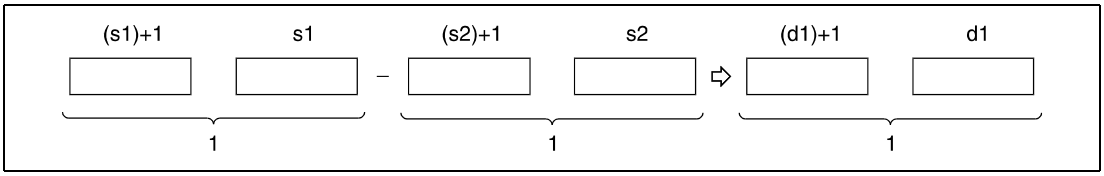
Die in s angegebene Gleitkommazahl wird von der Gleitkommazahl in d subtrahiert. Das Subtraktionsergebnis wird in d gespeichert.



[1] Gleitkommazahl, Datentyp reelle Zahl

- 2. Variante:

Die in s2 angegebene Gleitkommazahl wird von der Gleitkommazahl in s1 subtrahiert. Das Subtraktionsergebnis wird an den in d1 angegebenen Operanden ausgegeben.



[1] Gleitkommazahl, Datentyp reelle Zahl

Der in s, d, s1, s2 und d1 angegebene bzw. zu speichernde Datenwert muss Null oder eine Gleitkommazahl in folgenden Grenzen sein:

$\pm 2^{-127} \leq$ Gleitkommazahl (s, d, s1, s2, d1) $< \pm 2^{129}$

Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

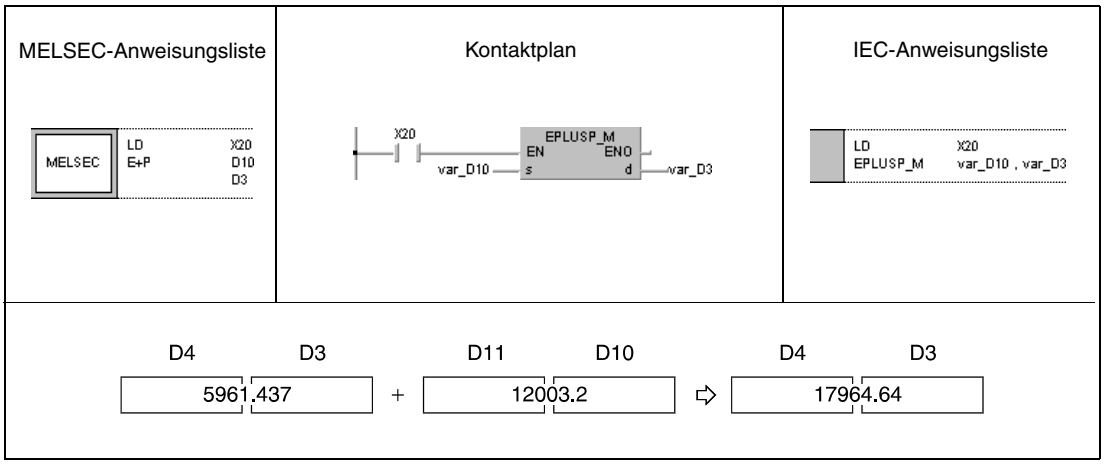
- Die in s, d, s1, s2 und d1 angegebenen Gleitkommazahlen oder die Additions- und Subtraktionsergebnisse sind von Null verschieden oder liegen außerhalb der folgenden Grenzen (Fehlercode 4100):

  $\pm 2^{-127} \leq$ Gleitkommazahl/Operationsergebnis (s, d, s1, s2, d1) $< \pm 2^{129}$

Beispiel 1

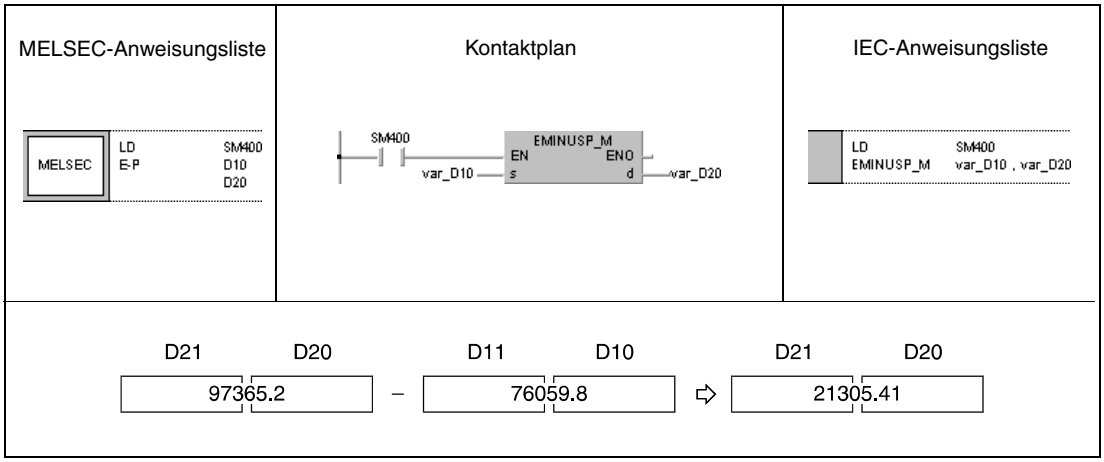
E+P (s, d)

Das folgende Programm addiert mit positiver Flanke von X20 die Gleitkommazahl in D3 und D4 mit der Gleitkommazahl in D10 und D11. Das Additionsergebnis wird in D3 und D4 gespeichert.



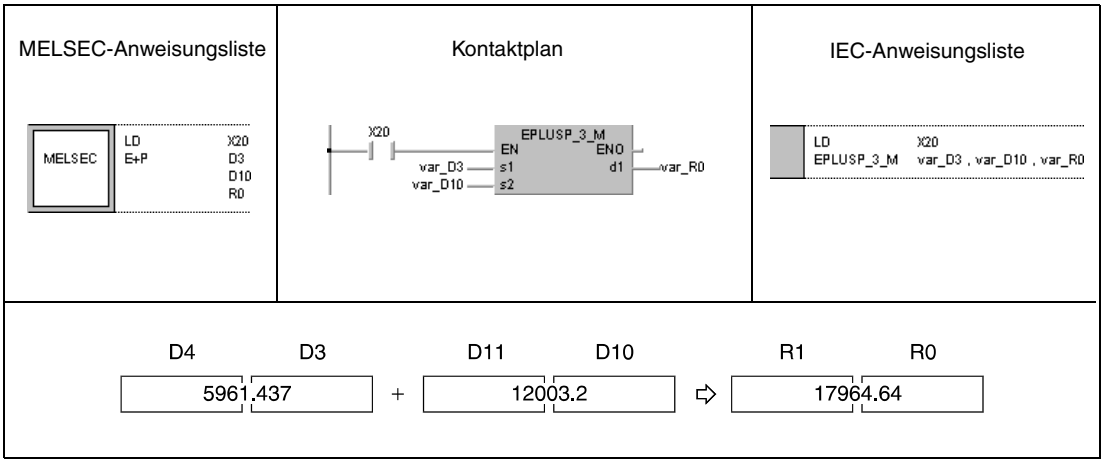
Beispiel 2 E-P (s, d)

Das folgende Programm subtrahiert mit positiver Flanke von SM400 die Gleitkommazahl in D10 und D11 von der Gleitkommazahl in D20 und D21. Das Subtraktionsergebnis wird in D20 und D21 gespeichert.



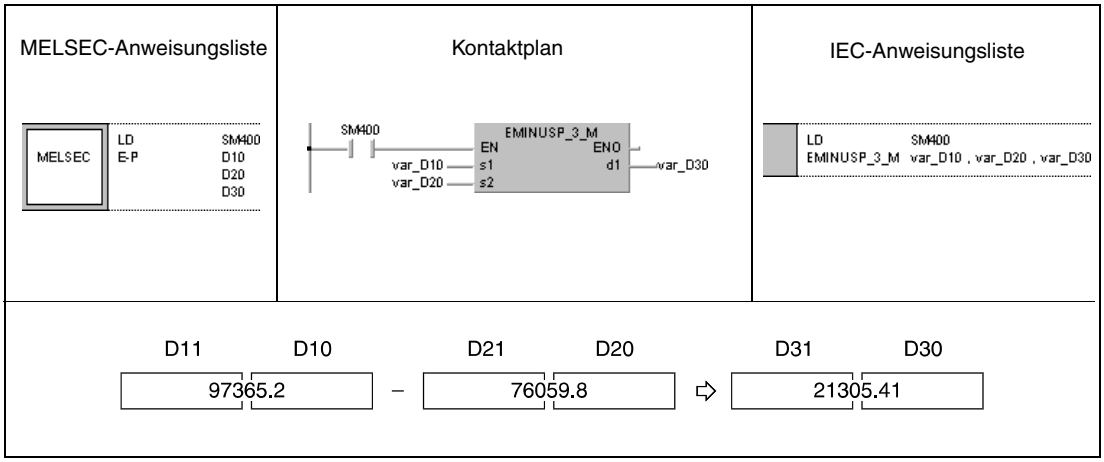
Beispiel 3 E+P (s1, s2, d)

Das folgende Programm addiert mit positiver Flanke von X20 die Gleitkommazahlen in D3 und D4 mit den Gleitkommazahlen in D10 und D11. Das Additionsergebnis wird in R0 und R1 gespeichert.



Beispiel 4 E-P (s1, s2, d)

Das folgende Programm subtrahiert mit positiver Flanke von SM400 die Gleitkommazahlen in D20 und D21 von den Gleitkommazahlen in D10 und D11. Das Ergebnis wird in D30 und D31 gespeichert.



HINWEIS *Diese Programmbeispiele sind ohne Variablendefinition im Header der Programm-Organisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

6.2.10 Ex, ExP, E/, E/P

CPU

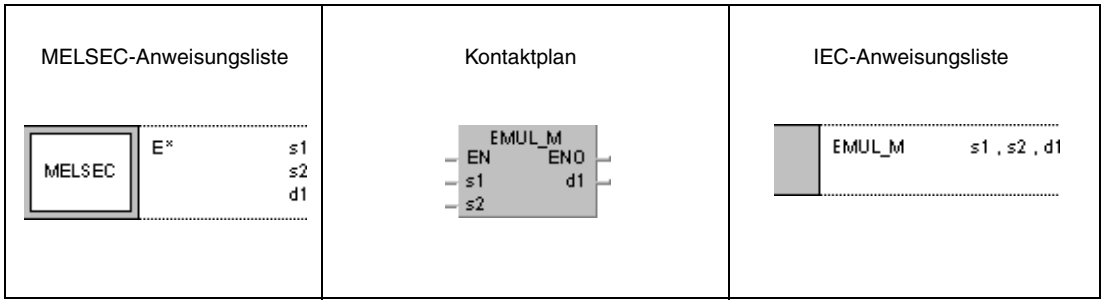
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●[1]

[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□N□		Sondermodule U□\G□	Index-Register Zn	Konstanten E	Andere		
	Bit	Wort		Bit	Wort				U		
s1	—	●	●	—	●	●	—	●	—	SM0	4
s2	—	●	●	—	●	●	—	●	—		
d1	—	●	●	—	●	●	—	—	—		

GX IEC Developer



GX Developer

Variablen

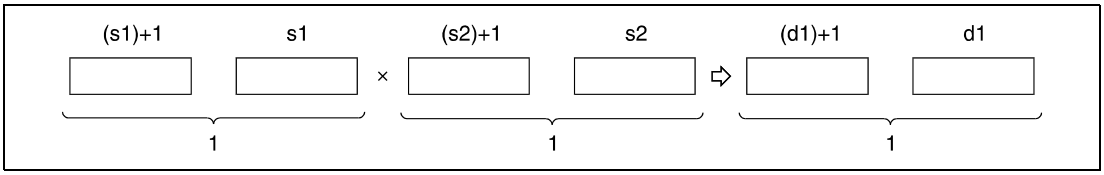
Operand	Befehlswert	Datentyp
s1	Multiplikatoren- oder Dividendendaten oder erste Adresse, ab der die Multiplikatoren- oder Dividendendaten gespeichert sind.	reelle Zahl
s2	Multiplikatoren- oder Divisorendaten oder erste Adresse, ab der die Multiplikatoren- oder Divisorendaten gespeichert sind.	
d1	Erste Adresse, ab der das Produkt oder der Quotient gespeichert ist.	

Funktionsweise

Multiplikation und Division von Gleitkommazahlen

Ex Multiplikationsanweisung für Gleitkommazahlen

Die in s1 angegebene Gleitkommazahl wird mit der Gleitkommazahl in s2 multipliziert, und das Multiplikationsergebnis wird in d1 gespeichert.



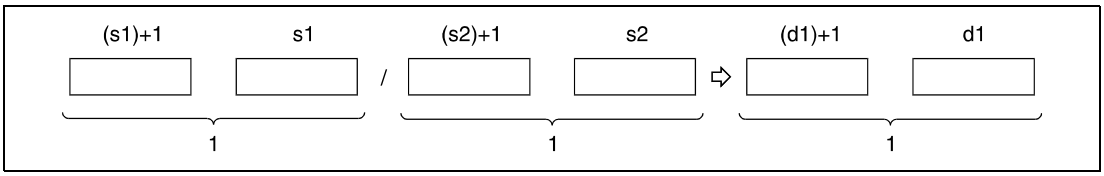
[1] Gleitkommazahl, Datentyp reelle Zahl

Der in s1, s2 und d1 angegebene bzw. zu speichernde Datenwert muss Null oder eine Gleitkommazahl in folgenden Grenzen sein:

$\pm 2^{-127} \leq$ Gleitkommazahl (s1, s2, d1) $< \pm 2^{129}$

E/ Divisionsanweisung für Gleitkommazahlen

Die in s1 angegebenen Gleitkommazahl wird durch die Gleitkommazahl in s2 dividiert, und das Divisionsergebnis wird in d1 gespeichert.



[1] Gleitkommazahl, Datentyp reelle Zahl

Der in s1, s2 und d1 angegebene bzw. zu speichernde Datenwert muss Null oder eine Gleitkommazahl in folgenden Grenzen sein:

$\pm 2^{-127} \leq$ Gleitkommazahl (s1, s2, d1) $< \pm 2^{129}$

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

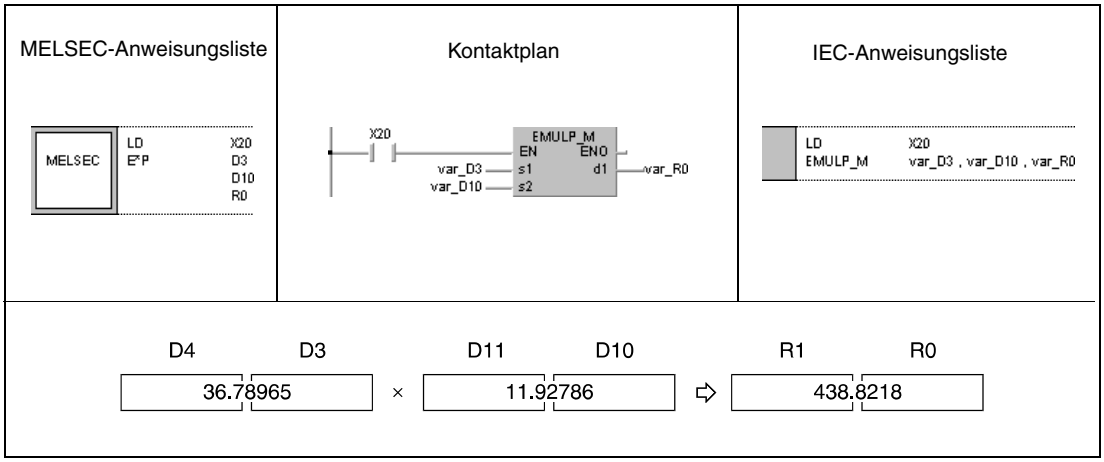
- Die in s1, s2 und d1 angegebenen Gleitkommazahlen oder die Multiplikations- und Divisionsergebnisse sind von Null verschieden oder liegen außerhalb der folgenden Grenzen (Fehlercode 4100):

  $\pm 2^{-127} \leq$ Gleitkommazahl/Operationsergebnis (s1, s2, d1) $< \pm 2^{129}$

- Der Divisor s2 ist gleich 0 (Fehlercode 4100).

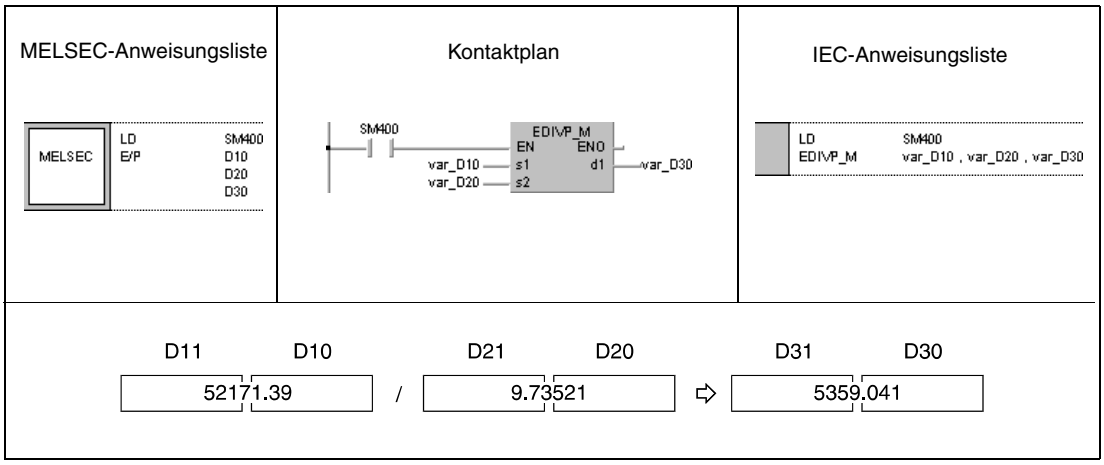
Beispiel 1 ExP

Das folgende Programm multipliziert mit positiver Flanke von X20 die Gleitkommazahl in D3 und D4 mit der Gleitkommazahl in D10 und D11. Das Multiplikationsergebnis wird in R0 und R1 gespeichert.



Beispiel 2 E/P

Das folgende Programm dividiert mit positiver Flanke von SM400 die Gleitkommazahl in D10 und D11 durch die Gleitkommazahl in D20 und D21. Das Divisionsergebnis wird in D30 und D31 gespeichert.



HINWEIS *Diese Programmbeispiele sind ohne Variablendefinition im Header der Programm-Organisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

6.2.11 BK+, BK+P, BK-, BK-P

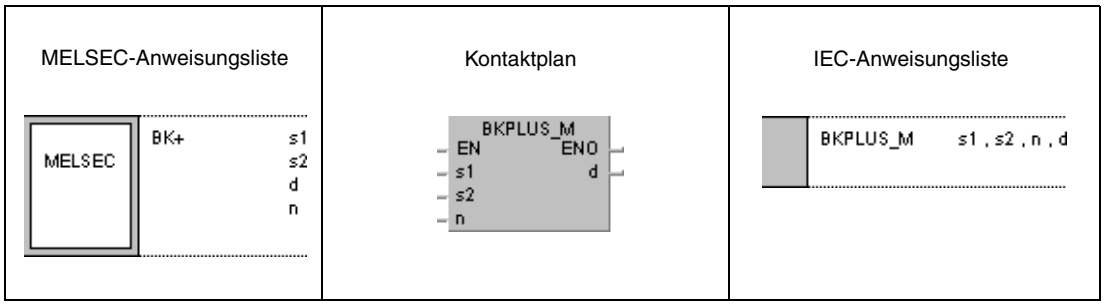
CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

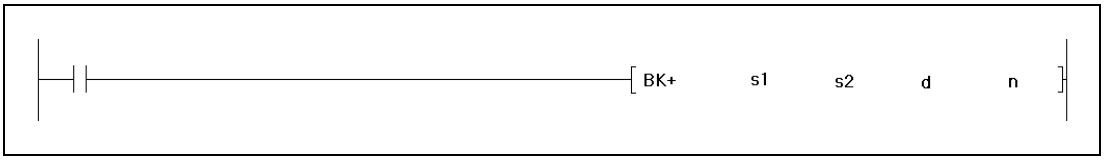
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	—	●	●	—	—	—	—	—	—	SM0	5
s2	—	●	●	—	—	—	—	●	—		
d	—	●	●	—	—	—	—	—	—		
n	●	●	●	●	●	●	●	●	—		

GX IEC Developer



GX Developer



Variablen

Operand	Befehlswert	Datentyp
s1	Additions- und Subtraktionsdaten oder erste Adresse, ab der die Additions- und Subtraktionsdaten gespeichert sind.	BIN-16-Bit
s2	Erste Adresse, ab der die Additions- und Subtraktionsdaten gespeichert sind.	
d	Erste Adresse, ab der die Ergebnisdaten des Vergleichs gespeichert werden sollen.	
n	Anzahl der Datenblöcke, mit denen die Operation durchgeführt wird.	

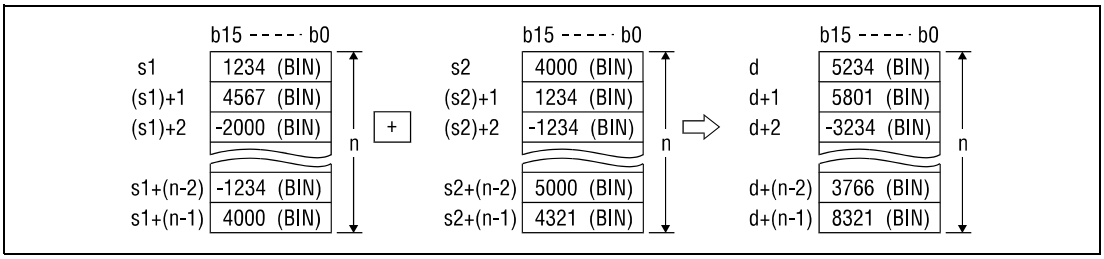
Funktions-weise

Blockweise Addition und Subtraktion von Binärdaten

BK+ Additionsanweisung für Binärdatenblöcke

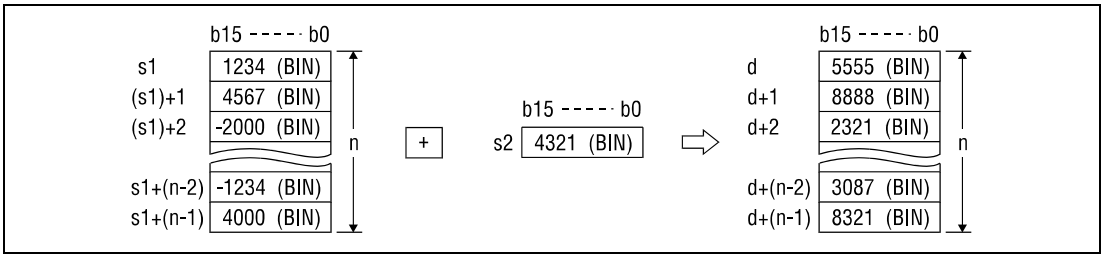
Eine Additionsanweisung für Binärdatenblöcke besteht aus der Anweisung selbst, den Daten s1 und s2, die addiert werden sollen, der Zielbezeichnung d, in dem die Ergebnisse abgelegt werden, und der Anzahl n, der zu addierenden Datenblöcke.

Addiert wird jeweils der n-te 16-Bit-Block von s1 und der n-te 16-Bit-Block von s2, beginnend bei dem niedrigstwertigen 16-Bit-Block. Das Ergebnis einer jeden Blockaddition wird in d gespeichert.



Die Additionsoperationen werden in Einheiten zu 16 Bit durchgeführt.

Eine in s2 abgelegte Konstante muss eine 16-Bit-Binärzahl zwischen -32768 und 32767 sein.



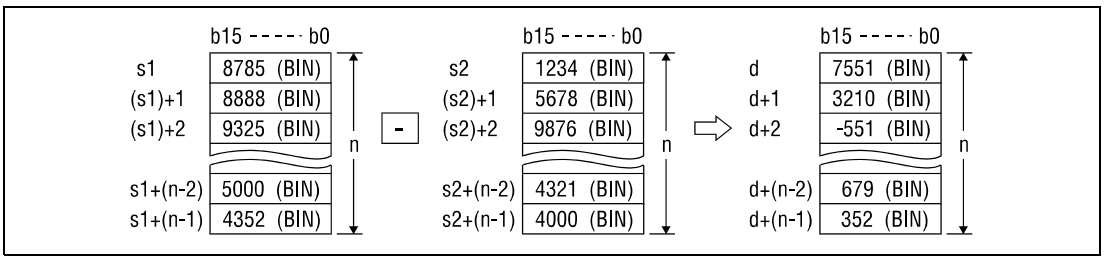
Das jeweils höchstwertige Bit des Blockes legt fest, ob die Datenwerte in dem entsprechenden Block von s1, s2 oder d positiv (Bit = 0) oder negativ (Bit = 1) sind.

Wenn das niedrigstwertige Bit des Blockes unterschritten oder das höchstwertige Bit des Blokkes überschritten wird, wird das Carry Flag nicht gesetzt.

BK- Subtraktionsanweisung für Binärdatenblöcke

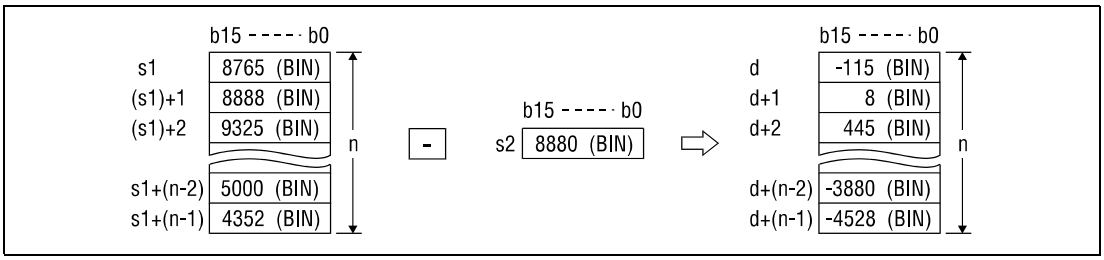
Eine Subtraktionsanweisung für Binärdatenblöcke besteht aus der Anweisung selbst, den Daten s1 und s2, die subtrahiert werden sollen, der Zielbezeichnung d, in dem die Ergebnisse abgelegt werden, und der Anzahl n, der zu subtrahierenden Datenblöcke.

Subtrahiert wird jeweils der n-te 16-Bit-Block von s2 von dem n-ten 16-Bit-Block von s1, beginnend bei dem niedrigstwertigen 16-Bit-Block. Das Ergebnis einer jeden Blocksubtraktion wird in d gespeichert.



Die Subtraktionsoperationen werden in Einheiten zu 16 Bit durchgeführt.

Eine in s2 abgelegte Konstante muss eine 16-Bit-Binärzahl zwischen -32768 und 32767 sein.



Das jeweils höchstwertige Bit des Blockes legt fest, ob die Datenwerte in dem entsprechenden Block von s1, s2 oder d positiv (Bit = 0) oder negativ (Bit = 1) sind.

Wenn das niedrigstwertige Bit des Blockes unterschritten oder das höchstwertige Bit des Blokkes überschritten wird, wird das Carry Flag nicht gesetzt.

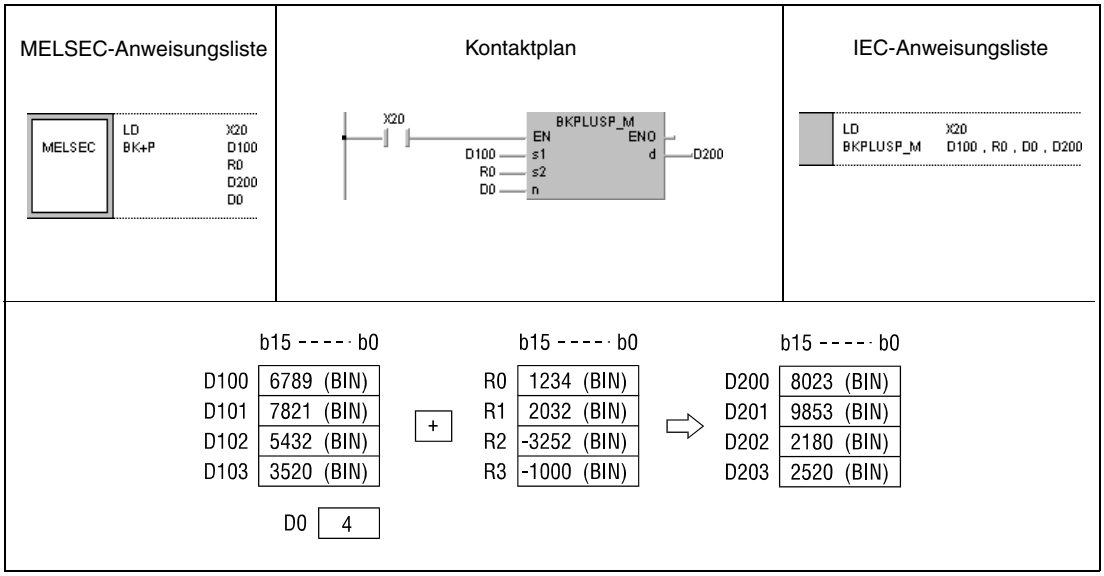
Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die Anzahl der Datenblöcke in s1, s2 oder d übersteigt die zulässige Anzahl.
- Der für die Speicherung vorgesehene Datenbereich von s1 überlappt mit denen von s2 oder d.

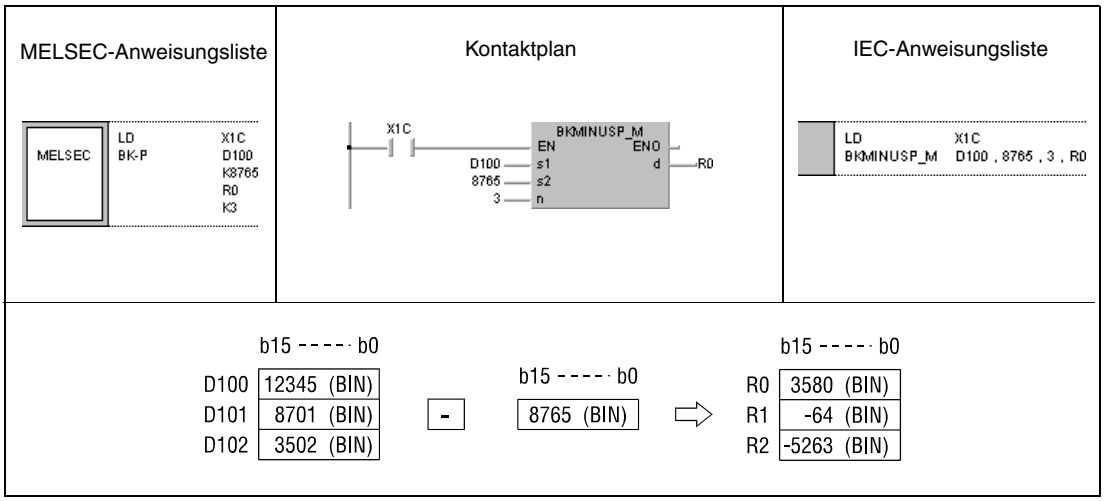
Beispiel 1 BK+P

Das folgende Programm addiert mit positiver Flanke von X20 die Datenblöcke beginnend bei D100 mit den Datenblöcken beginnend bei R0 und speichert die Blockergebnisse beginnend bei D200. Die Anzahl der zu addierenden Blöcke (4) ist in D0 hinterlegt.



Beispiel 2 BK-P

Das folgende Programm subtrahiert mit positiver Flanke von X1C die Konstante 8765 von den Datenblöcken beginnend bei D100 und speichert die Blockergebnisse beginnend bei R0. Die Anzahl der Datenblöcke (3) gibt die Konstante K3 an.



6.2.12 $+, $+P

CPU

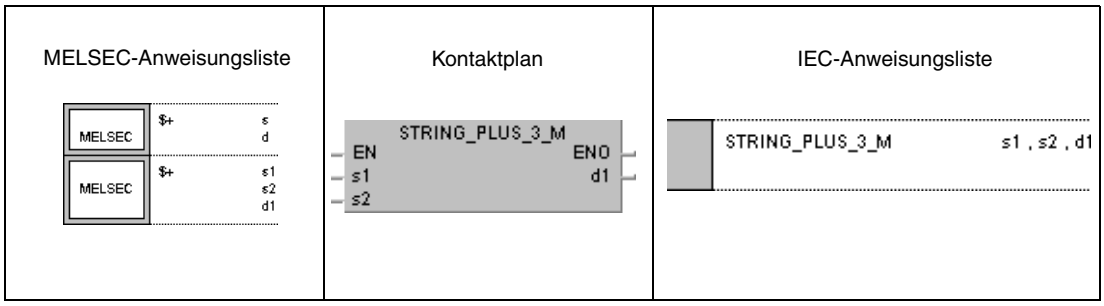
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●[1]

[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□N□		Sondermodule U□NG□	Index-Register Zn	Konstanten $	Andere		
	Bit	Wort		Bit	Wort				U		
s	—	●	●	—	—	—	—	●	—	SM0	3
d	—	●	●	—	—	—	—	—	—		
s1	—	●	●	—	—	—	—	●	—	SM0	4
s2	—	●	●	—	—	—	—	●	—		
d1	—	●	●	—	—	—	—	—	—		

GX IEC Developer



GX Developer

Variablen

Operand	Befehlswert	Datentyp
s	Daten oder erste Adresse, ab der die zu verbindenden Daten gespeichert sind.	Zeichenfolge
d	Erste Adresse, ab der die verbundenen Daten gespeichert sind.	
s1	Daten oder erste Adresse, ab der die zu verbindenden Daten gespeichert sind.	
s2	Erste Adresse, ab der die verbundenen Daten gespeichert sind.	
d1	Erste Adresse, ab der das Ergebnis der Datenverbindung gespeichert sind.	

Funktionsweise

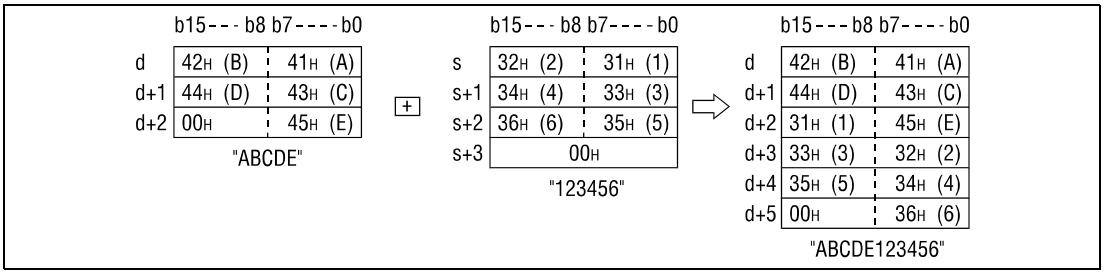
Verknüpfung von Zeichenfolgen

$+ Verknüpfungsanweisung für Zeichenfolgen

- 1. Variante:

Die in s angegebene Zeichenfolge wird an die in d angegebene Zeichenfolge angehängt. Die verknüpfte Zeichenfolge wird in d gespeichert.

Diese verknüpfte Zeichenfolge beginnt mit dem Zeichen des niedrigstwertigen Bytes der vor der Operation in d angegebenen Zeichenfolge und endet mit dem Code "00H" der in s angegebenen Zeichenfolge.

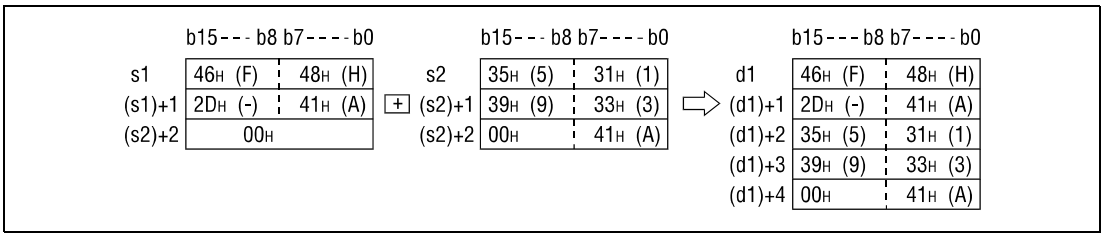


Bei der Verknüpfung wird der Code "00H", der das Ende der Zeichenfolge in d definiert, ignoriert. Die in s angegebene Zeichenfolge wird an das letzte Zeichen der in d angegebenen Zeichenfolge angehängt.

- 2. Variante:

Die in s1 angegebene Zeichenfolge wird mit der in s2 angegeben Zeichenfolge verknüpft. Die verknüpfte Zeichenfolge wird in d1 gespeichert.

Diese verknüpfte Zeichenfolge beginnt mit dem Zeichen des niedrigstwertigen Bytes der in s1 abgelegten Zeichenfolge und endet mit dem Code "00H" der Zeichenfolge in s2.



Bei der Verknüpfung wird der Code "00H", der das Ende der Zeichenfolge in s1 definiert, ignoriert. Die in s2 angegebene Zeichenfolge wird an das letzte Zeichen der in s1 angegebenen Zeichenfolge angehängt.

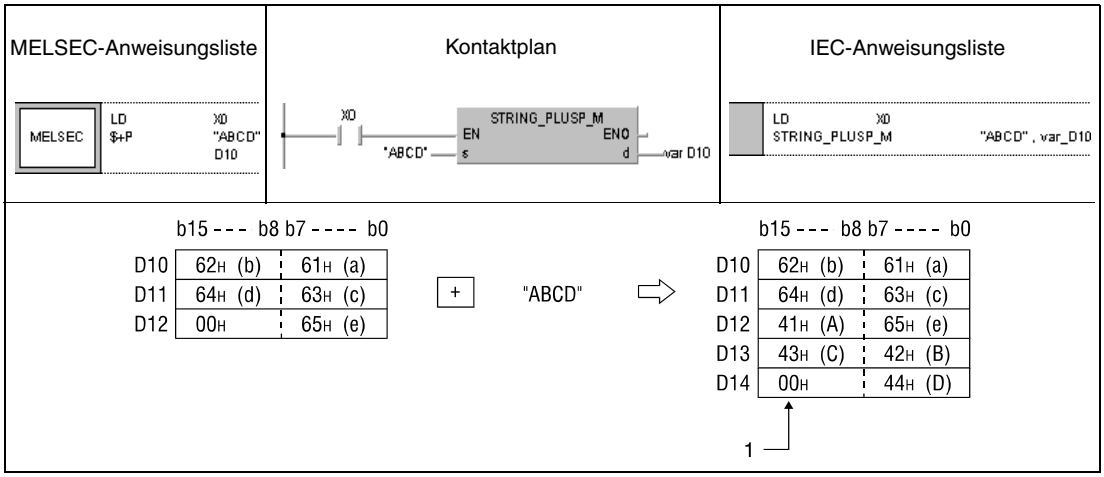
Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die vollständige, verknüpfte Zeichenfolge kann nicht gespeichert werden (Fehlercode 4100).
- Die für die Speicherung vorgesehenen Datenbereiche von s, s1 oder s2 überlappen mit denen von d oder d1 (Fehlercode 4101).

Beispiel 1 S+P

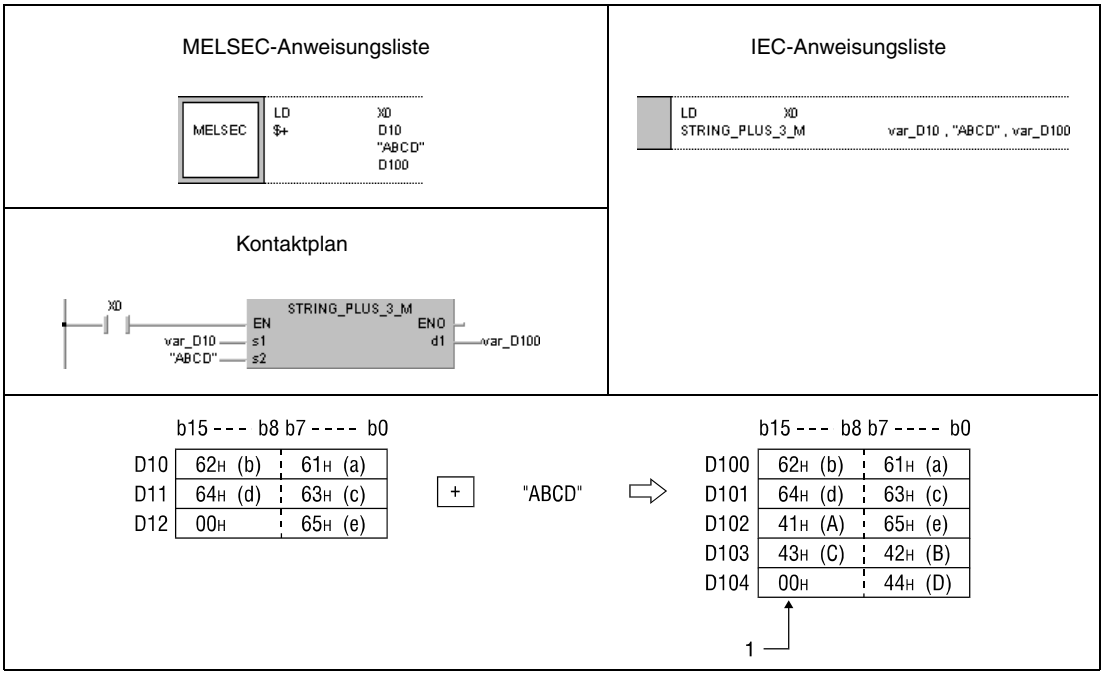
Das folgende Programm verknüpft mit positiver Flanke von X0 die in D10 - D12 gespeicherte Zeichenfolge mit der Zeichenfolge "ABCD". Die verknüpfte Zeichenfolge wird in D10 - D14 abgespeichert.



[1] Zur Kennzeichnung des Endes der Zeichenfolge wird dieses Byte automatisch mit "00H" beschrieben.

Beispiel 2 S+

Das folgende Programm verknüpft für die Einschaltdauer von X0 die in D10 - D12 gespeicherte Zeichenfolge mit der Zeichenfolge "ABCD". Die verknüpfte Zeichenfolge wird in D101 - D104 abgespeichert.



[1] Zur Kennzeichnung des Endes der Zeichenfolge wird dieses Byte automatisch mit "00H" beschrieben.

HINWEIS Diese Programmbeispiele sind ohne Variablendefinition im Header der Programm-Organisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.

6.2.13 INC, INCP, DEC, DECP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

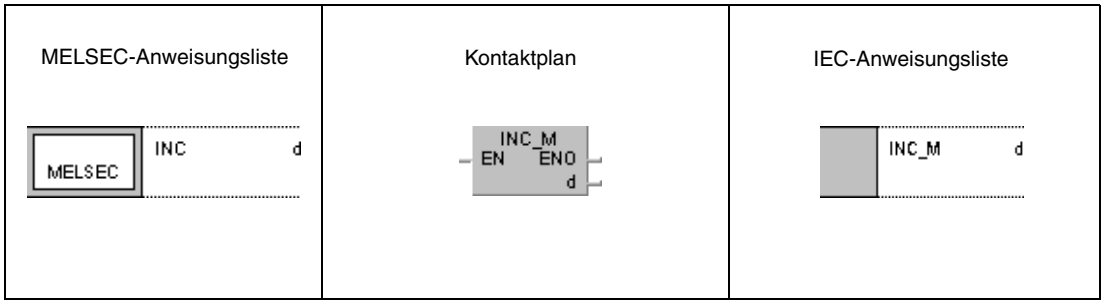
	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
d		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●						K1 ↓ K4	3 ●[1]	●		●

[1] Die Anzahl der Schritte bei Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
d	●	●	●	●	●	●	●	—	—	—	2

GX IEC Developer



GX Developer

Variablen

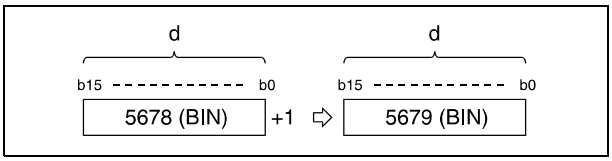
Operand	Befehlswert	Datentyp
d	Erste Adresse, auf die die INC- oder DEC-Anweisung angewendet werden soll.	BIN-16-Bit

Funktions-weise

Inkrementieren und Dekrementieren von Binärdaten (16-Bit)

INC Binärdaten inkrementieren (16 Bit)

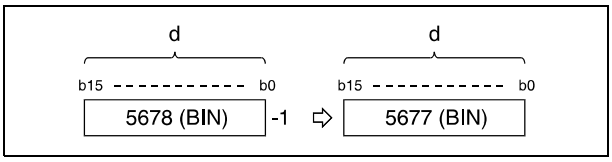
Der in d vorgegebene Operand (16 Bit) wird betragsmäßig um 1 erhöht.



Lautet der Inhalt von d bei Ausführung einer INC- oder INCP-Anweisung 32767, wird in d der Wert -32768 abgelegt.

DEC Binärdaten dekrementieren (16 Bit)

Der in d vorgegebene Operand (16 Bit) wird betragsmäßig um 1 vermindert.



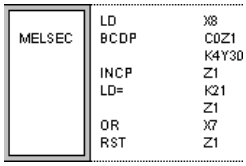
Ist der Inhalt von d bei Ausführung einer DEC- oder DECP-Anweisung gleich 0, wird in d eine -1 abgelegt.

Lautet der Inhalt von d bei Ausführung einer DEC- oder DECP-Anweisung -32768, wird in d der Wert 32767 abgelegt.

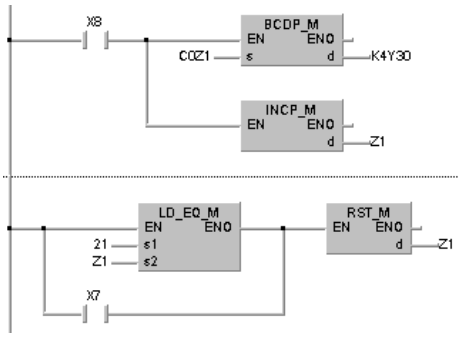
Beispiel 1 INCP

Das folgende Programm gibt mit jeder positiven Flanke von X8 den Istwert der Counter C0 bis C20 (C0 plus Z1) an Y30 bis Y3F in BCD aus (der Zählersollwert ist 9999). Z1 wird auf 0 gesetzt (RST Z1), wenn Z1 gleich 21 ist (LD= K21 Z1) oder der Reset-Eingang X7 eingeschaltet wird.

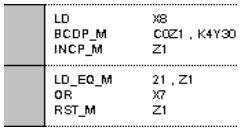
MELSEC-Anweisungsliste



Kontaktplan

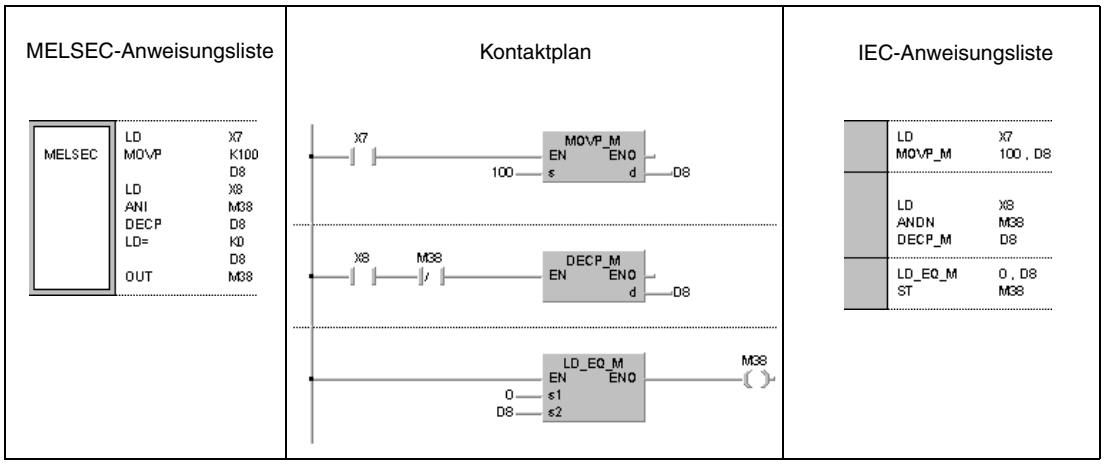


IEC-Anweisungsliste



Beispiel 2 DECP

Das folgende Programm zeigt ein Beispiel für einen Abwärtszähler. Mit der positiven Flanke von X7 wird der Wert 100 in D8 geschrieben. Vorausgesetzt M38 ist nicht gesetzt, wird der Wert in D8 mit positiver Flanke von X8 um 1 vermindert. M38 wird gesetzt, sobald D8 gleich 0 ist.



6.2.14 DINC, DINCP, DDEC, DDECP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
d		●	●	●	●	●	●	●	●	●	●	●	●		●	●[1]						K1 ↓ K8	3 ●[2]	●		●

[1] Nicht für AnN CPU

[2] Die Anzahl der Schritte bei Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

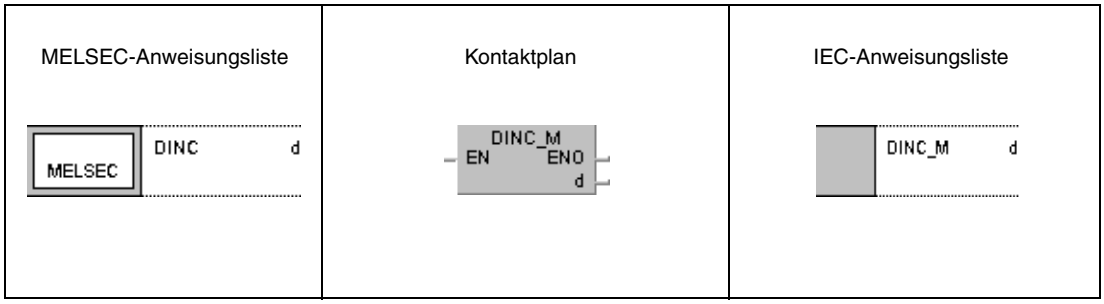
Operanden MELSEC Q

	Operanden								Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sonder-module U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort				U		
d	●	●	●	●	●	●	●	—	—	—	2[1)]

[1] Die Anzahl der Programmschritte ist abhängig vom Typ der CPU und von den verwendeten Operanden.

Bei Verwendung einer QnA-CPU oder einer Single-Prozessor-Q-CPU:	2
Bei Verwendung einer Multi-Prozessor-Q-CPU und interner Wortoperanden (außer File-Register ZR):	3
Konstanten:	3
Bit-Operanden, deren Adresse ein Vielfaches von 16 ist, die die Bit-Blockbezeichnung K8 haben und die nicht durch Index-Vergabe bearbeitet werden:	3
anderer Operanden als oben aufgeführt:	2

GX IEC Developer



GX Developer

Variablen

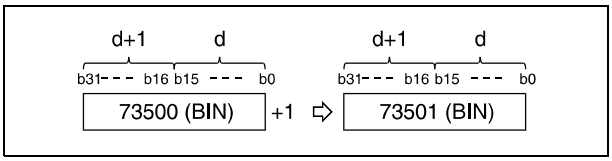
Operand	Befehlswert	Datentyp
d	Erste Adresse, auf die die DINC- oder DDEC-Anweisung angewendet werden soll.	BIN-32-Bit

Funktions-weise

Inkrementieren und Dekrementieren von Binärdaten (32-Bit)

DINC Binärdaten inkrementieren (32 Bit)

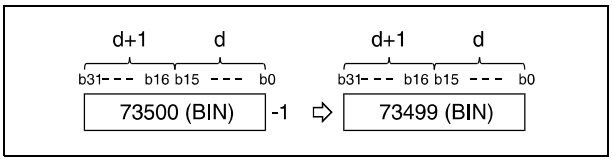
Der in d vorgegebene Operand (32 Bit) wird betragsmäßig um 1 erhöht.



Lautet der Inhalt von d bei Ausführung einer DINC- oder DINCP-Anweisung 2147483647, wird in d der Wert -2147483648 abgelegt.

DDEC Binärdaten dekrementieren (32 Bit)

Der in d vorgegebene Operand (32 Bit) wird betragsmäßig um 1 vermindert.

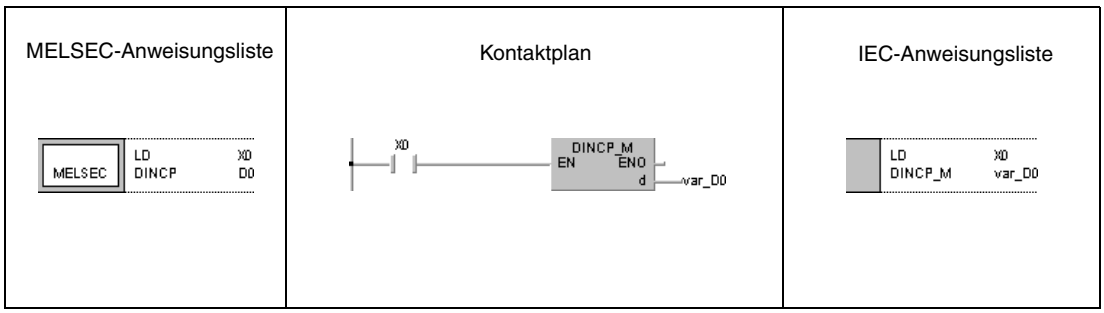


Ist der Inhalt von d bei Ausführung einer DDEC- oder DDECP-Anweisung gleich 0, wird in d eine -1 abgelegt.

Lautet der Inhalt von d bei Ausführung einer DDEC- oder DDECP-Anweisung -2147483648, wird in d der Wert 2147483647 abgelegt.

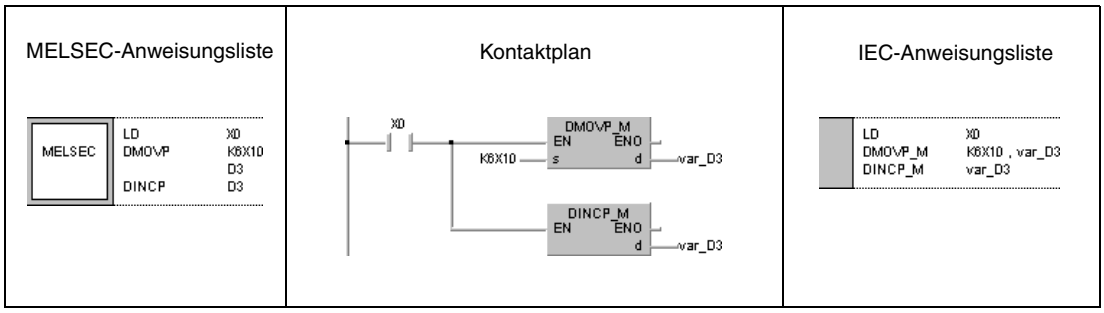
Beispiel 1 DINCP

Das folgende Programm erhöht den Wert in D0 mit positiver Flanke von X0 um 1.



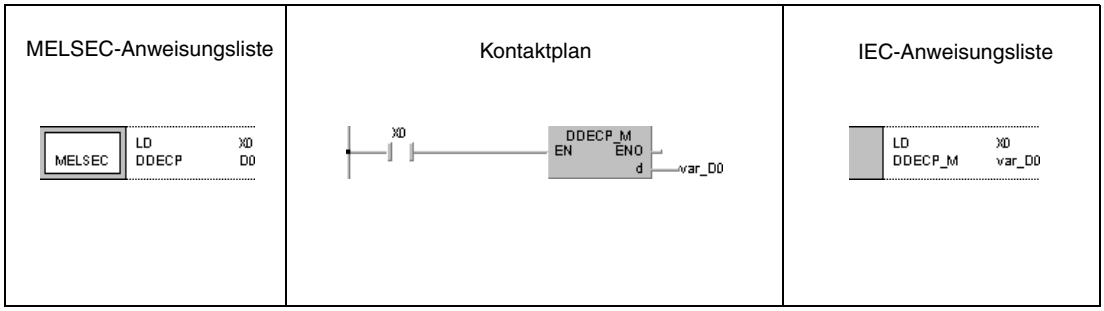
Beispiel 2 DINCP

Das folgende Programm erhöht mit positiver Flanke von X0 die Datenwerte von X10 bis X27 um 1 und speichert die Ergebnisse in den Registern D3 und D4.



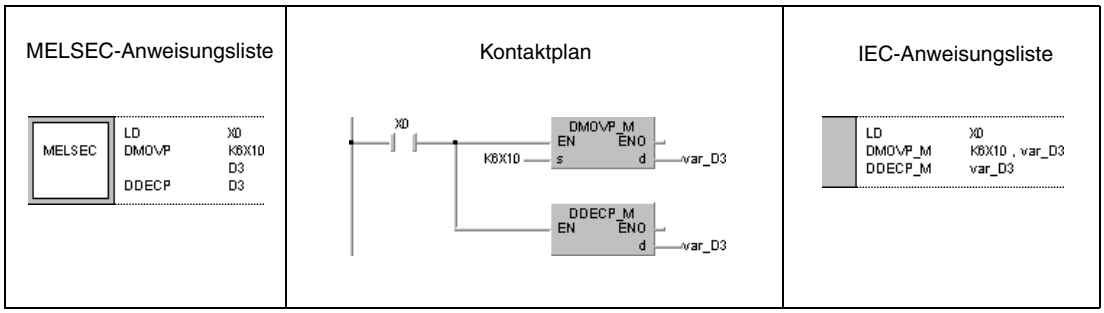
Beispiel 3 DDECP

Das folgende Programm vermindert mit positiver Flanke von X0 den Wert in D0 um 1.



Beispiel 4 DDECP

Das folgende Programm vermindert mit positiver Flanke von X0 die Datenwerte von X10 bis X27 und speichert das Ergebnis in D3 und D4.



HINWEIS *Diese Programmbeispiele sind ohne Variablendefinition im Header der Programm-Organisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

6.3 Konvertierungsanweisungen

Die im vorliegenden Abschnitt beschriebenen Konvertierungsanweisungen wandeln unterschiedliche Datenformate um.

HINWEIS *Sie sollten in den IEC-Editoren die IEC-Befehle nutzen.*

Konvertierung	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor
BIN (16-/32-Bit) ⇓ BCD (4-/8-stellig)	BCD	BCD_M
	BCDP	BCDP_M
	DBCD	DBCD_M
	DBCDP	DBCDP_M
BCD (4-/8-stellig) ⇓ BIN (16-/32-Bit)	BIN	BIN_M
	BINP	BINP_M
	DBIN	DBIN_M
	DBINP	DBINP_M
BIN (16-/32-Bit) ⇓ Gleitkommazahl	FLT	FLT_M
	FLTP	FLTP_M
	DFLT	DFLT_M
	DFLTP	DFLTP_M
Gleitkommazahl ⇓ BIN (16-/32-Bit)	INT	INT_MD
		INT_E_MD
	INTP	INT_P_MD
		INT_P_E_MD
	DINT	DINT_MD
		DINT_E_MD
	DINTP	DINT_P_MD
		DINT_P_E_MD
BIN-16-Bit ⇓ BIN-32-Bit	DBL	DBL_M
	DBLP	DBLP_M
BIN-32-Bit ⇓ BIN-16-Bit	WORD	WORD_M
	WORDP	WORDP_M
BIN (16-/32-Bit) ⇓ GRAY-CODE	GRY	GRY_M
	GRYP	GRYP_M
	DGRY	DGRY_M
	DGRYP	DGRYP_M
GRAY-CODE ⇓ BIN (16-/32-Bit)	GBIN	GBIN_M
	GBINP	GBINP_M
	DGBIN	DGBIN_M
	DGBINP	DGBINP_M
Vorzeichenumkehr BIN (16-/32-Bit) (Zweierkomplement)	NEG	NEG_M
	NEGP	NEGP_M
	DNEG	DNEG_M
	DNEGP	DNEGP_M
Vorzeichenumkehr Gleitkommazahl	ENEG	ENEG_M
	ENEGP	ENEGP_M

Konvertierung	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor
BIN-Blöcke (16-Bit) ⇓ BCD-Blöcke (4-stellig)	BKBCD	BKBCD_M
	BKBCDP	BKBCDP_M
BCD-Blöcke (4-stellig) ⇓ BIN-Blöcke (16-Bit)	BKBIN	BKBIN_M
	BKBINP	BKBINP_M

6.3.1 BCD, BCDP, DBCD, DBCDP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

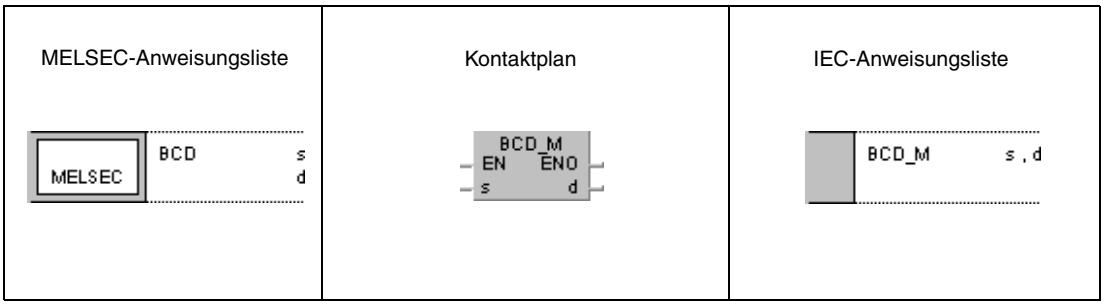
	Operanden																					Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
BCD																										
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●						K1 ↓ K4	5	●		●
d		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●							●[1]			
DBCD																										
s	●	●	●	●	●	●	●	●	●	●	●	●	●		●							K1 ↓ K8	9	●		●
d		●	●	●	●	●	●	●	●	●	●	●	●		●								●[1]			

[1] Die Anzahl der Schritte bei Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

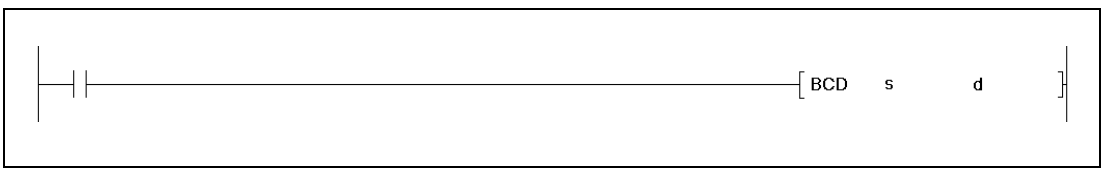
Operanden MELSEC Q

	Operanden								Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere	
	Bit	Wort		Bit	Wort					
s	●	●	●	●	●	●	●	●	—	SM0	3
d	●	●	●	●	●	●	●	—	—		

GX IEC Developer



GX Developer



Variablen

Operand	Befehlswert	Datentyp
s	BIN-Daten oder Operand, unter dem die BIN-Daten gespeichert sind.	BIN-16-/32-Bit
d	Operand, unter dem die BCD-Daten gespeichert werden.	BCD-4-/8-stellig

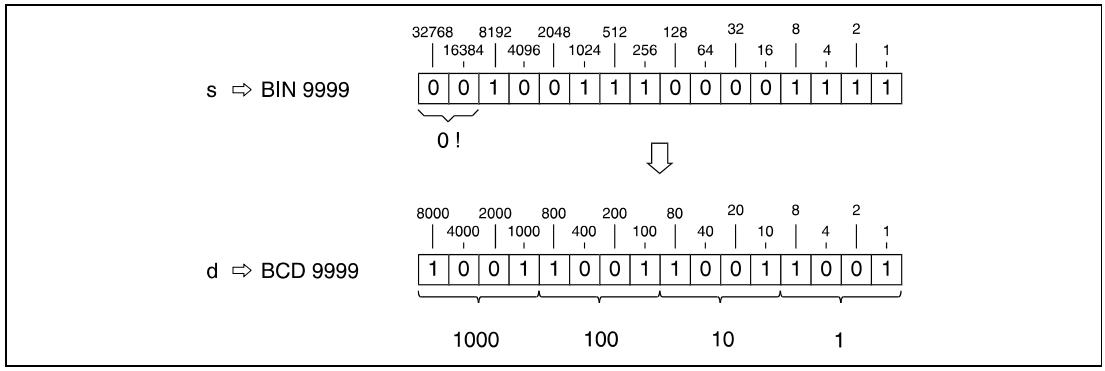
Funktions-weise

Konvertierung von BIN- in BCD-Daten (4-/8-stellig)

BCD Konvertierung von BIN- in BCD-Daten (4-stellig)

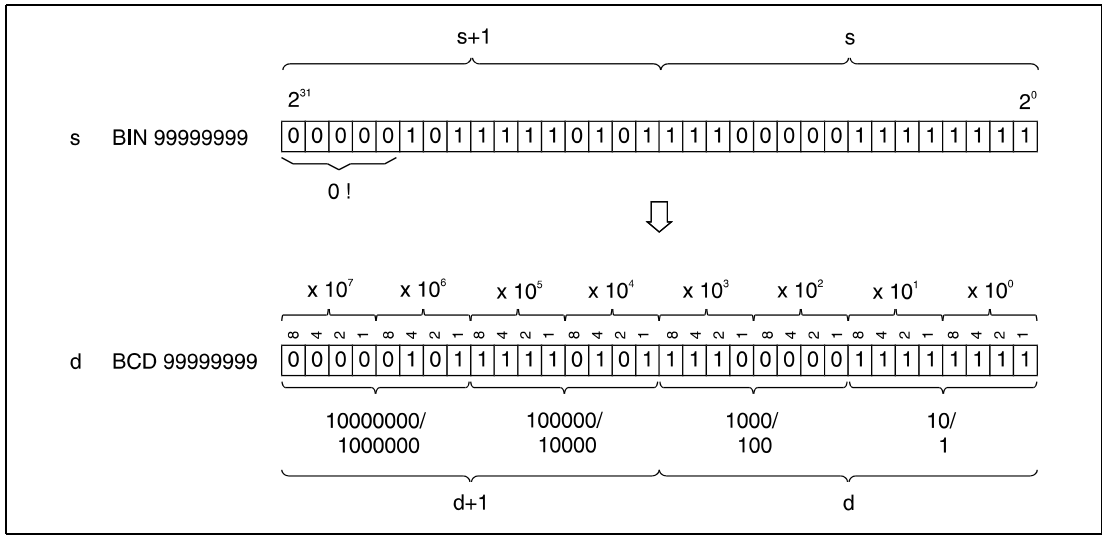
Die BCD-Anweisung konvertiert den in s angegebenen Binärdatenwert (0 bis 9999) in einen BCD-Wert und überträgt das Ergebnis an den in d angegeben Operanden.

Die beiden höchstwertigen Bits des Binärwertes in s müssen bei einem 4-stelligen Datenwert 0 sein.



DBCD Konvertierung von BIN- in BCD-Daten (8-stellig)

Die DBCD-Anweisung arbeitet in der gleichen Weise wie die BCD-Anweisung. Der Binärdatenwert kann jedoch 8-stellig (0 bis 99999999) sein. Die höchstwertigen fünf Bits des Binärwertes in s müssen 0 sein.



Fehler-quellen

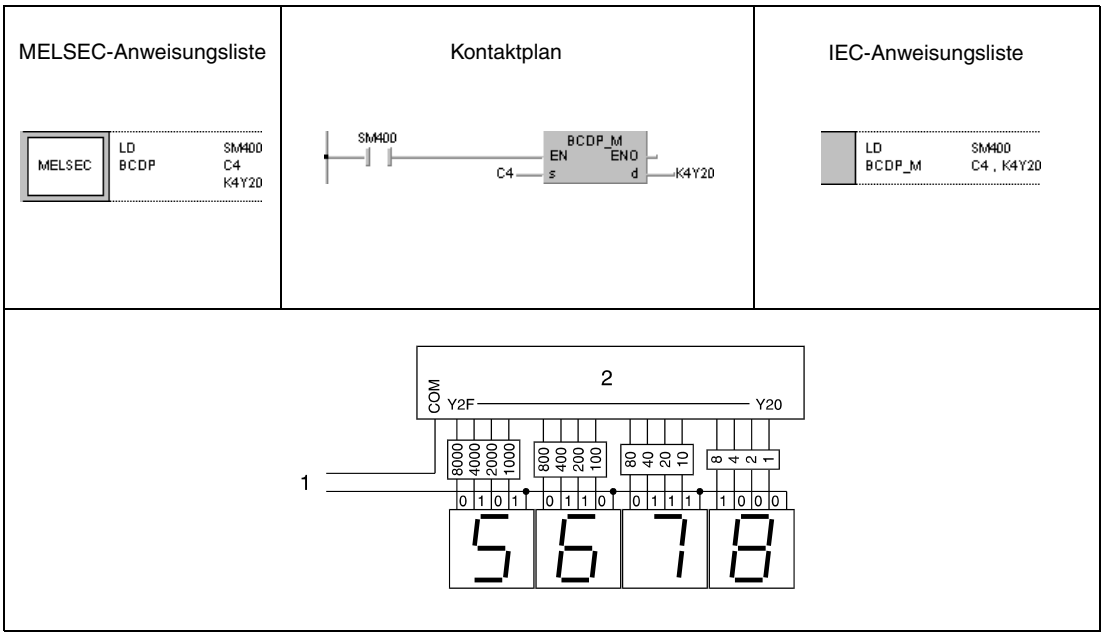
In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Bei Programmierung einer BCD-Anweisung liegen die Quelldaten in s außerhalb des Bereiches von 0 bis 9999 (Q-Serie/System Q = Fehlercode 4100).
- Bei Programmierung einer DBCD-Anweisung liegen die Quelldaten in s+1 oder s außerhalb des Bereiches von 0 bis 99999999 (Q-Serie/System Q = Fehlercode 4100).

Beispiel

BCDP

Das folgende Programm gibt mit positiver Flanke von SM400 den Istwert von C4 (5678) an Y20 bis Y2F zur BCD-Anzeige aus.



[1] Spannungsversorgung

[2] Ausgangsmodul

6.3.2 BIN, BINP, DBIN, DBINP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

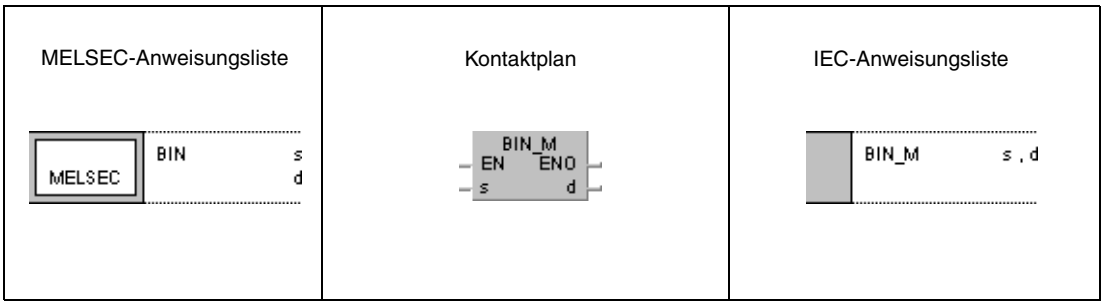
	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
BIN																										
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●						K1 ↓ K4	5	●		●
d		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●							●[1]			
DBIN																										
s	●	●	●	●	●	●	●	●	●	●	●	●	●		●							K1 ↓ K8	9	●		●
d		●	●	●	●	●	●	●	●	●	●	●	●		●								●[1]			

[1] Die Anzahl der Schritte bei Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

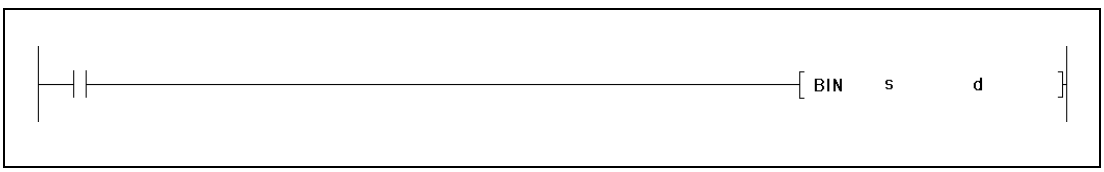
Operanden MELSEC Q

	Operanden								Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sonder-module U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere	
	Bit	Wort		Bit	Wort				U	
s	●	●	●	●	●	●	●	●	—	SM0	3
d	●	●	●	●	●	●	●	—	—		

GX IEC Developer



GX Developer



Variablen

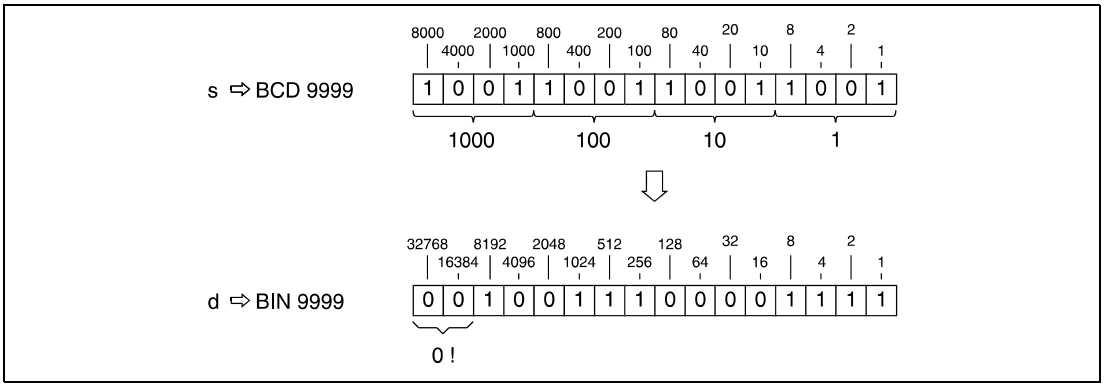
Operand	Befehlswert	Datentyp
s	BCD-Daten oder Operand, unter dem die BCD-Daten gespeichert sind.	BCD-4-/8-stellig
d	Operand, unter dem die BIN-Daten gespeichert werden.	BIN-16-/32-Bit

Funktions-weise

Konvertierung von BCD-Daten (4-/ 8-stellig) in BIN-Daten

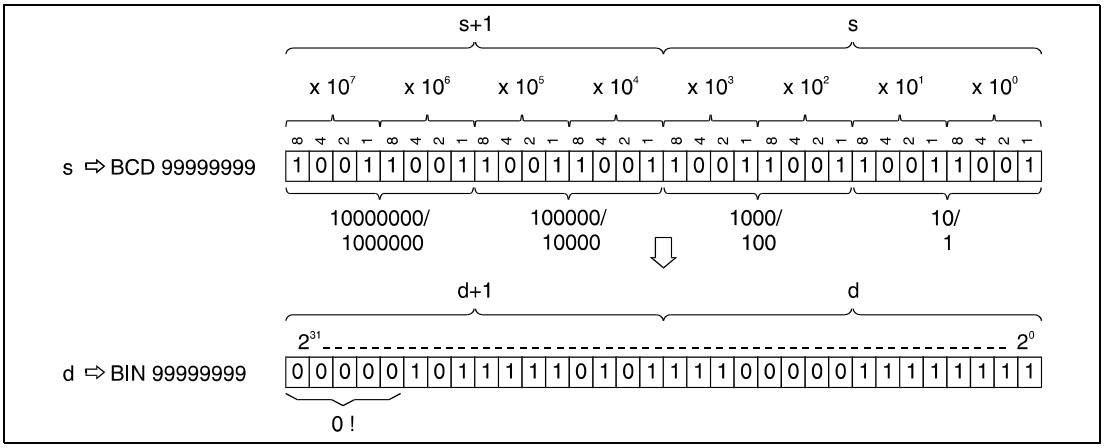
BIN Konvertierung von BCD- (4-stellig) in BIN-Daten

Die BIN-Anweisung konvertiert den in s angegebenen BCD-Datenwert (0 bis 9999) in einen Binärwert und überträgt das Ergebnis an den in d angegeben Operanden. Die höchstwertigen beiden Bits des Binärwertes in d müssen bei einem 4-stelligen binären Datenwert immer 0 sein.



DBIN Konvertierung von BCD- (8-stellig) in BIN-Daten

Die DBIN-Anweisung wird in gleicher Weise wie die BIN-Anweisung verarbeitet. Der BCD-Datenwert kann jedoch 8-stellig (0 bis 99999999) sein. Die höchstwertigen fünf Bit des Binärwertes in d müssen 0 sein.



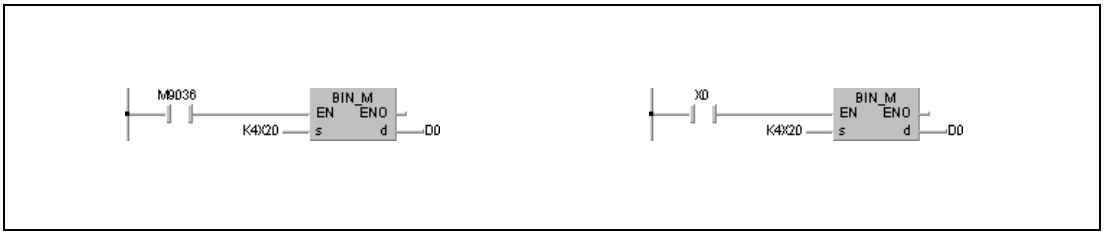
Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Eine Ziffernstelle des Quelldatenwertes in s liegt nicht im Bereich zwischen 0 und 9.

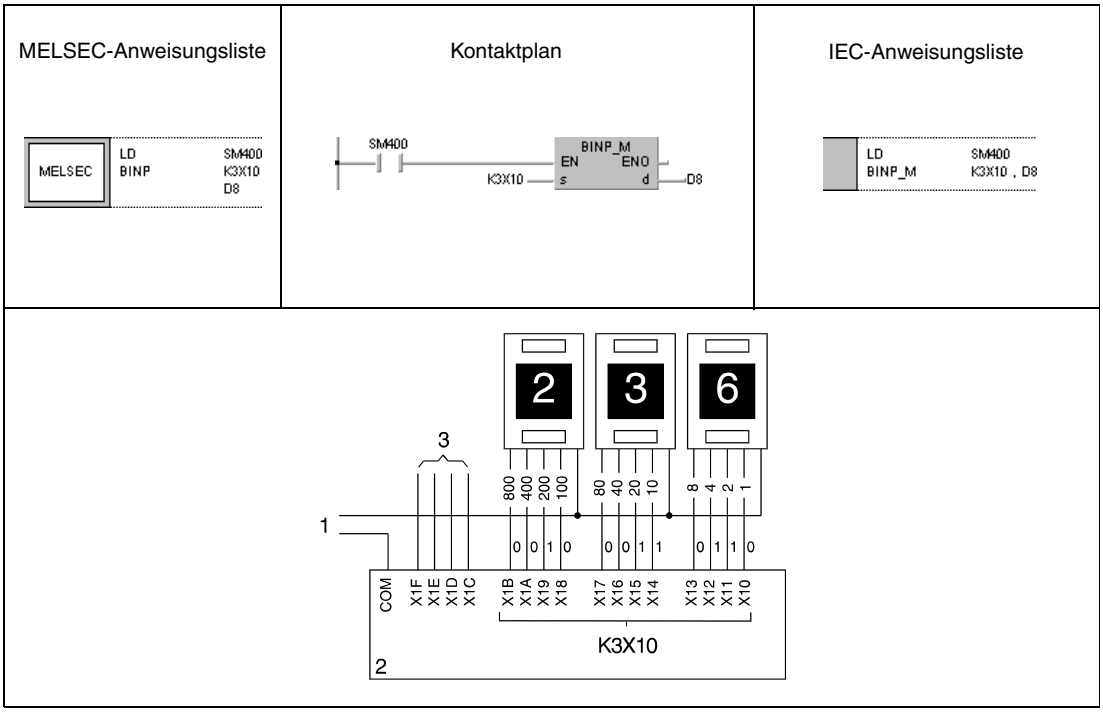
Bei Verwendung der Q-CPU kann die Fehlermeldung unterdrückt werden, wenn SM722 gesetzt wird. Die Anweisung wird unabhängig von Zustand von SM722 aber nicht ausgeführt, wenn der zulässige Bereich für s überschritten wird.

HINWEIS *Unter Umständen kann bei Ausführung einer BIN- oder DBIN-Anweisung mit Sondermerker M9036 oder M9037 als Eingangsbedingung aufgrund der Schaltverzögerung von BCD-Anzeigen ein Fehler in der Programmverarbeitung auftreten. In diesem Fall empfiehlt es sich, die binäre Konvertierung erst nach dem Setzen der BCD-Daten über einen gewöhnlichen Eingangskontakt vorzunehmen (gilt nur für die A-Serie).*



Beispiel 1 BINP

Das folgende Programm konvertiert die BCD-Daten der Eingänge X10 bis X1B in BIN-Datenwerte und speichert das Ergebnis mit positiver Flanke von SM400 in D8.



[1] Spannungsversorgung

[2] Eingangsmodul

[3] Eingänge zur freien Verfügung

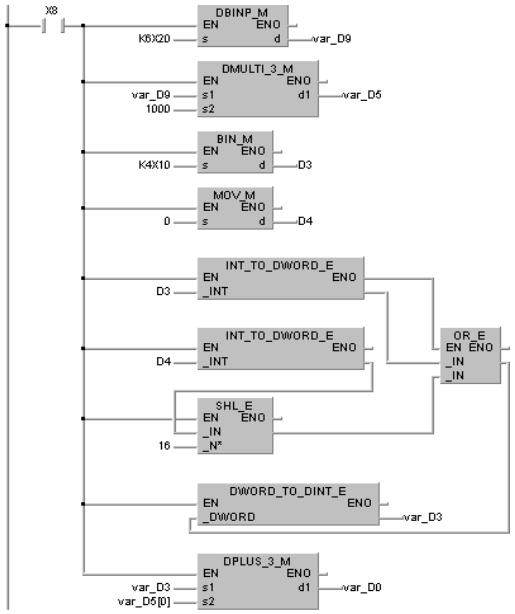
Beispiel 2 DBINP

Das folgende Programm konvertiert mit positiver Flanke von X8 die BCD-Daten der Eingänge X10 bis X37 in BIN-Datenwerte konvertiert. Das Ergebnis wird in D0 und D1 gespeichert.

MELSEC-Anweisungsliste

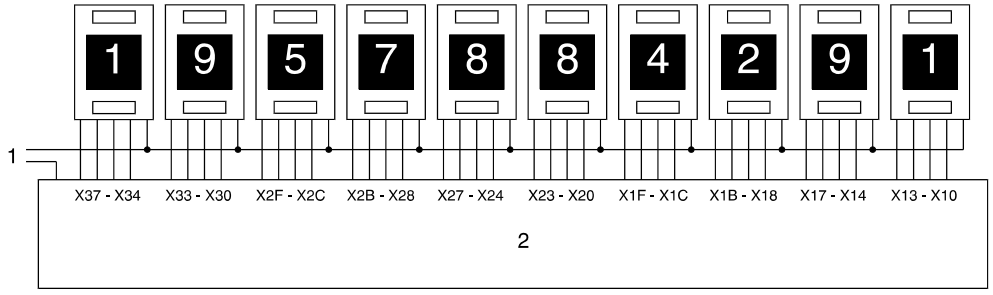
```
MELSEC   LD      X8
         DBINP   K6X20
                 D9
         D*      D9
                 K1000
                 D5
         BIN     K4X10
                 D3
         MOV     K0
                 D4
         D+      D3
                 D5
                 D0
```

Kontaktplan



IEC-Anweisungsliste

```
LD              X8
DBINP_M         K6X20 , Var_D9
DMULTI_3_M      var_D9 , K1000 , Var_D5
BIN_M           K4X10 , D3
MOV_M           K0 , D4
INT_TO_DWORD_E      D4 , var_D3
SHL_E           var_D3 , K16 , var_D3
INT_TO_DWORD_E      D3 , var_temp
LD              var_D3
OR              var_temp
ST              var_D3
LD              X8
DPLUS_3_M       var_D3 , var_D5[0] , var_D0
```



1 Spannungsversorgung

2 Eingangsmodul

HINWEISE *Stehen an den Eingängen X10 bis X37 BCD-Werte oberhalb von 2147483647 an, so liegen diese Werte außerhalb des Bereiches, der von 32-Bit-Operanden verarbeitet werden kann! Die Werte in D0 und D1 werden in diesem Fall negativ. Weitere Informationen hierzu sind dem Abs. 3.4 „Programmierung von Variablen“ dieser Programmieranleitung zu entnehmen.*

Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- und Checker-Fehlermeldungen können die Folge sein. Weitere Informationen hierzu sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.

6.3.3 FLT, FLTP, DFLT, DFLTP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●[1]

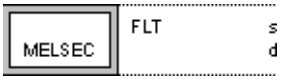
[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	●	●	●	●	●	●	●	●	—	—	3
d	—	●	●	—	●	●	—	—	—	—	

GX IEC Developer

MELSEC-Anweisungsliste



Kontaktplan

IEC-Anweisungsliste

GX Developer

Variablen

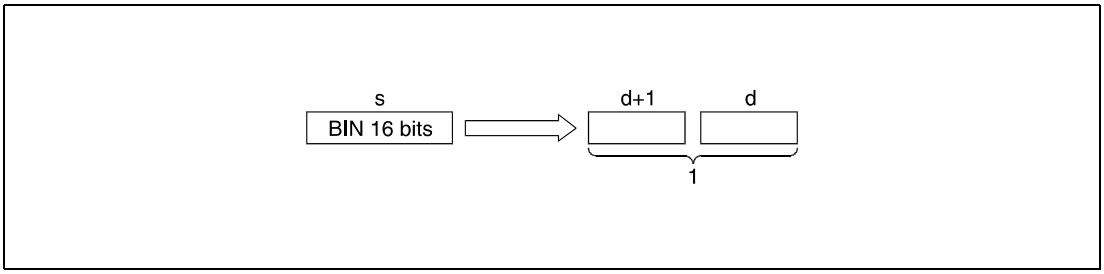
Operand	Befehlswert	Datentyp
s	Operand, in dem die zu konvertierenden BIN-Daten gespeichert sind.	BIN-16-/32-Bit
d	Operand, in dem die konvertierte Gleitkommazahl gespeichert wird.	Reelle Zahl

Funktions-weise

Konvertierung von 16-/32-Bit-Binärzahlen in Gleitkommazahlen

FLT Konvertierung von Binärdaten (16 Bit) in Gleitkommazahlen

Die in s angegebenen Binärdaten werden in eine Gleitkommazahl umgewandelt. Das Ergebnis wird in d gespeichert.

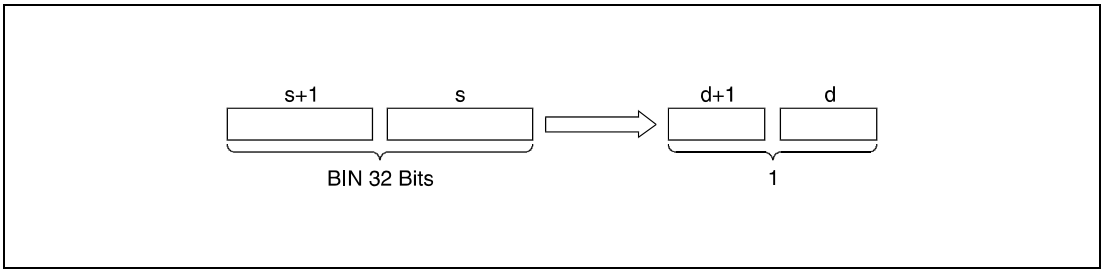


[1] Gleitkommazahl (reelle Zahl)

Der in s angegebene Datenwert muss zwischen -32768 und 32767 liegen.

DFLT Konvertierung von Binärdaten (32 Bit) in Gleitkommazahlen

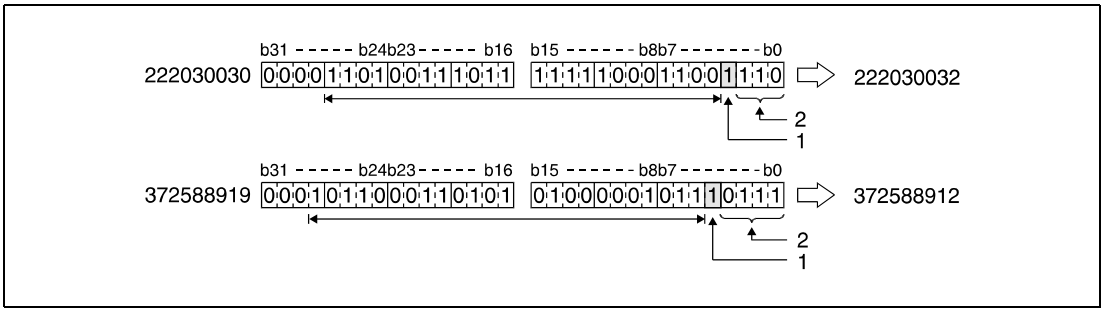
Die in s angegebenen 32-Bit-Binärdaten werden in eine Gleitkommazahl umgewandelt. Das Ergebnis wird in d gespeichert.



[1] Gleitkommazahl (reelle Zahl)

Der in s und s+1 angebene Datenwert muss zwischen -2147483648 und 2147483647 liegen.

Aufgrund der Tatsache, dass Gleitkommazahlen in einem 32-Bit-Prozess verarbeitet werden, reduziert sich die Datenbreite auf 24 Bits bei binärer Darstellung bzw. (annähernd) 7 Digits bei dezimaler Darstellung. Vor der Konvertierung wird der Binärdatenwert auf eine Breite von 25 Bits gerundet. Alle Bits nach dem 25 sten werden eliminiert. Liegt der gerundete Wert (Integer) außerhalb des mit 24 Bits darstellbaren Bereichs (-16777216 bis 16777215), können während der Umwandlung Fehler auftreten.

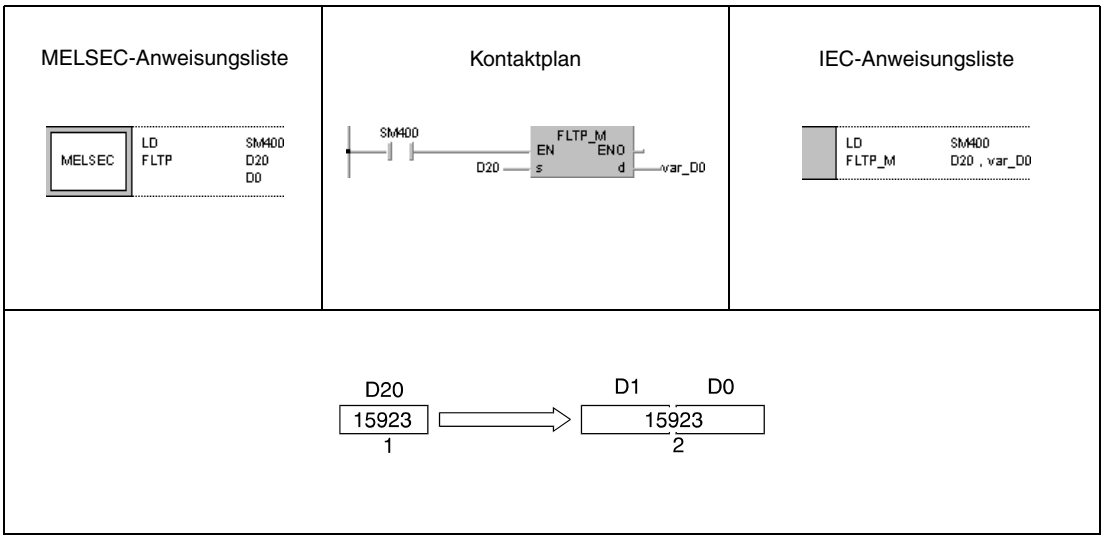


[1] Gerundete Stelle

[2] Eliminierte Bits

Beispiel 1 FLTP

Das folgende Programm wandelt mit positiver Flanke von SM400 die in D20 gespeicherte 16-Bit-Binärzahl in eine Gleitkommazahl um und speichert das Ergebnis in D0 und D1 ab.

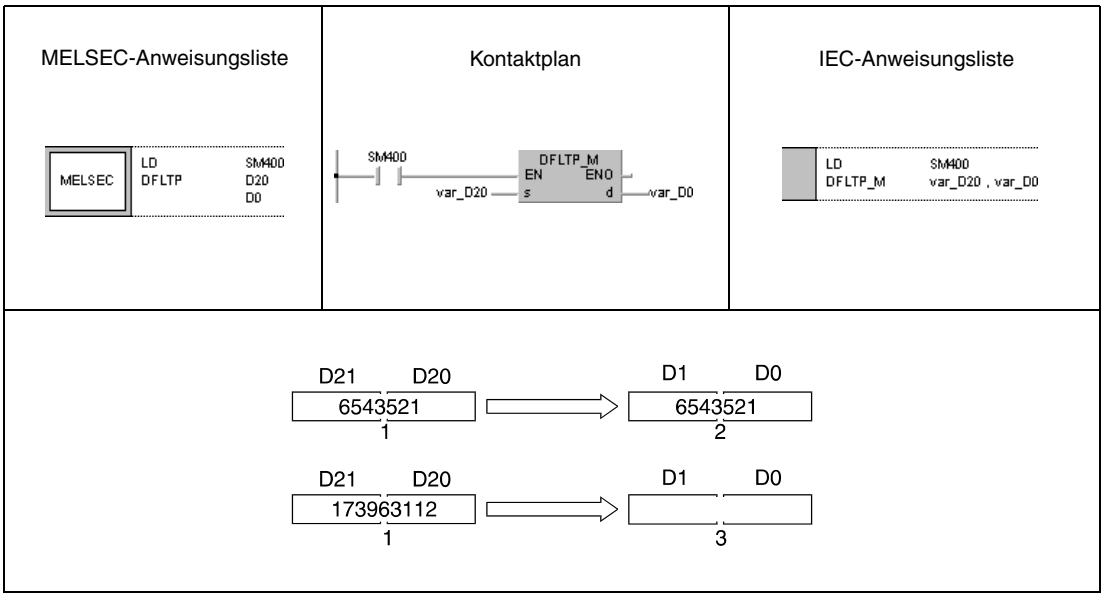


[1] Binärdaten

[2] Gleitkommazahl (reelle Zahl)

Beispiel 2 DFLTP

Das folgende Programm wandelt mit positiver Flanke von SM400 die in D20 und D21 gespeicherte 32-Bit-Binärzahl in eine Gleitkommazahl um und speichert das Ergebnis in D0 und D1 ab.



[1] Binärdaten

[2] Gleitkommazahl (reelle Zahl)

[3] Es tritt ein Umwandlungsfehler auf, da der zu konvertierende Wert eine Datenbreite von 7 Digits aufweist.

HINWEIS *Diese Programmbeispiele sind ohne Variablendefinition im Header der Programmorganisationeinheit (POE) nicht lauffähig. Compiler- und Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

6.3.4 INT, INTP, DINT, DINTP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●[2]

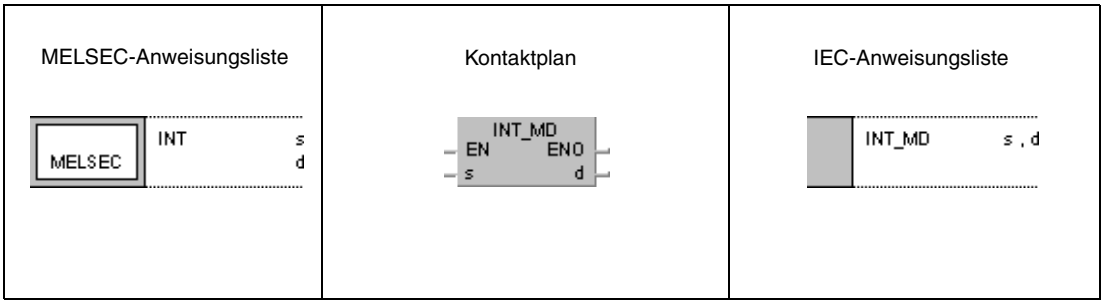
[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R Anweisungen programmiert werden.

[2] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten E	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	●	●	—	●	—	SM0	3
d	●	●	●	●	●	●	●	—	—		

GX IEC Developer



GX Developer

Variablen

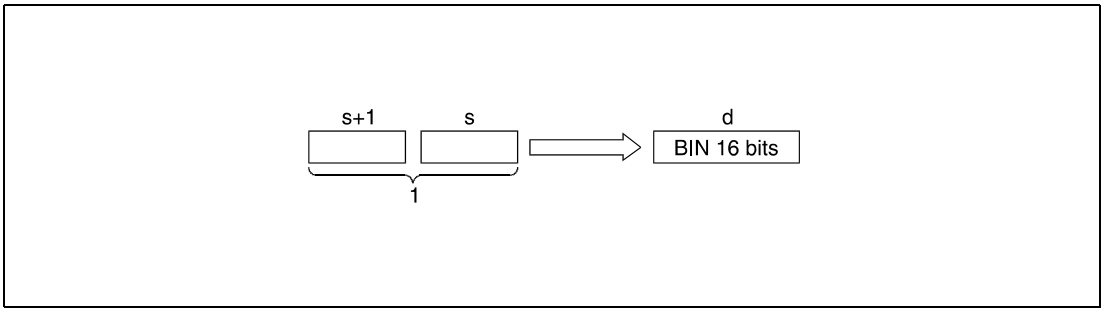
Operand	Befehlswert	Datentyp
s	Operand, in dem die zu konvertierende Gleitkommazahl gespeichert ist.	Reelle Zahl
d	Operand, in dem die konvertierten BIN-Daten gespeichert werden.	BIN-16-/32-Bit

Funktionsweise

Konvertierung von Gleitkommazahlen in 16-/32-Bit-Binärdaten

INT Konvertierung von Gleitkommazahlen in 16-Bit-Binärdaten

Die in s gespeicherte Gleitkommazahl wird in eine 16-Bit-Binärzahl umgewandelt und das Ergebnis in d gespeichert.



[1] Gleitkommazahl (reelle Zahl)

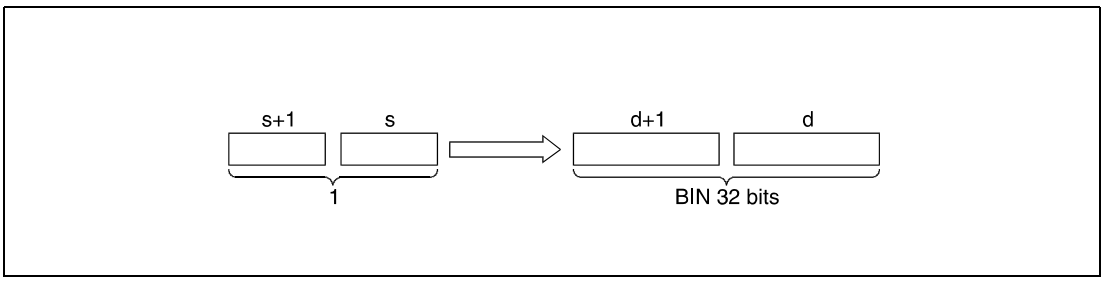
Die in s und s+1 eingetragene Gleitkommazahl muss im Bereich zwischen -32768 und 32767 liegen.

Der umgewandelte Integer-Wert wird in d im 16-Bit-Binärdatenformat gespeichert.

Der in d abgespeicherte Wert gibt die auf eine Stelle vor dem Komma gerundete und konvertierte Gleitkommazahl wieder.

DINT Konvertierung von Gleitkommazahlen in 32-Bit-Binärdaten

Die in s gespeicherte Gleitkommazahl wird in eine 32-Bit-Binärzahl umgewandelt und das Ergebnis in d gespeichert.



[1] Gleitkommazahl (reelle Zahl)

Die in s und s+1 abgelegte Gleitkommazahl muss zwischen -2147483648 und 2147483647 liegen.

Nach der Umwandlung wird der Integer-Wert im 32-Bit-Binärdatenformat gespeichert.

Der in d angegebene Datenwert gibt die auf eine Stelle vor dem Komma gerundete und umgewandelte Gleitkommazahl wieder.

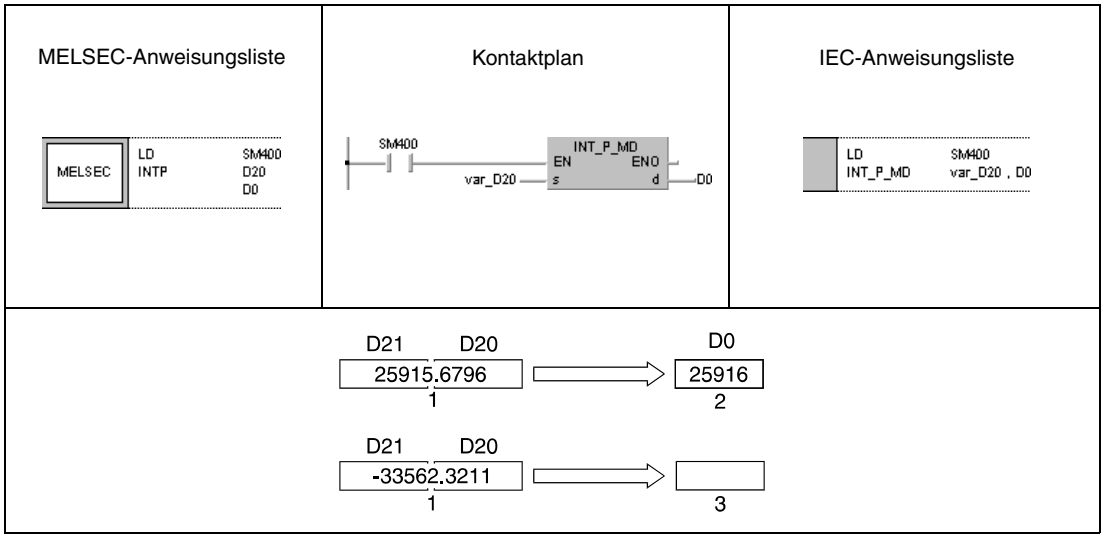
Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Der in s eingetragene Datenwert liegt bei Verwendung der INT-Anweisung außerhalb des Bereichs von -32768 bis 32767.
- Der in s angegebene Datenwert liegt bei Verwendung der DINT-Anweisung außerhalb des Bereichs von -2147483648 bis 2147483647.

Beispiel 1 INTP

Das folgende Programm wandelt mit positiver Flanke von SM400 die in D20 und D21 gespeicherte Gleitkommazahl in eine 16-Bit-Binärzahl um und speichert das Ergebnis in D0.



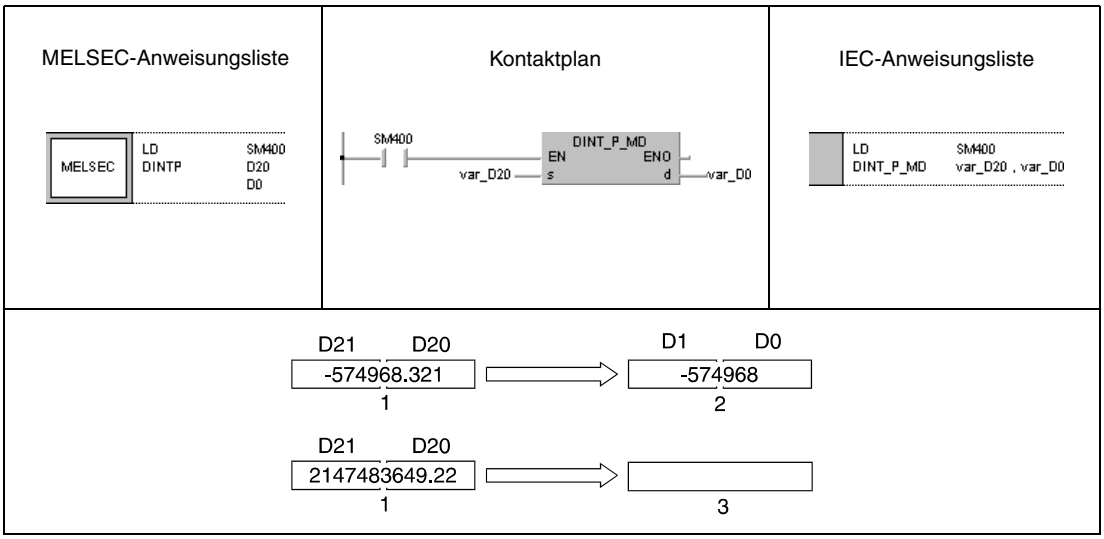
[1] Gleitkommazahl (reelle Zahl)

[2] Binärdaten

[3] Der zu konvertierende Wert liegt außerhalb des für INT-Anweisungen gültigen Datenbereichs. Dadurch findet keine Umwandlung statt, und es wird ein Fehler gemeldet.

Beispiel 2 DINTP

Das folgende Programm konvertiert mit positiver Flanke von SM400 die in D20 und D21 abgelegte Gleitkommazahl in eine 32-Bit-Binärzahl. Das Ergebnis wird in D0 und D1 gespeichert.



[1] Gleitkommazahl (reelle Zahl)

[2] Binärdaten

[3] Der zu konvertierende Wert liegt außerhalb des für DINT-Anweisungen gültigen Datenbereichs. Dadurch findet keine Umwandlung statt, und es wird ein Fehler gemeldet.

HINWEIS *Diese Programmbeispiele sind ohne Variablendefinition im Header der Programmorganisationeinheit (POE) nicht lauffähig. Compiler- und Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

6.3.5 DBL, DBLP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

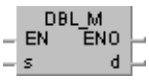
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	●	●	●	●	●	●	●	●	—	—	3
d	●	●	●	●	●	●	●	—	—	—	

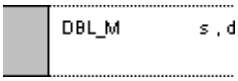
GX IEC Developer

MELSEC-Anweisungsliste

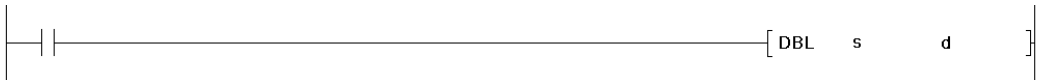
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

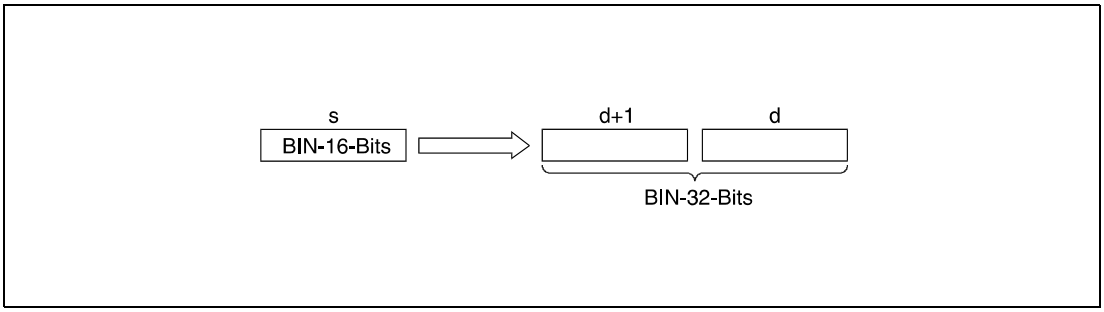
Operand	Befehlswert	Datentyp
s	Operand, in dem die zu konvertierenden BIN-Daten gespeichert sind.	BIN-16-Bit
d	Operand, in dem die konvertierten BIN-Daten gespeichert werden.	BIN-32-Bit

Funktionsweise

Konvertierung von 16-Bit-Binärdaten in 32-Bit-Binärdaten

DBL Konvertierung von 16-Bit-Binärdaten in 32-Bit-Binärdaten

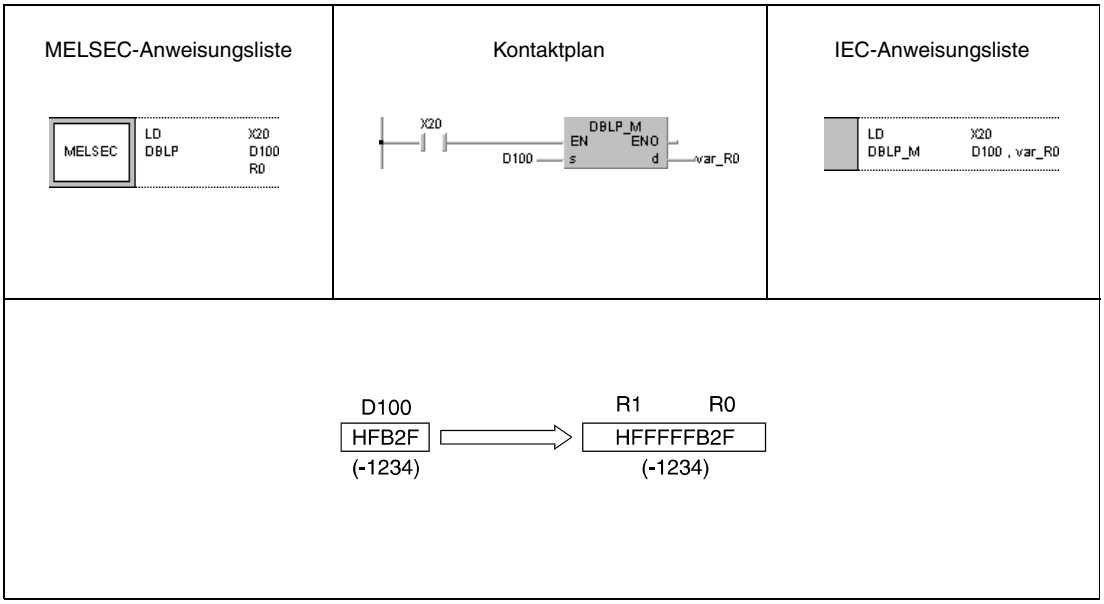
Die in s gespeicherten 16-Bit-Binärdaten werden in 32-Bit-Binärdaten konvertiert. Das Ergebnis wird mit Vorzeichen in d gespeichert.



Beispiel

DBLP

Das folgende Programm wandelt mit positiver Flanke von X20 eine in D100 gespeicherte 16-Bit-Binärzahl in eine 32-Bit-Binärzahl um. Das Ergebnis wird in R0 und R1 gespeichert.



HINWEIS

Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- und Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.

6.3.6 WORD, WORDP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

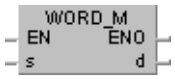
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□N□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	●	●	●	●	●	●	●	●	—	SM0	3
d	●	●	●	●	●	●	●	—	—		

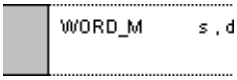
GX IEC Developer

MELSEC-Anweisungsliste

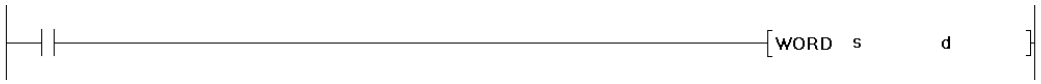
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

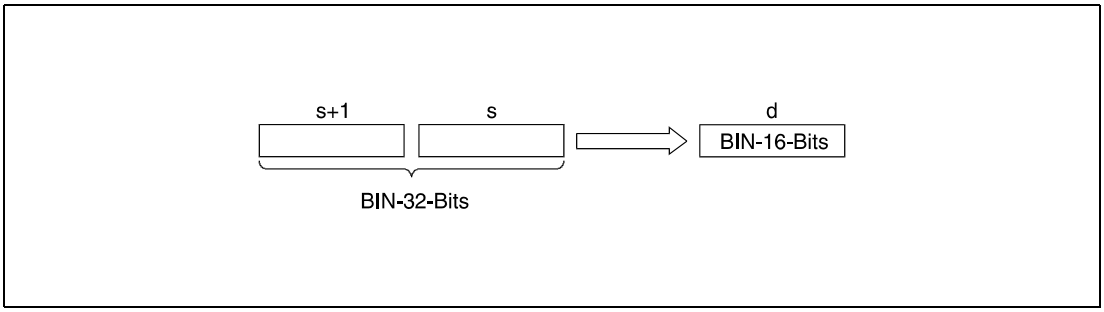
Operand	Befehlswert	Datentyp
s	Operand, in dem die zu konvertierenden Binärdaten gespeichert sind.	BIN-32-Bit
d	Operand, in dem die konvertierten Binärdaten gespeichert werden.	BIN-16-Bit

Funktionsweise

Konvertierung von 32-Bit-Binärdaten in 16-Bit-Binärdaten

WORD Konvertierung von 32-Bit-Binärdaten in 16-Bit-Binärdaten

Die in s gespeicherten 32-Bit-Binärdaten werden in 16-Bit-Binärdaten konvertiert. Das Ergebnis wird mit Vorzeichen in d gespeichert.



Fehlerquellen

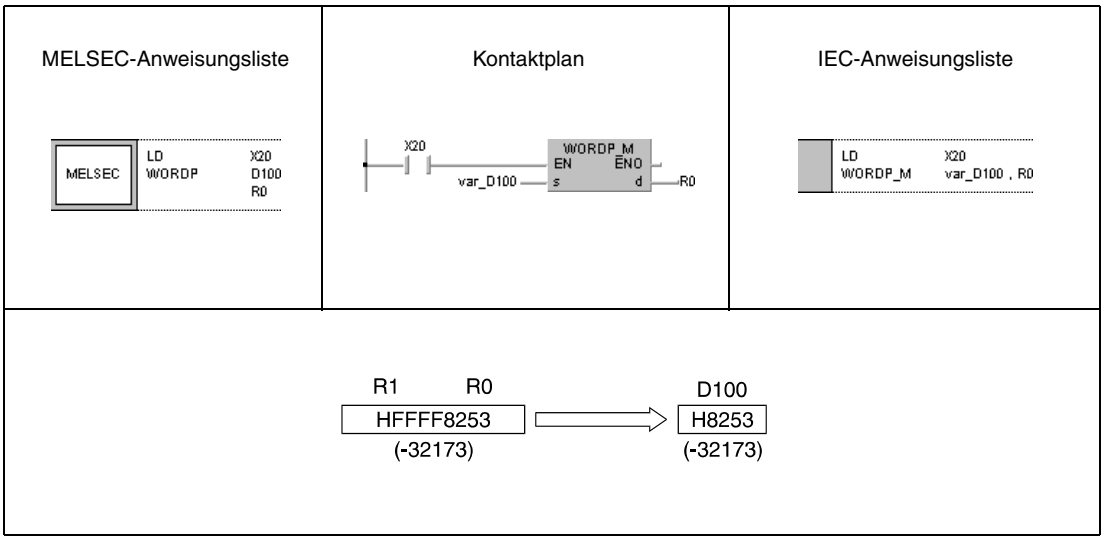
In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Der in s und s+1 eingetragene Datenwert liegt außerhalb des Bereiches von -32768 bis 32767 (Fehlercode 4100).

Beispiel

WORDP

Das folgende Programm wandelt mit positiver Flanke von X20 eine in D100 und D101 gespeicherte 32-Bit-Binärzahl in eine 16-Bit-Binärzahl um. Das Ergebnis wird in R0 gespeichert.



HINWEIS

Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- und Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.

6.3.7 GRY, GRYP, DGRY, DGRYP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Operanden MELSEC Q

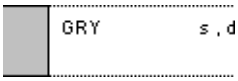
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	●	●	●	●	●	●	●	●	—	SM0	3
d	●	●	●	●	●	●	●	—	—		

GX IEC Developer

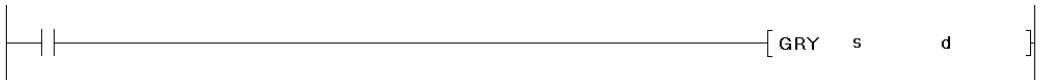
MELSEC-Anweisungsliste

Kontaktplan

IEC-Anweisungsliste



GX Developer



Variablen

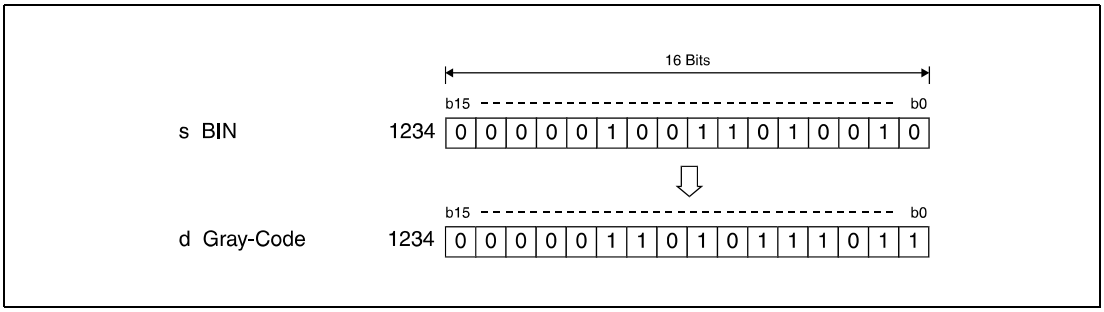
Operand	Befehlswert	Datentyp
s	Binärdaten oder Operand, in dem die Binärdaten gespeichert sind.	BIN-16-/32-Bit
d	Operand, in dem die konvertierten Gray-Code-Daten gespeichert werden.	Gray-Code 16-/32-Bit

Funktions-weise

Konvertierung von 16-/32-Bit-Binärdaten in den Gray-Code

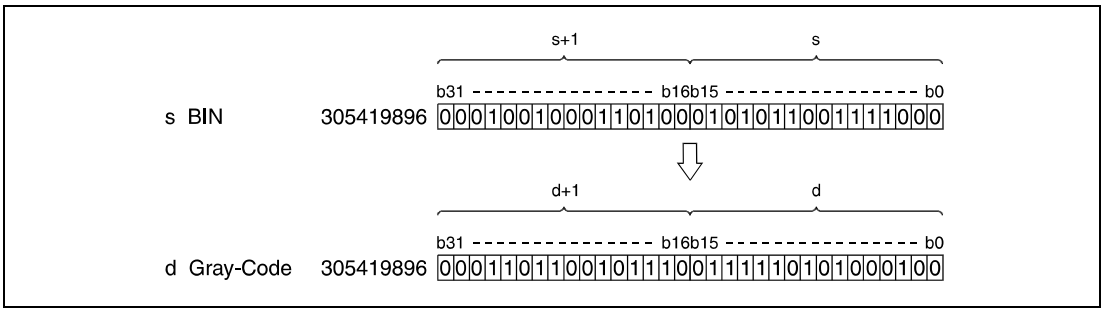
GRY Konvertierung von 16-Bit-Binärdaten in den Gray-Code

Die in s gespeicherten 16-Bit-Binärdaten werden im Gray-Code kodiert. Das Ergebnis wird in d gespeichert.



DGRY Konvertierung von 32-Bit-Binärdaten in den Gray-Code

Die in s gespeicherten 32-Bit-Binärdaten werden im Gray-Code kodiert. Das Ergebnis wird in d gespeichert.



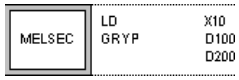
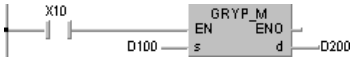
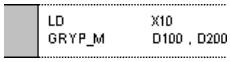
Fehler-quellen

In folgenden Fälllen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Der in s eingetragene Datenwert ist negativ.

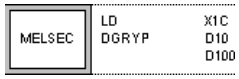
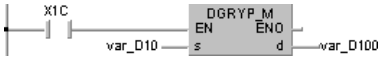
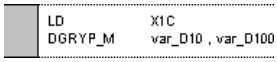
Beispiel 1 GRYP

Das folgende Programm wandelt mit positiver Flanke von X10 eine in D100 gespeicherte 16-Bit-Binärzahl in den Gray-Code um. Das Ergebnis wird in D200 gespeichert.

MELSEC-Anweisungsliste	Kontaktplan	IEC-Anweisungsliste
		

Beispiel 2 DGRYP

Das folgende Programm wandelt mit positiver Flanke von X1C eine in D10 und D11 gespeicherte 32-Bit-Binärzahl in den Gray-Code um. Das Ergebnis wird in D100 und D101 gespeichert.

MELSEC-Anweisungsliste	Kontaktplan	IEC-Anweisungsliste
		

HINWEIS *Das Programmbeispiel 2 ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- und Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

6.3.8 GBIN, GBINP, DGBIN, DGBINP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

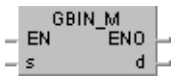
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	●	●	●	●	●	●	●	●	—	SM0	3
d	●	●	●	●	●	●	●	—	—		

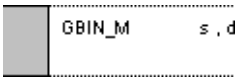
GX IEC Developer

MELSEC-Anweisungsliste

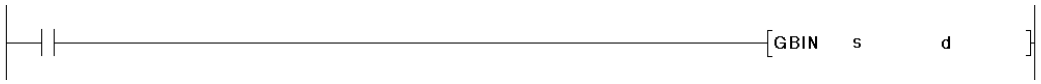
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

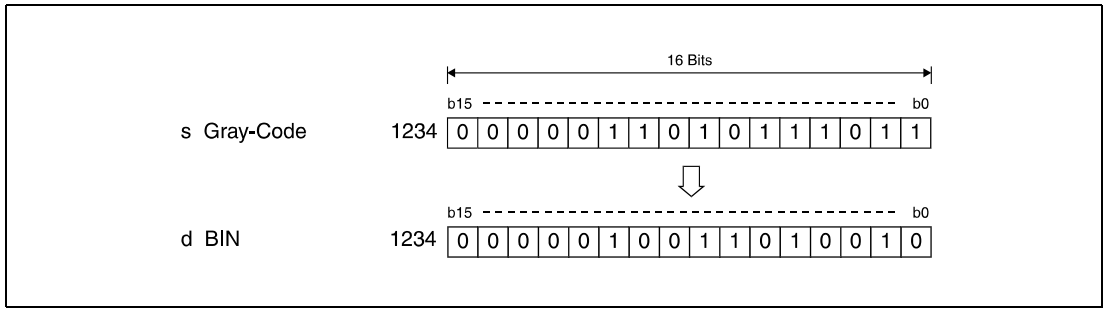
Operand	Befehlswert	Datentyp
s	Gray-Code-Daten oder Operand, in dem die Gray-Code-Daten gespeichert sind.	Gray-Code 16-/32-Bit
d	Operand, in dem die konvertierten Binärdaten gespeichert werden.	BIN-16-/32-Bit

Funktions-weise

Konvertierung von Gray-Code-Daten in 16-/32-Bit-Binärdaten

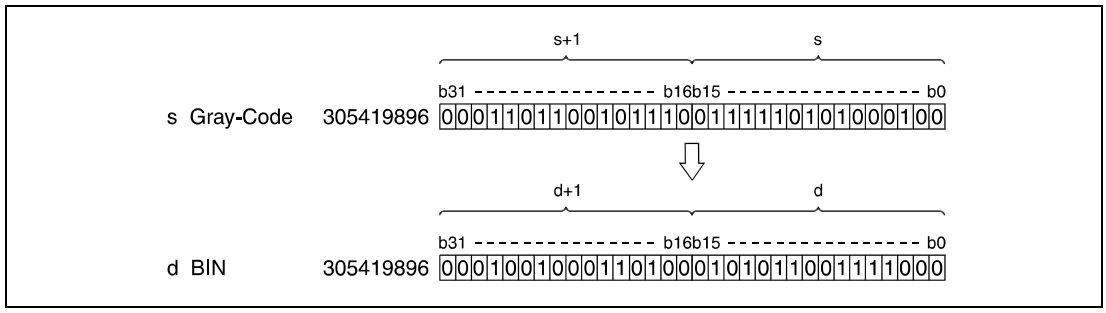
GBIN Konvertierung von Gray-Code-Daten in 16-Bit-Binärdaten

Die in s gespeicherten Gray-Code-Daten werden in 16-Bit-Binärdaten umgewandelt. Das Ergebnis wird in d gespeichert.



DGBIN Konvertierung von Gray-Code-Daten in 32-Bit-Binärdaten

Die in s gespeicherten Gray-Code-Daten werden in 32-Bit-Binärdaten umgewandelt. Das Ergebnis wird in d gespeichert.



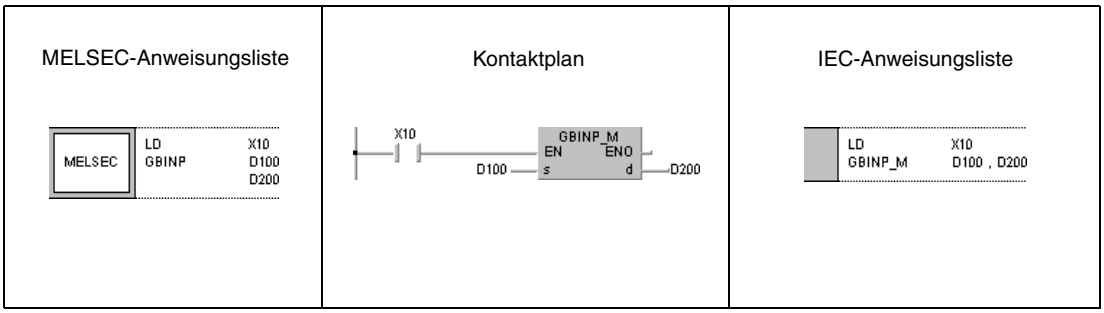
Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Der in s eingetragene Datenwert liegt bei Verwendung der GBIN-Anweisung außerhalb des Bereiches von 0 bis 32767.
- Der in s eingetragene Datenwert liegt bei Verwendung der DGBIN-Anweisung außerhalb des Bereiches von 0 bis 2147483647.

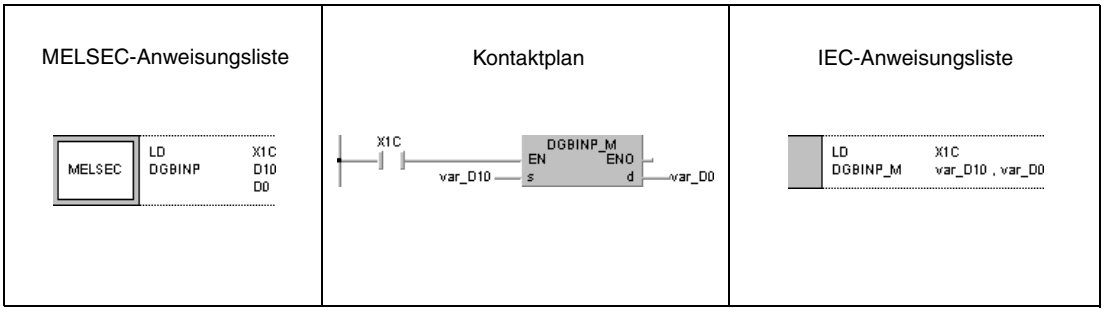
Beispiel 1 GBINP

Das folgende Programm dekodiert mit positiver Flanke von X10 die in D100 gespeicherten Gray- Code-Daten und speichert die umgewandelte 16-Bit-Binärzahl in D200 ab.



Beispiel 2 DGBINP

Das folgende Programm konvertiert mit positiver Flanke von X1C die in D10 und D11 gespeicherten Gray-Code-Daten und speichert die umgewandelte 32-Bit-Binärzahl in D0 und D1 ab.



HINWEIS *Das Programmbeispiel 2 ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- und Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser der Programmieranleitung zu entnehmen.*

6.3.9 NEG, NEGP, DNEG, DNEGP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
d		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●[2]						K1 ↓ K4	3 ●[1]	●		●

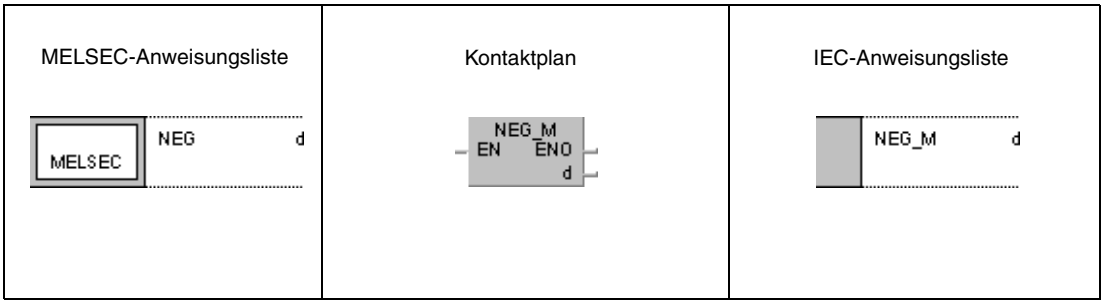
[1] Die Anzahl der Schritte bei Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

[2] Bei DNEG und DNEGP nicht gültig für AnN und AnS

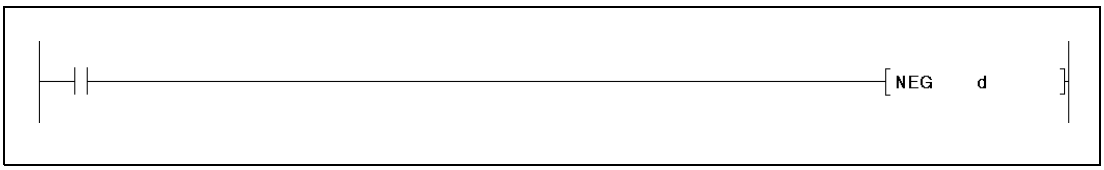
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sonder-module U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
d	●	●	●	●	●	●	●	—	—	—	2

GX IEC Developer



GX Developer



Variablen

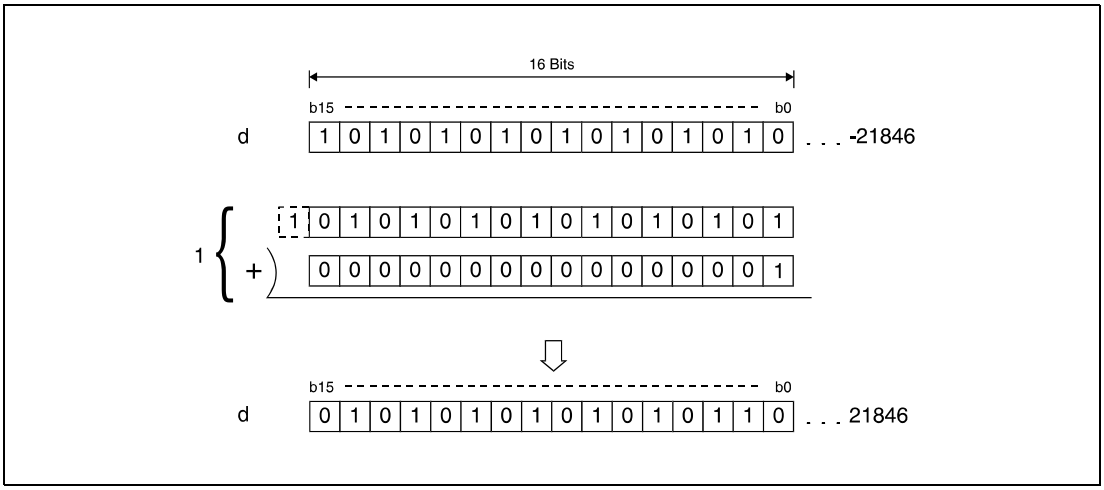
Operand	Befehlswert	Datentyp
d	Operand, in dem die Binärdaten, von denen das Zweierkomplement gebildet werden soll, gespeichert sind.	BIN-16-/32-Bit

Funktions-weise

Zweierkomplementbildung von 16-/32-Bit-Binärdaten (Vorzeichenumkehr)

NEG Negation von 16-Bit-Binärdaten

Die NEG-Funktion (Zweierkomplement oder auch NICHT-Logik genannt) negiert den Wert eines 16-Bit-Datenwortes. Dabei wird das 16-Bit-Datenwort in d zunächst invertiert und im nächsten Schritt der Wert "1" dazu addiert. Das Ergebnis (Zweierkomplement) wird wieder in d gespeichert.

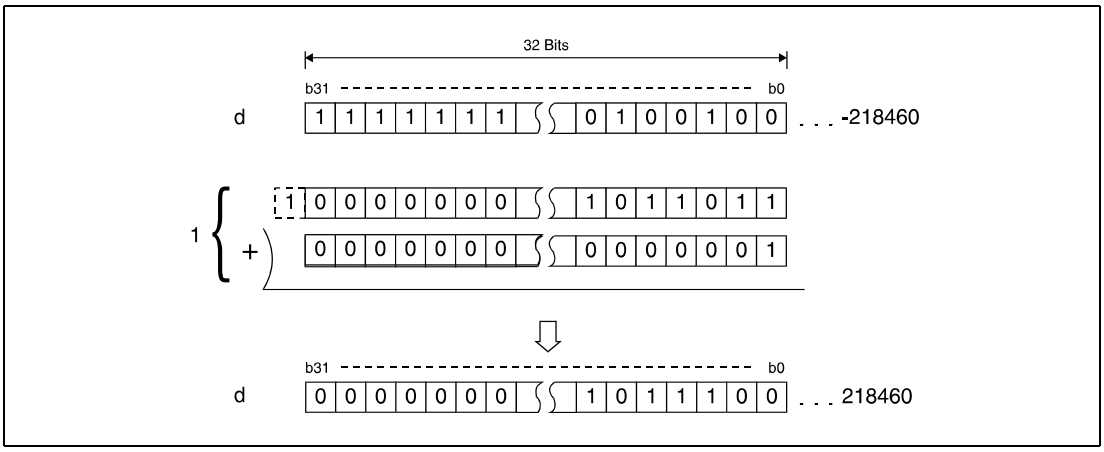


[1] Inversion mit anschließender Addition

Die Anweisung dient dazu, das negative Vorzeichen einer Zahl in ein positives Vorzeichen oder ein positives in ein negatives zu wandeln.

DNEG Negation von 32 Bit Binärdaten (Nur Q-Serie und System Q)

Die DNEG-Funktion (Zweierkomplement oder auch NICHT-Logik genannt) negiert den Wert eines 32-Bit-Datenwortes. Dabei wird das 32-Bit-Datenwort in d zunächst invertiert und im nächsten Schritt der Wert "1" addiert. Das Ergebnis (Zweierkomplement) wird wieder in d gespeichert.



[1] Inversion mit anschließender Addition

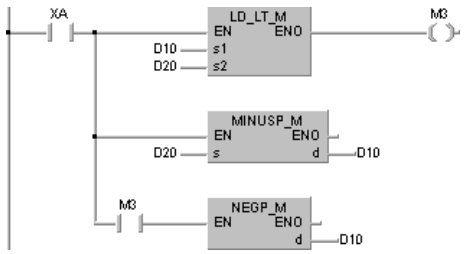
Beispiel NEGP

Das folgende Programm subtrahiert mit positiver Flanke von XA den Wert aus D10 von dem Wert aus D20. Zuvor wird M3 gesetzt, wenn D10 kleiner als D20 ist. Ist das Ergebnis aus "D10 - D20" negativ (M3 ist gesetzt), erlangt das Ergebnis in D10 absoluten Wert (Zweierkomplement) und wird positiv.

MELSEC-Anweisungsliste

```
MELSEC   LD     XA
         AND<   D10
                D20
         OUT    M3
         LD     XA
         -P     D20
                D10
         AND    M3
         NEGP   D10
```

Kontaktplan



IEC-Anweisungsliste

```
         LD       XA
         AND_LT_M D10 , D20
         ST       M3
         LD       XA
         MINUSP_M D20 , D10
         AND      M3
         NEGP_M   D10
```

6.3.10 ENEG, ENEGP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●[1]

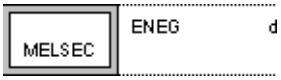
[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

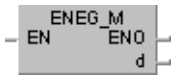
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
d	—	●	●	—	●	●	—	—	—	—	2

GX IEC Developer

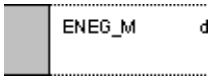
MELSEC-Anweisungsliste



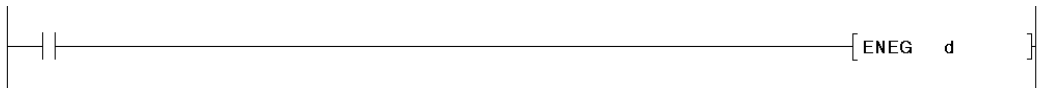
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

Operand	Befehlswert	Datentyp
d	Operand, in dem die Gleitkommazahl gespeichert ist, bei welcher die Vorzeichenumkehr vorgenommen werden soll.	Reelle Zahl

Funktions-weise

Vorzeichenumkehr bei Gleitkommazahlen

ENEG Negation von Gleitkommazahlen

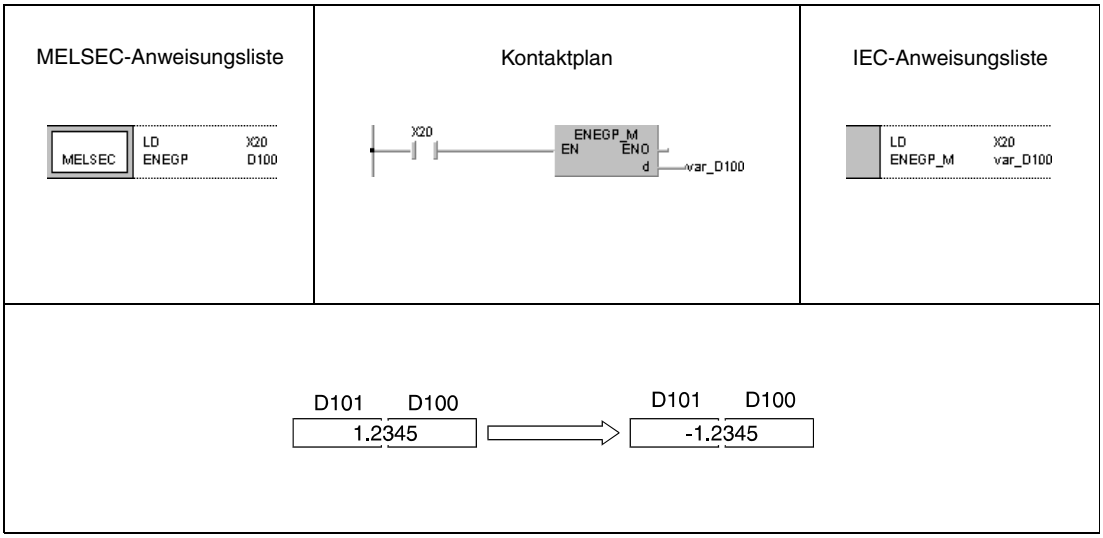
Diese Anweisung nimmt eine Negation der in d gespeicherten Gleitkommazahl vor. Das Ergebnis wird wieder in d gespeichert.

Die Anweisung dient dazu, das negative Vorzeichen einer Zahl in ein positives Vorzeichen oder ein positives in ein negatives zu wandeln.

Beispiel

ENEGP

Das folgende Programm negiert mit positiver Flanke von X20 die in D100 und D101 gespeicherte Gleitkommazahl und legt das Ergebnis wieder in D100 und D101 ab.



HINWEIS

Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationeinheit (POE) nicht lauffähig. Compiler- und Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.

6.3.11 BKBCD, BKBCDP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

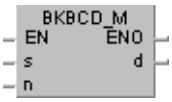
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	—	—	—	—	—	SM0	4
d	—	●	●	—	—	—	—	—	—		
n	●	●	●	●	●	●	●	●	—		

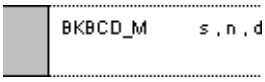
GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

Operand	Befehlswert	Datentyp
s	Operand, in dem die zu konvertierenden BIN-Datenblöcke gespeichert sind.	BIN-16-Bit
d	Operand, in dem die konvertierten BCD-Datenblöcke gespeichert werden.	
n	Anzahl der zu konvertierenden Datenblöcke.	

Funktions-weise

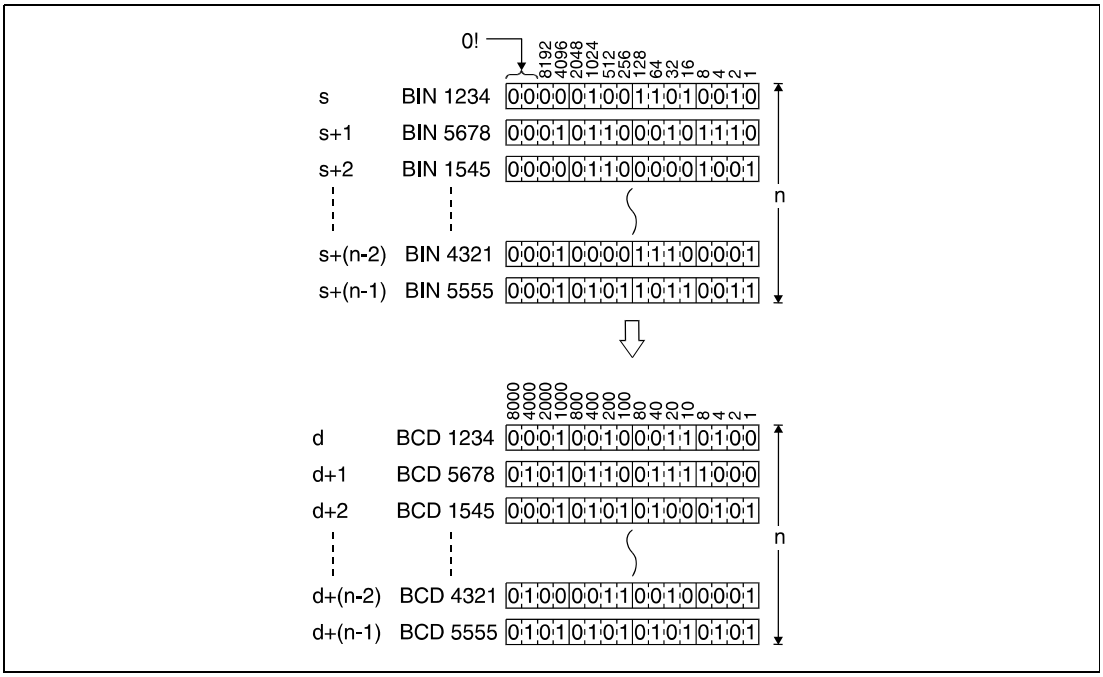
Blockweise Umwandlung von BIN-Daten in BCD-Daten

BKBCD Konvertierung von 16-Bit-BIN-Datenblöcken in 4-stellige BCD-Datenblöcke

Diese Anweisung konvertiert den jeweils n-ten 16-Bit-Block in s und speichert den entsprechenden umgewandelten n-ten 4-stelligen BCD-Datenblock in d.

Der Wert der in s angegebenen Binärdatenblöcke muss im Bereich zwischen 0 und 9999 liegen.

Die beiden höchstwertigen Bits der 16-Bit-Binärdatenblöcke müssen "0" lauten.



Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die in n angegebene Anzahl von Datenblöcken in s und d liegt außerhalb des für die Speicherung vorgesehenen Bereiches der Operanden (Fehlercode 4101).
- Die in s eingetragenen Binärdaten liegen außerhalb des Bereichs zwischen 0 und 9999 (Fehlercode 4100).
- Die Speicherbereiche von s und d überlappen (Fehlercode 4101).

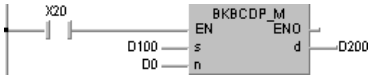
Informationen zur Verwendung der indizierten Adressierung enthält Kap. 3.6.

Beispiel BKBCDP

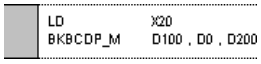
Das folgende Programm wandelt mit positiver Flanke von X20 die ab D100 gespeicherten 16-Bit-Binärdatenblöcke in 4-stellige BCD-Datenblöcke um. Die BCD-Datenblöcke werden ab D200 gespeichert. Die Anzahl der zu konvertierenden Datenblöcke (3) ist in D0 hinterlegt.

MELSEC-Anweisungsliste

Kontaktplan



IEC-Anweisungsliste



6.3.12 BKBIN, BKBINP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Operanden MELSEC Q

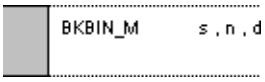
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	—	—	—	—	—	SM0	4
d	—	●	●	—	—	—	—	—	—		
n	●	●	●	●	●	●	●	●	—		

GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan

IEC-Anweisungsliste



GX Developer

Variablen

Operand	Befehlswert	Datentyp
s	Operand, in dem die zu konvertierenden BCD-Datenblöcke gespeichert sind.	BIN-16-Bit
d	Operand, in dem die konvertierten BIN-Datenblöcke gespeichert werden.	
n	Anzahl der zu konvertierenden Datenblöcke.	

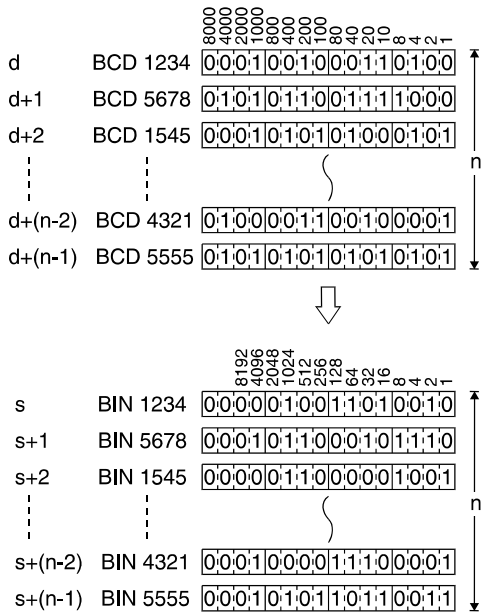
Funktions-weise

Blockweise Umwandlung von BCD-Daten in BIN-Daten

BKBIN, BKBINP **Konvertierung von 4-stelligen BCD-Datenblöcken in 16-Bit-BIN-Datenblöcken**

Diese Anweisung konvertiert den jeweils n-ten 4-stelligen BCD-Datenblock in s und speichert den entsprechenden umgewandelten n-ten 16-Bit-Block in d.

Der Wert der in s angegebenen BCD-Datenblöcke muss zwischen 0 und 9999 liegen.



Fehler-quellen

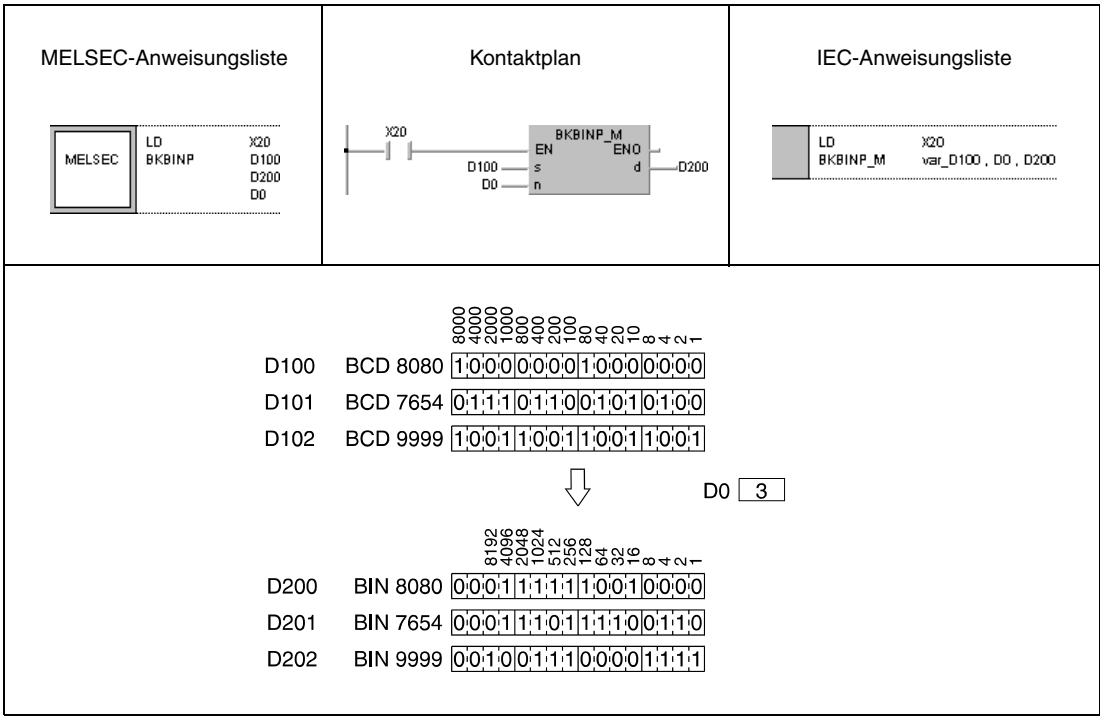
In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die in n angegebene Anzahl von Datenblöcken in s und d liegt außerhalb des für die Speicherung vorgesehenen Bereiches der Operanden (Fehlercode 4101).
- Die in s eingetragenen BCD-Daten liegen außerhalb des Bereichs zwischen 0 und 9999.
- Die Speicherbereiche von s und d überlappen.

Informationen zur Verwendung der indizierten Adressierung enthält Kap. 3.6 der Programmier-anleitung.

Beispiel BKBINP

Das folgende Programm wandelt mit positiver Flanke von X20 die ab D100 gespeicherten 4-stellige BCD-Datenblöcke in 16-Bit-Binärdatenblöcke um. Die Binärdatenblöcke werden ab D200 gespeichert. Die Anzahl der zu konvertierenden Datenblöcke (3) ist in D0 hinterlegt.



HINWEIS *Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationeinheit (POE) nicht lauffähig. Compiler- und Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

6.4 Transferanweisungen

Die Anweisungen zum Datentransfer ermöglichen es, Daten zu übertragen, auszutauschen oder zu invertieren. Es stehen insgesamt 24 verschiedene Anweisungen zur Verfügung. Eine Übersicht enthält die nachstehende Tabelle.

HINWEIS *Die mit Hilfe der Transferanweisungen übertragenen Daten bleiben so lange aktuell, bis sie durch neu übertragene Daten ersetzt werden. Aus diesem Grund bleiben die Daten auch dann erhalten, wenn die Eingangsbedingung der Transferanweisung wegfällt.*

Funktion	MELSEC-Anweisung im MELSEC-EDITOR	MELSEC-Anweisung im IEC-EDITOR
Übertragung (16-/32-Bit)	MOV	MOV_M
	MOVP	MOVP_M
	DMOV	DMOV_M
	DMOVP	DMOVP_M
Übertragung von Gleitkommazahlen	EMOV	EMOV_M
	EMOVP	EMOVP_M
Übertragung von Zeichenfolgen	$MOV	STRING_MOV_M
	$MOVP	STRING_MOVP_M
Inverse Übertragung (16-/32-Bit)	CML	CML_M
	CMLP	CMLP_M
	DCML	DCML_M
	DCMLP	DCMLP_M
Block-übertragung	BMOV	BMOV_M
	BMOVP	BMOVP_M
Blocküübertragung identischer Daten	FMOV	FMOV_M
	FMOVP	FMOVP_M
Austausch (16-/32-Bit)	XCH	XCH_M
	XCHP	XCHP_M
	DXCH	DXCH_M
	DXCHP	DXCHP_M
Austausch (16-Bit-8Blöcke)	BXCH	BXCH_M
	BXCHP	BXCHP_M
Austausch (höherwertiges und niedrigwertiges Byte)	SWAP	SWAP_MD
	SWAPP	SWAP_P_MD

HINWEIS *Nutzen Sie in den IEC-Editoren die IEC-Befehle.*

6.4.1 MOV, MOVP, DMOV, DMOVP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
MOV																										
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				K1 ↓ K4	5	●		●
d		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●							●[1]			
DMOV																										
s	●	●	●	●	●	●	●	●	●	●	●	●	●		●		●	●				K1 ↓ K8	7	●		●
d		●	●	●	●	●	●	●	●	●	●	●	●		●								●[1]			

[1] Die Anzahl der Schritte bei Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

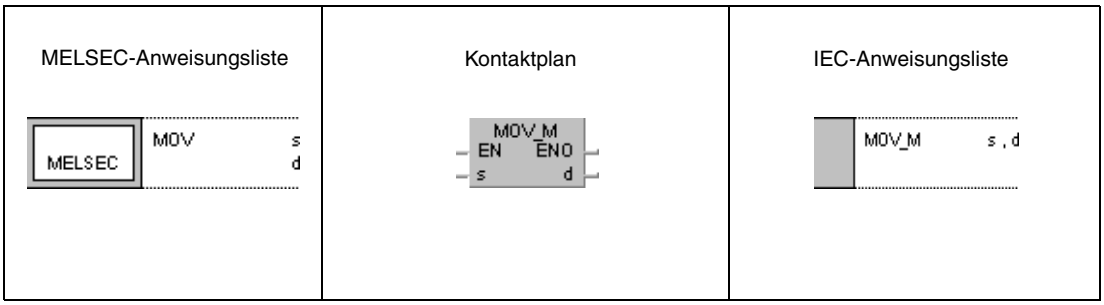
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sonder-module U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	●	●	●	●	●	●	●	●	—	—	3[1)]
d	●	●	●	●	●	●	●	—	—	—	3[2)]

[1] Die Anzahl der Programmschritte ist abhängig vom Typ der CPU und von den verwendeten Operanden.
Bei Verwendung einer QnA-CPU oder Single-Prozessor-Q-CPU: 3
Bei einer Multi-Prozessor Q-CPU und interner Wortoperanden (außer File-Register ZR) oder Konstanten: 2
Bei einer Multi-Prozessor-Q-CPU und Bit-Operanden, deren Adresse ein Vielfaches von16 ist, die die Bit-Blockbezeichnung 4 haben und die nicht durch Index-Vergabe bearbeitet werden: 2
Bei Verwendung einer Multi-Prozessor-Q-CPU und anderer Operanden als oben aufgeführt: 3

[2] Die Anzahl der Programmschritte ist abhängig vom Typ der CPU:
Bei einer QnA-CPU oder einer Multi-Prozessor-Q-CPU: 3
Bei einer Single-Prozessor-Q-CPU: 2

GX IEC Developer



GX Developer

Variablen

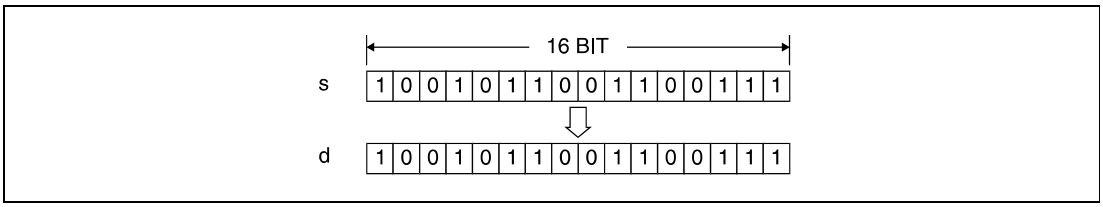
Operand	Befehlswert	Datentyp
s	Binäraten oder Operand, in dem die Binärdaten gespeichert sind.	BIN-16-/32-Bit
d	Operand, an welchen die Binärdaten übertragen werden.	

Funktionsweise

Datenübertragung von 16-/32-Bit-Binärdaten

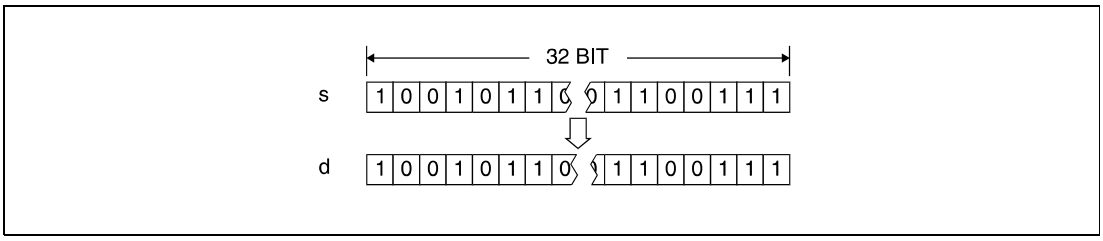
MOV Datenübertragung (16 Bit)

Die MOV-Anweisung überträgt die in s vorgegebenen 16-Bit-Daten an den Operanden in d.



DMOV Datenübertragung (32 Bit)

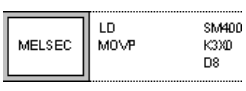
Die DMOV-Anweisung überträgt die in s vorgegebenen 32-Bit-Daten an den Operanden in d.



Beispiel 1 MOVP

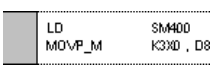
Das folgende Programm überträgt mit positiver Flanke von SM400 die Daten von X0 bis XB nach D8.

MELSEC-Anweisungsliste



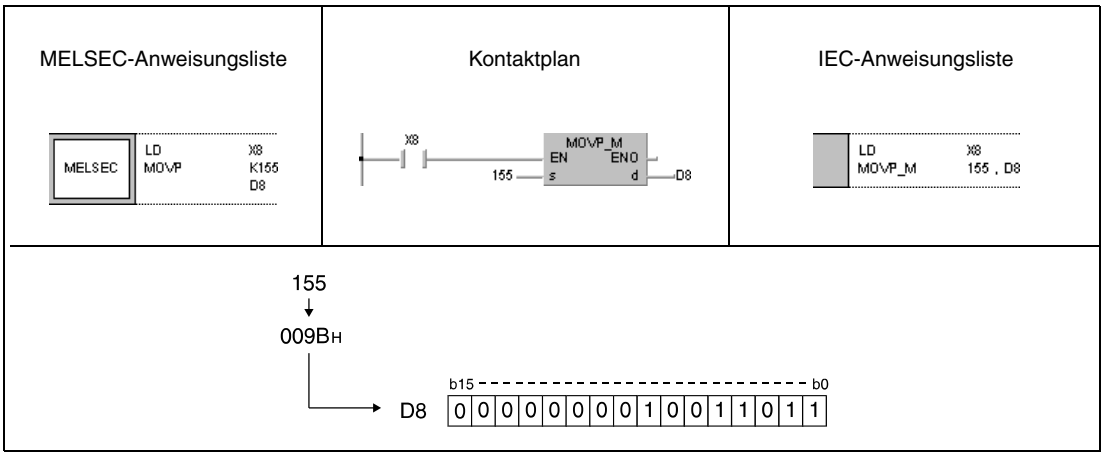
Kontaktplan

IEC-Anweisungsliste



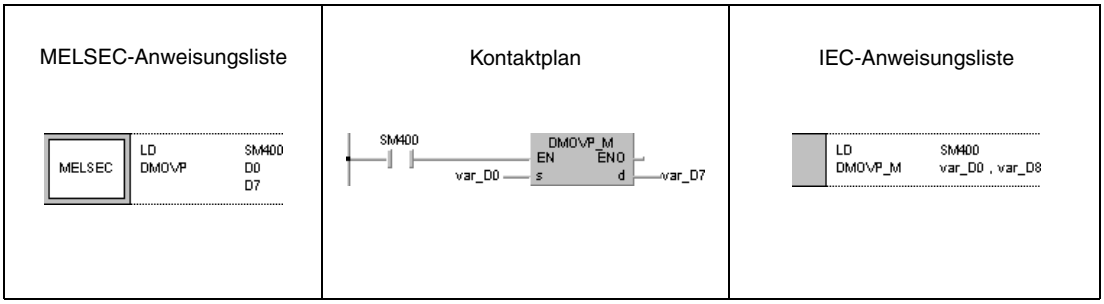
Beispiel 2 MOVP

Das folgende Programm schreibt mit positiver Flanke von X8 der Wert 155 in binärer Form in das Register D8.



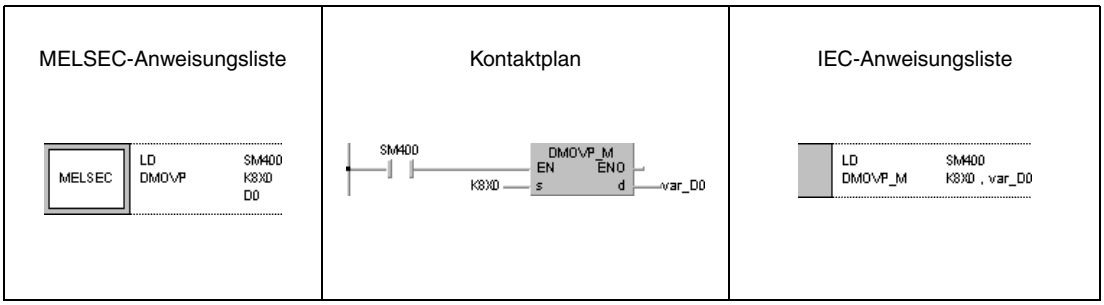
Beispiel 3 DMOVP

Das folgende Programm überträgt mit positiver Flanke von SM400 die Daten von D0 und D1 nach D7 und D8.



Beispiel 4 DMOVP

Das folgende Programm überträgt mit positiver Flanke von SM400 die Daten von X0 bis X1F nach D0 und D1.



HINWEIS *Die Programmbeispiele 3 und 4 sind ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- und Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

6.4.2 EMOV, EMOVP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●[1]

[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

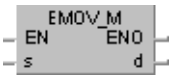
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten E	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	●	●	—	●	—	—	3
d	—	●	●	—	●	●	—	—	—	—	

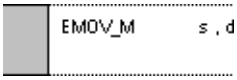
GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

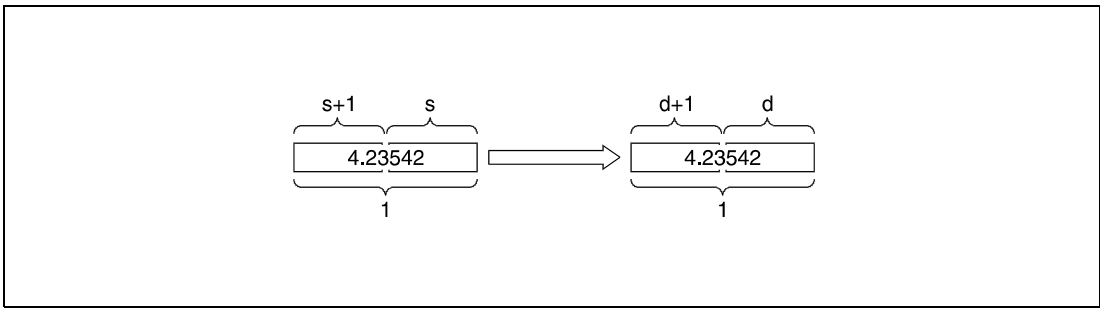
Operand	Befehlswert	Datentyp
s	Gleitkommazahl oder Operand, in dem die zu übertragende Gleitkommazahl gespeichert ist.	Reelle Zahl
d	Operand, an welchen die Gleitkommazahl übertragen wird.	

Funktions-weise

Übertragung von Gleitkommazahlen

EMOV/EMOVP Übertragung von Gleitkommazahlen

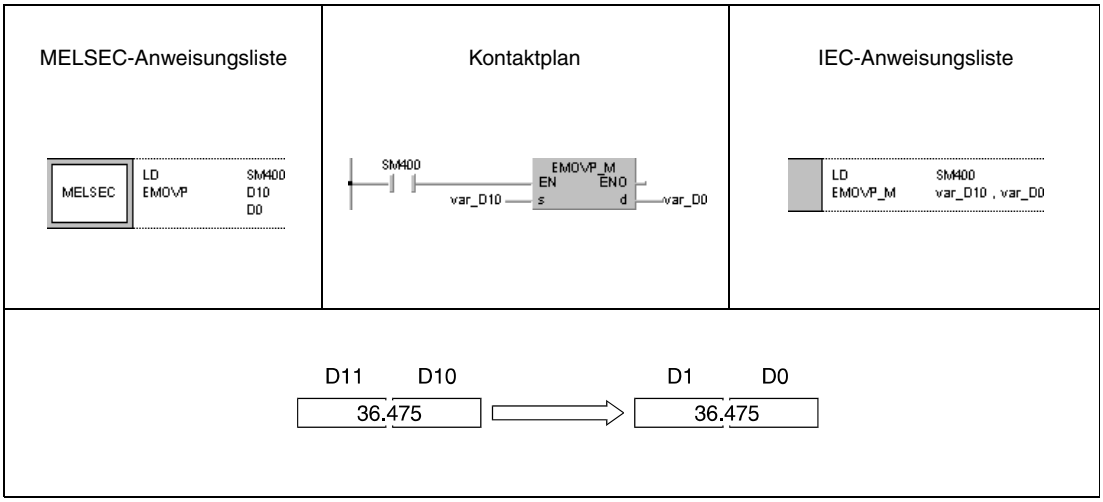
Die EMOV-Anweisung überträgt die in s angegebene Gleitkommazahl an den Operanden in d.



[1] Gleitkommazahl (reelle Zahl)

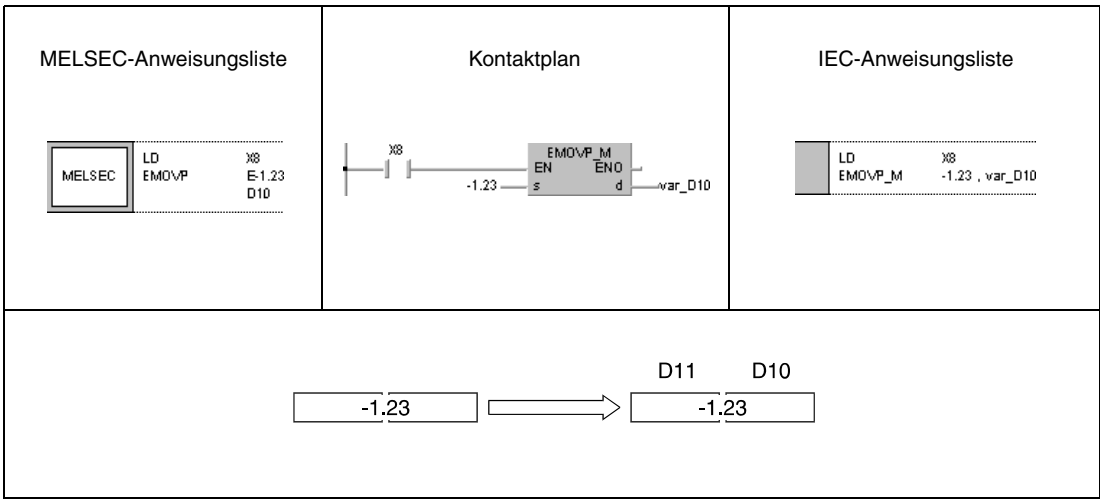
Beispiel 1 EMOVP

Das folgende Programm überträgt mit positiver Flanke von SM400 die Gleitkommazahl in D10 und D11 nach D0 und D1.



Beispiel 2 EMOVP

Das folgende Programm überträgt mit positiver Flanke von X8 die reelle Zahl -1,23 nach D10 und D11.



HINWEIS *Diese Programmbeispiele sind ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- und Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

6.4.3 $MOV, $MOVP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

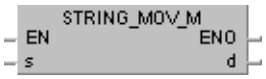
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten $	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	—	—	—	●	—	SM0	3
d	—	●	●	—	—	—	—	—	—		

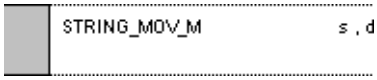
GX IEC Developer

MELSEC-Anweisungsliste

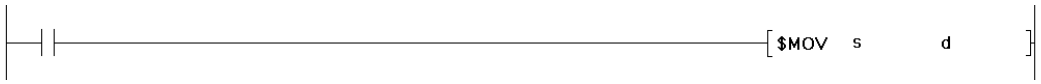
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

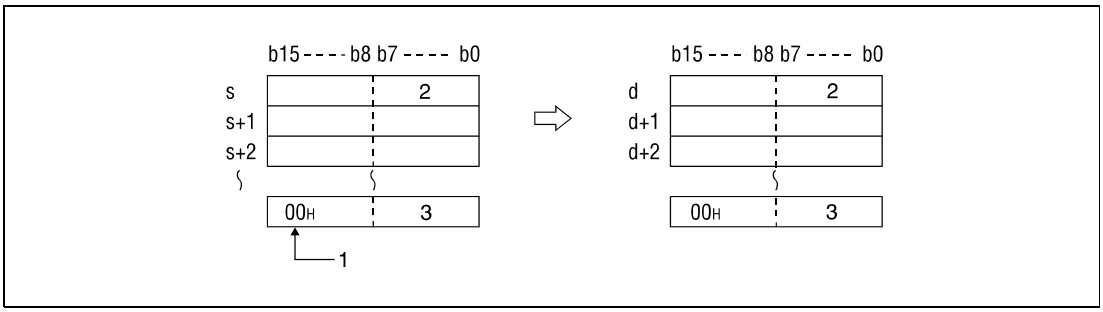
Operand	Befehlswert	Datentyp
s	Zeichenfolge oder Operand, in dem die zu übertragende Zeichenfolge gespeichert ist.	Zeichenfolge
d	Operand, an den die Zeichenfolge übertragen wird.	

Funktions-weise

Übertragung von Zeichenfolgen

$MOV Übertragung von Zeichenfolgen

Die $MOV-Anweisung überträgt die in s gespeicherten Bytes der Zeichenfolge nach d. Bei der Datenübertragung wird die gesamte Zeichenfolge, beginnend bei dem ersten Zeichen (Byte) bis zu dem mit "00H" beschriebenen Byte (Ende der Zeichenfolge) in einem Arbeitsgang übertragen.

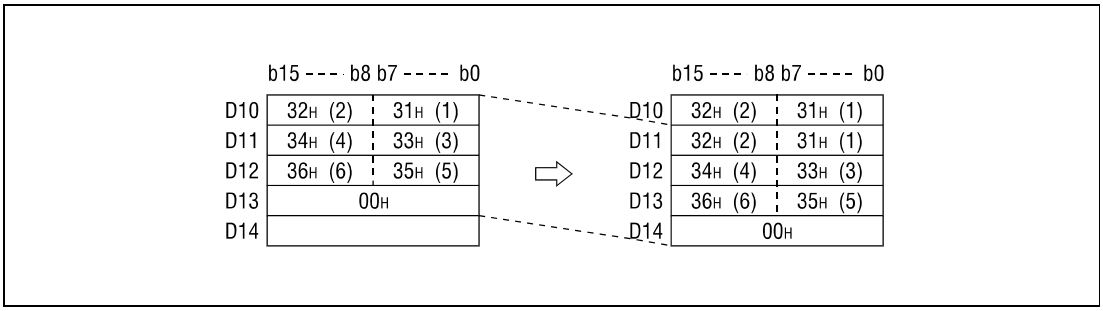


[1] Kennzeichnet das Ende der Zeichenfolge

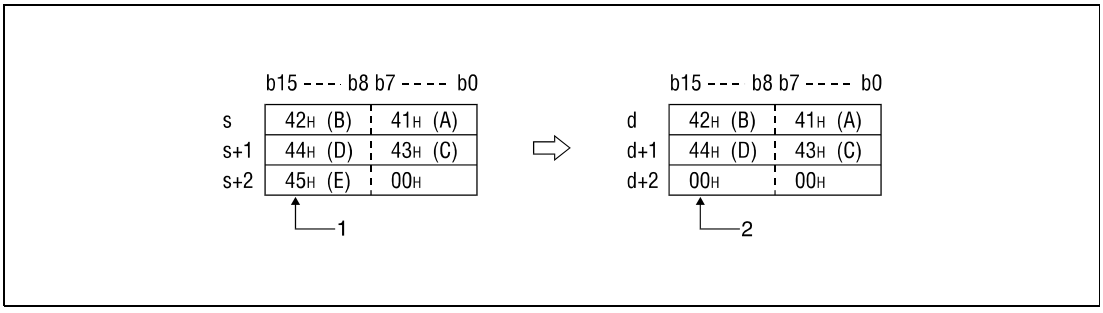
[2] 1. Zeichen

[3] n-tes Zeichen

Die Abarbeitung der $MOV-Anweisung wird selbst dann ohne Fehlermeldung ausgeführt, wenn die für die Speicherung vorgesehenen Datenbereiche von s bis s+n mit denen von d bis d+n überlappen. Das folgende Ergebnis tritt auf, wenn die in D10 bis D13 gespeicherte Zeichenfolge nach D11 bis D14 übertragen wird.



Befindet sich in der Zeichenfolge der Code "00H" in s+n vor dem Zeichen im höherwertigen Byte, so wird das darauffolgende Zeichen bei der Übertragung nicht berücksichtigt und das entsprechende Byte in d+n ebenfalls mit "00H" beschrieben.



[1] Dieses Zeichen wird nicht übertragen.

[2] Dieses Byte wird automatisch mit "00H" beschrieben.

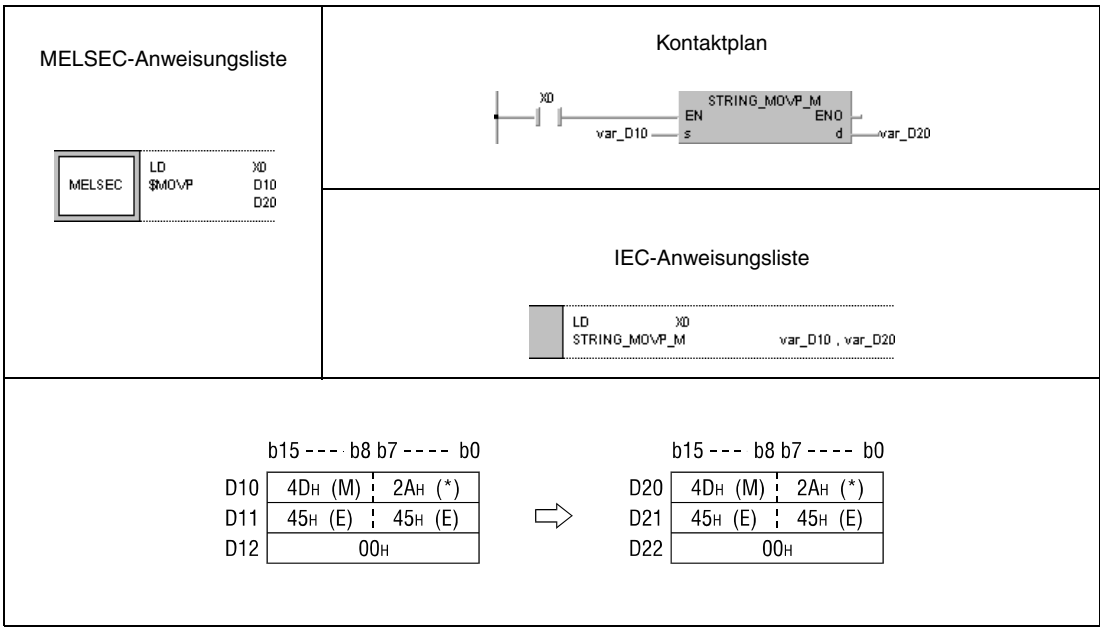
Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Der Code "00H" existiert nicht in der in s angebenen Zeichenfolge (Fehlercode 4101).
- Die vollständige Zeichenfolge kann nicht nach d übertragen werden.

Beispiel

Das folgende Programm überträgt mit positiver Flanke von X0 die in D10 bis D12 abgelegte Zeichenfolge nach D20 bis D22.



HINWEIS

Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationeinheit (POE) nicht lauffähig. Compiler- und Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.

6.4.4 CML, CMLP, DCML, DCMLP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
CML																										
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				K1 ↓ K4	5 ●[1]	●		●
d		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●										
DCML																										
s	●	●	●	●	●	●	●	●	●	●	●	●	●		●		●	●				K1 ↓ K8	7 ●[1]	●		●
d		●	●	●	●	●	●	●	●	●	●	●	●		●											

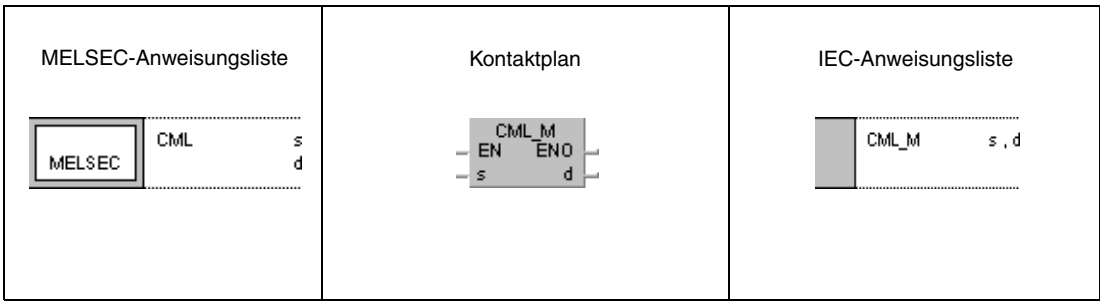
[1] Die Anzahl der Schritte bei Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

Operanden MELSEC Q

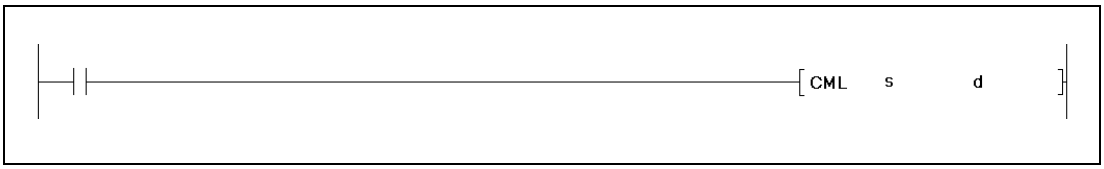
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	●	●	●	●	●	●	●	●	—	—	3[1)]
d	●	●	●	●	●	●	●	—	—	—	

[1] Die Anzahl der Programmschritte ist abhängig vom Typ der CPU und von den verwendeten Operanden.
Bei Verwendung einer QnA-CPU oder Single-Prozessor-Q-CPU: 3
Bei einer Multi-Prozessor-Q-CPU und interner Wortoperanden (außer File-Register ZR) oder Konstanten: 2
Einer Multi-Prozessor-Q-CPU und Bit-Operanden, deren Adresse ein Vielfaches von16 ist, die die Bit-Blockbezeichnung 4 haben und die nicht durch Index-Vergabe bearbeitet werden: 2
Bei Verwendung einer Multi-Prozessor-Q-CPU und anderer Operanden als oben aufgeführt: 3

GX IEC Developer



GX Developer



Variablen

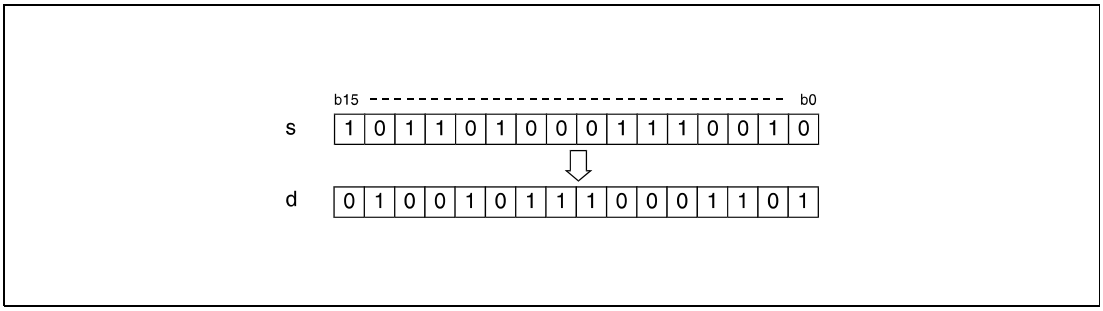
Operand	Befehlswert	Datentyp
s	Binärdaten oder Operand, in dem die zu invertierenden Binärdaten gespeichert sind.	BIN-16-/32-Bit
d	Operand, in dem die invertierten Binärdaten gespeichert werden.	

Funktionsweise

Inversion von 16-/32-Bit-Binärdaten

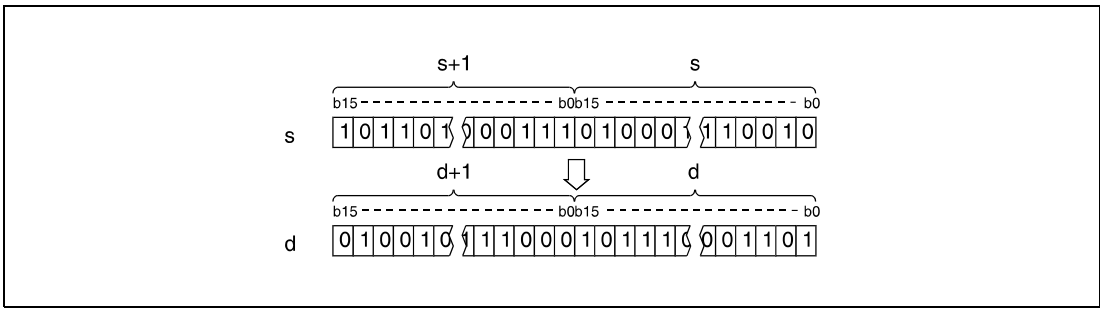
CML Dateninversion (16 Bit)

Der in der CML-Anweisung angegebene 16-Bit-Datenwert in s wird bitweise negiert. Das Ergebnis wird nach d übertragen.



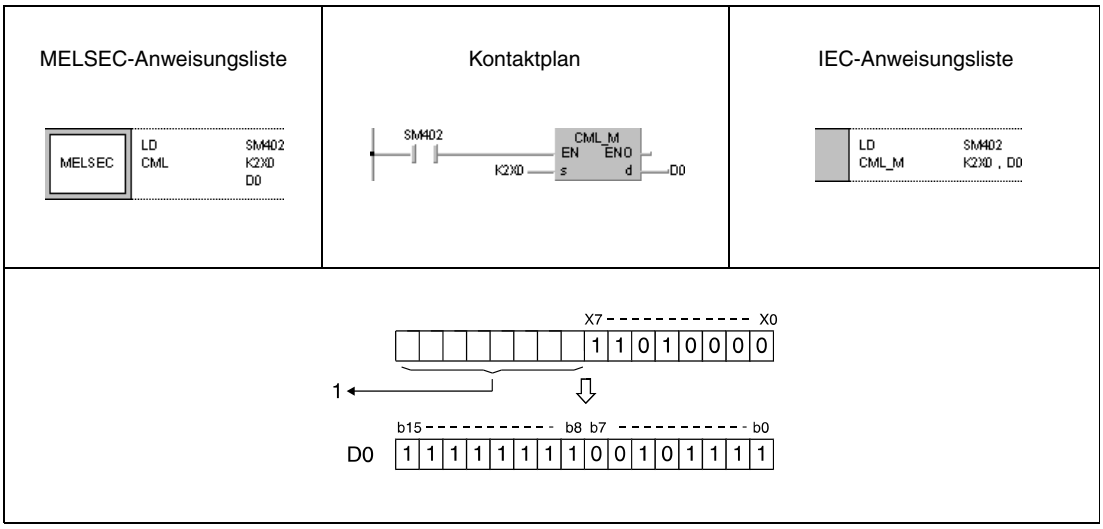
DCML Dateninversion (32 Bit)

Der in der DCML-Anweisung angegebene 32-Bit-Datenwert in s wird bitweise negiert. Das Ergebnis wird nach d übertragen.



Beispiel 1 CML

Das folgende Programm überträgt für die Einschaltdauer von SM402 die Daten X0 bis X7 negiert nach D0.

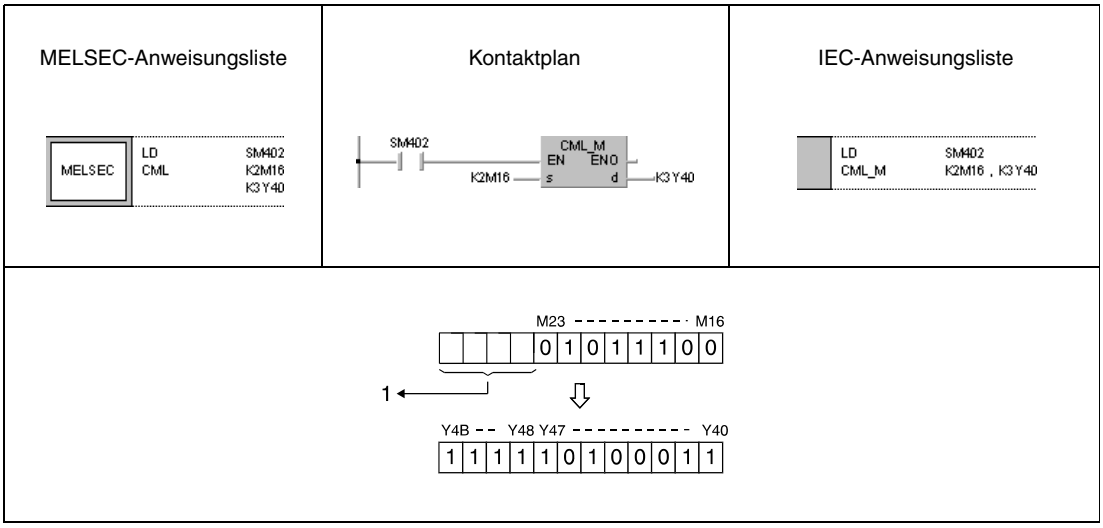


[1] Nicht beschriebene Bits werden als "0" eingelesen.

Beachten Sie , dass die Anzahl der Bits in s kleiner als die Anzahl der Bits in d sein muss.

Beispiel 2 CML

Das folgende Programm überträgt für die Einschaltdauer von SM402 die Daten von M16 bis M23 negiert nach K3 Y40 (Y40 bis Y4F). Y48 bis Y4B werden auf "1" gesetzt, da sie als "0" eingelesen werden.

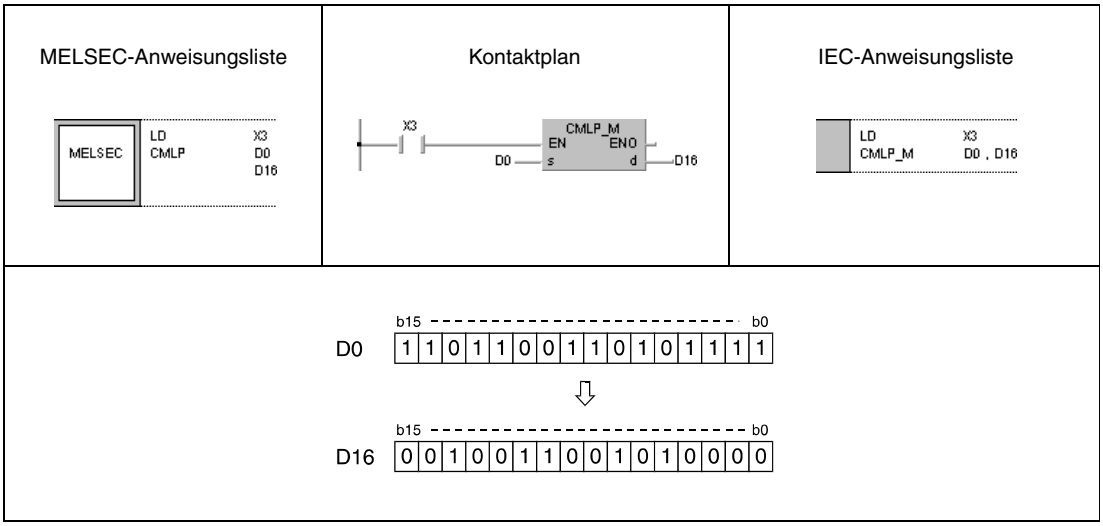


[1] Nicht beschriebene Bits werden als "0" eingelesen.

Die Anzahl der Bits in s muss kleiner sein als die Anzahl der Bits in d.

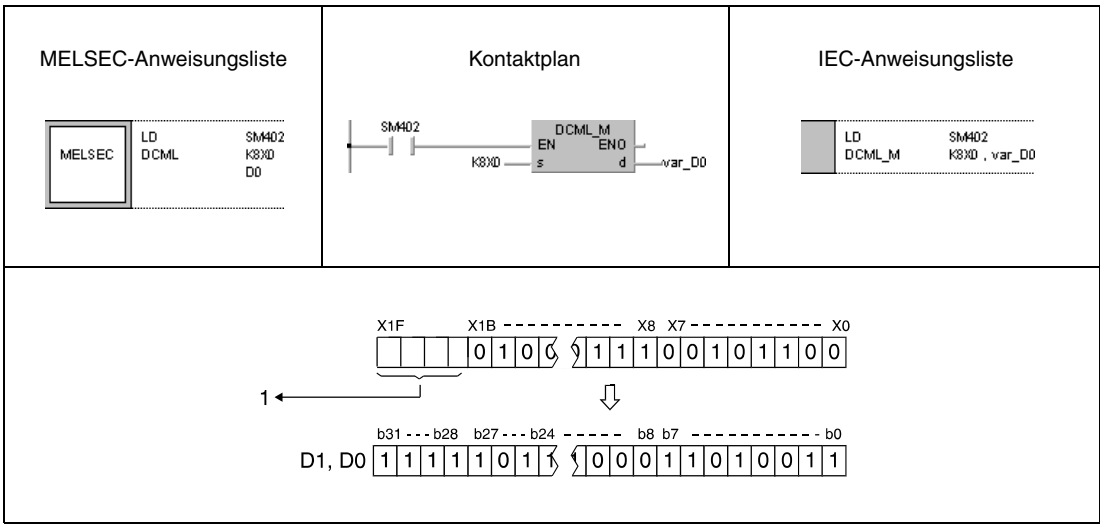
Beispiel 3 CMLP

Das folgende Programm überträgt mit positiver Flanke von X3 die Daten aus D0 negiert nach D16.



Beispiel 4 DCML

Das folgende Programm überträgt für die Einschaltdauer von SM402 die Daten von X0 bis X1FF negiert nach D0 und D1.

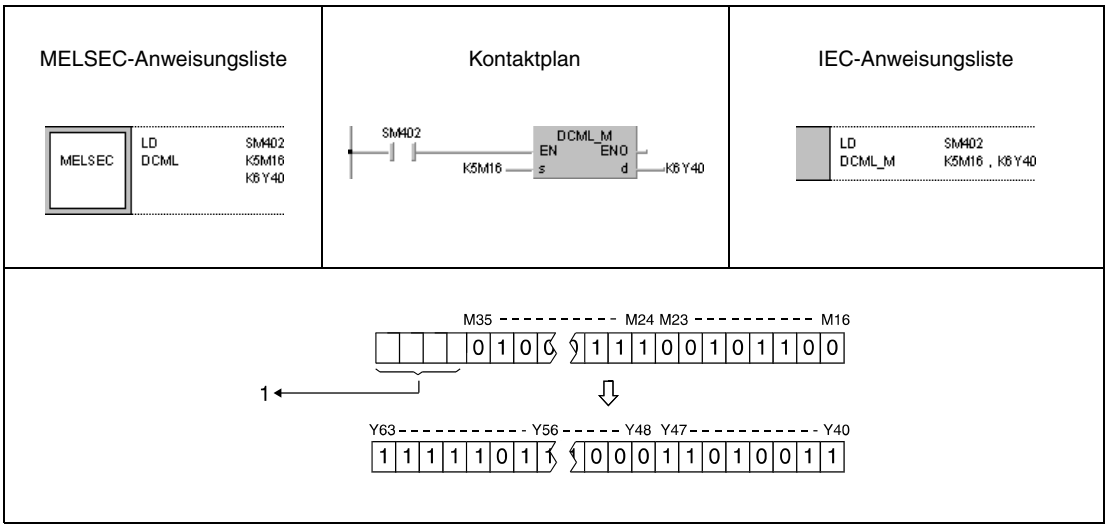


[1] Nicht beschriebene Bits werden als "0" eingelesen.

Beachten Sie, dass die Anzahl der Bits in s kleiner als die Anzahl der Bits in d sein muss.

Beispiel 5 DCML

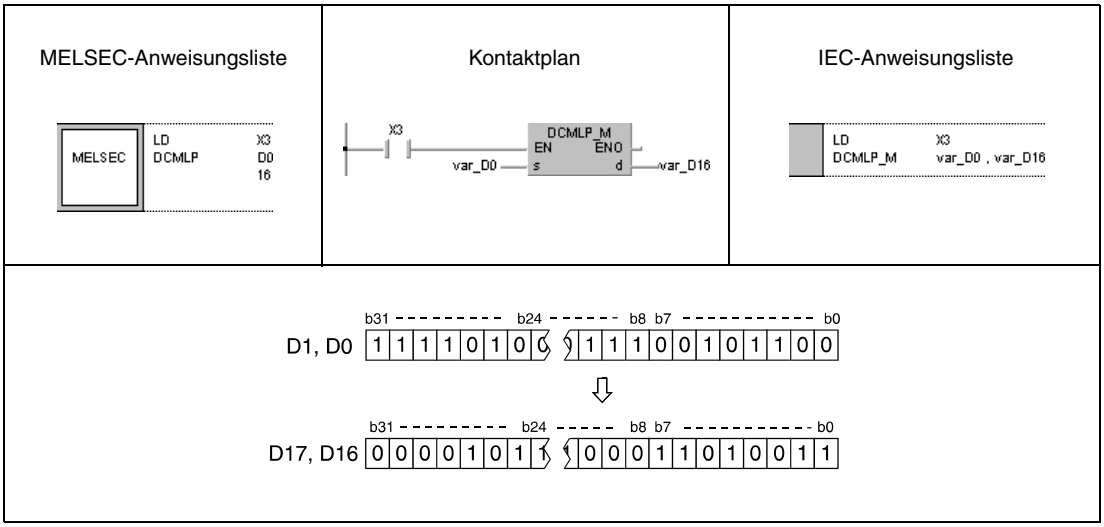
Das folgende Programm überträgt für die Einschaltdauer von SM402 die Daten von M16 bis M35 negiert nach Y40 und Y57.



[1] Nicht beschriebene Bits werden als "0" eingelesen.

Beispiel 6 DCMLP

Das folgende Programm überträgt mit positiver Flanke von X3 die Daten aus D0 und D1 negiert und nach D16 und D17.



HINWEIS *Die Anzahl der Bits in s muss kleiner sein als die Anzahl der Bits in d.*

Die Programmbeispiele 4 und 6 sind ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- und Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.

6.4.5 BMOV, BMOVP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

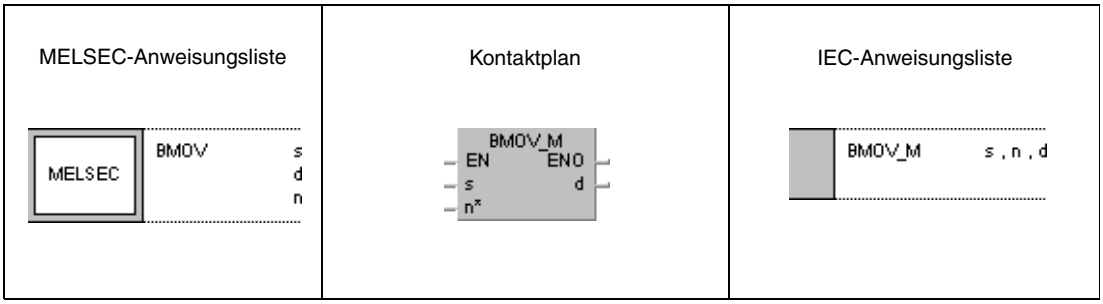
	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
s	●	●	●	●	●	●	●	●	●	●	●	●														
d		●	●	●	●	●	●	●	●	●	●	●										K1 ↓ K4	9 ●[1]	●		●
n																	●	●								

[1] Die Anzahl der Schritte bei Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sonder-module U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	●	●	●	●	●	●	—	—	—	SM0	4
d	●	●	●	●	●	●	—	—	—		
n	●	●	●	●	●	●	●	●	—		

GX IEC Developer



GX Developer

Variablen

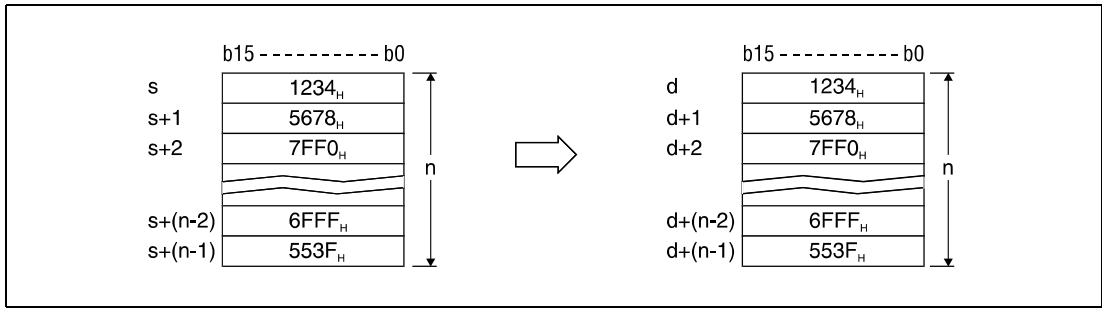
Operand	Befehlswert	Datentyp
s	Operand, in dem die zu übertragenden Daten gespeichert sind.	BIN-16-Bit
d	Operand, an den die Daten übertragen werden.	
n	Anzahl der zu übertragenden Datenblöcke.	

Funktions-weise

Übertragung von Binärdatenblöcken

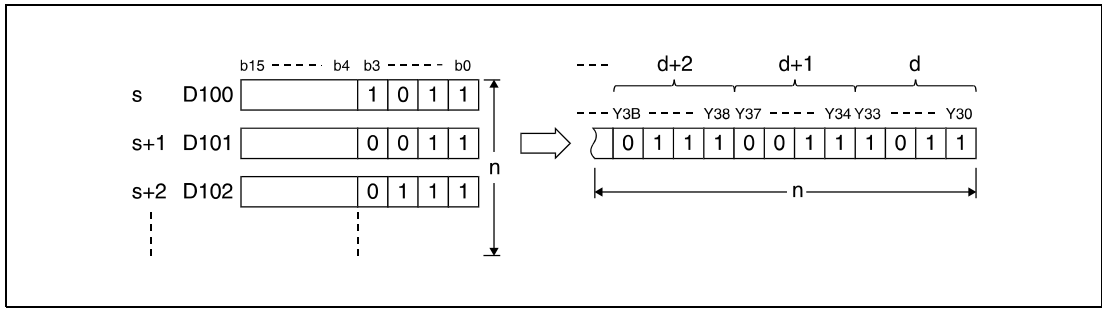
BMOV Blockweise Datenübertragung (16 Bit)

Mit Hilfe der BMOV-Anweisung kann ein Block von aufeinanderfolgenden Adressen gleichzeitig übertragen werden. In s wird die erste zu übertragende Adresse festgelegt. Der Wert in "n" gibt die Anzahl der aufeinanderfolgenden Adressen an. Die Daten werden in Blöcken zu "n" Adressen an die Zieladresse beginnend mit d übertragen.



Eine Datenübertragung ist auch dann möglich, wenn Quelle und Ziel die gleichen Adressen enthalten. Die Übertragung zu den Operanden mit der kleineren Adresse beginnt mit s und die Übertragung zu den Operanden mit der höheren Adresse s+(n-1).

Ist s ein Wort-Operand und d ein Bit-Operand, werden die über die Bit-Bestimmung angegebenen Stellen des Wort-Operanden an den Bit-Operanden übertragen. Wird zum Beispiel K1Y30 für d gesetzt, werden die niedrigstwertigen 4 Bits der über s festgelegten Wort-Operanden übertragen.



Handelt es sich bei den Adressen um Bit-Operanden, muss die Anzahl der Bits in d und s identisch sein.

Fehler-quellen

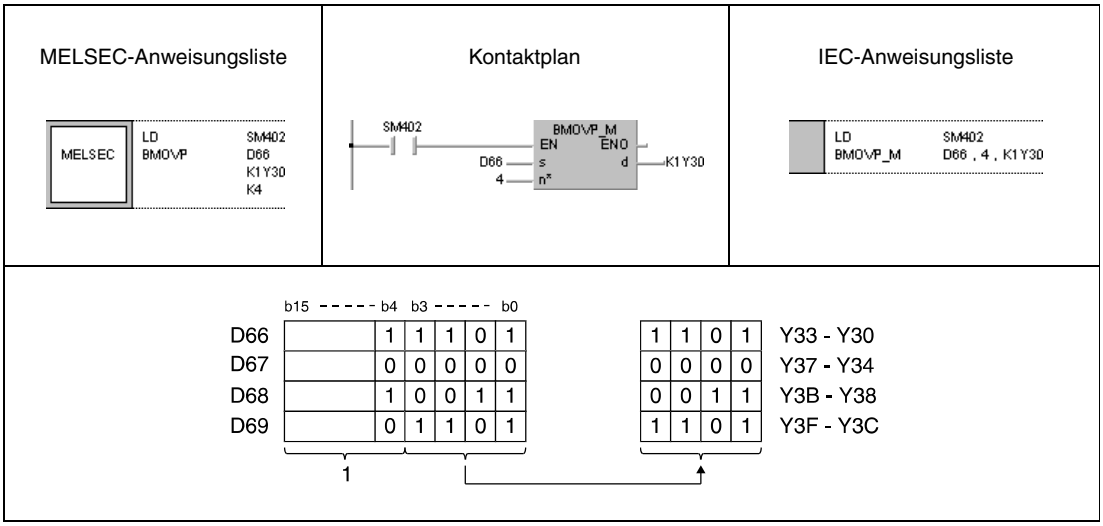
In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die in n angegebene Anzahl von Datenblöcken in s und d liegt außerhalb des für die Speicherung vorgesehenen Bereiches der Operanden
  (Q-Serie/System Q = Fehlercode 4101).

Beispiel 1 BMOVP

Das folgende Programm überträgt mit positiver Flanke von SM402 die Daten der untersten 4 Bits (von b0 bis b3) von D66 bis D69 an die Ausgänge Y30 bis Y3F. Die Anzahl der zu übertragenden Blöcke (4) ist durch die Konstante K4 angegeben.

Die Bit-Schemen zeigen die Bit-Struktur jeweils vor und nach der Übertragung.

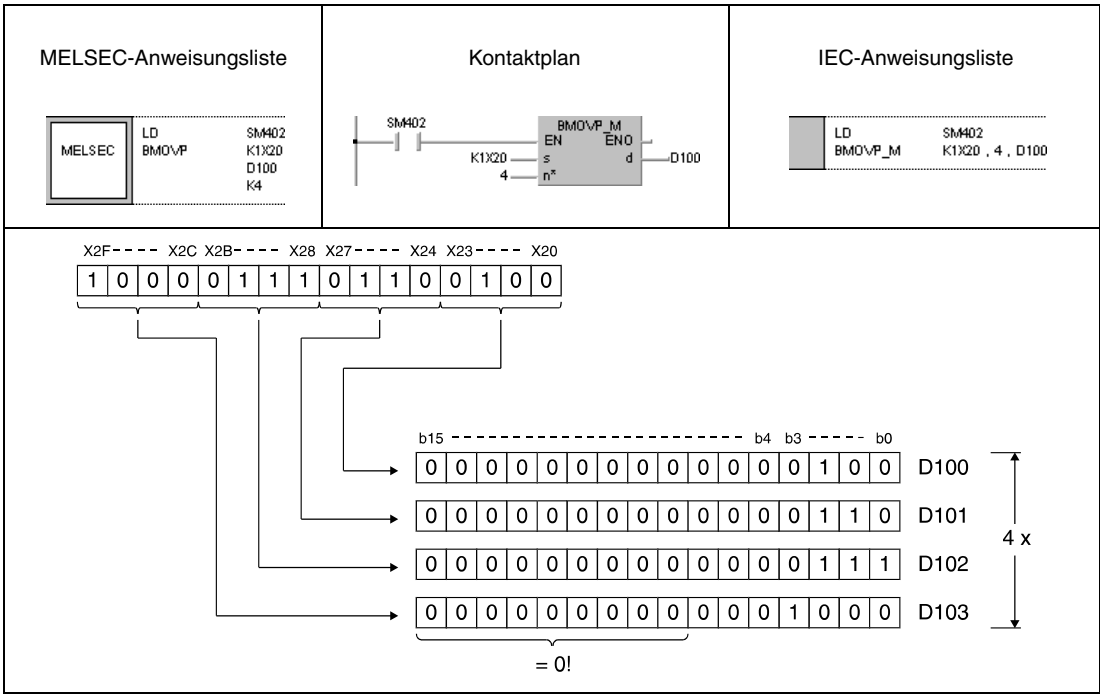


[1] Diese Bits bleiben bei der Operation unberücksichtigt.

Beispiel 2 BMOVP

Das folgende Programm überträgt mit positiver Flanke von SM402 die Daten von X20 bis X2F an die Register D100 bis D103. Die Anzahl der Datenblöcke (4) ist durch die Konstante K4 angegeben.

Die Bit-Schemen zeigen die Bit-Struktur jeweils vor und nach der Übertragung.



6.4.6 FMOV, FMOVP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

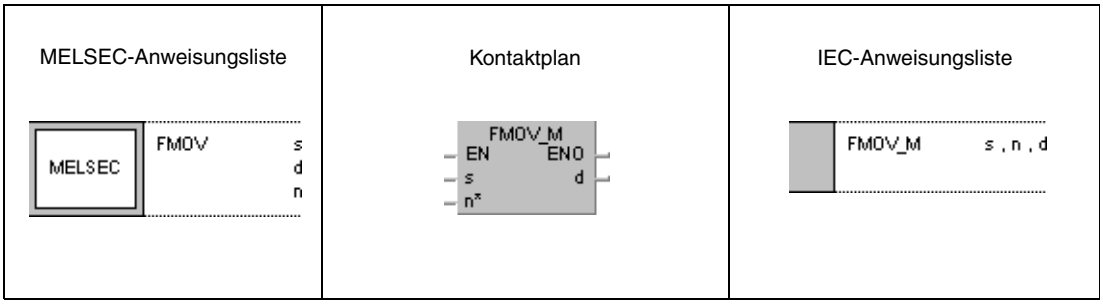
	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				K1 ↓ K4	9 ●[1]	●		●
d		●	●	●	●	●	●	●	●	●	●	●														
n																	●	●								

[1] Die Anzahl der Schritte bei Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□N□		Sonder-module U□G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	●	●	●	●	●	●	●	●	—	SM0	4
d	●	●	●	●	●	●	—	—	—		
n	●	●	●	●	●	●	●	●	—		

GX IEC Developer



GX Developer

Variablen

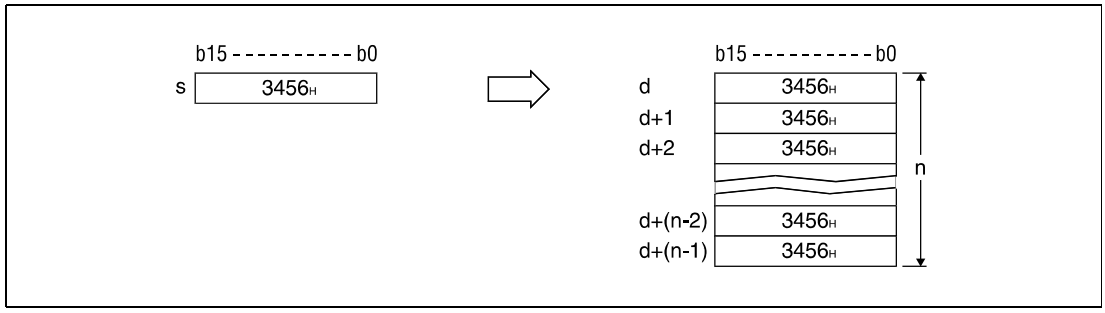
Operand	Befehlswert	Datentyp
s	Daten oder Operanden, in dem die zu übertragenden Daten gespeichert sind.	BIN-16-Bit
d	Operand, an den die Daten übertragen werden.	
n	Anzahl der Übertragungen.	

Funktions-weise

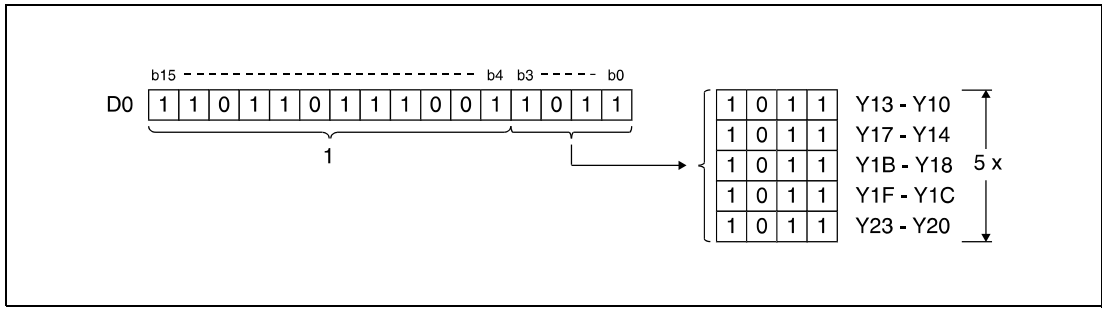
Übertragung eines Binärdatenblocks

FMOV Übertragung eines identischen Datenblocks (16-Bit)

Mit Hilfe der FMOV-Anweisung können die Daten des Operanden s in einen Datenbereich beginnend mit dem in d festgelegten Operanden bis zum Operanden d+(n-1) übertragen werden. Dadurch wird im Datenblock von d bis d+(n-1) jeder Operand auf den Wert des Operanden s gesetzt.



Ist s ein Wort-Operand und d ein Bit-Operand werden die über die Bit-Bestimmung angegebenen Stellen des Wort-Operanden an den Bit-Operanden übertragen.



[1] Diese Bits werden bei der Verarbeitung nicht berücksichtigt

Handelt es sich bei den Adressen um Bit-Operanden, muss die Anzahl der Bits in d und s identisch sein.

Fehler-quellen

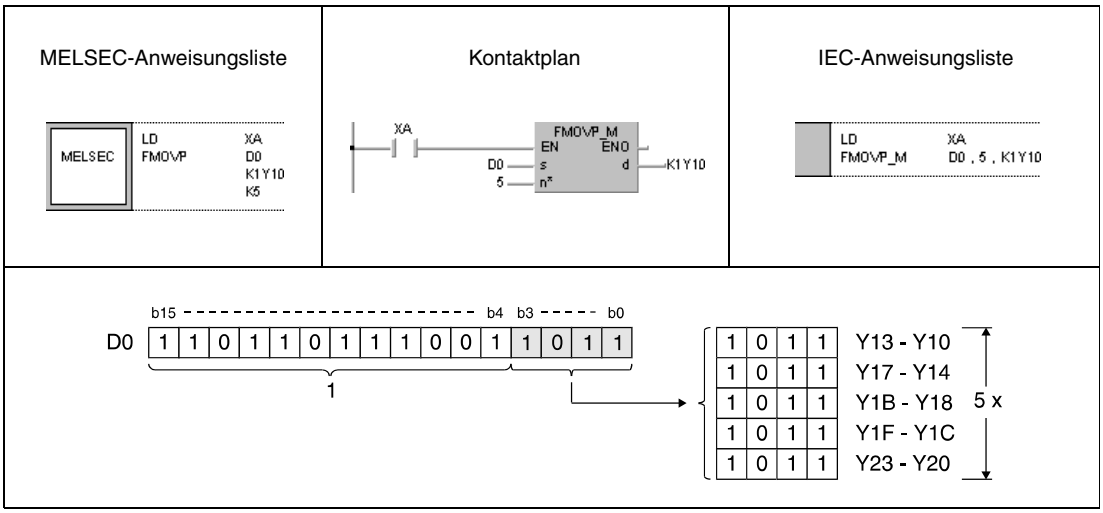
In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die in n angegebene Anzahl von Datenblöcken in s und d liegt außerhalb des für die Speicherung vorgesehenen Bereiches der Operanden (Q-Serie/System Q = Fehlercode 4101).

Beispiel 1 FMOVP

Das folgende Programm überträgt mit positiver Flanke von XA die Daten der untersten 4 Bits (b0 – b3) von D0 an die Ausgänge Y10 bis Y23. Die Anzahl der Blöcke (5) gibt die Konstante K5 an.

Die Bit-Schemen zeigen die Bit-Struktur jeweils vor und nach der Übertragung.

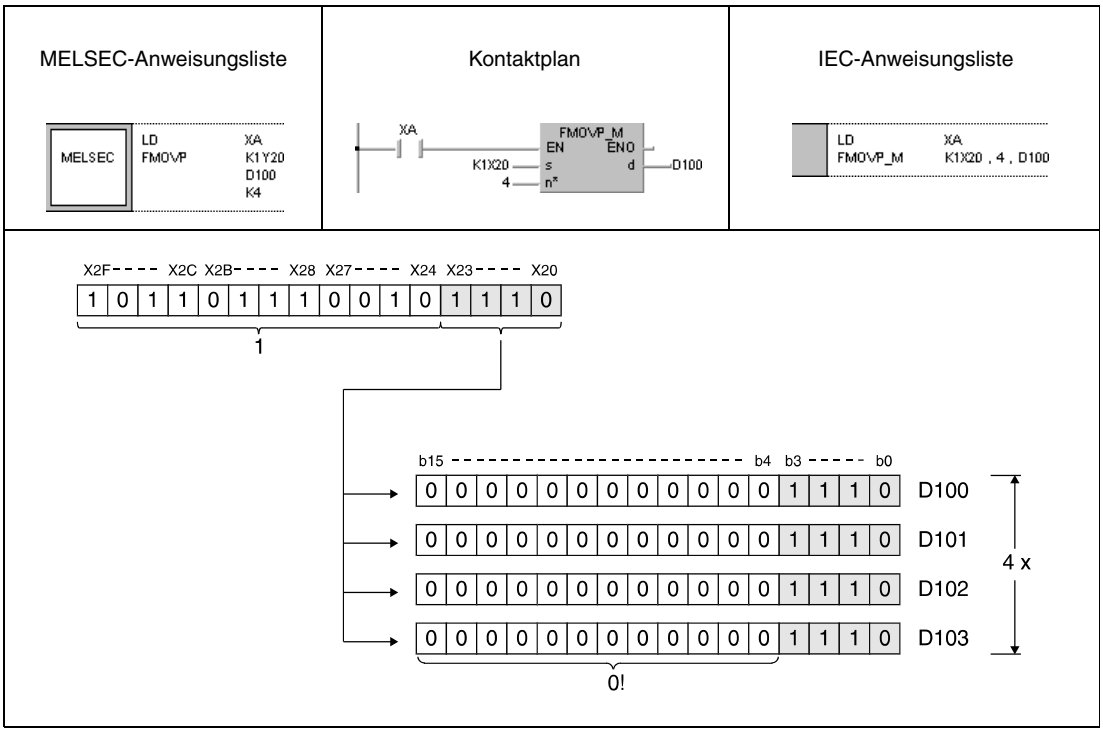


[1] Diese Bits werden bei der Verarbeitung nicht berücksichtigt.

Beispiel 2 FMOVP

Das folgenden Programm überträgt mit positiver Flanke von XA die Daten von X20 bis X23 in die Datenregister D100 bis D103 . Die Anzahl der Blöcke (4) gibt die Konstante K4 an.

Die Bit-Schemen zeigen die Bit-Struktur jeweils vor und nach der Übertragung.



[1] Diese Bits werden bei der Verarbeitung nicht berücksichtigt.

6.4.7 XCH, XCHP, DXCH, DXCHP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

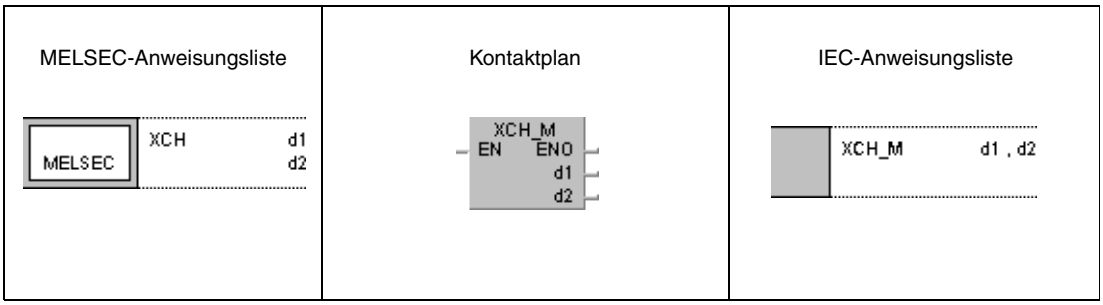
	Operanden																					Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
XCH																										
d1		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●						K1 ↓ K4	5 ●[1]	●		●
d2		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●										
DXCH																										
d1		●	●	●	●	●	●	●	●	●	●	●	●		●							K1 ↓ K8	7 ●[1]	●		●
d2		●	●	●	●	●	●	●	●	●	●	●	●		●											

[1] Die Anzahl der Schritte bei Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

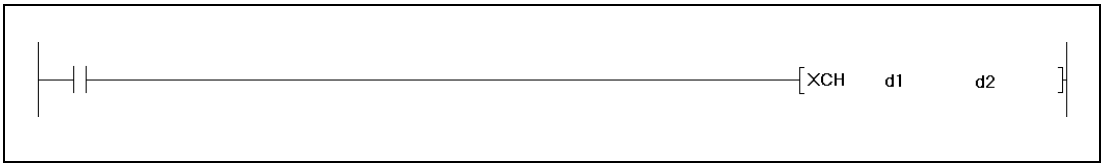
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
d1	●	●	●	●	●	●	●	—	—	—	3
d2	●	●	●	●	●	●	●	—	—	—	

GX IEC Developer



GX Developer



Variablen

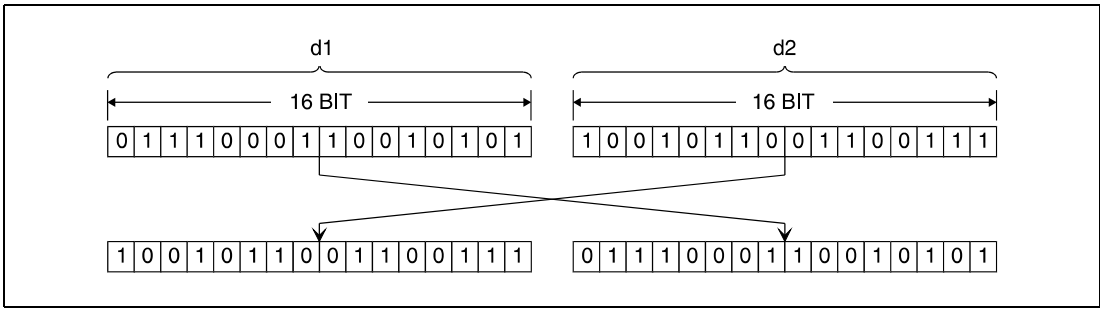
Operand	Befehlswert	Datentyp
d1	Operanden, in denen die zu tauschenden Daten gespeichert sind.	BIN-16-/32-Bit
d2		

Funktions-weise

Austausch von Binärdaten

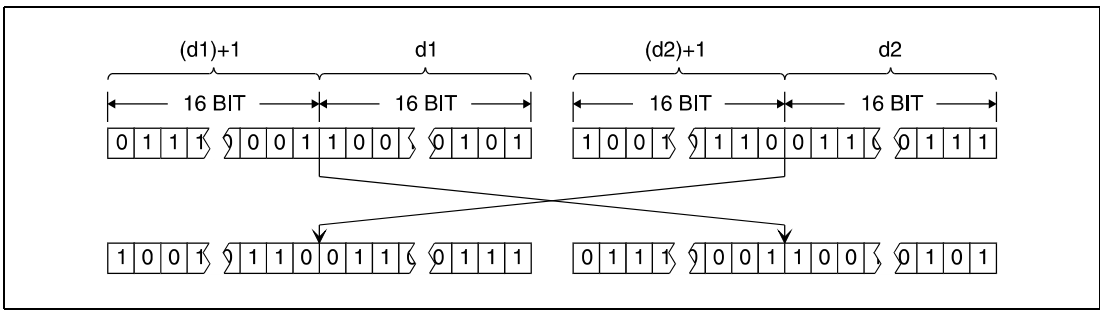
XCH Datenaustausch (16 Bit)

Die XCH-Anweisung (e**XCH**ange; austauschen) tauscht die 16-Bit-Daten von d1 und d2 gegeneinander aus.



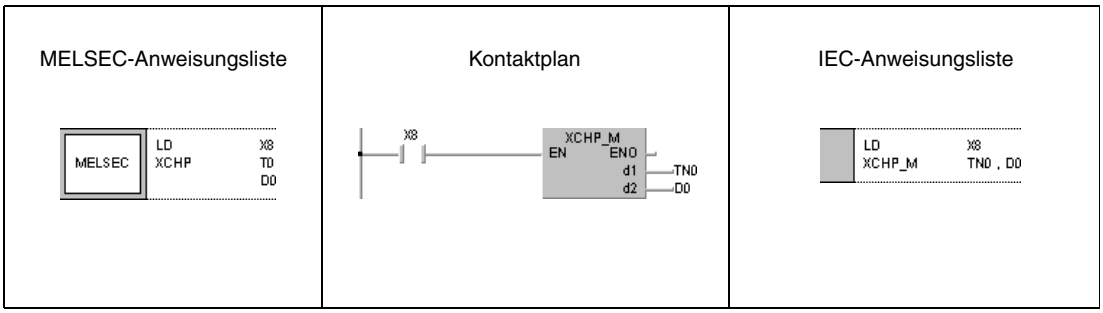
DXCH Datenaustausch (32 Bit)

Die DXCH-Anweisung tauscht die 32-Bit-Daten von (d1)+1, d1 und (d2)+1, d2 gegeneinander aus.



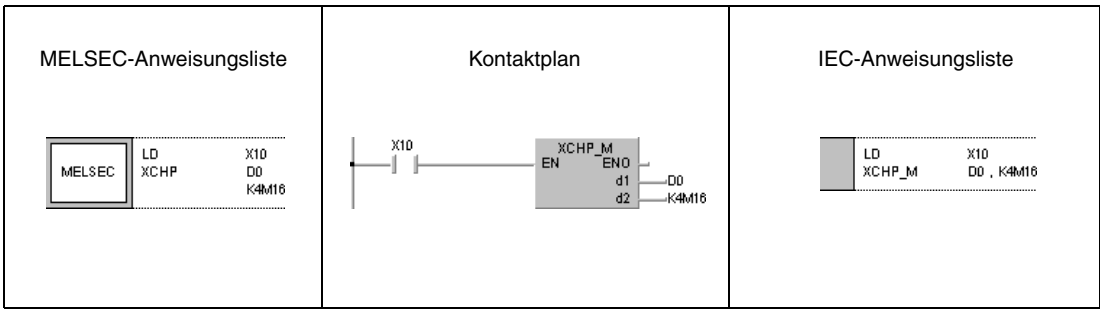
Beispiel 1 XCHP

Das folgende Programm tauscht mit positiver Flanke von X8 wird der Inhalt von D0 mit dem Istwert von T0 aus.



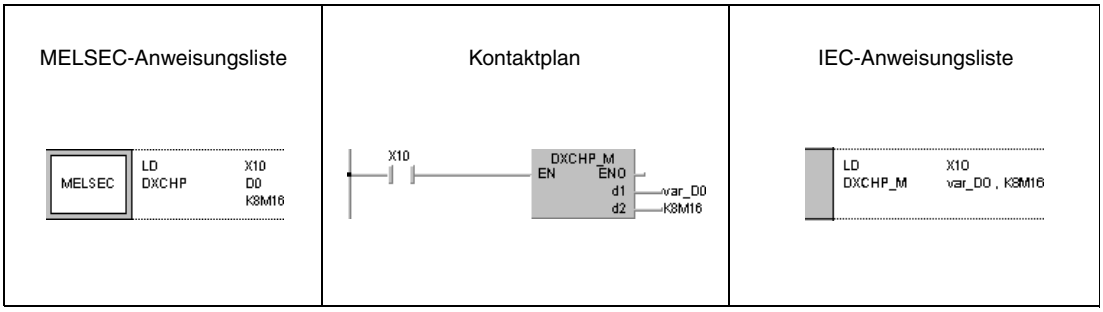
Beispiel 2 XCHP

Das folgende Programm tauscht mit positiver Flanke von X10 den Inhalt von D0 mit dem Inhalt von M16 bis M31 aus.



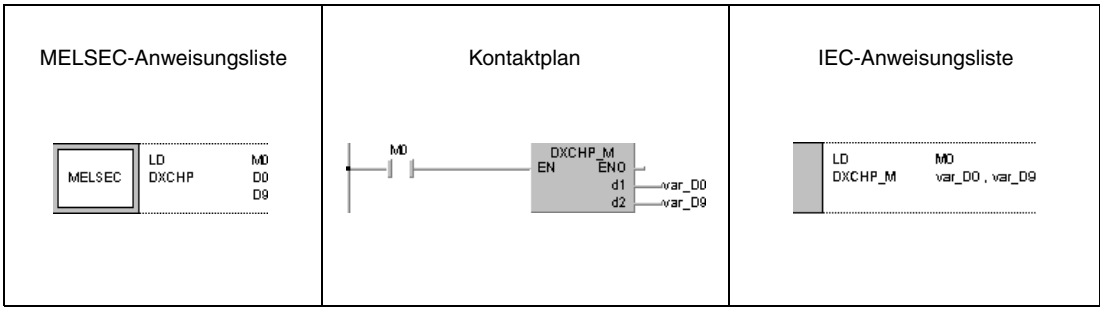
Beispiel 3 DXCHP

Das folgende Programm tauscht mit positiver Flanke von X10 den Inhalt von D0 und D1 mit dem Inhalt von M16 bis M47 aus.



Beispiel 4 DXCHP

Das folgende Programm tauscht mit positiver Flanke von M0 den Inhalt von D0 und D1 mit dem Inhalt von D9 und D10 aus.



HINWEIS *Die Programmbeispiele 3 und 4 sind ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- und Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

6.4.8 BXCH, BXCHP

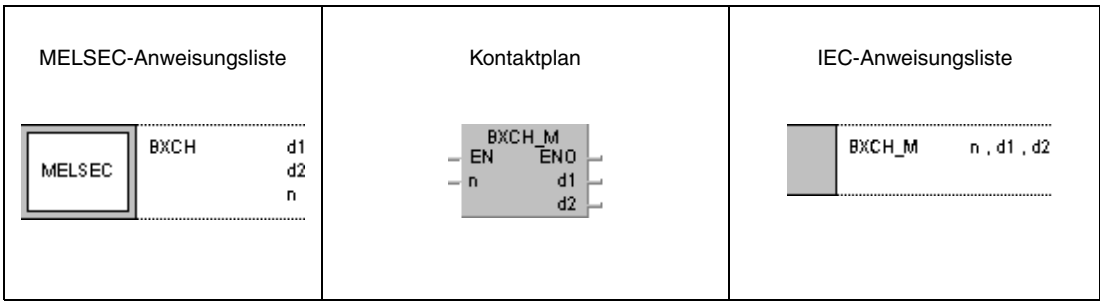
CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
d1	—	●	●	—	—	—	—	—	—	SM0	4
d2	—	●	●	—	—	—	—	—	—		
n	●	●	●	●	●	●	●	●	—		

GX IEC Developer



GX Developer

Variablen

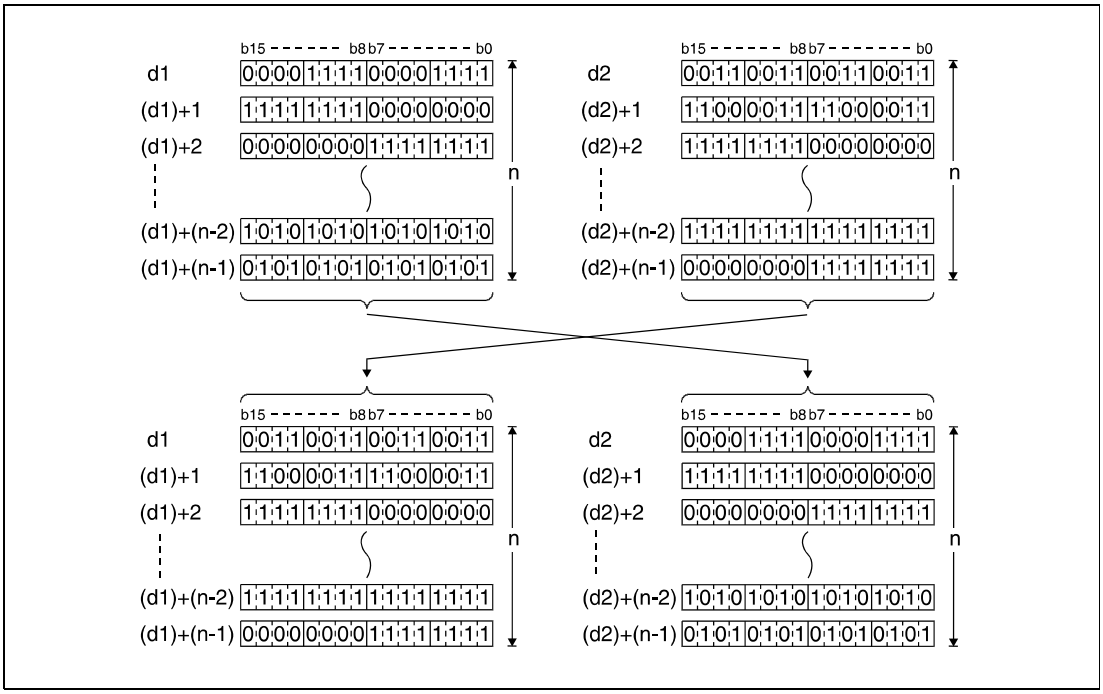
Operand	Befehlswert	Datentyp
d1	Operanden, in denen die zu tauschenden Daten gespeichert sind.	BIN-16-Bit
d2		
n	Anzahl der auszutauschenden Datenblöcke.	

Funktions-weise

Austausch von Binärdatenblöcken

BXCH Blockweiser Datenaustausch (16 Bit)

Die BXCH-Anweisung (e**XCH**ange; austauschen) tauscht die 16-Bit-Daten der Blöcke von d1 und d2 gegeneinander aus.



Fehler-quellen

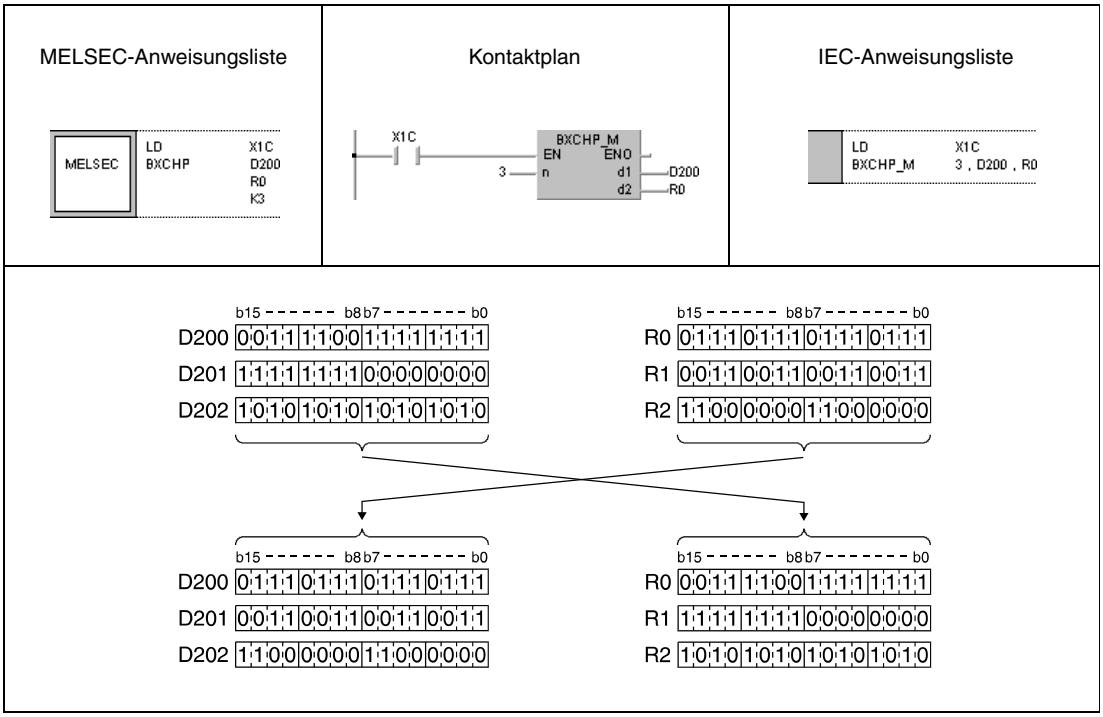
In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die in n angegebene Anzahl von Datenblöcken in d1 und d2 liegt außerhalb des für die Speicherung vorgesehenen Bereiches der Operanden (Q-Serie/System Q = Fehlercode 4101).
- Die Speicherbereiche von d1 und d2 überlappen (Fehlercode 4101).

Beispiel BXCHP

Das folgende Programm tauscht mit positiver Flanke von X1C die Datenblöcke beginnend bei D200 mit den Datenblöcken beginnend bei R0 aus. Die Anzahl der zu tauschenden Binärdatenblöcke (3) gibt die Konstante K3 an.

Die Bit-Schemen zeigen die Bit-Struktur jeweils vor und nach der Übertragung.



6.4.9 SWAP, SWAPP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●

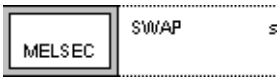
[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R Anweisungen programmiert werden.

Operanden MELSEC Q

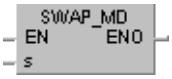
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	●	●	●	●	●	●	●	—	—	—	3

GX IEC Developer

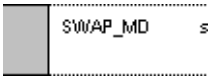
MELSEC-Anweisungsliste



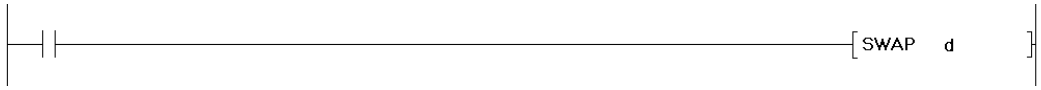
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

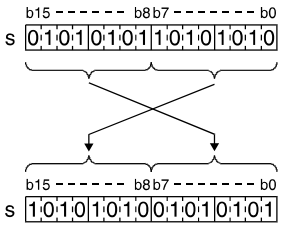
Operand	Befehlswert	Datentyp
s	Operand, in dem die Daten gespeichert sind.	BIN-16-Bit

Funktions-weise

Austausch der Bytes eines Wortes

SWAP Austausch des höherwertigen und des niedrigwertigen Bytes

Die SWAP-Anweisung tauscht die höherwertigen 8 Bits (höherwertiges Byte) mit den niedrigwertigen 8 Bits (niedrigwertiges Byte) der in s gespeicherten Wortes.



Beispiel

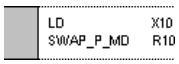
SWAPP

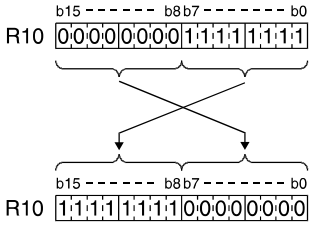
Das folgende Programm tauscht mit positiver Flanke von X10 die oberen 8 Bits mit den unteren 8 Bits des Operanden R10 aus.

MELSEC-Anweisungsliste

Kontaktplan

IEC-Anweisungsliste





6.5 Programmverzweigungsanweisungen

Programmverzweigungsanweisungen werden zusammen mit einem Sprungziel programmiert.

Funktion	MELSEC-Anweisung im MELSEC-EDITOR	MELSEC-Anweisung im IEC-EDITOR
Bedingter Sprung	CJ	CJ_M
Bedingter Sprung im nächsten Zyklus	SCJ	SCJ_M
Sprunganweisung	JMP	JMP_M
Sprung zum Programmende	GOEND	GOEND_M

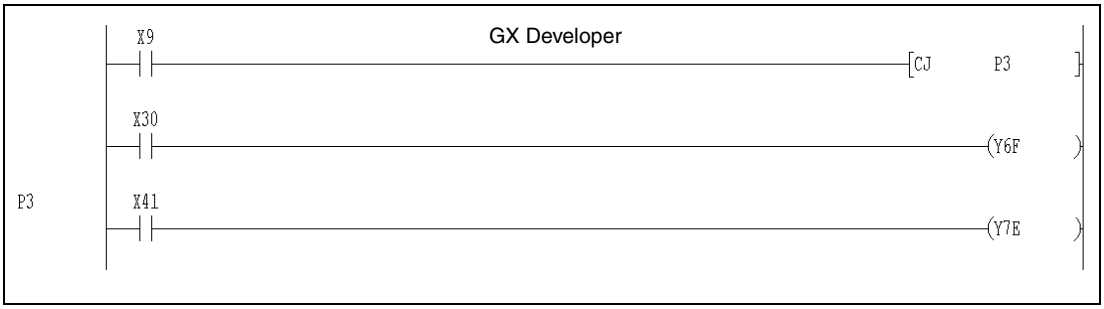
Sprungziele werden durch einen Pointer P (in MELSEC MEDOC und GX Developer) oder ein Label (im GX IEC Developer) gekennzeichnet.
Detaillierte Hinweise zur Programmierung eines Labels im GX IEC Developer erhalten Sie im Benutzerhandbch für den GX IEC Developer.

MELSEC MEDOC

```
LD    X9
CJ    P3
LD    X30
OUT   Y6F
P3
LD    X41
OUT   Y7E
```

GX IEC Developer

```
      LD    X9
      JMPC  Label_3
      LD    X30
      ST    Y6F
Label_3:
      LD    X41
      ST    Y7E
```



6.5.1 CJ, SCJ, JMP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
p																			●				3 ●[1]	●		●

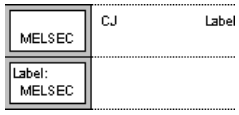
[1] Die Anzahl der Schritte bei Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

Operanden MELSEC Q

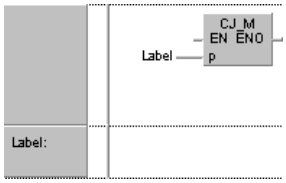
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sonder-module U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort				P		
p	—	—	—	—	—	—	—	—	●	SM0	2

GX IEC Developer

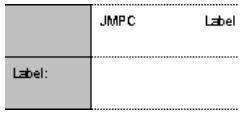
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

Operand	Befehlswert	Datentyp
p	Sprungzieladresse	Pointer/Label

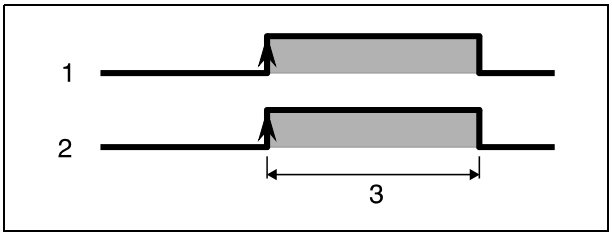
Funktionsweise

Sprunganweisungen

Eine Sprunganweisung besteht aus dem Sprungbefehl CJ, SCJ oder JMP (**C**onditional **J**ump = bedingter Sprung, **JuMP** = Sprung) und einem Pointer (Label) P, der das Sprungziel kennzeichnet. Die Pointer(Label)-Adresse kann bei der A-Serie zwischen P(Label)0 und P(Label)255 liegen, wobei P(Label)255 die Bedeutung einer END-Anweisung hat und nicht als Sprungzieladresse verwendet werden kann. Bei der Q-Serie und dem System Q kann die Pointer(Label)-Adresse zwischen P(Label)0 und P(Label)4095 liegen. Eine Sprungzieladresse P(Label)xx kann beliebig im Programm gesetzt werden.

CJ Bedingter Sprung

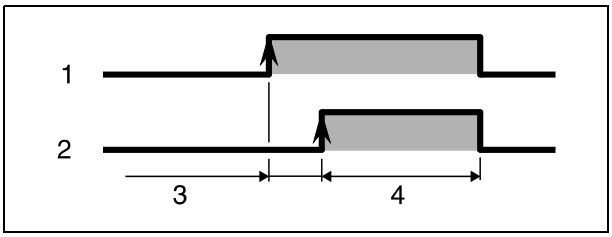
Die CJ-Anweisung führt nach gesetzter Eingangsbedingung den Programmteil an der angegebenen Sprungzieladresse aus. Ist die Eingangsbedingung nicht gegeben, wird der nächste Programmschritt ausgeführt.



[1] Eingangsbedingung
[2] Sprunganweisung
[3] Programmzyklus

SCJ Bedingter Sprung im nächsten Zyklus

Die SCJ-Anweisung führt nach gesetzter Eingangsbedingung den Programmteil an der angegebenen Sprungzieladresse mit dem folgenden Programmzyklus aus. Ist die Eingangsbedingung nicht gegeben, wird der nächste Programmschritt ausgeführt.



[1] Eingangsbedingung
[2] Sprunganweisung
[3] Ein Zyklus
[4] Programmzyklus

JMP Sprunganweisung

Die JMP-Anweisung führt den Programmteil an der angegebenen Sprungzieladresse ohne Eingangsbedingung aus (bedingungsloser Sprung).

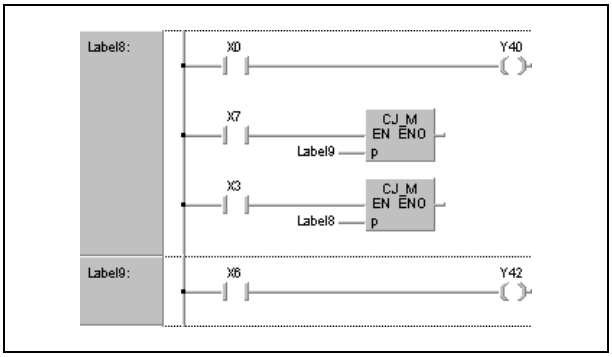
HINWEISE

Wird ein bereits gesetzter Timer durch eine CJ-, SCJ- oder JMP-Anweisung übersprungen, läuft der Timer weiterhin kontinuierlich ab.

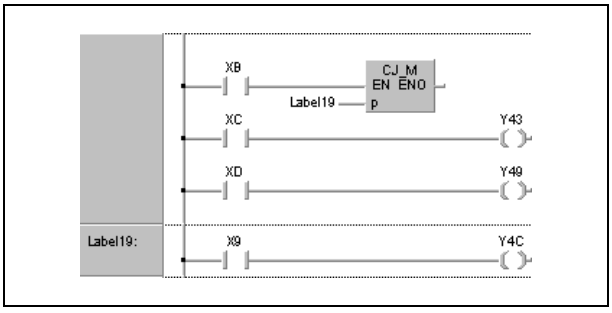
Wird eine OUT-Anweisung durch eine der Sprunganweisungen übersprungen, bleibt der Zustand des Ausgangs unverändert.

Durch die Ausführung einer Sprunganweisung verkürzt sich die Zykluszeit des Programms in Relation zu den übersprungenen Programmschritten. (Siehe Abs. A.2.1 „Liste der Verarbeitungszeiten (QnA-Serie und System Q)“ im Anhang)

Mit Hilfe der CJ-, SCJ- und JMP-Anweisung ist auch ein "Rücksprung" zu einer niedrigeren Sprungzieladresse möglich. Bei der Programmerstellung ist jedoch darauf zu achten, dass die Programmschleife verlassen wird, bevor der Watch Dog Timer abläuft (im Beispiel wird die Schleife verlassen, sobald X7 einschaltet).

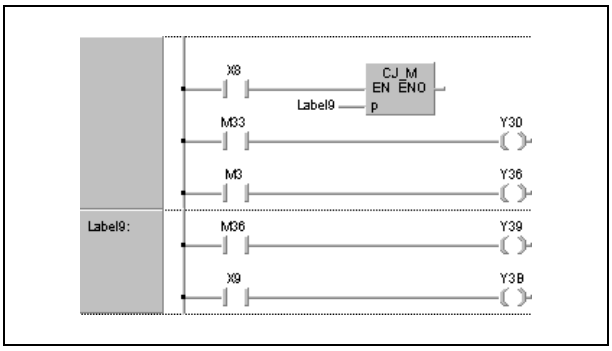


Ein mittels Sprunganweisung übersprungener Operand verändert seinen Zustand nicht. Das nachstehende Beispiel soll dies verdeutlichen.



Nach dem Einschalten von XB erfolgt der Sprung zur Sprungzieladresse Label19. Die Zustände der Ausgänge Y43 und Y49 bleiben in diesem Fall selbst dann unverändert, wenn XC oder XD ein- oder ausgeschaltet werden.

Die Sprungzieladresse Label9 belegt einen Schritt im Programm.



Durch eine CJ-, SCJ- oder JMP-Anweisung kann nur zu einer Sprungzieladresse innerhalb des selben Programms verzweigt werden.

Das Programm wird an der nächstmöglichen auf die Sprungadresse folgenden Adresse fortgesetzt, wenn die Sprungadresse innerhalb eines Bereichs liegt, der während einer Skip-Operation (Operation zum Überspringen eines Programmteils) genutzt wird.

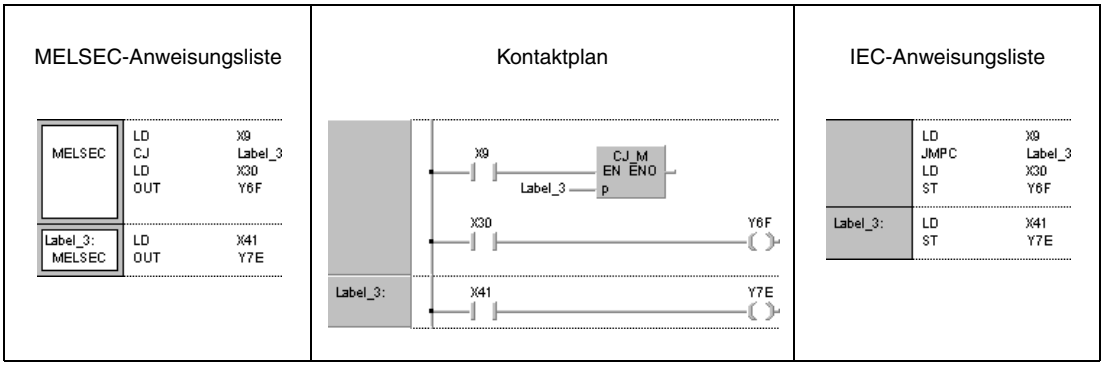
Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Eine Pointer-Adresse wird im Programm doppelt gesetzt und über eine Sprunganweisung angesprochen (Q-Serie/System Q = Fehlercode 4210).
- Die in der Sprunganweisung angegebene Sprungzieladresse ist im Programm nicht definiert (Sprungzieladresse bzw. Pointer fehlt) (Q-Serie/System Q = Fehlercode 4210).
- Das Sprungziel liegt hinter einer END-Anweisung (Q-Serie/System Q = Fehlercode 4210).
- Das Sprungziel liegt zwischen einer FOR-/NEXT-Routine.
- Das Sprungziel liegt innerhalb einer Unterroutine.

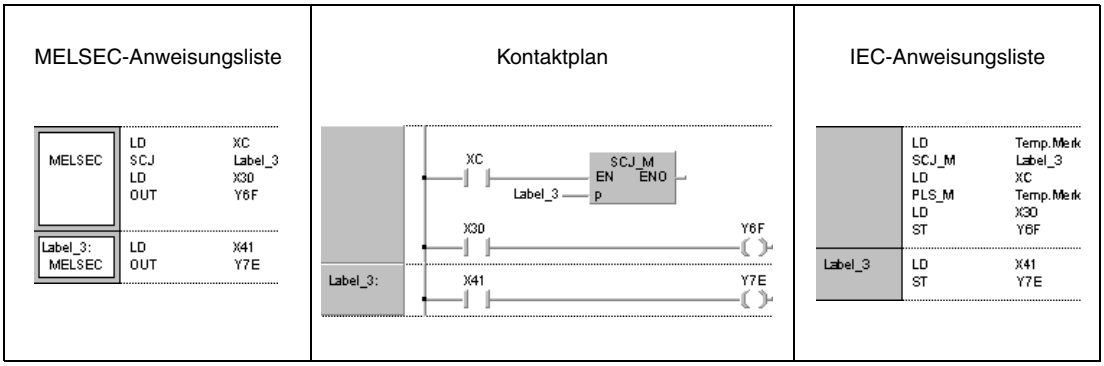
Beispiel 1 CJ

Im folgenden Programm wird bei Einschalten von X9 ein Sprung zur Sprungzieladresse Label_3 ausgeführt.



Beispiel 2 SCJ

Im Beispiel erfolgt der Programmsprung mit dem folgenden Zyklus zur Sprungzieladresse Label_3, sobald XC einschaltet.



6.5.2 GOEND

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
—	—	—	—	—	—	—	—	—	—	SM0	1

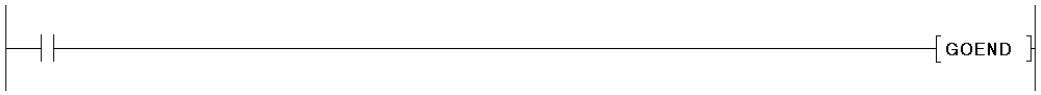
GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan

IEC-Anweisungsliste

GX Developer



Variablen

Operand	Befehlswert	Datentyp
—	—	—

Funktionsweise

GOEND Sprung zum Programmende

Bei der Sprunganweisung GOEND ist die Sprungzieladresse die FEND- oder END-Anweisung des Programms.

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

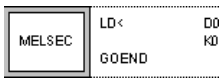
- Eine GOEND-Anweisung wurde nach der Ausführung einer CALL- oder ECALL-Anweisung und vor der Ausführung der RET-Anweisung gesetzt (Fehlercode 4211).
- Eine GOEND-Anweisung wurde nach der Ausführung einer FOR-Anweisung und vor der Ausführung der NEXT-Anweisung gesetzt (Fehlercode 4200).
- Eine GOEND-Anweisung wurde in einer Interrupt-Routine und vor der Ausführung der IRET-Anweisung gesetzt (Fehlercode 4221).
- Eine GOEND-Anweisung wurde zwischen einer CHKCIR- und einer CHKEND-Anweisung ausgeführt (Fehlercode 4230).
- Eine GOEND-Anweisung wurde zwischen einer IX- und einer IXEND-Anweisung ausgeführt (Fehlercode 4231).

Beispiel

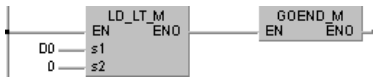
GOEND

Im folgenden Programm wird zur END-Anweisung gesprungen, wenn der Wert in D0 negativ ist.

MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste

```
LD      TRUE
LD_LT_M D0 , K0
GOEND_M
```

6.6 Anweisungen zum Interrupt-Programmaufruf

Die Anweisung zum Interrupt-Programmaufruf ermöglicht das Aufrufen von Interrupt-Routinen. Dabei können die Interrupts einzeln oder über Bit-Schemen aktiviert bzw. deaktiviert werden. Die folgende Tabelle enthält eine Übersicht der Anweisungen.

Funktion	MELSEC-Anweisung im MELSEC-EDITOR	MELSEC-Anweisung im IEC-EDITOR
Interrupt verhindern	DI	DI_M
Interrupt ermöglichen	EI	EI_M
Bit-Schema der Ausführungsbedingungen der Interruptprogramme	IMASK	IMASK_M
Ende eines Interrupt-Programms	IRET	IRET_M

6.6.1 DI, EI, IMASK

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●[1]	●[1]	●	●	●	●

[1] Bei einer AnN oder AnS-CPU ist die EI/DI-Anweisung nur dann ausführbar, wenn der Sondermerker M9053 nicht gesetzt ist.

Operanden MELSEC A

	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
																							1 ●[1]			

[1] Nur für EI- und DI-Anweisung

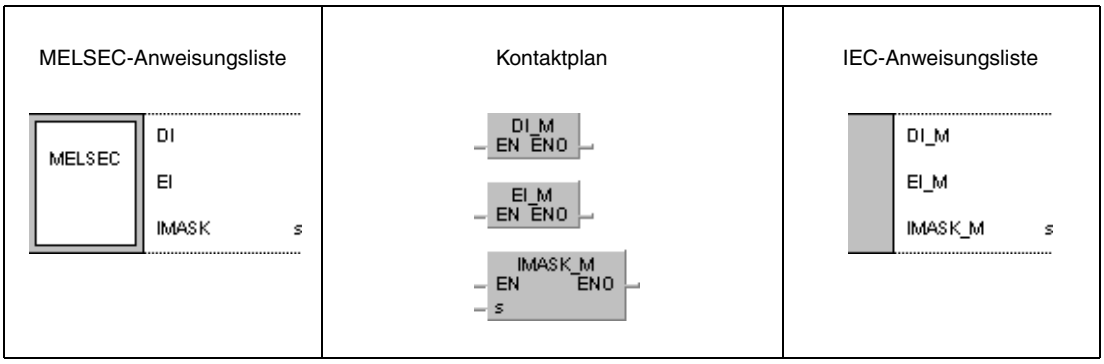
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	—	●[1]	●[1]	—	●[1]	●[1]	—	—	—	—	2 ●[1]
—	—	—	—	—	—	—	—	—	—	—	1 ●[2]

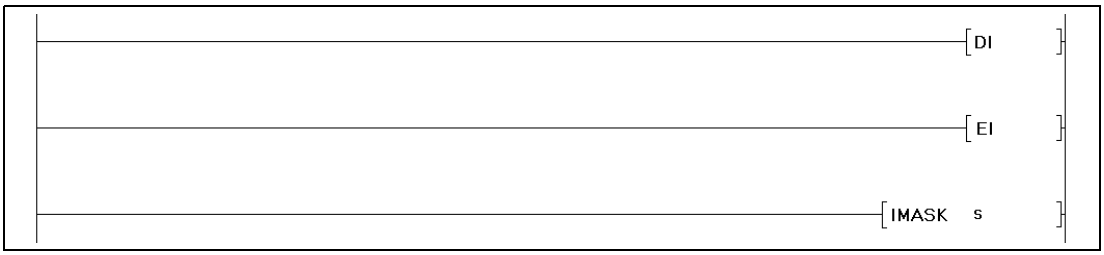
[1] Nur für IMASK-Anweisung

[2] Nur für EI- und DI-Anweisung

GX IEC Developer



GX Developer



Variablen

Operand	Befehlswert	Datentyp
s	Bit-Schema, in dem die Ausführungsbedingungen der Interrupts angegeben werden oder Startadresse, ab dem dieses Bit-Schema gespeichert ist.	BIN-16-Bit

Funktions-weise

Interrupt-Anweisungen

Ein Interrupt-Programm ist ein eingefügter Programmteil (gekennzeichnet durch die Interrupt-Adresse Ixx), der über ein externes Interrupt-Signal aufgerufen werden kann. Die Ausführung des Interrupt-Programms erfolgt in Abhängigkeit der EI-/DI-Anweisung. Die Bedeutung der EI/DI-Anweisungen hängt bei Verwendung einer AnN-CPU von dem Status des Sondermerkers M9053 ab. Nur wenn der Merker nicht gesetzt ist, dienen die Anweisungen als Ausführungsbedingungen für ein Interrupt-Programm. Ist der Merker M9053 gesetzt, werden die Anweisungen in Zusammenhang mit einem Link-Refresh genutzt (siehe Abs. „Datenaktualisierungsanweisungen“ der Programmieranleitung).

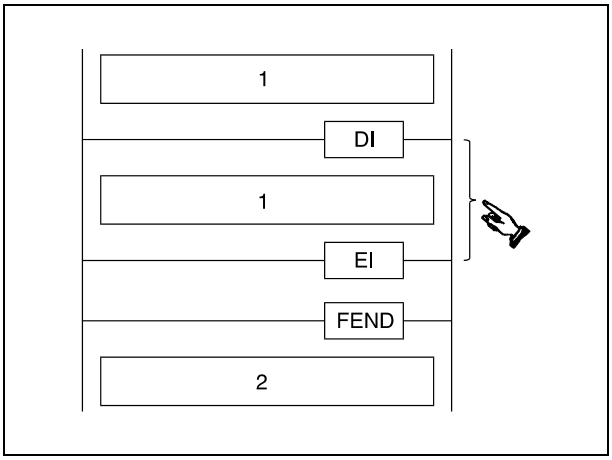
DI Interrupt verhindert

Die DI-Anweisung verhindert die Abarbeitung eines Interrupt-Programms so lange, bis eine EI-Anweisung in der Programmfolge erscheint. Der DI-Status ist nach dem Einschalten oder einem RESET der CPU aktiv.

EI Interrupt ermöglicht

Die EI-Anweisung ermöglicht den Aufruf eines Interrupt-Programms mittels Angabe der Interrupt-Adresse Ixx bzw. die Ausführung der IMASK-Anweisung.

Erfolgt der Interrupt zwischen einer DI- und EI-Anweisung, wird dieser Interrupt erst dann ausgeführt, wenn die zwischen dieser DI- und EI-Anweisung befindliche Programmsequenz abgearbeitet ist. Die nachfolgende Abbildung zeigt eine solche Programmierung.



[1] Ablaufprogramm

[2] Interruptprogramm

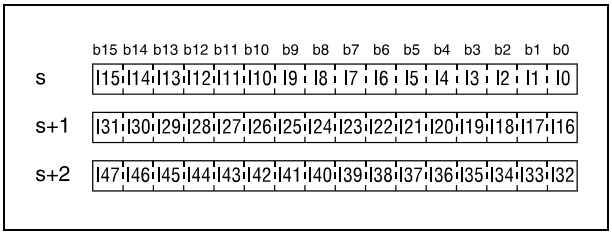
HINWEIS *Beim GX IEC Developer wird die FEND-Anweisung automatisch eingefügt. Das Ereignis Ixx muss einer Task zugewiesen werden.*

IMASK Bit-Schemen der Ausführungsbedingungen der Interruptprogramme (Nur Q-Serie und System Q)

In dem in s angegebenen Bit-Schema ist jedem Bit eine bestimmte Interrupt-Adresse zugeordnet. Ob der entsprechende Interrupt ausgeführt werden kann, ist von dem Zustand des jeweiligen Bits abhängig. Hat das Bit den Wert 0, kann das zugeordnete Interrupt-Programm nicht ausgeführt werden. Wenn das Bit den Zustand 1 aufweist, wird das Interruptprogramm ausgeführt.

QnA-CPU

Die Zuordnung der Bits in s bis s+2 zu den entsprechenden Interrupt-Adressen zeigt folgende Abbildung.

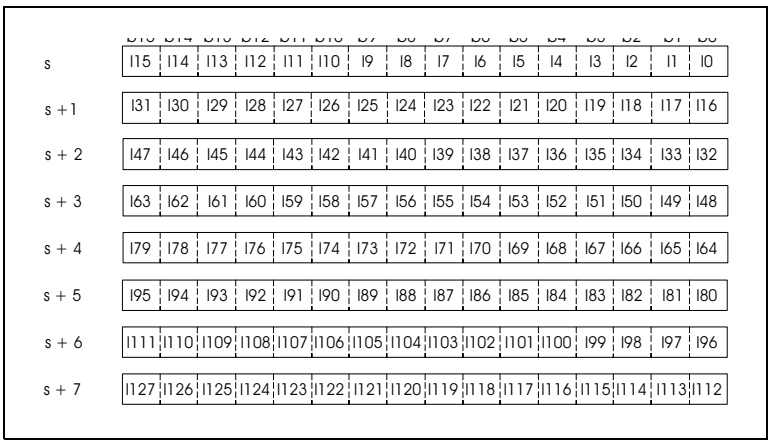


Nach dem Einschalten der CPU oder einem Reset mittels des RUN-STOP-Schlüsselschalters sind die Bits b0 - b31 (Interrupt-Adresse I0 - I31) auf 1 gesetzt, d.h. diese Interrupt-Programme können ausgeführt werden. Die Bits b32 - b47 (Interrupt-Adresse I32 - I47) haben den Zustand 0, wodurch die entsprechenden Interrupt-Programme nicht ausgeführt werden können.

Die in s bis s+2 angegebenen Bit-Schemen werden in den Diagnoseregistern SD715 bis SD717 gespeichert.

Single-Prozessor-Q-CPU

Die Bits in s bis s+7 sind den Interrupt-Adressen wie folgt zugeordnet:



Nach Einschalten der Versorgungsspannung der CPU oder nach einem Reset mit dem RUN/STOP-Schalter, ist die Ausführung der Interrupt-Programme I0 bis I31 freigegeben.

Die in s bis s+2 angegebenen Bit-Schemen werden in den Diagnoseregistern SD715 bis SD717 gespeichert. Die in s+3 bis s+7 angegebenen Bit-Schemen werden in den Diagnoseregistern SD781 bis SD785 gespeichert.
Obwohl die Diagnoseregister durch die Anordnung in SD715 bis SD717 und in SD781 bis SD785 getrennt sind, werden die Bit-Schemen mit s bis s+7 bezeichnet.

Multi-Prozessor-Q-CPU

Die folgende Abbildung zeigt die Zuordnung der Bits in s bis s+15 zu den Interrupt-Adressen.

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
s	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
s +1	I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16
s + 2	I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32
s + 3	I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48
s + 4	I79	I78	I77	I76	I75	I74	I73	I72	I71	I70	I69	I68	I67	I66	I65	I64
s + 5	I95	I94	I93	I92	I91	I90	I89	I88	I87	I86	I85	I84	I83	I82	I81	I80
s + 6	I111	I110	I109	I108	I107	I106	I105	I104	I103	I102	I101	I100	I99	I98	I97	I96
s + 7	I127	I126	I125	I124	I123	I122	I121	I120	I119	I118	I117	I116	I115	I114	I113	I112
s + 8	I143	I142	I141	I140	I139	I138	I137	I136	I135	I134	I133	I132	I131	I130	I129	I128
s + 9	I159	I158	I157	I156	I155	I154	I153	I152	I151	I150	I149	I148	I147	I146	I145	I144
s +10	I175	I174	I173	I172	I171	I170	I169	I168	I167	I166	I165	I164	I163	I162	I161	I160
s + 11	I191	I190	I189	I188	I187	I186	I185	I184	I183	I182	I181	I130	I129	I128	I127	I126
s + 12	I207	I206	I205	I204	I203	I202	I201	I200	I199	I198	I197	I196	I195	I194	I193	I192
s + 13	I223	I222	I221	I220	I219	I218	I217	I216	I215	I214	I213	I212	I211	I210	I209	I208
s + 14	I239	I238	I237	I236	I235	I234	I233	I232	I231	I230	I229	I228	I227	I226	I225	I224
s + 15	I255	I254	I253	I252	I251	I250	I249	I248	I247	I246	I245	I244	I243	I242	I241	I240

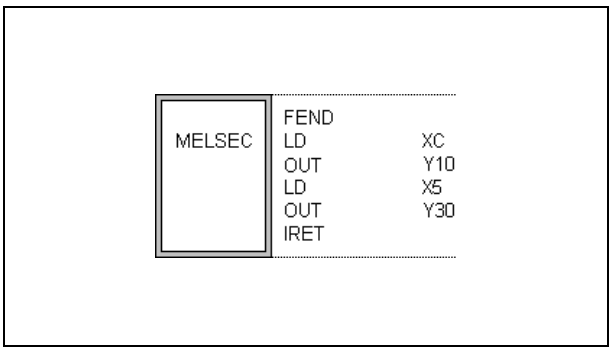
Nach Einschalten der Versorgungsspannung der CPU oder nach einem Reset (mit dem RUN/STOP-Schalter), ist die Ausführung der Interrupt-Programme I0 bis I31 und I48 bis I255 freigegeben. Die Interrupt-Programme I32 bis I47 sind gesperrt und können nicht ausgeführt werden.

Die in s bis s+2 angegebenen Bit-Schemen werden in den Diagnoseregistern SD715 bis SD717 gespeichert. Die in s+3 bis s+15 angegebenen Bit-Schemen werden in den Diagnoseregistern SD781 bis SD793 gespeichert.
Obwohl die Diagnoseregister durch die Anordnung in SD715 bis SD717 und in SD781 bis SD793 getrennt sind, werden die Bit-Schemen mit s bis s+15 bezeichnet.

HINWEISE *Wird ein Counter innerhalb eines Interrupt-Programms benötigt, sind hierfür die speziellen Interrupt-Counter zu programmieren. Die CPU-Typen A3H, A3M, AnA, AnAS und AnU verfügen nicht über Counter, die in einem Interrupt-Programm verwendet werden können.*

Die Interrupt-Adresse (Interrupt-Pointer) zur Kennzeichnung des Interrupt-Programms belegt einen Programmschritt.

Schachteln Sie keine Interrupt-Programme d.h. rufen Sie in einem Interrupt-Programm kein anderes Interrupt-Programm auf.



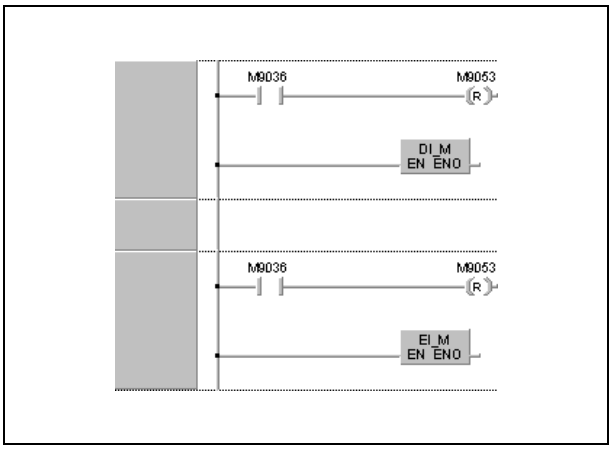
Wenn das Programm im GX Developer oder mit dem GX IEC Developer im Melsec-Modus erstellt wird die Anweisungen FEND und IRET vom Benutzer programmiert werden.

Eine Alternative dazu ist die Programmierung im IEC-Editor. Der Interrupt wird einem Task zugeordnet, und die FEND und IRET Anweisungen werden vom Compiler des GX IEC Developers automatisch gesetzt (siehe Programmierbeispiel).

Während der Ausführung eines Interrupt-Programms wird intern der DI-Status gesetzt, so das keine weiteren Interrupt-Programme gleichzeitig ausgeführt werden können. Der Aufruf eines weiteren Interrupt-Programms ist erst wieder nach Setzen der EI-Anweisung möglich.

Befindet sich eine EI- oder DI-Anweisung innerhalb einer MC-Anweisung, werden diese EI- oder DI-Anweisungen ohne Berücksichtigung der MC-Anweisung ausgeführt.

Bei einer AnN oder AnS CPU ist die EI-/DI-Anweisung nur dann ausführbar, wenn der Sondermerker M9053 nicht gesetzt ist. Ist der Sondermerker gesetzt, ist die EI-/DI-Anweisung die Ausführungsbedingung eines Link-Refresh. Damit die EI-/DI-Anweisung in Verbindung mit einer AnN oder AnS CPU als Bedingung für einen Interrupt-Programmaufruf verarbeitet werden kann, muss vor der EI-/DI-Anweisung der Sondermerker M9053 zurückgesetzt werden.



Beispiel EI, DI, IMASK (GX IEC Developer)

Im nachfolgenden Beispielprogramm wird die Ausführung eines Interrupt-Programms nach dem Einschalten von X0 ermöglicht. Bei ausgeschaltetem X0 wird die Abarbeitung des Interrupt-Programms verhindert.

Die untere Abbildung zeigt die im IEC-Modus zu programmierenden Tasks, die die Interrupt-Programme I1 und I2 aufrufen.

Interrupt_1 (I1) und Interrupt_2 (I2) sind die Interrupt-Programme. Die IRET-Anweisung muss nicht programmiert werden, da sie automatisch im Compiler des GX IEC Developers gesetzt wird.

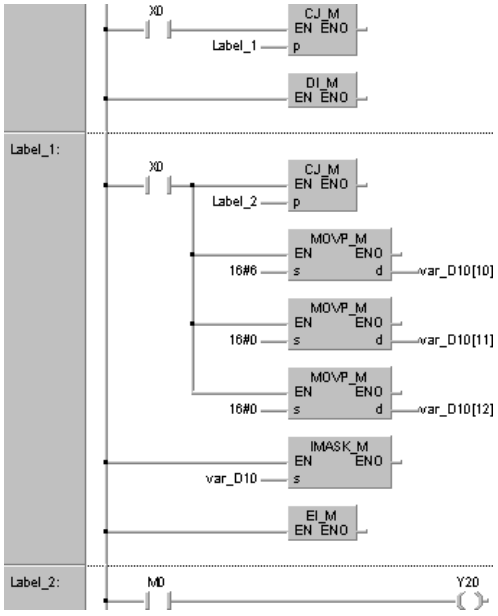
MELSEC-Anweisungsliste

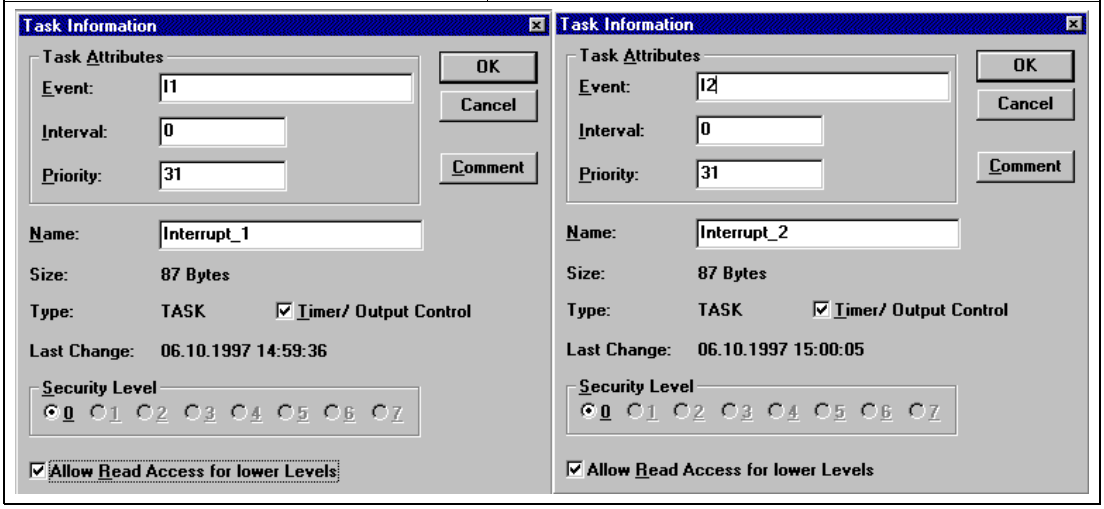
```
MELSEC      LD      X0
            CJ      Label_1
            DI
Label_1:    LD      X0
MELSEC      CJ      Label_2
            LD      X0
            MOVP    H6
                    D10
            MOVP    H0
                    D11
            MOVP    H0
                    D12
            IMASK   D10
            EI
Label_2:    LD      M0
MELSEC      OUT     Y20
```

IEC-Anweisungsliste

```
            LD      X0
            JMPC    Label_1
            DI_M
Label_1:    LD      X0
            JMPC    Label_2
            LD      X0
            MOVP_M  H6 , var_D10[10]
            MOVP_M  H0 , var_D10[11]
            MOVP_M  H0 , var_D10[12]
            IMASK_M var_D10
            EI_M
Label_2:    LD      M0
            ST      Y20
```

Kontaktplan





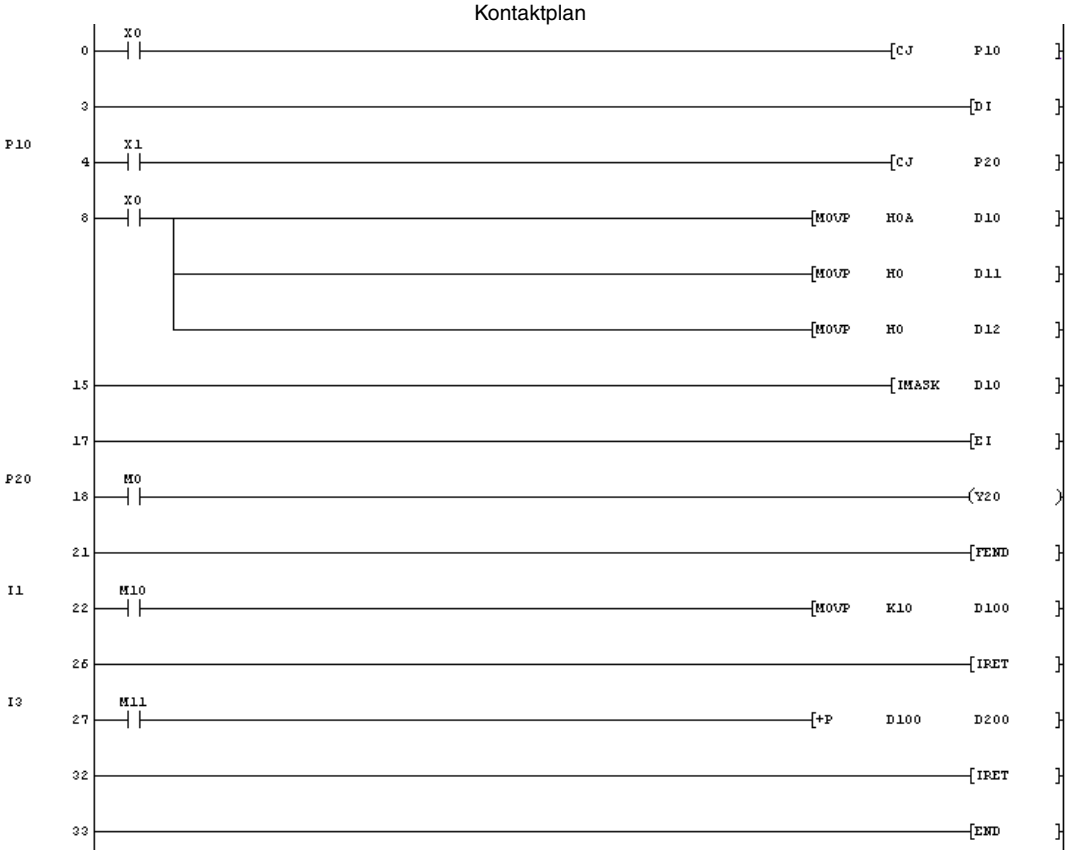
HINWEIS *Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

Beispiel EI, DI, IMASK bei einer Q-CPU (GX Developer)

Im nachfolgenden Beispielprogramm wird die Ausführung eines Interrupt-Programms nach dem Einschalten von X0 ermöglicht. Bei ausgeschaltetem X0 wird die Abarbeitung des Interrupt-Programms verhindert.

I1 und I3 bilden die beiden Interrupt-Programme.

Kontaktplan



Anweisungsliste

```
 0  LD      X0
 1  CJ      P10
 3  DI
 4  P10
 5  LD      X1
 6  CJ      P20
 8  LD      X0
 9  MOVP    H0A     D10
11  MOVP    H0      D11
13  MOVP    H0      D12
15  IMASK   D10
17  EI
18  P20
19  LD      M0
20  OUT     Y20
21  FEND
22  I1
23  LD      M10
24  MOVP    K10     D100
26  IRET
27  I3
28  LD      M11
29  +P      D100    D200
32  IRET
33  END
```

6.6.2 IRET

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
																							1			

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sonder-module U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
	—	—	—	—	—	—	—	—	—	—	1

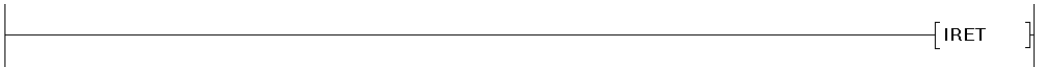
GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan

IEC-Anweisungsliste

GX Developer



Variablen

Operand	Befehlswert	Datentyp
—	—	—

HINWEIS In den IEC-Editoren wird die IRET-Anweisung autonatisch erzeugt.

Funktions-weise

Rücksprung aus dem Interrupt-Programm ins Hauptprogramm

IRET Ende eines Interrupt-Programms

Das Ende eines Interrupt-Programms wird durch eine IRET-Anweisung gekennzeichnet.

Die Verarbeitung von Countern wird während des Interrupts fortgesetzt.

Die Rückkehr zum Hauptprogramm erfolgt nach Ausführung der IRET-Anweisung.

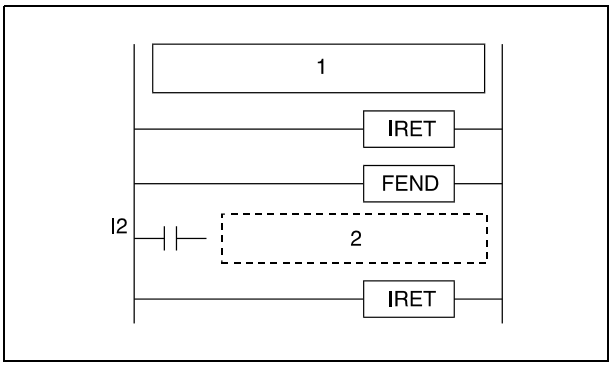
Bei den CPU-Typen A3H, A3M, AnA, AnAS und AnU erfolgt keine Verarbeitung von Interrupt-Countern.

Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Es gibt keine Zieladresse zu dem Interrupt-Aufruf (Q-Serie/System Q = Fehlercode 4220).
- Ist eine IRET-Anweisung in der Ablauffolge vor einem Interrupt-Programm programmiert, bricht die CPU die Verarbeitung an dieser Stelle ab (Q-Serie/System Q = Fehlercode 4223).
- Nach einem Interrupt-Aufruf und vor der Ausführung der IRET-Anweisung ist eine END-, FEND-, GOEND- oder STOP-Anweisung gesetzt worden (Q-Serie/System Q = Fehlercode 4221).

HINWEIS *Das nachfolgende Beispiel zeigt eine fehlerhafte Programmierung!*



[1] Ablaufprogramm

[2] Interrupt-Programm

Beispiel Die Verwendung einer IRET-Anweisung in einem Programm ist dem Beispiel der EI-, DI-, IMASK-Anweisung zu entnehmen.

6.7 Datenaktualisierungsanweisungen

Die Datenaktualisierungsanweisungen bieten die Möglichkeit, Daten an Ein-/Ausgabe-Schnittstellen oder Daten einer Datenübertragung zu aktualisieren. Die folgende Tabelle enthält eine Übersicht der Datenaktualisierungsanweisungen.

Funktion	MELSEC-Anweisung im MELSEC-EDITOR	MELSEC-Anweisung im IEC-EDITOR
E/A-Teilaktualisierung (Q-Serie)	RFS	RFS_M
	RFSP	RFSP_M
E/A-Teilaktualisierung (A-Serie)	SEG	SEG_M
Aktualisierungsanweisung für Netzwerk- und Schnittstellendaten	COM	COM_M
Ausführungsbedingungen für die Aktualisierungsanweisung von Netzwerk- und Schnittstellendaten	EI	EI_M
	DI	DI_M

6.7.1 RFS, RFSP

CPU

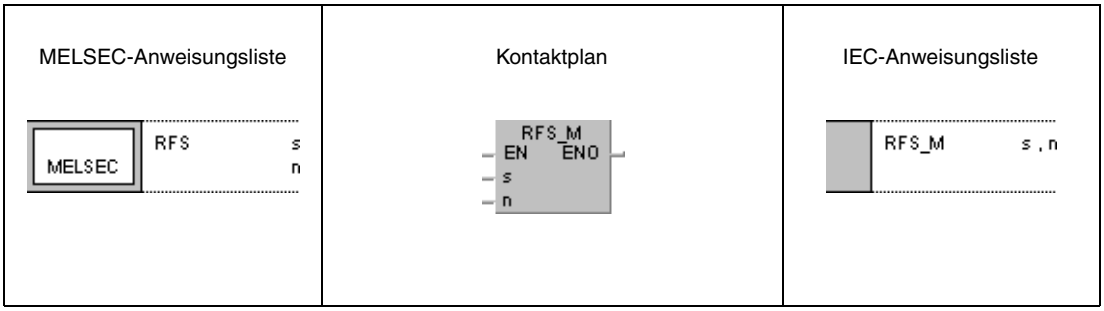
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	●[1]	—	—	—	—	—	—	—	—	SM0	3
n	●	●	●	●	●	●	●	●	—		

[1] Nur X und Y

GX IEC Developer



GX Developer

Variablen

Operand	Befehlswert	Datentyp
s	Startadresse des Ein-/Ausgangs, auf den die Aktualisierungsanweisung angewendet wird.	Bit
n	Anzahl der zu aktualisierenden Ein-/Ausgangsbits.	BIN-16-Bit

Funktionsweise

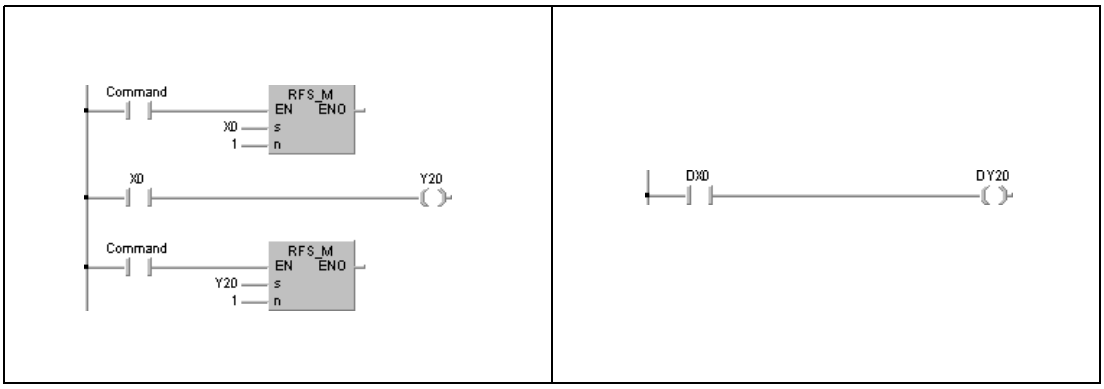
E/A-Teilaktualisierung (Q-Serie und System Q)

RFS Aktualisierungsanweisung

Die RFS-Anweisung aktualisiert die Ein- und Ausgänge des ausgewählten Bereichs für einen Programmzyklus und dient dem Einlesen von Daten einer externen Quelle, bzw. der Ausgabe von Daten an ein Ausgangsmodul.

Das Einlesen von Daten von einer externen Quelle oder das Ausgeben der Daten an eine externe Quelle erfolgt stapelweise nach Ausführung der END-Anweisung. Demzufolge können im Verlauf eines Programmzyklus keine gepulsten Signale ausgegeben werden. Bei der Teilaktualisierung mit Hilfe der SEG-Anweisung werden die vorgegebenen E-/A-Adressen der Eingänge (X) und Ausgänge (Y) separat aktualisiert. Auf diese Weise können auch gepulste Signale ausgegeben werden.

Bei Verwendung direkt adressierbarer Ein- und Ausgänge (DX/DY) werden die Eingänge (X) und Ausgänge (Y) bitweise aktualisiert.



Im linken Beispiel werden der Eingang X0 und der Ausgang Y20 über die RFS-Anweisung aktualisiert.

Im rechten Beispiel wird die gleiche Funktion bei Verwendung von DX und DY ohne Aktualisierungsanweisung ausgeführt.

Fehlerquellen

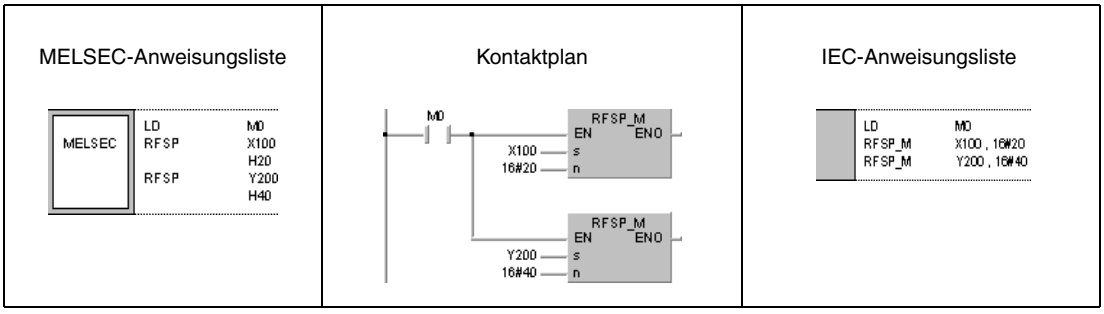
In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die mit n angegebene Anzahl der zu aktualisierenden Bits ist außerhalb des Ein-/Ausgabeadressbereichs.

Beispiel 1

RFSP

Das folgende Programm aktualisiert mit positiver Flanke von M0 die Eingänge X100 bis X11F und die Ausgänge Y200 bis Y23F.



6.7.2 SEG

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●[1]	●[1]	●[1]	●[1]		

[1] Eine Teilaktualisierung ist nur bei gesetztem Merker M9052 ausführbar.

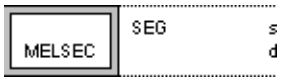
Operanden MELSEC A

	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
s	●	●	●	●	●	●	●	●	●	●	●	●	●		●		●	●				K1 ↓ K8	9	●		●
d		●	●	●	●	●	●	●	●	●	●	●	●		●								●[1]			

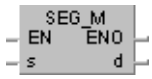
[1] Die Anzahl der Schritte bei Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

GX IEC Developer

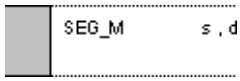
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



Variablen

Operand	Befehlswert	Datentyp
s	Startadresse des Ein-/Ausgangs, auf den die Aktualisierungsanweisung angewendet wird.	Bit
d	Anzahl der zu aktualisierenden Ein-/Ausgangsbits.	BIN-16-Bit

Funktions-weise

E/A-Teilaktualisierung (A-Serie)

Allgemeines

Die Ausführung einer Teilaktualisierung ist nur bei gesetztem Sondermerker M9052 möglich. Ist der Sondermerker nicht gesetzt, hat die SEG-Anweisung die Funktion der 7-Segment-Decodierung.

SEG Teilaktualisierung

Die SEG-Anweisung ermöglicht bei gegebener Eingangsbedingung die Aktualisierung eines bestimmten E/A-Adressenbereiches.

Bei einer Teilaktualisierung werden die vorgegebenen E-/A-Adressen nur für einen Programmzyklus aktualisiert. Eingangssignale werden währenddessen empfangen und abgehende Signale an die Ausgangsmodule weitergeleitet.

Mit Hilfe der Teilaktualisierung wird der Einschaltzustand von Eingängen (X) und Ausgängen (Y) während der E-/A-Verarbeitung im Refresh-Betrieb für einen Programmzyklus geändert.

Während eines gewöhnlichen Link-Refreshs werden die Ein- und Ausgangssignale stapelweise nach Ausführung der END-Anweisung verarbeitet. Demzufolge können im Verlauf eines Programmzyklus keine gepulsten Signale ausgegeben werden. Bei der Teilaktualisierung mit Hilfe der SEG-Anweisung werden die vorgegebenen E-/A-Adressen der Eingänge (X) und Ausgänge (Y) separat aktualisiert. Auf diese Weise können auch gepulste Signale ausgegeben werden.

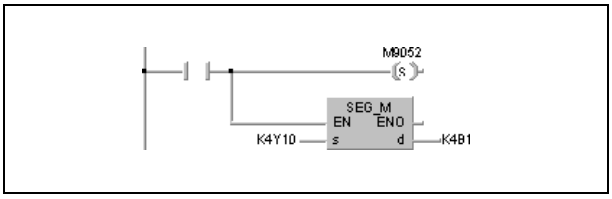
HINWEIS *In Verbindung mit einer A2C CPU können innerhalb eines Zyklus keine gepulsten Signale während der Kommunikation mit den E/A-Modulen ausgegeben werden. Dies gilt auch bei Teilaktualisierung der Ausgänge (Y) mittels SEG-Anweisung. Nähere Angaben hierzu sind der Hardware-Beschreibung der A2C-Serie zu entnehmen.*

Ausführungsbedingungen

Programmstruktur

Die Programmierung muss mit dem Setzen des Sondermerkers M9052 beginnen. Die dem SEG-Befehl folgenden Quelldaten s bestimmen die Startadresse (nur für Ein- und Ausgänge X, Y) der Teilaktualisierung. Neben den Quelldaten wird die Anzahl der Ein-/Ausgänge in Blöcken zu 8 Ein- oder Ausgängen festgelegt.

Die Abbildung zeigt das Programmschema zur SEG-Anweisung.



Startadresse

Das Festlegen der Startadresse erfolgt stets mit der ersten Adresse einer Adressenfolge von Eingangs- oder Ausgangsoperanden (z.B. X0, X10, Y20 usw.).

Wird eine Adresse zwischen Yn0 und Yn7 (Xn0 und Xn7) angegeben, erfolgt der Aktualisierung ab Adresse Yn0 (Xn0). Wird eine Adresse zwischen Yn8 und YnF (Xn8 und XnF) angegeben, erfolgt die Aktualisierung ab Adresse Yn8.

Anzahl der Adressen

Die Anzahl der Adressen, die für die Aktualisierung vorgesehen sind, kann im Bereich von 8 bis 2048 festgelegt werden. Die Festlegung in Blöcken zu 8 Adressen geschieht wie folgt:

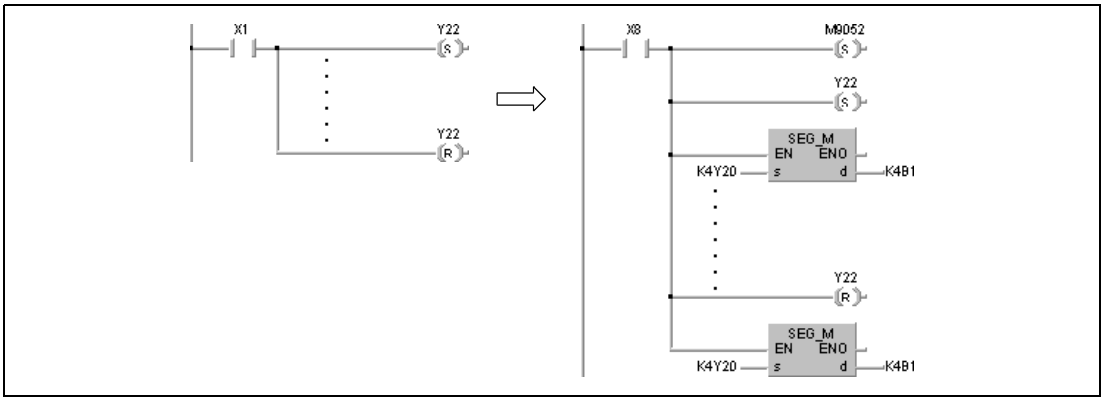
B1 = 8 Adressen
B2 = 16 Adressen
…
BA = 80 Adressen
BB = 88 Adressen
…
B10 = 128 Adressen
…
BFF = 2048 Adressen

Die Vorgabe von B0 aktualisiert sämtliche Adressen der SPS ab der vorgegebenen Startadresse.

Die Teilaktualisierung wird auch dann weitergeführt, wenn die SEG-Anweisung in der Direktverarbeitung der CPU ausgeführt wird. In diesem Fall werden die Schaltzustände der Ein- und Ausgänge jedoch nicht verändert.

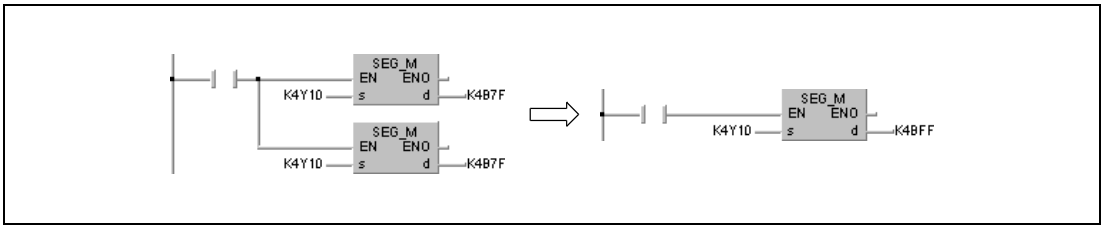
Bei gepulster Ausgabe mittels SET- und RST-Anweisungen in der Direktverarbeitung müssen die Programmblöcke für einen Link-Refresh wie folgt geändert werden.

HINWEIS *Von einer A2C-CPU kann das Programm nicht verarbeitet werden.*



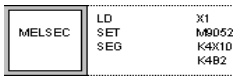
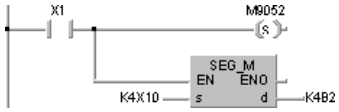
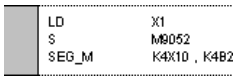
HINWEIS *Werden in Verbindung mit einer AnA oder AnU CPU alle 2048 Adressen mit einem Mal durch die SEG-Anweisung angesprochen, kann es zu einer fehlerhaften Verarbeitung der Ein-/Ausgangsaktualisierung kommen. Die Ausführung der Aktualisierung muss daher aufgeteilt in 2 x 1024 Adressen erfolgen.*

Das nachstehende Beispiel zeigt das Programmsplitting für eine Aktualisierung von 2048 Adressen in einem Programmzyklus einer AnA oder AnU CPU.



Beispiel SEG

Im folgenden Programm werden die Eingänge X10 bis X1F aktualisiert.

MELSEC-Anweisungsliste	Kontaktplan	IEC-Anweisungsliste
		

6.7.3 COM

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
																							3			

Operanden MELSEC Q

	Operanden								Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere	
	Bit	Wort		Bit	Wort					
—	—	—	—	—	—	—	—	—	—	—	1

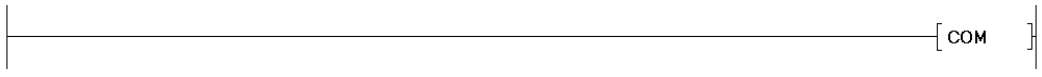
GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan

IEC-Anweisungsliste

GX Developer



Variablen

Operand	Befehlswert	Datentyp
—	—	—

Funktions-weise

Link-Refresh

COM Aktualisierungsanweisung für Netzwerk- und Schnittstellendaten

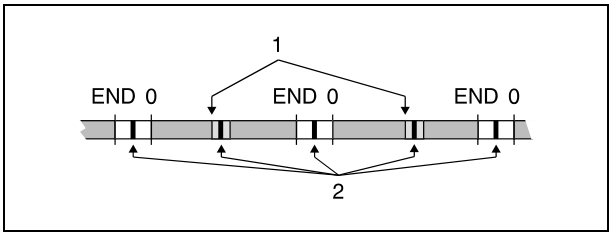
Die Art der Ausführung der COM-Anweisung ist bei CPUs der QnA-Serie und des System Q vom Zustand des Diagnosemerkers SM775 abhängig:

- Ist SM775 nicht gesetzt (0), wird eine Aktualisierung der Netzwerk- und Schnittstellen-Daten (Link Refresh) und eine Gesamtdatenverarbeitung (END-Verarbeitung) ausgeführt.
- Ist SM775 gesetzt (1), wird nur eine Gesamtdatenverarbeitung (END-Verarbeitung) ausgeführt.

Folgende Ausführungen gelten für die QnA-Serie und das System Q bei nicht gesetztem SM775 (0) und für die A-Serie:

Die COM-Anweisung wird z.B. zur Beschleunigung der Datenkommunikation mit einer Remote-EA-Station eingesetzt. Ist die Programmzykluszeit einer Master-Station länger als die einer lokalen Station, ermöglicht die COM-Anweisung die korrekte Verarbeitung der empfangenen Ein- und Ausgangsdaten.

Bei Ausführung der COM-Anweisung unterbricht die CPU vorübergehend das Ablaufprogramm und führt eine Gesamtdatenverarbeitung (END-Verarbeitung) und eine Aktualisierung der Netzwerk- und Schnittstellen-Daten (Link Refresh) durch.



[1] COM-Anweisung

[2] Gesamtdatenverarbeitung/Link-Refresh

Eine COM-Anweisung kann beliebig oft im Ablaufprogramm programmiert werden. Beachten Sie aber, dass die Programmzykluszeit um den Zeitraum der Gesamtdatenverarbeitung und des Link-Refreshs zunimmt.

HINWEISE

Während der Gesamtdatenverarbeitung laufen folgende Prozesse ab:

– Kommunikation zwischen der SPS und den Kommunikationsmodulen.

– Überwachung und Kontrolle der übrigen Stationen.

– Auslesen des Pufferspeichers der übrigen Sondermodule über ein Computer-Link-Modul.

Bei der Aktualisierung der Netzwerk- und Schnittstellen-Daten werden die folgenden Prozesse ausgeführt:

– Aktualisierung des CC-Link

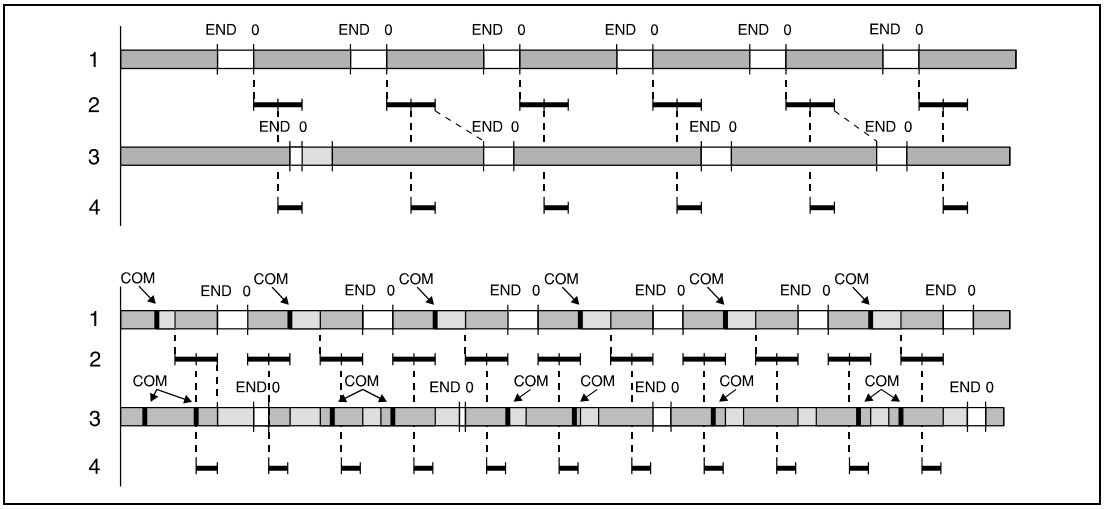
– Automatische Aktualisierung von Sondermodulen

– Aktualisierung des MELSECNET/10 und MELSECNET/H

– Automatische Aktualisierung des gemeinsamen Speicherbereichs in einem Multi-CPU-System (nur bei Q02-, Q02H-, Q06H-, Q12H, Q12PH-, Q25H- und Q25PHCPU ab Software Version B).

Ausführungsbedingungen:

Die Abbildung zeigt im oberen Diagramm die Zeitverläufe der Datenübertragung ohne COM-Anweisung. Im unteren Diagramm werden die Zeitverläufe bei Programmierung einer COM-Anweisung dargestellt.



[1] Programm der Master-Station

[2] Datenübertragung

[3] Programm der lokalen Station

[4] Remote E/A-Station, E/A-Refresh

Die Datenübertragung im Netzwerk wird mit Hilfe der COM-Anweisung im Ablaufprogramm der Master-Station beschleunigt, da die Anzahl der Kommunikationsvorgänge mit den Remote-E/A-Stationen zunimmt.

Ein korrekter Datenempfang entsprechend dem oberen Beispiel ist nicht möglich, wenn die Programmzykluszeit einer lokalen Station länger als die einer Master-Station ist. Ein sicherer Datenaustausch wird hier durch Programmierung der COM-Anweisung im Ablaufprogramm der lokalen Station gewährleistet.

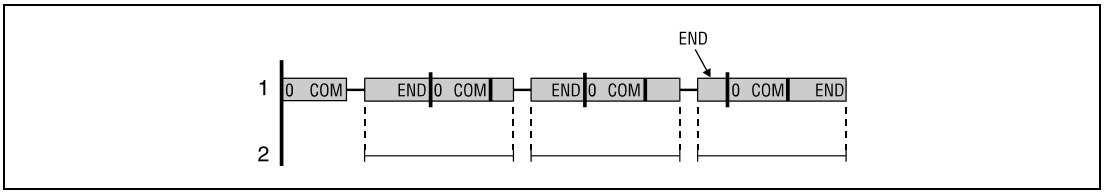
Wird eine COM-Anweisung im Ablaufprogramm einer lokalen Station programmiert, erfolgt eine Aktualisierung (Link-Refresh) jedesmal dann, wenn die lokale Station die Anweisung von der Master-Station zwischen den folgenden Anweisungen empfängt:

– Schritt 0 und COM-Anweisung,

– zwei COM-Anweisungen,

– COM-Anweisung und END-Anweisung.

Eine Beschleunigung der Datenkommunikation ist nicht möglich, wenn die Zykluszeit im Netzwerk länger als die Programmzykluszeit der Master-Station ist. Dies gilt auch dann, wenn die COM-Anweisung im Ablaufprogramm der Master-Station programmiert wurde.



[1] Programm der Master-Station

[2] Zykluszeit der Slave-Station

6.7.4 EI, DI

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●[1]	●[1]				

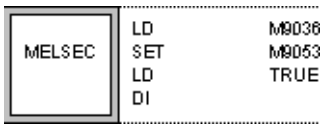
[1] Ein Link-Refresh ist nur bei gesetztem Merker M9053 ausführbar.

Operanden MELSEC A

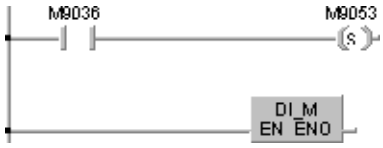
	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
																							1			

GX IEC Developer

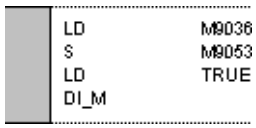
MELSEC-Anweisungsliste

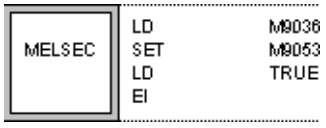


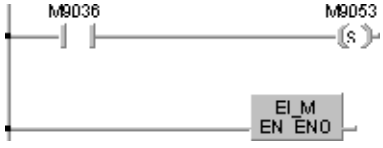
Kontaktplan

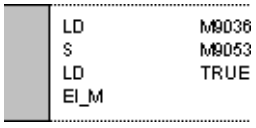


IEC-Anweisungsliste









Variablen

Operand	Befehlswert	Datentyp
—	—	—

Funktions-weise

Ausführungsbedingung für Link-Refresh

Allgemeines

Die Ausführung eines Link-Refreshs (siehe COM-Anweisung) erfolgt in Abhängigkeit von der EI-/DI-Anweisung. Die Bedeutung der EI-/DI-Anweisung im Zusammenhang mit einer AnN oder A2C-CPU hängt vom Status des Sondermerkers M9053 ab. Nur wenn der Merker gesetzt ist, dienen die Anweisungen als Ausführungsbedingungen für einen Link-Refresh. Ist der Merker M9053 nicht gesetzt, werden die Anweisungen als Ausführungsbedingungen für ein Interrupt-Programm genutzt.

DI Verhindern der Link-Refresh-Ausführung

Die DI-Anweisung verhindert die Ausführung eines Link-Refreshs so lange, bis eine EI-Anweisung in der Programmfolge erscheint. Nach dem Einschalten oder einem RESET der CPU ist der DI-Status inaktiv.

Ein Link-Refresh ist bei jeder END-Verarbeitung möglich.

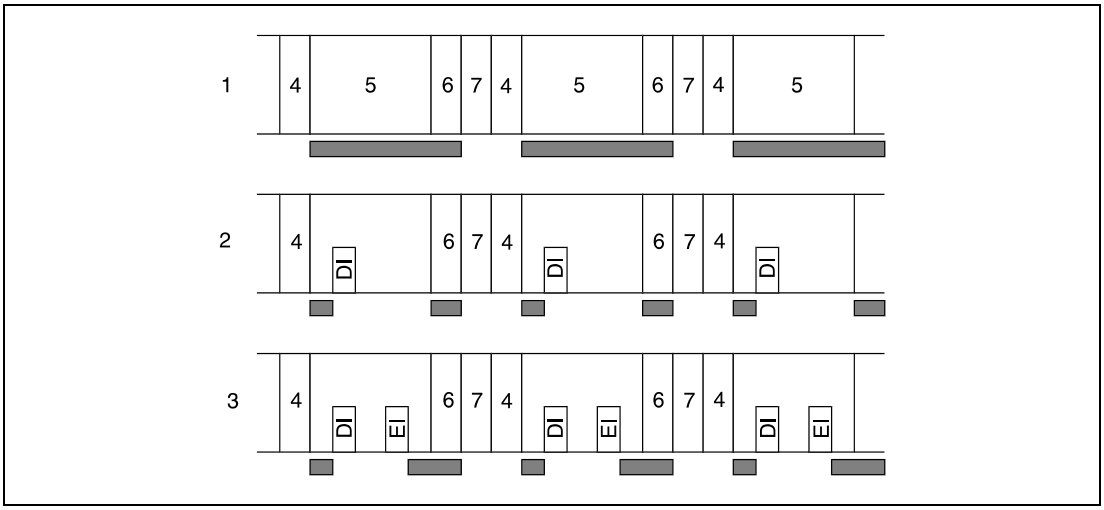
EI Ermöglichen der Link-Refresh-Ausführung

Die Ausführung eines Link-Refreshs wird nach dem Setzen einer EI-Anweisung möglich.

Ausführungsbedingungen

Die Abbildung zeigt die Ausführungsbedingungen der EI-/DI-Anweisung.

Die Verarbeitung von Netzwerkdaten ist an den schraffierten Bereichen möglich. Der Wartezyklus entfällt, wenn die Verarbeitung nicht mit konstanter Zykluszeit erfolgt. Bei Direktverarbeitung ist keine E-/A-Aktualisierung möglich.



[1] Programmverarbeitung ohne EI-/DI-Anweisung

[2] Programmverarbeitung mit DI-Anweisung

[3] Programmverarbeitung mit EI-/DI-Anweisung

[4] E/A-Refresh

[5] Ablaufverarbeitung

[6] END-Verarbeitung

[7] Wartezyklus

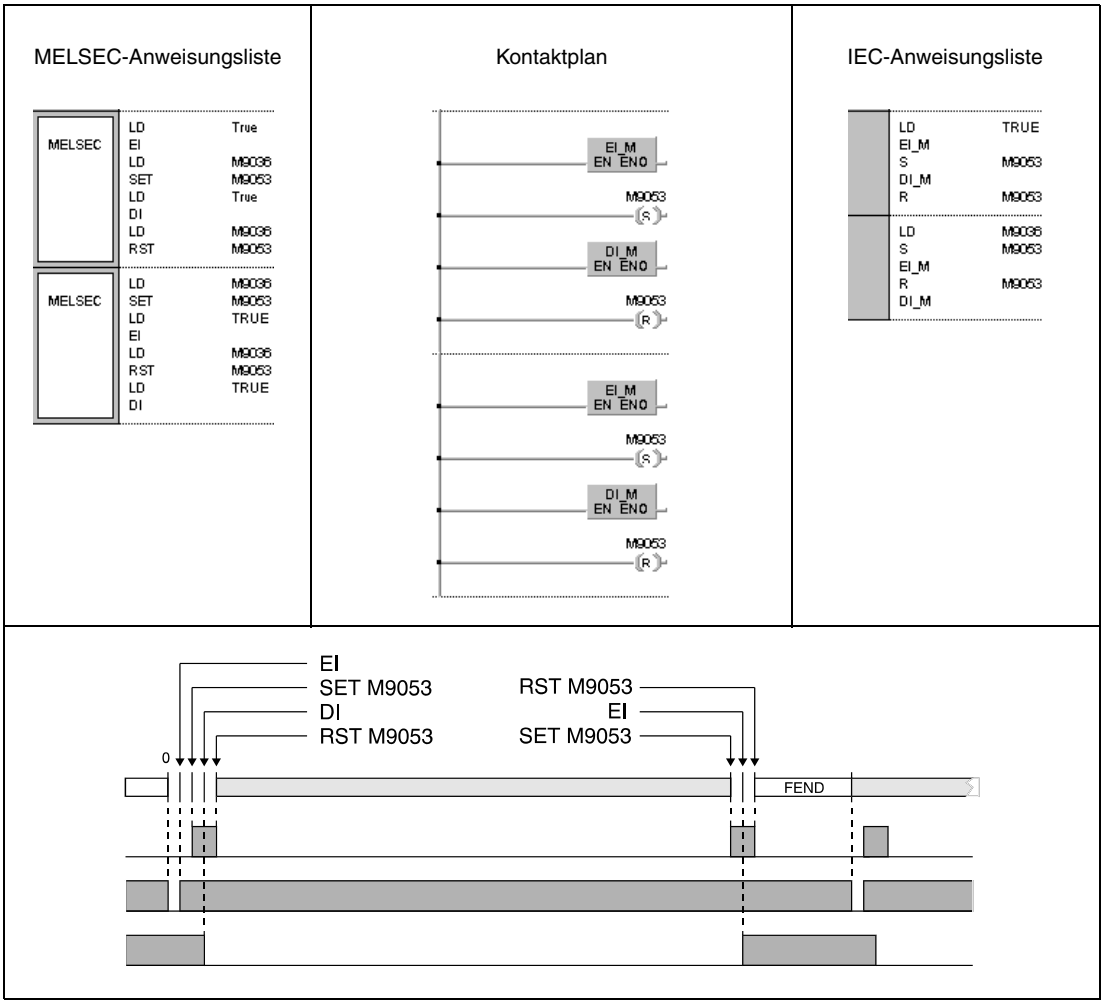
Die Verarbeitung erfolgt bei gegebener Ausführungsbedingung.

Die Bedeutung einer EI-/DI-Anweisung ist vom Zustand des Sondermerkers M9053 abhängig. Nach Ausführung einer EI-/DI-Anweisung kann M9053 ein- oder ausgeschaltet sein.

Befindet sich eine EI- oder DI-Anweisung innerhalb einer MC-Anweisung, erfolgt die Verarbeitung unabhängig von der Ausführung der MC-Anweisung.

Beispiel EI

Das folgende Programm unterbindet den Link-Refresh so lange, bis die EI-Anweisung kurz vor der Verarbeitung der FEND-Anweisung ausgeführt wird. Der Aufruf eines Interrupt-Progamms ist jederzeit möglich. Die Abbildung zeigt den Zeitverlauf der Programmverarbeitung.



6.8 Weitere Anweisungen

Die in der Tabelle aufgeführten Anweisungen ermöglichen die Programmierung von speziellen Timern und Countern, Impuls-Zählern und -Gebern. Ferner sind Anweisungen zur Positionierung von Rotationstischen und zur Bildung von Eingabe-Matrizen angegeben.

Funktion	MELSEC-Anweisung im MELSEC-EDITOR	MELSEC-Anweisung im IEC-EDITOR
Einphasiger Auf-/Abwärtszähler	UDCNT1	UDCNT1_M
Zweiphasiger Auf-/Abwärtszähler	UDCNT2	UDCNT2_M
Programierbarer Timer	TTMR	TTMR_M
Sonderfunktions-Timer	STMR	STMR_M
	STMRH	STMRH_M
Positionieranweisung für Rotationstische	ROTC	ROTC_M
Rampensignal	RAMP	RAMP_M
Impulszähler	SPD	SPD_M
Impulsausgang mit einstellbarer Anzahl von Impulsen	PLSY	PLSY_M
Puls-Weiten-Modulation	PWM	PWM_M
Bildung einer Eingabe-Matrix	MTR	MTR_M

6.8.1 UDCNT1

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●[1]

[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	●[1]	—	—	—	—	—	—	—	—	—	4
d	—	●[2]	—	—	—	—	—	—	—	—	
n	●	●	●	●	●	●	●	●	—	—	

[1] Nur X

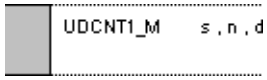
[2] Nur C

GX IEC Developer

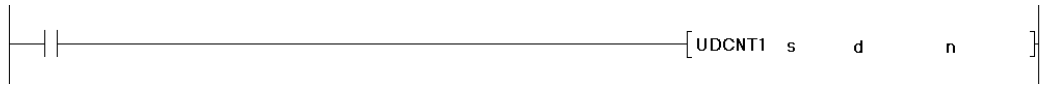
MELSEC-Anweisungsliste

Kontaktplan

IEC-Anweisungsliste



GX Developer



Variablen

Operand	Befehlswert	Datentyp	
		MELSEC	IEC
s	s+0: Zähleingang (Gepulstes Signal, Phase)	Bit	Array [1..2] of BOOL
	s+1: Zählrichtungsfestlegung 0 = Zählrichtung aufwärts 1 = Zählrichtung abwärts		
d	Zähler, der die UDCNT1-Anweisung ausführt.	BIN-16-Bit (nur Counter)	ANY16
n	Vorgabewert für Zählerkontakt.	BIN-16-Bit	ANY16

Funktionsweise

Einphasiger Auf-/Abwärtszähler

UDCNT1 Zähleranweisung

Wenn der in s+0 (Array_s [0]) angegebene Eingang von 0 auf 1 wechselt, wird der aktuelle Zählerstand des in d angegebenen Zählers aktualisiert. Das bedeutet, dass nur die ansteigenden Flanken gezählt werden.

Die Zählrichtung ist vom Zustand des in s+1 (Array_s [1]) angegebenen Eingangs abhängig:

Hat der Eingang den Zustand 0, werden die Impulse des in s+0 (Array_s [0]) angegebenen Eingangs zu dem aktuellen Zählerstand hinzuaddiert.

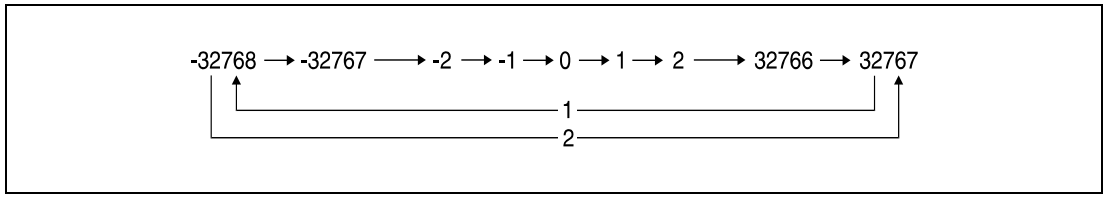
Hat dieser Eingang den Zustand 1, so werden die Impulse von dem aktuellen Zählerstand subtrahiert.

Der Zählvorgang wird wie nachfolgend beschrieben ausgeführt:

Der Zählerkontakt des in d angegebenen Zählers wird eingeschaltet (1), wenn der Zählerstand bei Aufwärtszählung mit dem in n angegebenen Zählerstand übereinstimmt. Der Zählvorgang wird auch bei gesetztem Zählerkontakt fortgesetzt (siehe Beispiel).

Bei Abwärtszählung wird der Zählerkontakt bei Übereinstimmung des Zählerstandes mit dem Wert n-1 wieder zurückgesetzt (siehe Beispiel).

Der in d angegebene Zähler ist ein Endloszähler. Lautet der aktuelle Zählerstand 32767, und wird dieser um 1 erhöht, springt der Zähler auf -32768 um. Ebenso springt der Zähler bei einem Zählerstand von -32768 und Verringerung um 1 auf 32767 um. Nachfolgende Abbildung veranschaulicht diesen Sachverhalt.



[1] Bei Aufwärtszählung

[2] Bei Abwärtszählung

Die UDCNT1-Anweisung wird bei Einschalten der Ausführungsbedingung gestartet und stoppt bei Ausschalten der Ausführungsbedingung. Bei erneutem Einschalten startet der Zähler an der Stelle, an der er zuvor gestoppt hat.

Mit der RST-Anweisung wird der Zählerinhalt des in d angegebenen Zählers gelöscht und der entsprechende Zählerkontakt ausgeschaltet.

HINWEISE

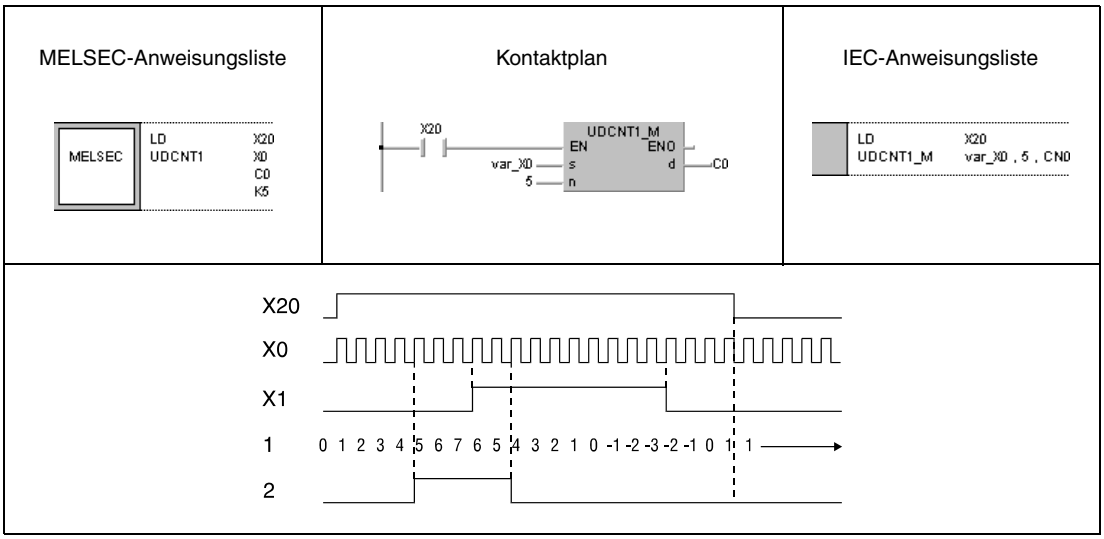
Der Zählvorgang für Zähler bei Nutzung einer UDCNT1-Anweisung wird während eines CPU-Interrupts (1 ms bei Multi-Prozessor-Q-CPU und 5 ms bei QnA-CPU) ausgeführt. Aus diesem Grund sollte die Tastdauer und die Pausenzeit der Impulse größer als 1 ms bzw. 5 ms sein, um den korrekten Zählvorgang zu ermöglichen.

Die Zählereinstellungen können während des Zählvorgangs nicht verändert werden (der in s+0 (Array_s [0]) *angegebene Eingang ist 1). Um Veränderungen vorzunehmen, muss der in s+0* (Array_s [0]*) angegebene Eingang auf 0 gesetzt sein.*

Mit einer UDCNT1-Anweisung verwendete Zähler können nicht gleichzeitig von anderen Anweisungen benutzt werden. Ist das der Fall, ist ein korrektes Zählergebnis der Zähler nicht mehr gewährleistet.

Beispiel UDCNT1

Im folgenden Programm wird der Zähler C0 (Auf-/Abwärtszähler) verwendet, um die positiven Flanken von X0 zu zählen, wenn X20 eingeschaltet wird.



[1] Zählerstand

[2] Zählerkontakt des Zählers C0

HINWEIS *Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

6.8.2 UDCNT2

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●[1]

[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	●[1]	—	—	—	—	—	—	—	—	—	4
d	—	●[2]	—	—	—	—	—	—	—	—	
n	●	●	●	●	●	●	●	●	—	—	

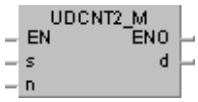
[1] Nur X

[2] Nur C

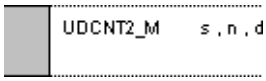
GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

Operand	Befehlswert	Datentyp	
		MELSEC	IEC
s	s+0: Zähleingang (Gepulstes Signal, Phase A)	Bit	Array [1..2] of BOOL
	s+1: Zähleingang (Gepulstes Signal, Phase B)		
d	Zähler, der die UDCNT2-Anweisung ausführt.	BIN-16-Bit (nur Counter)	ANY16
n	Vorgabewert für Zählerkontakt.	BIN-16-Bit	ANY16

Funktionsweise

Zweiphasiger Auf-/Abwärtszähler

UDCNT2 Zähleranweisung

Der Zählerstand des in d angegebenen Zählers wird abhängig von dem Zustand der beiden in s+0 (Array_s [0]) und s+1 (Array_s [1]) angegebenen Eingänge verändert.

Die Zählrichtung wird wie folgt bestimmt:

Wenn der in s+0 (Array_s [0]) angegebene Eingang den Zustand 1 hat und der in s+1 (Array_s [1]) angegebene Eingang von 0 auf 1 wechselt, wird der aktuelle Zählerstand um 1 erhöht.

Wenn der in s+0 (Array_s [0]) angegebene Eingang den Zustand 1 hat, der in s+1 (Array_s [1]) angegebene Eingang jedoch von 1 auf 0 wechselt, wird der aktuelle Zählerstand um 1 verringert.

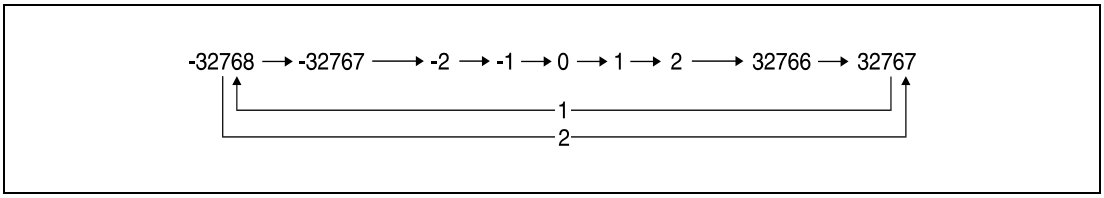
Wenn der in s+0 (Array_s [0]) angegebene Eingang den Zustand 0 hat, wird keine Zähloperation ausgeführt.

Der Zählvorgang wird wie nachfolgend beschrieben ausgeführt:

Der Zählerkontakt des in d angegebenen Zählers wird eingeschaltet (1), wenn der Zählerstand bei Aufwärtszählung mit dem in n angegebenen Zählerstand übereinstimmt. Der Zählvorgang wird auch bei gesetztem Zählerkontakt fortgesetzt (siehe Beispiel).

Bei Abwärtszählung wird der Zählerkontakt bei Übereinstimmung des Zählerstandes mit dem Wert n-1 wieder zurückgesetzt (siehe Beispiel).

Der in d angegebene Zähler ist ein Endloszähler. Lautet der aktuelle Zählerstand 32767 und wird dieser um 1 erhöht, springt der Zähler auf -32768 um. Ebenso springt der Zähler bei einem Zählerstand von -32768 und Verringerung um 1 auf 32767 um. Nachfolgende Abbildung veranschaulicht diesen Sachverhalt.



[1] Bei Aufwärtszählung

[2] Bei Abwärtszählung

Die UDCNT2-Anweisung wird bei Einschalten der Ausführungsbedingung gestartet und stoppt bei Ausschalten der Ausführungsbedingung. Bei erneutem Einschalten startet der Zähler an der Stelle, an der er zuvor gestoppt hat.

Mit der RST-Anweisung wird der Zählerinhalt des in d angegebenen Zählers gelöscht und der entsprechende Zählerkontakt ausgeschaltet.

HINWEISE

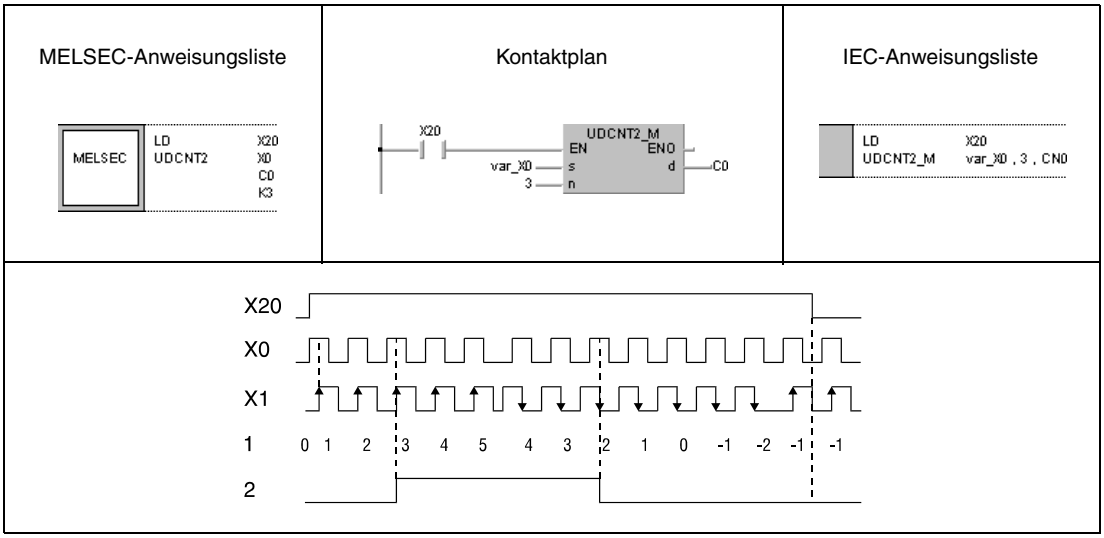
Der Zählvorgang für Zähler bei Nutzung einer UDCNT2 -Anweisung wird während eines CPU-Interrupts (1 ms bei Multi-Prozessor-Q-CPU und 5 ms bei QnA-CPU) ausgeführt. Aus diesem Grund sollten die Tastdauer und die Pausenzeit der Impulse größer als 1 ms bzw. 5 ms sein, um den korrekten Zählvorgang zu ermöglichen.

Die Zählereinstellungen können während des Zählvorgangs (der in s+0 (Array_s [0]) angegebene Eingang ist 1) nicht verändert werden. Um Veränderungen vorzunehmen, muss der in s+0 (Array_s [0]) angegebene Eingang auf 0 gesetzt sein.

Mit einer UDCNT2-Anweisung verwendete Zähler können nicht gleichzeitig von anderen Anweisungen benutzt werden. Ist das der Fall, ist ein korrektes Zählergebnis der Zähler nicht mehr gewährleistet.

Beispiel UDCNT2

Das folgende Programm führt eine Zähloperation mit dem Zähler C0 nach Einschalten von X20 durch. Das Zählergebnis und die Zählrichtung (auf-/abwärts) sind von den Zuständen von X0 und X1 abhängig.



HINWEIS *Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

6.8.3 TTMR

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●[1]

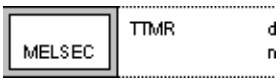
[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

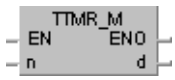
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
d	—	●	●	—	—	—	—	—	—	SM0	3
n	—	●	●	●	●	●	●	●	—		

GX IEC Developer

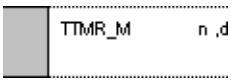
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

Operand	Befehlswert	Datentyp	
		MELSEC	IEC
d	d+0: Operand, in dem der Messwert gespeichert wird.	BIN-16-Bit	Array [1..2] of ANY16
	d+1: Wird CPU-systemintern genutzt.		
n	Wert, mit dem der Messwert multipliziert wird.		ANY16

Funktions-weise

Programmierbarer Timer

TTMR Timeranweisung

Der mit der TTMR-Anweisung programmierbare Timer misst die Zeit in Sekunden, die am Eingang der Anweisung ansteht. Das Messergebnis wird mit dem in n angegebenen Faktor multipliziert und in dem in d (Array_d [0]+[1]) angegebenen Operanden gespeichert.

Bei der ansteigenden Flanke des Messbefehls am Eingang der Anweisung wird der Inhalt von d+0 (Array_d [0]) und d+1 (Array_d [1]) gelöscht.

Angabe des Multiplikationsfaktor in n:

n = 0, Faktor 1

n = 1, Faktor 10

n = 2, Faktor 100

HINWEISE

Die Zeitmessung erfolgt während der Ausführung der TTMR-Anweisung. Die Anwendung einer JMP-Anweisung oder einer ähnlichen Anweisung auf die TTMR-Anweisung hat eine fehlerhafte Zeitmessung zur Folge.

Eine Veränderung des in n angegebenen Faktors während der Ausführung der TTMR-Anweisung verfälscht das Messergebnis.

Die TTMR-Anweisung kann auch in Programmen mit geringer Verarbeitungsgeschwindigkeit verwendet werden.

Der in d+1 (Array_d [1]) angegebene Wert dient der systeminternen Verarbeitung und sollte nicht verändert werden. Eine Veränderung dieses Wertes führt zu einer Verfälschung des in d+0 (Array_d [0]) abgespeicherten Ergebnisses.

Fehler-quellen

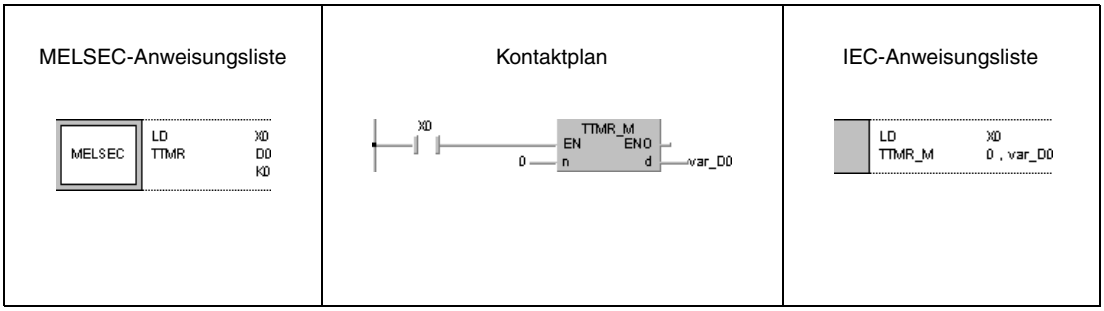
In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Der in n abgegebene Wert liegt außerhalb des Bereichs zwischen 0 und 2 (Fehlercode 4100).

Beispiel

TTMR

Das folgende Programm misst die Einschaltdauer von X0 in Sekunden (Faktor 1) und speichert das Ergebnis in D0.



HINWEIS

Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.

6.8.4 STMR, STMRH

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●[1]

[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

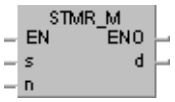
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sonder-module U□\G□	Index-Register Zn	Konstanten	Andere		
	Bit	Wort		Bit	Wort						
s	—	●[1]	—	—	—	—	—	—	—	—	3
n	●	—	—	—	—	—	—	—	—	—	
d	—	●	●	●	●	●	●	●	—	—	

[1] Kann nur zur Auswahl der Timer (T) verwendet werden.

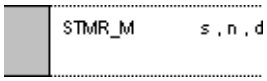
GX IEC Developer

MELSEC-Anweisungsliste

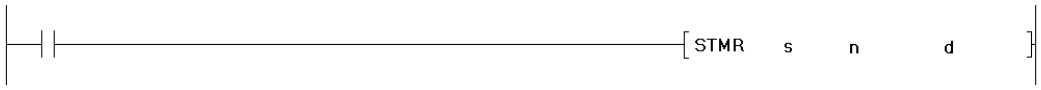
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

Operand	Befehlswert	Datentyp	
		MELSEC	IEC
s	Timer-Nummer	BIN-16-Bit (nur Timer)	ANY16
n	Zeiteinstellung	BIN-16-Bit	ANY16
d	d+0: Ausschaltverzögerter Timerausgang.	Bit	Array [1..4] of BOOL
	d+1: Timerausgang mit Setzimpuls (Setzen durch abfallende Flanke des Befehls).		
	d+2: Timerausgang mit Setzimpuls (Setzen durch ansteigende Flanke des Befehls).		
	d+3: Einschaltverzögerter Timerausgang.		

Funktionsweise

Sonderfunktions-Timer

STMR	**Timeranweisung für langsame Timer (Low-Speed-Timer)**
STMRH	**Timeranweisung für schnelle Timer (High-Speed-Timer)**

Die STMR-Anweisung nutzt die in d+0 (Array_d [0]) bis d+3 (Array_d [3]) angegebenen Ausgänge, um vier unterschiedliche Timerfunktionen zu realisieren:

Ausschaltverzögerter Timerausgang (d+0) (Array_d [0])

Der in d+0 (Array_d [0]) angegebene Ausgang wird nach der ansteigenden Flanke der Ausführungsbedingung gesetzt und fällt nach der abfallenden Flanke der Ausführungsbedingung um die durch n bestimmte Zeit verzögert ab.

Timerausgang mit Setzimpuls (Setzen bei abfallender Flanke des Befehls, d+1 (Array_d [1]))

Der in d+1 (Array_d [1]) angegebene Ausgang wird bei der abfallenden Flanke der Ausführungsbedingung gesetzt. Der Ausgang wird nach Ablauf der durch n bestimmten Zeit oder bei ansteigender Flanke der Ausführungsbedingung zurückgesetzt.

Timerausgang mit Setzimpuls
(Setzen bei ansteigender Flanke der Ausführungsbedingung, d+2 (Array_d [2]))

Der in d+2 (Array_d [2]) angegebene Ausgang wird bei der ansteigenden Flanke der Ausführungsbedingung gesetzt. Das Rücksetzen erfolgt nach Ablauf der durch n bestimmten Zeit oder bei der abfallenden Flanke der Ausführungsbedingung.

Einschaltverzögerter Timerausgang (d+3 (Array [3]))

Der in d+3 (Array_d [3]) angegebene Ausgang wird bei der abfallenden Flanke der Timerspule gesetzt. Das entspricht einer Einschaltverzögerung um die durch n bestimmte Zeit. Der Ausgang wird ebenfalls bei der abfallenden Flanke der Ausführungsbedingung gesetzt und fällt erst nach Ablauf der durch n bestimmten Zeit wieder ab.

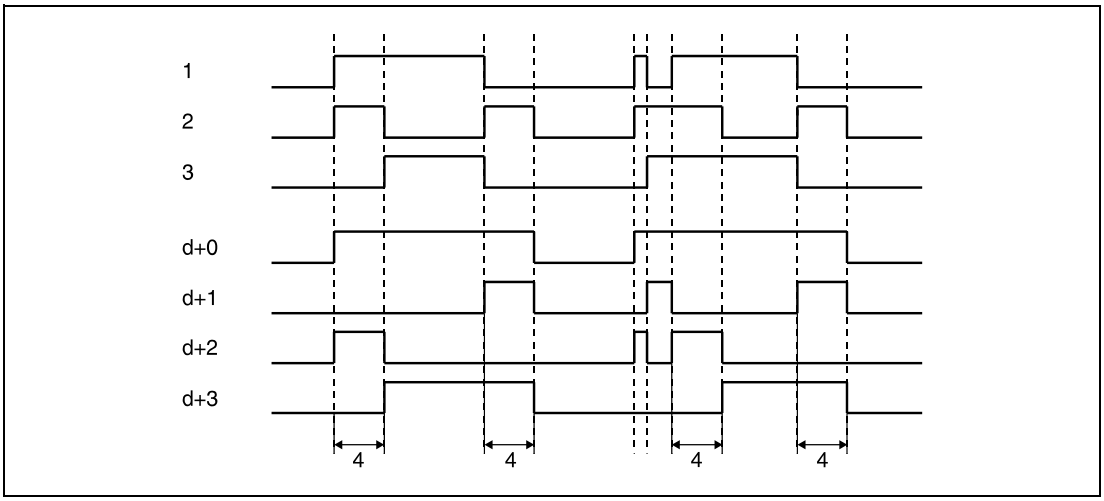
Die Timerspule des in s angegebenen Timers wird bei der ansteigenden Flanke der Ausführungsbedingung gesetzt und beginnt mit der Messung der durch n bestimmten Zeit.

Die Timerspule misst bis zu dem Punkt, an dem die gemessene Zeit mit der durch n bestimmten Zeit übereinstimmt, und fällt ab.

Wird die Ausführungsbedingung vor Ablauf der durch n bestimmten Zeit zurückgesetzt, bleibt die Timerspule gesetzt, und die Zeitmessung wird an dieser Stelle unterbrochen.

Bei erneutem Einschalten der Ausführungsbedingung wird der Messwert gelöscht und die Zeitmessung beginnt von neuem.

Der Timerkontakt des in s angegebenen Timers wird entweder bei gesetzer Timerspule und abfallender Flanke der Ausführungsbedingung oder bei gesetzter Ausführungsbedingung und abfallender Flanke der Timerspule gesetzt. Der Timerkontakt wird bei nicht gesetzter Timerspule und abfallender Flanke der Ausführungsbedingung zurückgesetzt. Der Timerkontakt dient ausschließlich der CPU-systeminternen Verarbeitung.



[1] Ausführungsbedingung

[2] Spule des in s angegebenen Timers

[3] Kontakt des in s angegebenen Timers

[4] In n angegebene Zeit

Die Messung der aktuell abgelaufenen Zeit erfolgt während der Ausführung der SMTR-Anweisung. Wenn auf die SMTR-Anweisung eine JMP-Anweisung oder eine ähnliche Anweisung angewendet wird, ist eine korrekte Zeitmessung nicht mehr möglich.

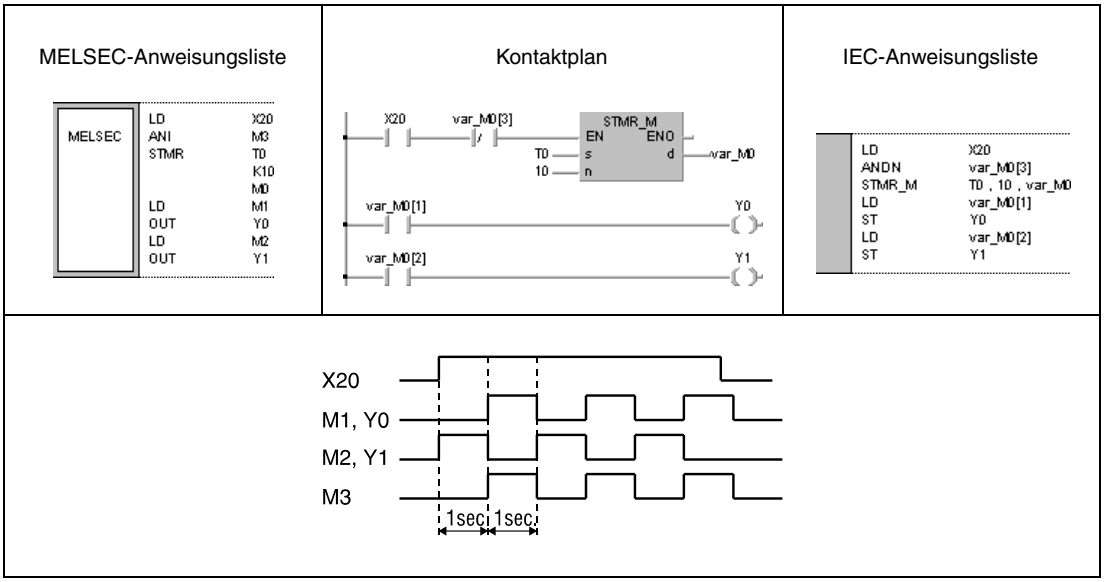
Die tatsächlichen Zeitangaben für die in d angegebenen Timerausgänge berechnen sich aus der in n angegebenen Konstante multipliziert mit der Maßeinheit für langsame Timer (Standardeinstellung 100 ms).

Die in n angegebene Konstante kann zwischen 1 und 32767 liegen.

Der in s angegebene Timer kann nicht in einer OUT-Anweisung verwendet werden. Verwenden eine OUT-Anweisung und eine SMTR-Anweisung den gleichen Timer, kann die SMTR-Anweisung nicht fehlerfrei ausgeführt werden.

Beispiel SMTR

Das folgende Programm schaltet die Ausgänge Y0 und Y1 abwechselnd für jeweils 1 Sekunde ein, wenn X20 eingeschaltet wird. Der hier verwendete Timer ist ein 100-ms-Timer. Die Einschaltdauer von einer Sekunde berechnet sich aus der Konstante K10 multipliziert mit 100 ms.



HINWEIS *Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

6.8.5 ROTC

CPU

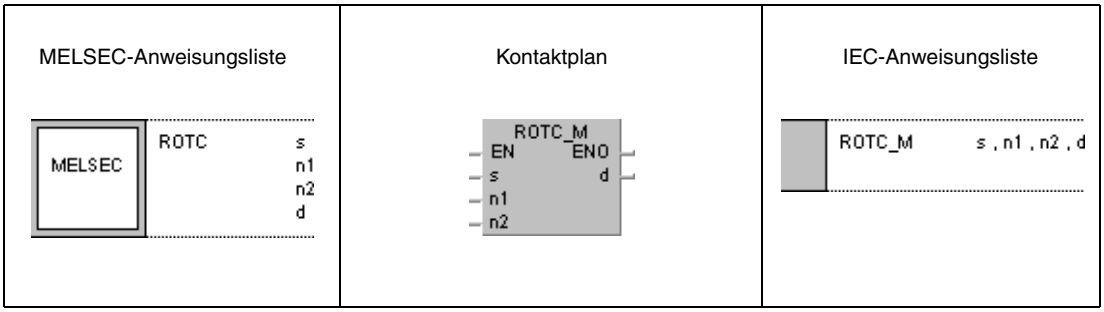
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●[1]

[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

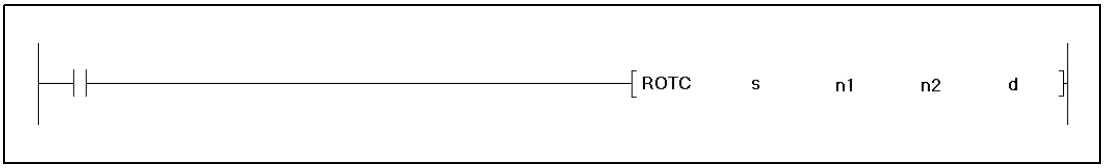
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	—	—	—	—	—	SM0	5
n1	●	●	●	●	●	●	●	●	—		
n2	●	●	●	●	●	●	●	●	—		
d	●	—	—	—	—	—	—	—	—		

GX IEC Developer



GX Developer



Variablen

Operand	Befehlswert	Datentyp	
		MELSEC	IEC
s	s+0: Messung der Umdrehungen pro Minute (systemintern).	BIN 16 Bit	Array [1..3] of ANY16
	s+1: Nummer der Position.		
	s+2: Nummer des Sektors.		
n1	Teilung (von 2 bis 32767)		ANY16
n2	Anzahl der Abschnitte für niedrige Rotationsgeschwindigkeit (Schleichgang) (von 0 bis n1).		ANY16

Variablen

Operand	Befehlswert	Datentyp	
		MELSEC	IEC
d	d+0: Eingangssignal (Phase A)	Bit	Array [1..8] of Bool
	d+1: Eingangssignal (Phase B)		
	d+2: Eingangssignal der Nullpunktsbestimmung.		
	d+3: Ausgangssignal vorwärts, hohe Drehzahl (systemintern).		
	d+4: Ausgangssignal vorwärts, niedrige Drehzahl (systemintern).		
	d+5: Ausgangssignal Stopp (systemintern).		
	d+6: Ausgangssignal rückwärts, hohe Drehzahl (systemintern).		
	d+7: Ausgangssignal rückwärts, niedrige Drehzahl (systemintern).		

Funktionsweise

Positionieranweisung für Rotationstische

ROTC Positionieranweisung

Mit der ROTC-Anweisung ist es möglich, einen Rotationstisch mit der in n1 angegebenen Anzahl von Sektoren (Teilung) derart zu positionieren, das ein in s+2 (Array_s [2]) angegebener Sektor an eine in s+1 (Array_s [1]) angegebene Position gefahren wird.

Die Numerierung der Positionen und der Sektoren auf dem Rotationstisch erfolgt gegen den Uhrzeigersinn.

Der in s+0 (Array_s [0]) angegebene Wert wird systemintern als Zähler genutzt und gibt dem System die Information, welcher Sektor sich relativ zur Position 0 an welcher Stelle befindet. Der in s+0 (Array_s [0]) angegebene Wert darf nicht verändert werden, da sonst eine Positionierung nicht mehr möglich ist.

Der in n2 angegebene Wert gibt die Anzahl der Sektoren an, in denen der Rotationstisch mit niedriger Geschwindigkeit (Schleichgang) verfahren wird. Er muss kleiner als die in n1 angegebene Teilung sein.

Die in d+0 (Array_d [0]) und d+1 (Array_d [1]) angegebenen Eingänge (Phase A/B) dienen der Drehrichtungsermittlung. Beide Eingänge werden mit Impulsen beschaltet. Hat der in d+0 (Array_d [0]) angegebene Eingang (Phase A) den Zustand 1, wird die Drehrichtung über die Flanke des Impulses des in d+1 (Array_d [1]) angegebenen Eingangs (Phase B) wie folgt ermittelt:

Hat die Phase B zu diesem Zeitpunkt eine ansteigende Flanke, ist die Drehrichtung im Uhrzeigersinn (rechtsdrehend).

Hat die Phase B zu diesem Zeitpunkt eine abfallende Flanke, ist die Drehrichtung gegen den Uhrzeigersinn (linksdrehend).

Der in d+2 (Array_d [2]) angegebene Eingang dient der Nullpunktbestimmung. Dieser Eingang wird gesetzt, wenn Sektor 0 die Position 0 erreicht. Wird dieser Eingang während der Ausführung der ROTC-Anweisung gesetzt, wird der Wert in s+0 (Array_s [0]) zurückgesetzt. Es ist empfehlenswert, diese Rücksetzoperation des in s+0 (Array_s [0]) angegebenen Wertes vor der eigentlichen Positionierung mit der ROTC-Anweisung vorzunehmen, um eine einwandfreie Funktion zu gewährleisten.

Die Daten in d+3 (Array_d [3]) bis d+7 (Array_d [7]) beinhalten Ausgangssignale zur Ansteuerung des Rotationstisches. Welches Ausgangssignal gesetzt wird, hängt von dem aktuellen Ausführungsergebnis der ROTC-Anweisung ab.

Wenn unmittelbar vor der Ausführung der ROTC-Anweisung alle Operationsergebnisse 0 waren, werden die in d+3 (Array_d [3]) bis d+7 (Array_d [7]) angegebenen Ausgänge ohne Ausführung einer Positionierung zurückgesetzt. Nach Abschalten der Ausführungsbedingung werden diese Ausgänge ebenfalls zurückgesetzt.

Die ROTC-Anweisung kann nur einmal in einem Programm benutzt werden. Mehrmalige Verwendung in einem Programm führt zu einer fehlerhaften Ausführung der Anweisung.

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Der in s+0 (Array_s [0]) bis s+2 (Array_s [2]) oder n2 angegebene Wert ist größer als der in n1 angegebene Wert (Fehlercode 4100).

Beispiel ROTC

Im folgenden Programm werden die Merker zur Drehrichtungs- und Nullpunktsbestimmung M0 (var_M0 Array [0]) bis M2 (var_M0 Array [2]) durch die Kontakte X0, X1 (Inkrementalgeber) und X2 angesprochen. Der Kontakt X2 wird aktiviert, wenn sich Sektor 0 an Position 0 befindet (Nullpunktsbestimmung).

Auf dem hier abgebildeten Rotationstisch befinden sich 10 Teile (Sektoren) .

Welches Teil (Sektor) zu welchem Greifer (Position) gefahren wird, ist vor Ausführung der ROTC-Anweisung in D201 (var_D200 Array [1]) und D202 (var_D200 Array [2]) anzugeben.

Durch Angabe des Wertes n1=10 gibt der Kontakt des Zählregisters bei einer Umdrehung des Rotationstisches 10 Impulse aus (Teilung). Mit n2=2 ist die Anzahl der Teilungen angegeben, die im Schleichgang gefahren werden.

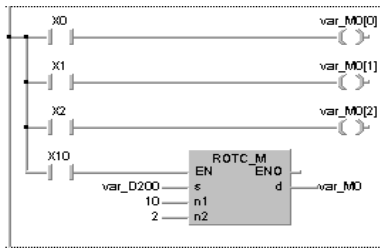
Wird z.B. in das Register D201 (var_D200 Array [1]) der Wert 0 und in Register D202 (var_D200 [2]) der Wert 3 angegeben, fährt der Rotationstisch auf dem kürzesten Weg (rechtsdrehend) das Teil 3 (Sektor 3) zu dem Greifer 0 (Position 0), wobei der Weg von Sektor 1 bis Sektor 3 im Schleichgang zurückgelegt wird.

Eine Zuordnung der einzelnen Register, Merker bzw. Arrayelemente zu den entsprechenden Funktionen ist in der auf das Beispiel folgenden Tabelle angegeben.

MELSEC-Anweisungsliste

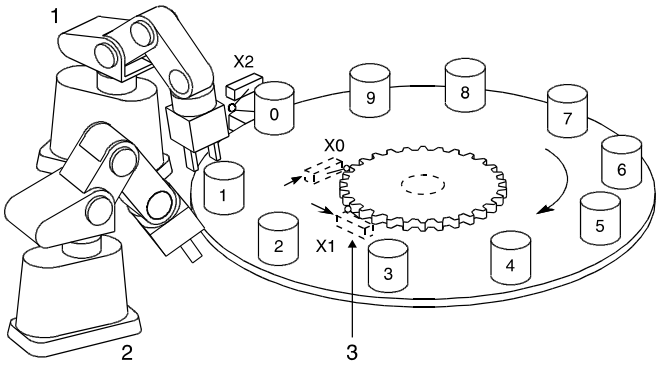
```
MELSEC
LD    X0
OUT   M0
LD    X1
OUT   M1
LD    X2
OUT   M2
LD    X10
ROTC  D200
      K10
      K2
      M0
```

Kontaktplan



IEC-Anweisungsliste

```
LD    X0
ST    var_M0[0]
LD    X1
ST    var_M0[1]
LD    X2
ST    var_M0[2]
LD    X10
ROTC_M  var_D200 , 10 , 2 , var_M0
```



[1] Position 0

[2] Position 1

[3] Inkrementalgeber

Datenregister	Beschreibung	Anmerkung
D200 (var_D200 Array [0])	Zählregister	
D201 (var_D200 Array [1])	Position des Greifers.	Die Werte werden zuvor über die MOV-Anweisung in die Datenregister D200 (var_D200 Array [0]) bis D202 (var_D200 Array [2]) geschrieben.
D202 (var_D200 Array [2])	Position des gewünschten Teiles.	
M0 (var_M0 Array [0])	A-Phasen-Signal	Die Merker M0 (var_M0 Array [0]) bis M2 (var_M0 Array [2]) werden über die Eingänge X0 bis X2 gesetzt (siehe Programm-beispiel).
M1 (var_M0 Array [1])	B-Phasen-Signal	
M2 (var_M0 Array [2])	Nullpunktsignal	
M3 (var_M0 Array [3])	Vorwärtsdrehung (High Speed)	Nach Einschalten von X10 wird die ROTC-Anweisung aktiviert, und den Merkern M3 (var_M0 Array [3]) bis M7 (var_M0 Array [7]) werden automatisch feste Funktionen zugeordnet. Nach Ab-schalten von X10 werden diese Merker zurückgesetzt.
M4 (var_M0 Array [4])	Vorwärtsdrehung (Low Speed) Schleichgang.	
M5 (var_M0 Array [5])	Stopp	
M6 (var_M0 Array [6])	Rückwärtsdrehung (High Speed)	
M7 (var_M0 Array [7])	Rückwärtsdrehung (Low Speed) Schleichgang.	

HINWEIS *Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

6.8.6 RAMP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●[1]

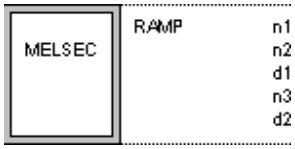
[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

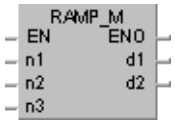
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
n1	●	●	●	●	●	●	●	●	—	—	6
n2	●	●	●	●	●	●	●	●	—	—	
d1	●	●	●	●	●	●	●	—	—	—	
n3	●	●	●	●	●	●	●	●	—	—	
d2	●	—	—	—	—	—	—	—	—	—	

GX IEC Developer

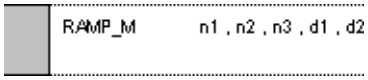
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

Operand	Befehlswert	Datentyp	
		MELSEC	IEC
n1	Startwert der Operation.	BIN 16 Bit	ANY16
n2	Endwert der Operation.		ANY16
d1	(d1)+0: Operand, in dem der aktuelle Wert gespeichert ist.		Array [1..2] of ANY16
	(d1)+1: Operand, in dem die Anzahl der ausgeführten Schritte gespeichert ist (systemintern).		
n3	Anzahl der auszuführenden Schritte.		ANY16
d2	(d2)+0: Bit, das nach vollständiger Ausführung der Anweisung gesetzt wird.	Bit	Array [1..2] of Bool
	(d2)+1: Bit, dessen Zustand die Speicherung des Operationsergebnisses festlegt.		

Funktionsweise

Rampensignal

RAMP Anweisung zum stufenweisen Erhöhen eines Speicherinhalts

Die RAMP-Anweisung erhöht stufenweise den Inhalt in (d1)+0 (Array_d1 [0]) von dem in n1 angegebenen Startwert bis zu dem in n2 angegebenen Endwert.

In n3 ist die Anzahl der Schritte angegeben, in denen die oben genannte Erhöhung durchgeführt werden soll.

In (d1)+1 (Array_d1 [1]) wird die Anzahl der bereits durchgeführten Schritte zur systeminternen Verarbeitung gespeichert.

Wenn die Operation abgeschlossen ist, wird der in (d2)+0 (Array_d2 [0]) angegebene Operand gesetzt.

Der Signalzustand von dem in (d2)+0 (Array_d2 [0]) angegebenen Operanden und der Inhalt von (d1)+0 (Array_d1 [0]) sind von dem Signalzustand des in (d2)+1 (Array_d2 [1]) angegebenen Operanden wie folgt abhängig:

Wenn der in (d2)+1 (Array_d2 [1]) angegebene Operand nicht gesetzt ist, wird der in (d2)+0 (Array_d2 [0]) angegebene Operand beim nächsten Schritt zurückgesetzt, und die RAMP-Anweisung führt eine erneute Erhöhung des zuletzt in (d1)+0 (Array_d1 [0]) gespeicherten Wertes durch.

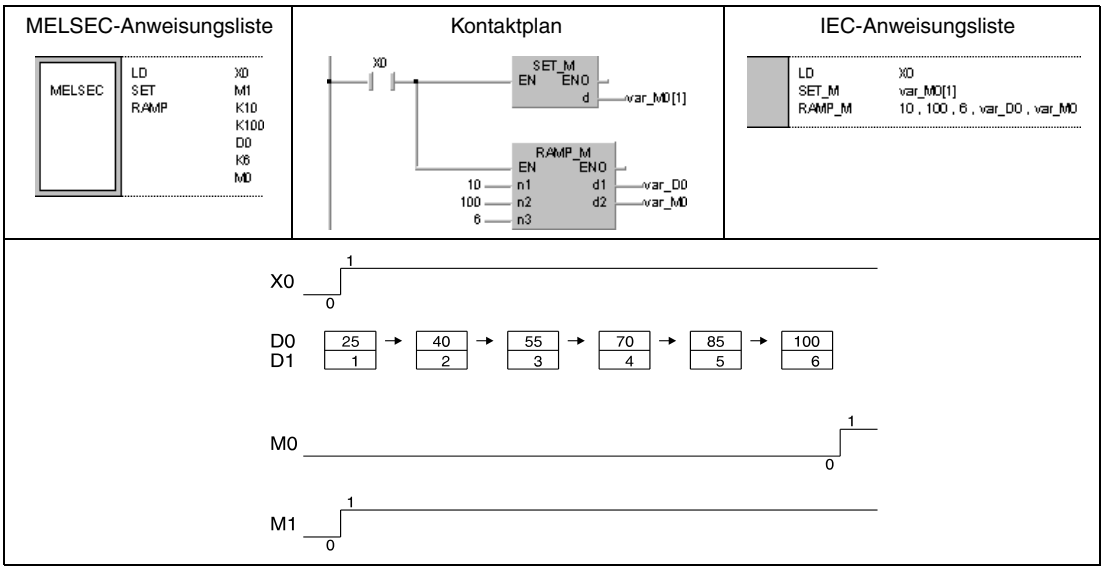
Wenn der in (d2)+1 (Array_d2 [1]) angegebene Operand gesetzt ist, bleibt auch der in (d2)+0 (Array_d2 [0]) angegebene Operand gesetzt, und der in (d1)+0 (Array_d1 [0]) angegebene Wert verändert sich nicht (Speicherung). Wenn die Ausführungsbedingung während der Operation auf 0 gesetzt wird, ändern sich die Inhalte in (d1)+0 (Array_d1 [0]) nicht. Wird die Ausführungsbedingung erneut gesetzt, führt die RAMP-Anweisung eine Erhöhung des vor dem Rücksetzen der Ausführungsbedingung gespeicherten Inhaltes von (d1)+0 (Array_d1 [0]) durch.

Die Werte in n1 und n2 dürfen während der Ausführung der Anweisung nicht verändert werden.

Beispiel

RAMP

Das folgende Programm erhöht den Inhalt von D0 in 6 Schritten von 10 auf 100 und speichert den Inhalt von D0, wenn die Operation abgeschlossen ist.



HINWEIS *Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

6.8.7 SPD

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●[1]

[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

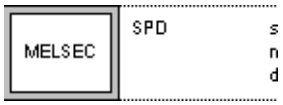
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten	Andere		
	Bit	Wort		Bit	Wort						
s	●[1]	—	—	●	●	●	●	●	—	—	4
n	●	●	●	—	—	—	—	—	—	—	
d	—	●	●	●	●	●	●	●	—	—	

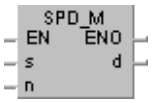
[1] Nur X

GX IEC Developer

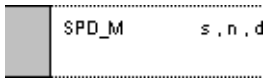
MELSEC-Anweisungsliste



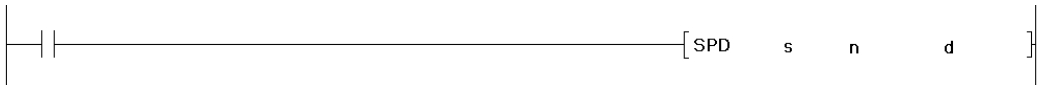
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

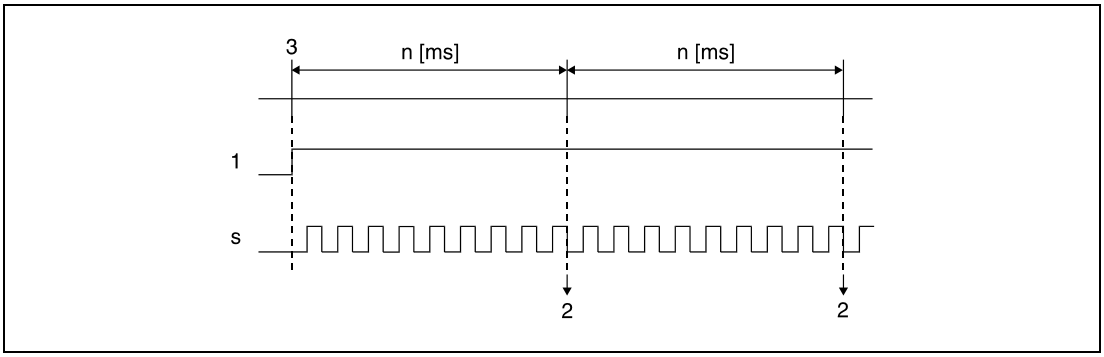
Operand	Befehlswert	Datentyp
s	Getaktetes Eingangssignal.	Bit
n	Messdauer in ms.	BIN-16-Bit
d	Startadresse, ab welcher das Messergebnis gespeichert wird.	

Funktions-weise

Impulszähler

SPD Zählanweisung für Impulszähler

Die SPD-Anweisung zählt die Impulse des in s angegebenen Eingangs für die in n angegebene Dauer und speichert das Ergebnis der Messung in dem in d angegebenen Operanden.



[1] Ausführungsbedingung

[2] Das Messergebnis wird in d gespeichert.

[3] Beginn der Messung

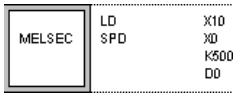
So lange die Ausführungsbedingung gesetzt ist, beginnt die Messung nach Ablauf der Messzeit erneut und die Zählung beginnt wieder bei 0. Um die von der SPD-Anweisung durchgeführte Messung zu stoppen, muss der Ausführungsbefehl zurückgesetzt werden.

HINWEIS *Die SPD-Anweisung speichert die Daten der Operanden im CPU-Arbeitsbereich und führt die Messung während eines Systeminterrupts aus. Um alle Impulse zu erfassen, muss deren Dauer gleich lang oder länger als die Zeit zwischen den Systeminterrupts sein. Bei Multi-Prozessor-Q-CPUs beträgt diese Zeit 1 ms, bei QnA-CPUs wird alle 5 ms ein Interrupt ausgelöst.*
Die Anweisung wird nicht ausgeführt, wenn bei Verwendung der Q-CPU n = 0 ist oder wenn bei der QnA-CPU n = 0 ist oder wenn n nicht ein Vielfaches von 5 ist.
Die SPD-Anweisung kann maximal sechs mal im gesamten Programm benutzt werden. Die siebte und alle folgenden SPD-Anweisungen werden nicht ausgeführt.

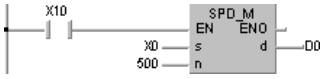
Beispiel SPD

Das folgende Programm misst nach dem Einschalten von X10 die Impulse des Eingangs X0 für die Dauer von 500 ms und speichert das Ergebnis in D0.

MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste

```
LD    X10
SPD_M X0 , 500 , D0
```

6.8.8 PLSY

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●[1]

[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

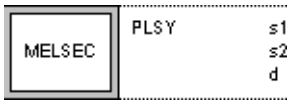
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten	Andere		
	Bit	Wort		Bit	Wort						
s1	●	●	●	●	●	●	●	●	—	—	4
s2	●	●	●	●	●	●	●	●	—	—	
d	●[1]	—	—	—	—	—	—	—	—	—	

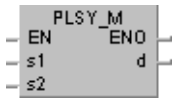
[1] Nur Y

GX IEC Developer

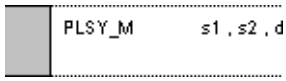
MELSEC-Anweisungsliste



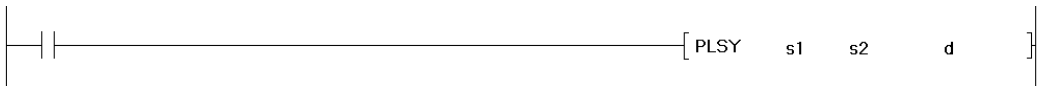
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

Operand	Befehlswert	Datentyp
s1	Operand, in dem die Frequenz gespeichert ist.	BIN-16-Bit
s2	Operand, in dem die Anzahl der Ausgangsimpulse gespeichert ist.	
d	Operand, in dem der Ausgang gespeichert ist.	Bit

Funktionsweise

Impulsausgang mit einstellbarer Anzahl von Impulsen

PLSY Impulsanweisung

Die PLSY-Anweisung gibt Impulse mit der in s1 angegebenen Frequenz und der in s2 angegebenen Anzahl an den in d angegebenen Ausgang aus.

Die in s1 angegebene Frequenz kann zwischen 1 und 100 Hz liegen. Wurde in s1 der Wert 0 eingetragen, gibt die PLSY-Anweisung ein Dauersignal aus.

Die in s2 angegebene Anzahl von Impulsen kann zwischen 1 und 32767 liegen.

In d können nur Ausgänge angegeben werden, die auch den Ausgangsmodulen zugeordnet sind.

Die Impulsausgabe beginnt mit der ansteigenden Flanke der Ausführungsbedingung der PLSY-Anweisung. Während der Impulsausgabe darf die Ausführungsbedingung nicht zurückgesetzt werden. Ein Zurücksetzen der Ausführungsbedingung beendet die Impulsausgabe.

HINWEIS

Die PLSY-Anweisung speichert die Daten der Operanden im CPU-Arbeitsbereich. Die Auffrischung der mit der PLSY-Anweisung ausgegebenen Impulse wird während Systeminterrupts ausgeführt. Um alle Impulse zu erfassen, muss deren Dauer gleich lang oder länger als die Zeit zwischen den Systeminterrupts sein. Bei den Multi-Prozessor-Q-CPUs beträgt diese Zeit 1 ms, bei den QnA-CPUs wird alle 5 ms ein Interrupt ausgelöst.

Die PLSY-Anweisung darf nicht verändert werden, wenn die Ausführungsbedingung erfüllt ist. Ansonsten kann die Ausgabe der Impulse nicht gestoppt werden, wenn die Ausführungsbedingung rückgesetzt wird.

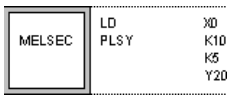
Die PLSY-Anweisung kann nur einmal im gesamten Programm verwendet werden.

Beispiel

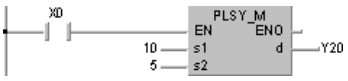
PLSY

Das folgende Programm gibt nach dem Einschalten von X0 fünf 10 Hz Impulse an Y20 aus.

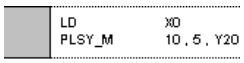
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



6.8.9 PWM

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●[1]

[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

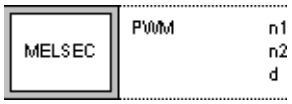
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten	Andere		
	Bit	Wort		Bit	Wort						
n1	●	●	●	●	●	●	●	●	—	—	4
n2	●	●	●	●	●	●	●	●	—	—	
d	●[1]	—	—	—	—	—	—	—	—	—	

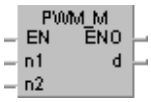
[1] Nur Y

GX IEC Developer

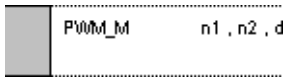
MELSEC-Anweisungsliste



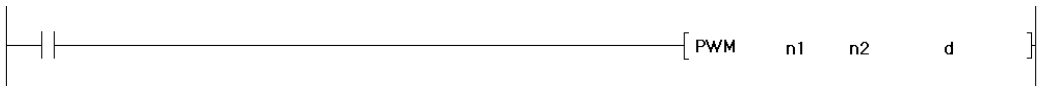
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

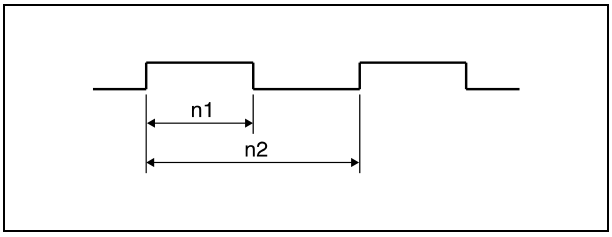
Operand	Befehlswert	Datentyp
n1	Operand, in dem die Impulsdauer gespeichert ist.	BIN-16-Bit
n2	Operand, in dem die Periodendauer gespeichert ist.	
d	Operand, an dem der Impuls ausgegeben wird	Bit

Funktionsweise

Puls-Weiten-Modulation

PWM Modulationsanweisung

Die PWM-Anweisung gibt Impulse mit der in n2 angegebenen Periodendauer und der in n1 angegebenen Impulsdauer an den in d angegebenen Ausgang aus.



Die in n1 und n2 angegebenen Werte können bei Multi-Prozessor-Q-CPUs zwischen 1 und 65535 ms und bei QnA-CPUs zwischen 5 und 65535 ms liegen. Der in n1 angegebene Wert muss kleiner als der in n2 angegebene Wert sein.

HINWEIS

Die PWM-Anweisung speichert die Daten der Operanden im CPU-Arbeitsbereich und führt die Ausgabeoperationen während eines Systeminterrupts aus (1 ms bei Multi-Prozessor-Q-CPUs *und 5 ms bei QnA-CPUs). Die PWM-Anweisung kann nur einmal im gesamten Programm verwendet werden.*

In den folgenden Fällen wird eine PWM-Anweisung nicht ausgeführt:

– Wenn n1 und n2 Null sind.

– Wenn n2 gleich oder kleiner als n1 ist.

– Wenn bei Verwendung der QnA-CPU n1 und n2 nicht ein Vielfaches von 5 sind.

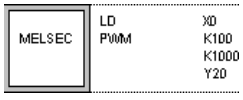
Während der Ausführung der PWM-Anweisung dürfen n1, n2 und d nicht verändert werden.

Beispiel

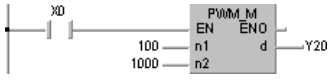
PWM

Das folgende Programm gibt nach dem Einschalten von X0 Impulse mit einer Periodendauer von 1 Sekunde und einer Tastdauer von 100 ms an Y20 aus.

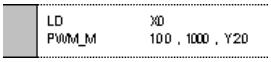
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



6.8.10 MTR

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●[1]

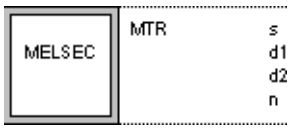
[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

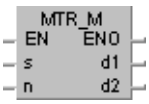
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sonder-module U□\G□	Index-Register Zn	Konstanten	Andere		
	Bit	Wort		Bit	Wort						
s	●	—	—	—	—	—	—	—	—	SM0	5
d1	●	—	—	—	—	—	—	—	—		
d2	●	—	—	—	—	—	—	—	—		
n	●	●	●	●	●	●	●	●	—		

GX IEC Developer

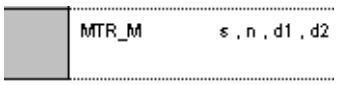
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

Operand	Befehlswert	Datentyp
s	Adresse des ersten Eingangs.	Bit
d1	Adresse des ersten Ausgangs.	
d2	Startadresse, ab welcher die Matrix-Eingangsdaten gespeichert werden.	
n	Anzahl der Eingangsabfragen.	BIN-16-Bit

Funktionsweise

Bildung einer Eingabe-Matrix

MTR Anweisung zum n-fachen Einlesen von Informationen unter Bildung einer Eingabe-Matrix

Die MTR-Anweisung liest 16 Informationen (0/1) beginnend bei dem in s angegebenen Operanden ein. Die Anzahl der Wiederholungen dieses Vorgangs (Reihen) ist in n angegeben. Die jeweiligen Zustände der eingelesenen Informationen werden beginnend in dem in d2 angegebenen Operanden gespeichert. Dadurch wird eine Matrix mit 16 Bits und n Reihen gebildet.

Pro Schritt können 16 Informationen eingelesen werden.

Das Einlesen von der ersten bis zur n-ten Reihe wird fortlaufend wiederholt.

Aufgrund der Tatsache, das bei der MTR-Anweisung eine Matrix von 16 Bit mal n Reihen gebildet wird, ist auch in dem in d2 angegebenen Operanden ein Raum von 16 Bit mal n Reihen für die Speicherung erforderlich.

Beginnend mit dem in d1 angegebenen Ausgang werden die einzelnen Reihen ausgewählt. Der zu einer einzulesenden Reihe mit 16 Bits gehörende Ausgang wird vom System automatisch gesetzt oder zurückgesetzt. Die Anzahl der Ausgänge ist mit der Anzahl der Reihen identisch. Dadurch ist gewährleistet, dass das System jede einzelne Reihe gezielt ansprechen kann.

Die in s, d1 und d2 angegebenen Operanden dürfen nur Adressen besitzen, die durch 16 teilbar sind.

Die Anzahl der Reihen kann zwischen 2 und 8 Reihen liegen.

Es ist zu beachten, das die MTR-Anweisung unmittelbar mit den aktuellen Ein- und Ausgabedaten arbeitet.

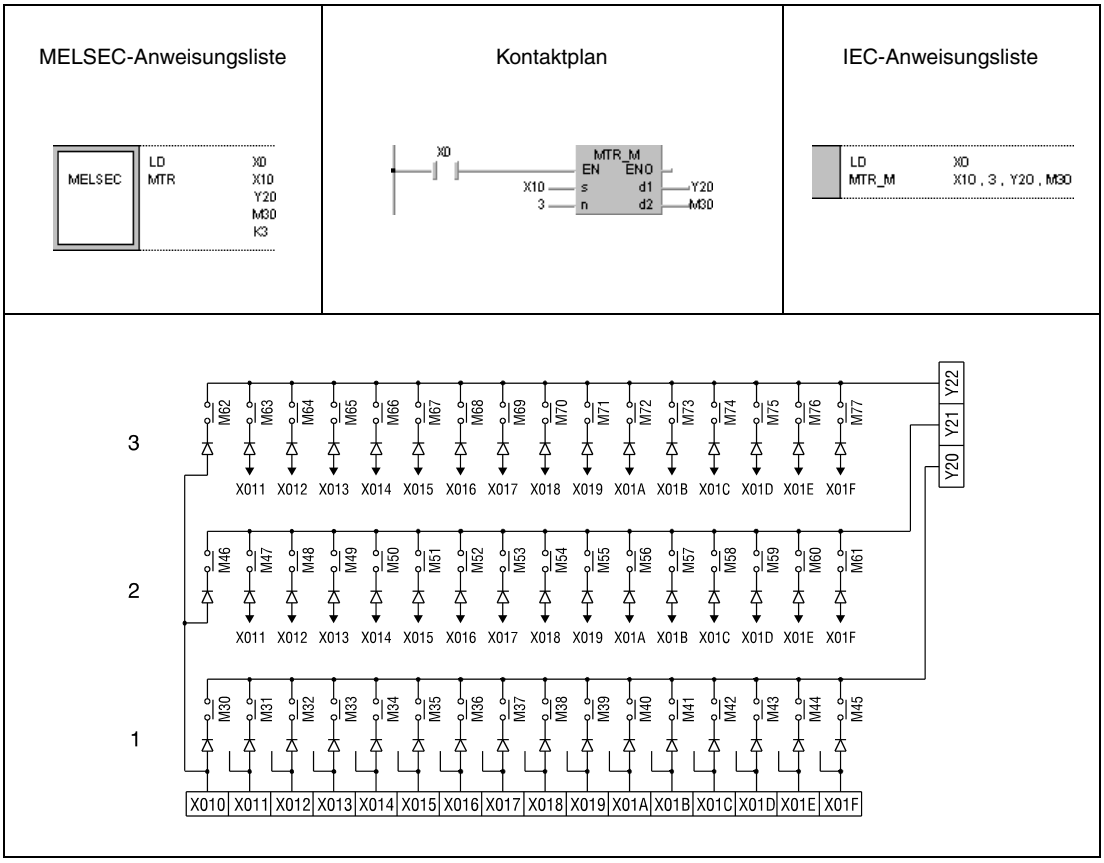
Fehlerquellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die Adressen der in s, d1 und d2 angegebenen Operanden sind nicht durch 16 teilbar (Fehlercode 4101).
- Der in s angegebene Operand liegt außerhalb des aktuellen Eingabebereichs (Fehlercode 4101).
- Der in d1 angegebene Operand liegt außerhalb des aktuellen Ausgabebereichs (Fehlercode 4101).
- Der Raum der 16-Bits-mal-n-Reihen-Matrix liegt außerhalb des für die Speicherung vorgesehenen Bereiches des d2 (Fehlercode 4101).
- Der für n eingetragene Wert liegt nicht zwischen 2 und 8 (Fehlercode 4100).

Beispiel MTR

Das folgende Programm liest nach Einschalten von X0 die Eingänge X10 bis X1F dreimal ein und speichert die Ergebnisse in M30 bis M77. Dadurch wird eine Matrix mit 16 Bits mal 3 Reihen gebildet. Die Reihen können über die Ausgänge Y20 bis Y22 angesprochen werden.



[1] 1. Reihe

[2] 2. Reihe

[3] 3. Reihe

7 Applikationsanweisungen Teil II

Die Applikationsanweisungen in Teil II sind anwendungsspezifische Anweisungen, mit deren Hilfe eine Reihe von Sonderfunktionen ausgeführt werden können. Eine Unterteilung der einzelnen Funktionen enthält die nachfolgende Tabelle.

Einteilung	Bedeutung
Logische Funktionsanweisungen	UND-/ ODER-Logik, Exklusiv-ODER-/ NOR-Logik
Rotationsanweisungen	Datenrotation rechts/links mit 16- und 32- Bit
Verschiebeanweisungen	Bit- oder blockweises Verschieben innerhalb eines Datenwortes
Bit-Verarbeitungsanweisungen	Setzen und Rücksetzen von Bits, Bitabfrage
Datenverarbeitungsanweisungen	Daten in definierten Bereichen suchen, Daten kodieren und dekodieren, Datenwerte auftrennen und zusammenführen
Strukturierte Programmanweisung	Wiederholungsanweisung, Unterprogrammaufruf, Unterprogrammaufruf zwischen Programmdateien, Umschaltung zwischen Main- und Sub-Programmbereich, Mikrocomputer-Programmaufruf, indizierte Adressierung eines gesamten Programmbereichs, Speicherung indizierter Operandenadressen in einer Index-Liste
Verarbeitungsanweisungen für Datenlisten	Daten zur weiteren Verarbeitung in und aus einer Daten-Liste schreiben und lesen, bestimmte Datenblöcke in der Datenliste löschen und einfügen
Anweisungen für den Pufferspeicherzugriff	Zugriff auf den Pufferspeicher eines Sondermoduls oder eines Remote-Moduls
Display-Anweisungen	Ausgabe von ASCII-Zeichen an die Ausgänge eines Moduls oder die LED-Anzeige einer CPU
Anweisungen zur Fehlerdiagnose und Fehlerbeseitigung	Fehlerkontrolle, Status Latch, Abtastüberwachung (Sampling Trace), Programmüberwachung (Program Trace)
Verarbeitungsanweisungen für Zeichenfolgen	Zeichenfolgenverarbeitung (ASCII-Code)
Anweisungen für Sonderfunktionen	Anweisungen zu Winkelfunktionen, Wurzel- und Exponentialrechnungen mit BCD-Daten und Gleitkommazahlen
Datenkontrollanweisungen	Überprüfung von Eingangsdaten auf Einhaltung vorgegebener Wertebereiche und Speicherung der überprüften Daten
Umschaltanweisungen für Datenregisterblöcke und Dateien	Umschaltung zwischen File-Registerblöcken und Dateien
Uhr-Anweisungen	Schreiben und Lesen von Uhr-Daten
Anweisungen für periphere Baugruppen	Ausgabe von Meldungen und Tastatureingabe an peripheren Baugruppen
Programmanweisungen	Anweisungen zum Wechsel der Programm-ausführungsmodi
Weitere Anweisungen	WDT zurücksetzen, Übertragstelle (Carry) Setzen und Rücksetzen, Impulsgeber, direktes Lesen und Schreiben von Bytes, Tastatureingabe, Sichern und Wiederherstellen von Index-Registerinhalten, Schreiben von Daten in EEPROM-Register

7.1 Logikanweisungen

Mit Hilfe der Logikanweisungen sind Logik-Verknüpfungen wie logische Addition oder logisches Produkt möglich.

Funktion	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor
UND-Logik (logisches Produkt)	WAND	WAND_M, WAND_3_M
	WANDP	WANDP_M, WANDP_3_M
	DAND	DAND_M, DAND_3_M
	DANDP	DANDP_M, DANDP_3_M
	BKAND	BKAND_M
	BKANDP	BKANDP_M
ODER-Logik (logische Addition)	WOR	WOR_M, WOR_3_M
	WORP	WORP_M, WORP_3_M
	DOR	DOR_M, DOR_3_M
	DORP	DORP_M, DORP_3_M
	BKOR	BKOR_M
	BKORP	BKORP_M
Exklusiv-ODER-Logik	WXOR	WXOR_M, WXOR_3_M
	WXORP	WXORP_M, WXORP_3_M
	DXOR	DXOR_M, DXOR_3_M
	DXORP	DXORP_M, DXORP_3_M
	BKXOR	BKXOR_M
	BKXORP	BKXORP_M
Exklusiv-NOR-Logik	WXNR	WXNR_M, WXNR_3_M
	WXNRP	WXNRP_M, WXNRP_3_M
	DXNR	DXNR_M, DXNR_3_M
	DXNRP	DXNRP_M, DXNRP_3_M
	BKXNR	BKXNR_M
	BKXNRP	BKXNRP_M

HINWEIS *Nutzen Sie in den IEC-Editoren die IEC-Anweisungen.*

Die Verarbeitung der Logikanweisungen erfolgt bitweise im Binärsystem. Hierbei werden jeweils zwei Zustände (0 und 1) miteinander logisch verknüpft, und das Ergebnis dieser Verknüpfung wird an eine Zieladresse ausgegeben.

Nachfolgend sind die Verknüpfungsergebnisse der Zustände 0 und 1 in einer Wahrheitstabelle aufgeführt. A und B stehen hierbei für die Eingangsvariablen und Y für die Ausgangsvariable.

Logische Verknüpfung	**Verarbeitung**	**Verknüpfungs-algebra**	**Beispiel**		
			A	**B**	**Y**
UND-Funktion	Der Ausgang Y ist nur dann 1, wenn die Eingänge A **und** B gleichzeitig 1 sind.	Y = A x B	0	0	0
			0	1	0
			1	0	0
			1	1	1
ODER-Funktion	Der Ausgang Y ist 1, wenn mindestens einer der Eingänge A **oder** B 1 ist.	Y = A + B	0	0	0
			0	1	1
			1	0	1
			1	1	1
Exklusiv-ODER-Funktion	Der Ausgang Y ist 1, wenn A und B ungleich sind, und ist 0, wenn A und B gleich sind.	$Y = \overline{A} \times B + A \times \overline{B}$	0	0	0
			0	1	1
			1	0	1
			1	1	0
Exklusiv-NOR-Funktion	Der Ausgang Y ist 1, wenn A und B gleich sind, und ist 0, wenn A und B ungleich sind.	$Y = (\overline{A} + B) (A + \overline{B})$	0	0	1
			0	1	0
			1	0	0
			1	1	1

7.1.1 WAND, WANDP, DAND, DANDP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
WAND																										
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				K1 ↓ K4	5 ●[1]	●		●
d		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●										
s1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					7 ●[1]			
s2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●								
d1		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●										
DAND																										
s	●	●	●	●	●	●	●	●	●	●	●	●	●		●		●	●				K1 ↓ K8	9 ●[1]	●		●
d		●	●	●	●	●	●	●	●	●	●	●	●		●											

[1] Die Anzahl der Schritte bei Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

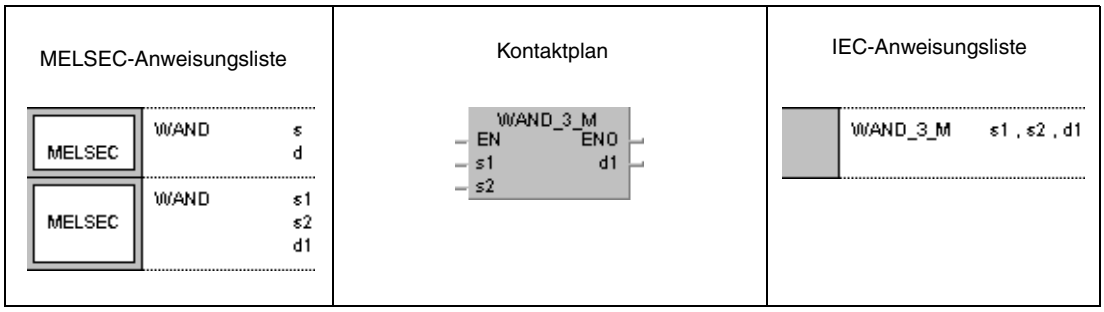
Operanden MELSEC Q

	Operanden										Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort							
WAND												
s	●	●	●	●	●	●	●	●	—	—		3
d	●	●	●	●	●	●	●	—	—	—		
s1	●	●	●	●	●	●	●	●	—	—		4
s2	●	●	●	●	●	●	●	●	—	—		
d1	●	●	●	●	●	●	●	—	—	—		
DAND												
s	●	●	●	●	●	●	●	●	—	—		4[1)]
d	●	●	●	●	●	●	●	—	—	—		
s1	●	●	●	●	●	●	●	●	—	—		4[2)]
s2	●	●	●	●	●	●	●	●	—	—		
d	●	●	●	●	●	●	●	—	—	—		

[1] Die Anzahl der Programmschritte ist abhängig vom Typ der CPU und von den verwendeten Operanden.
Bei Verwendung einer QnA-CPU: 4
Bei einer Single-Prozessor-Q-CPU 3
Bei Multi-Prozessor-Q-CPU, interne Wortoperanden (außer File-Register ZR) oder Konstanten: 6
Bei Multi-Prozessor-Q-CPU und Bit-Operanden, deren Adresse ein Vielfaches von16 ist, die die Bit-Blockbezeichnung 4 haben und die nicht durch Index-Vergabe bearbeitet werden: 6
Bei Verwendung einer Multi-Prozessor-Q-CPU und anderer Operanden als oben aufgeführt: 4

[2] Die Anzahl der Programmschritte ist abhängig vom Typ der CPU und von den verwendeten Operanden.
Bei Verwendung einer QnA-CPU: 4
Bei einer Q-CPU und interne Wortoperanden (außer File-Register ZR) oder Konstanten: 6
Bei einer Q-CPU und Bit-Operanden, deren Adresse ein Vielfaches von16 ist, die die Bit-Blockbezeichnung 4 haben und die nicht durch Index-Vergabe bearbeitet werden: 6
Bei Verwendung einer Q-CPU und anderer Operanden als oben aufgeführt: 4

GX IEC Developer



GX Developer

Variablen

Operand	Befehlswert	Datentyp
s	Daten, mit denen das logische Produkt gebildet wird, oder Startadresse, ab der diese Daten gespeichert sind.	BIN-16-/32-Bit
d		
s1	Daten, mit denen das logische Produkt gebildet wird, oder Startadresse, ab der diese Daten gespeichert sind.	
s2		
d1 (bei DAND d)	Startadresse, ab der das Ergebnis gespeichert wird.	

Funktions-weise

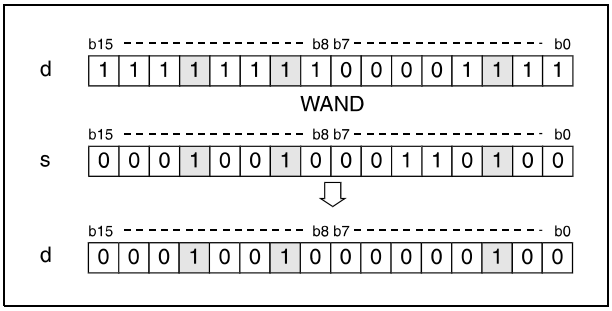
UND-Logik

WAND 16-Bit-Daten

Die UND-Logik (englisch: AND) bildet das logische Produkt aus zwei Eingangsvariablen.

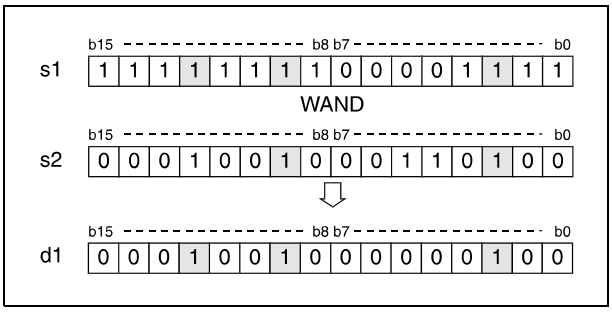
- 1.Variante:

Aus den in s und d angegebenen 16-Bit-Daten wird das logische Produkt bitweise gebildet. Das Ergebnis wird an den in d angegebenen Operanden ausgegeben.



- 2. Variante:

Aus den in s1 und s2 angegebenen 16-Bit-Daten wird das logische Produkt bitweise gebildet. Das Ergebnis wird an den in d1 angegebenen Operanden ausgegeben.

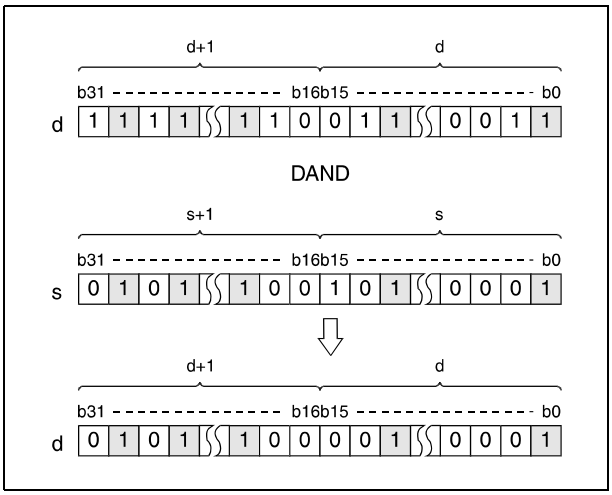


Bits oberhalb der Blocklänge werden auf 0 gesetzt. Ist die Blocklänge beispielsweise mit K2 festgelegt, werden die oberen 8 Bits (b8 bis b15) mit 0 verarbeitet.

DAND 32-Bit-Daten

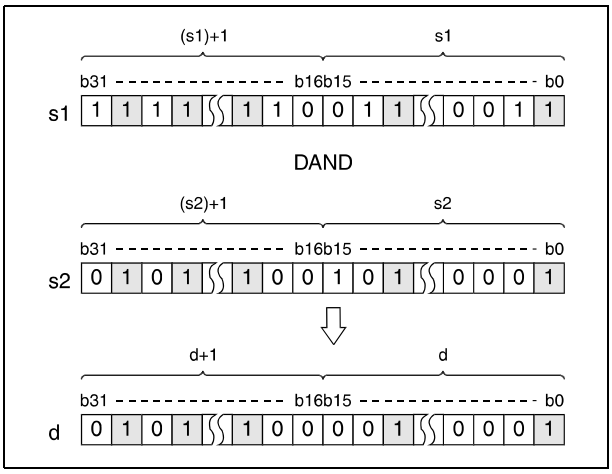
- 1.Variante:

Aus den in s und d angegebenen 32-Bit-Daten wird das logische Produkt bitweise errechnet. Das Ergebnis wird an den in d angegebenen Operanden ausgegeben.



- 2. Variante (QnA-Serie/System Q):

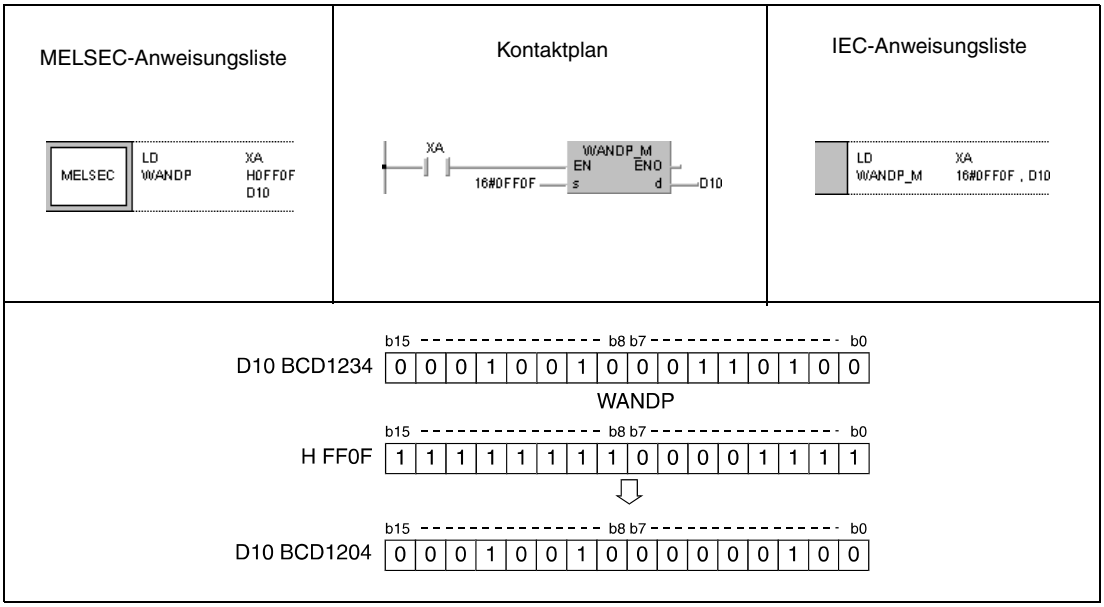
Aus den in s1 und s2 angegebenen 32-Bit-Daten wird das logische Produkt bitweise gebildet. Das Ergebnis wird an den in d angegebenen Operanden ausgegeben.



Nach Durchführung der Verknüpfung werden alle Bits, die außerhalb des Blockbereiches liegen, auf 0 gesetzt.

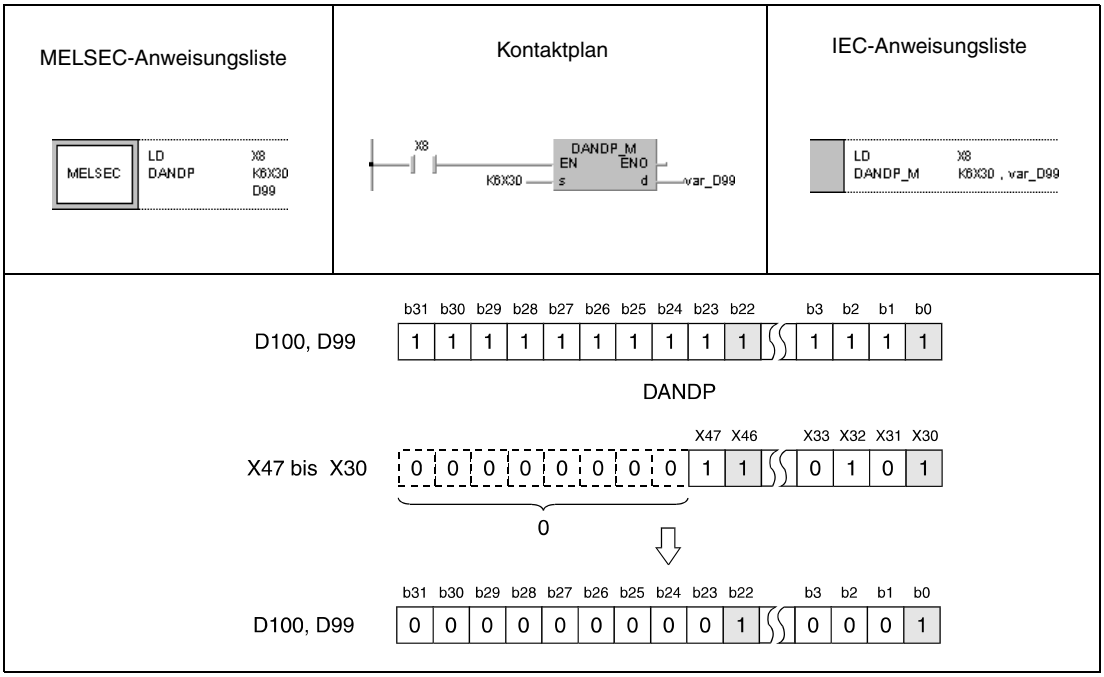
Beispiel 1 WANDP (s, d)

Das folgende Programm setzt mit positiver Flanke (ansteigender Flanke) von XA die Zehnerstelle (b5-b7) des in D10 angegebenen BCD-Datenwertes auf 0 und speichert das Ergebnis wieder in D10.



Beispiel 2 DANDP (s, d)

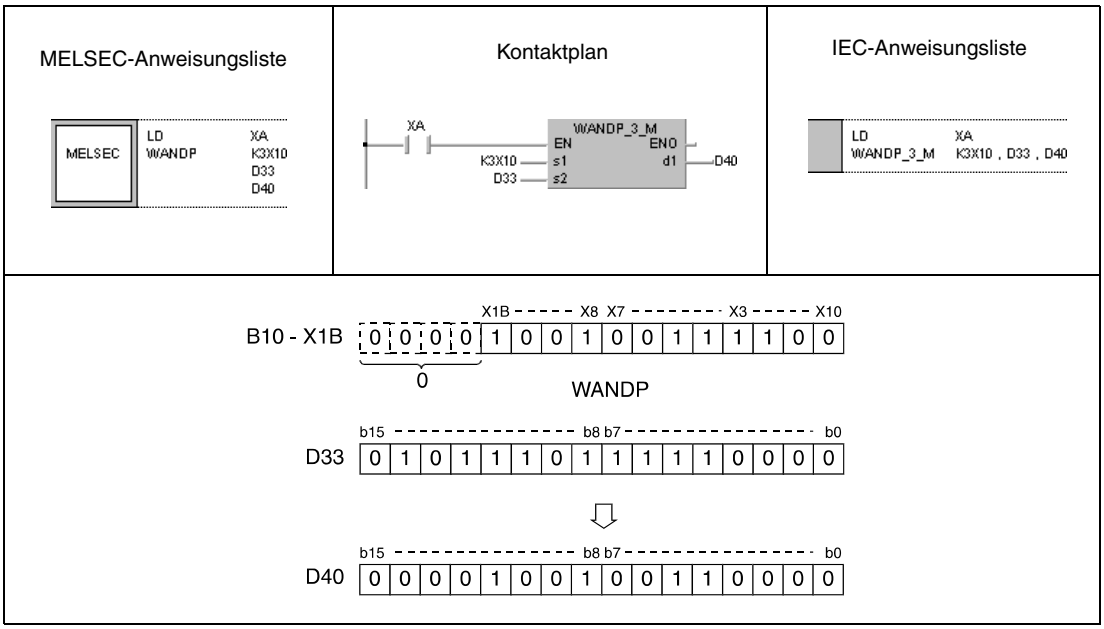
Das folgende Programm bildet mit positiver Flanke von X8 das logische Produkt mit dem 32-Bit-Datenwert in D99 und D100 und dem 24-Bit-Datenwert von X30 bis X47 und speichert das Ergebnis wieder in D99 und D100 ab.



[1] Diese Bits werden als 0 eingelesen.

Beispiel 3 WANDP (s1, s2, d1)

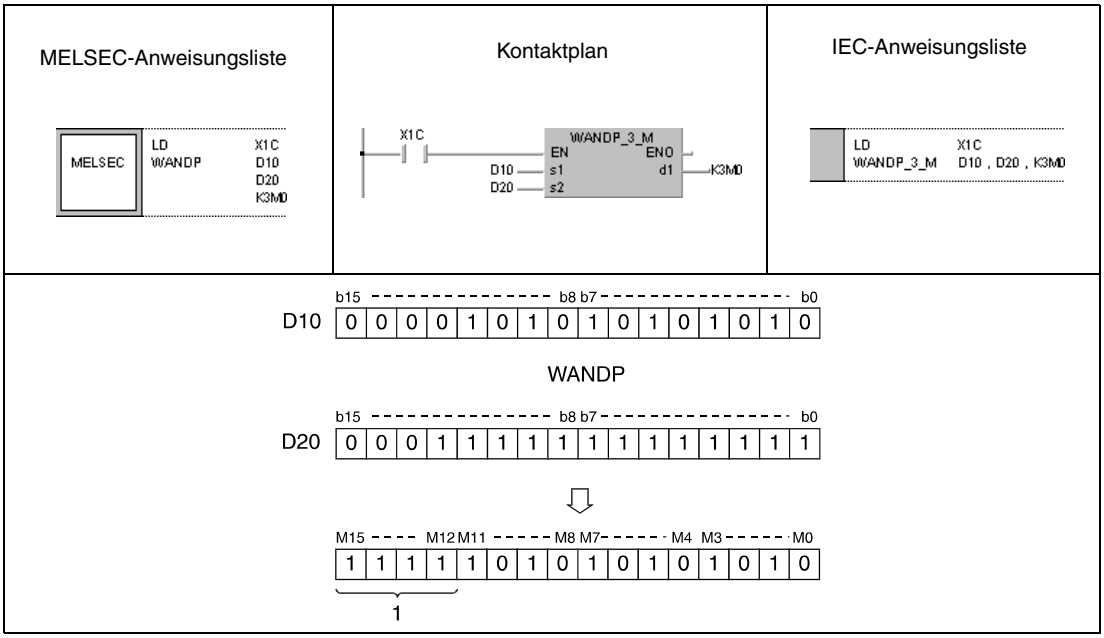
Das folgende Programm bildet mit positiver Flanke von XA das logische Produkt der Daten von X10 bis X1B und den Daten in D33 und speichert das Ergebnis in D40.



[1] Diese Bits werden als 0 eingelesen.

Beispiel 4 WANDP (s1, s2, d1)

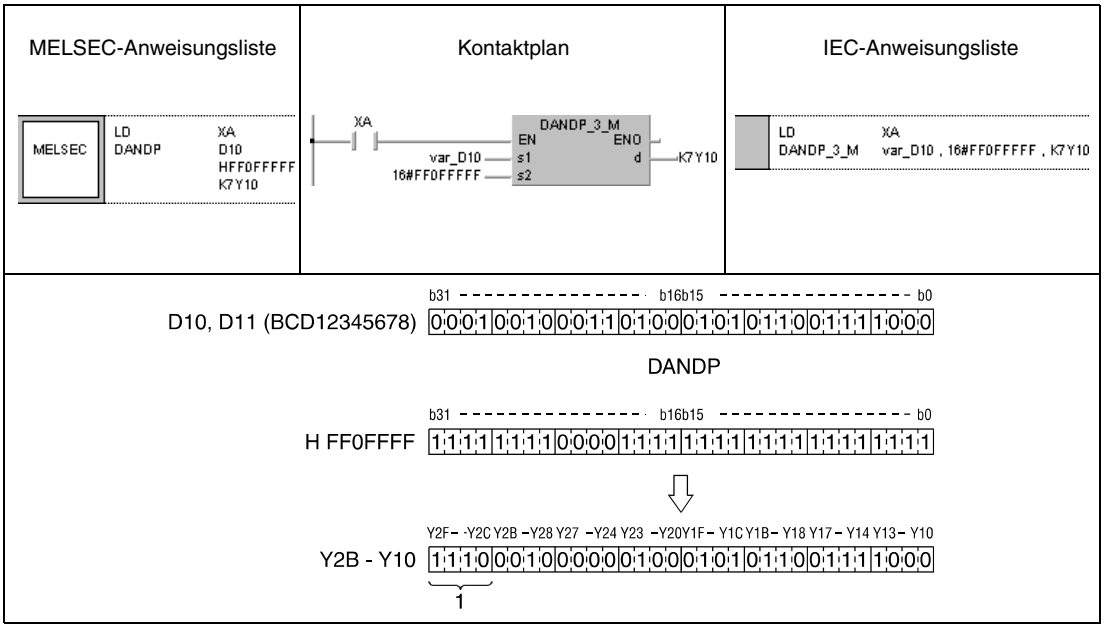
Das folgende Programm bildet mit positiver Flanke von X1C das logische Produkt der Daten in D10 und D20 und speichert das Ergebnis in M0 bis M11.



[1] Diese Bits verändern ihren Zustand nicht.

Beispiel 5 DANDP (s1, s2, d)

Das folgende Programm setzt mit positiver Flanke von XA die Hunderttausenderstelle des in D10 und D11 angegebenen BCD-Datenwertes auf 0 und gibt das Ergebnis an Y10 bis Y2B aus.



[1] Diese Bits verändern ihren Zustand nicht.

HINWEIS *Die Programmbeispiele 2 und 5 sind ohne Variablendefinition im Header der Programm-Organisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.1.2 BKAND, BKANDP

CPU

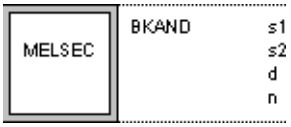
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Operanden MELSEC Q

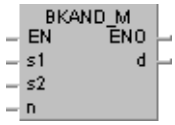
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	–	●	●	–	–	–	–	–	–	SM0	5
s2	–	●	●	–	–	–	–	●	–		
d	–	●	●	–	–	–	–	–	–		
n	●	●	●	●	●	●	●	●	–		

GX IEC Developer

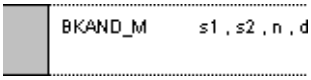
MELSEC-Anweisungsliste



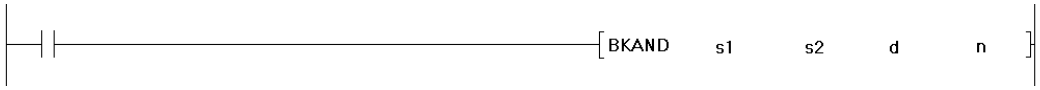
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

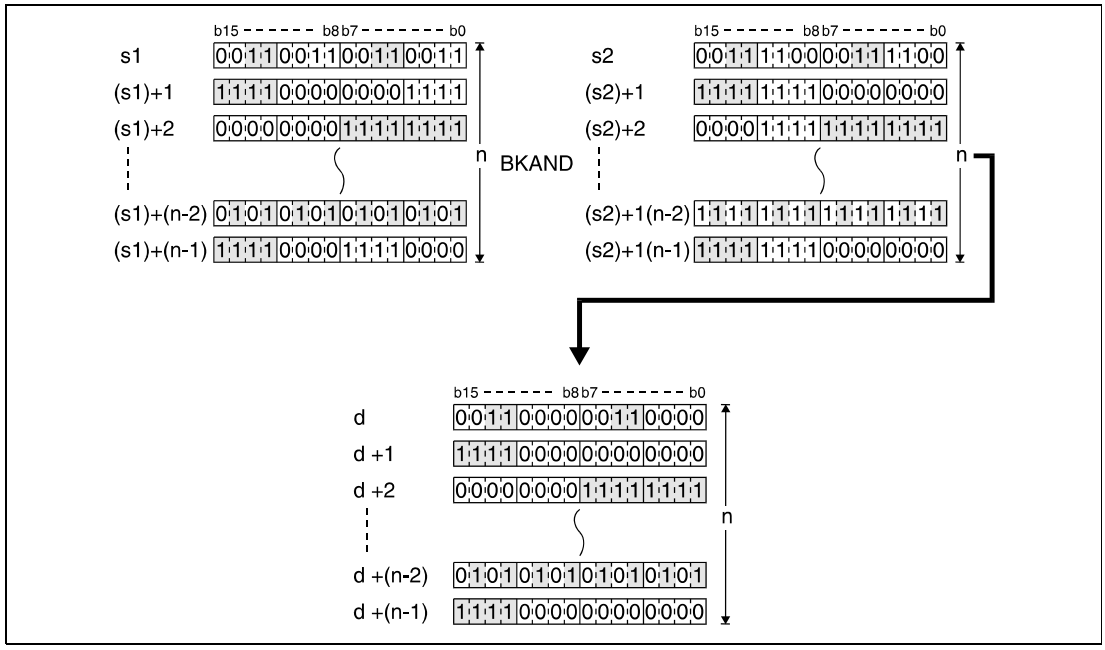
Operand	Befehlswert	Datentyp
s1	Erste Adresse des Operanden, in dem die Daten für die Operation gespeichert sind.	BIN-16-Bit
s2	Erste Adresse der Daten für die Operation oder erste Adresse des Operanden, in dem diese Daten gespeichert sind.	
d	Erste Adresse des Operanden, in dem das Ergebnis gespeichert wird.	
n	Anzahl der Datenblöcke, mit denen die Bildung des logischen Produkts vorgenommen wird.	

Funktions-weise

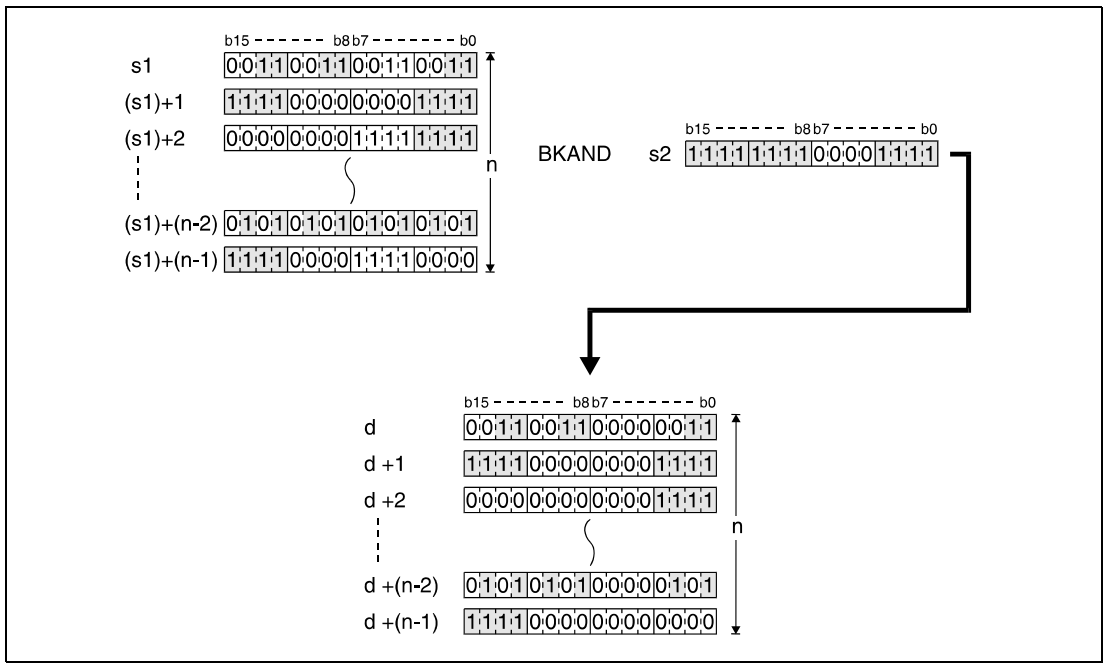
Bildung eines logischen Produkts mit 16-Bit-Datenblöcken

BKAND Blockweise Bildung eines logischen Produktes

Die BKAND-Anweisung bildet das logische Produkt aus den n-ten 16-Bit-Blöcken ab s1 und den n-ten 16-Bit-Blöcken ab s2. Der entsprechende 16-Bit-Block des Ergebnisses wird bei dem in d angegebenen Operanden beginnend gespeichert. Die Anzahl der Blöcke, mit denen die Operation durchgeführt wird, ist in n angegeben.



Eine in s2 abgelegte Konstante muss einen Wert zwischen -32768 und 32767 besitzen.



Fehlerquellen

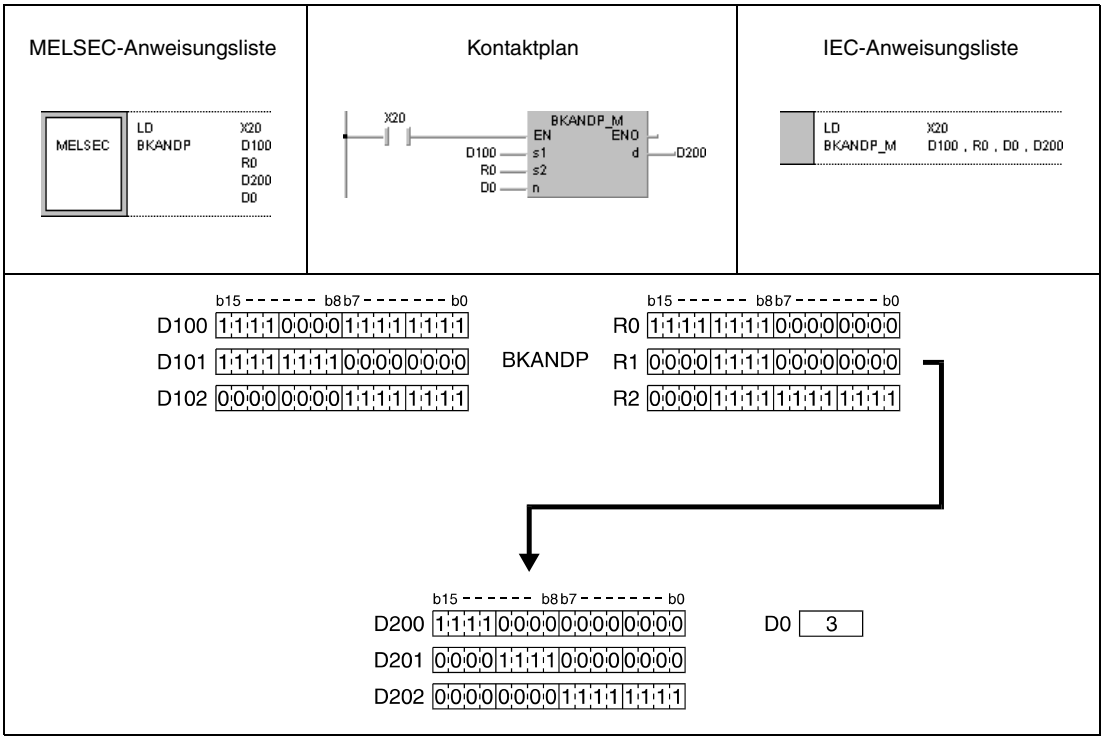
In den folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die in n angegebene Anzahl der Blöcke von s1, s2 oder d liegt außerhalb des für die Speicherung vorgesehenen Bereichs des Operanden (Fehlercode 4101).
- Die für die Speicherung vorgesehenen Bereiche von s1, s2 oder d überlappen (Fehlercode 4101).

Beispiel

BKANDP

Das folgende Programm bildet mit der positiven Flanke von X20 das logische Produkt der Daten in den Registern D100 bis D102 und den Daten in den Registern R0 bis R2. Das Ergebnis wird in den Registern D200 bis D202 gespeichert. Die Anzahl der an der Operation beteiligten 16-Bit-Datenblöcke (3) ist in D0 hinterlegt.



7.1.3 WOR, WORP, DOR, DORP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																					Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
WOR																										
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				K1 ↓ K4	5 ●[1]	●		●
d		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●										
s1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					7 ●[1]			
s2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●								
d1		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●										
DOR																										
s	●	●	●	●	●	●	●	●	●	●	●	●	●		●		●	●				K1 ↓ K8	9 ●[1]	●		●
d		●	●	●	●	●	●	●	●	●	●	●	●		●											

[1] Die Anzahl der Schritte bei Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

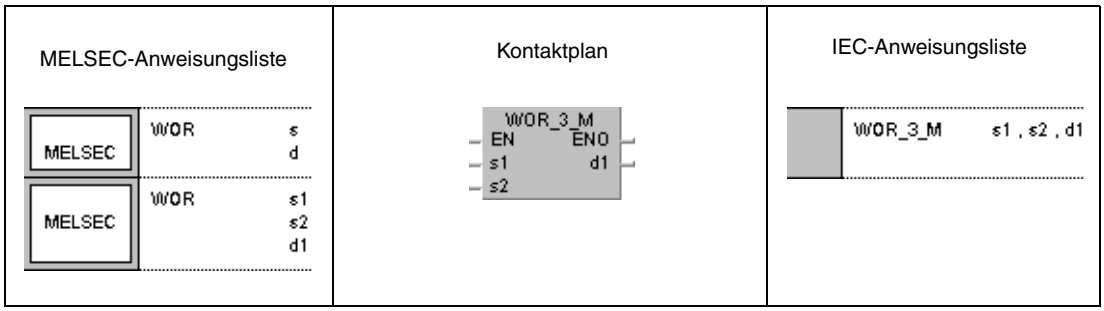
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort				U		
WOR											
s	●	●	●	●	●	●	●	●	—	—	3
d	●	●	●	●	●	●	●	—	—	—	
s1	●	●	●	●	●	●	●	●	—	—	4
s2	●	●	●	●	●	●	●	●	—	—	
d1	●	●	●	●	●	●	●	—	—	—	
DOR											
s	●	●	●	●	●	●	●	●	—	—	4[1)]
d	●	●	●	●	●	●	●	—	—	—	
s1	●	●	●	●	●	●	●	●	—	—	4[2)]
s2	●	●	●	●	●	●	●	●	—	—	
d	●	●	●	●	●	●	●	—	—	—	

[1] Die Anzahl der Programmschritte ist abhängig vom Typ der CPU und von den verwendeten Operanden.
Bei Verwendung einer QnA-CPU: 4
Bei einer Single-Prozessor-Q-CPU 3
Bei Multi-Prozessor-Q-CPU, interne Wortoperanden (außer File-Register ZR) oder Konstanten: 6
Bei Multi-Prozessor-Q-CPU und Bit-Operanden, deren Adresse ein Vielfaches von16 ist, die die Bit-Blockbezeichnung 4 haben und die nicht durch Index-Vergabe bearbeitet werden: 6
Bei Verwendung einer Multi-Prozessor-Q-CPU und anderer Operanden als oben aufgeführt: 4

[2] Die Anzahl der Programmschritte ist abhängig vom Typ der CPU und von den verwendeten Operanden.
Bei Verwendung einer QnA-CPU: 4
Bei einer Q-CPU und interne Wortoperanden (außer File-Register ZR) oder Konstanten: 6
Bei einer Q-CPU und Bit-Operanden, deren Adresse ein Vielfaches von16 ist, die die Bit-Blockbezeichnung 4 haben und die nicht durch Index-Vergabe bearbeitet werden: 6
Bei Verwendung einer Q-CPU und anderer Operanden als oben aufgeführt: 4

GX IEC Developer



GX Developer

Variablen

Operand	Befehlswert	Datentyp
s	Daten, mit denen die Bildung der logischen Summe vorgenommen wird, oder Startadresse des Operanden, in dem diese Daten gespeichert sind.	BIN-16-/32-Bit
d		
s1	Daten, mit denen die Bildung der logischen Summe vorgenommen wird, oder Startadresse des Operanden, in dem diese Daten gespeichert sind.	
s2		
d1 (bei DOR d)	Startadresse des Operanden, in dem das Ergebnis gespeichert wird.	

Funktions-weise

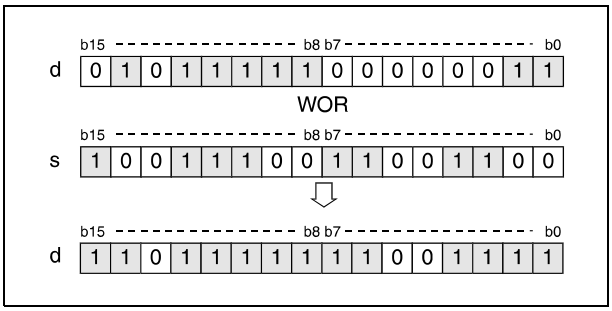
ODER-Logik

WOR 16-Bit-Daten

Die ODER-Logik (englisch: OR) bildet die logische Summe aus zwei Eingangsvariablen.

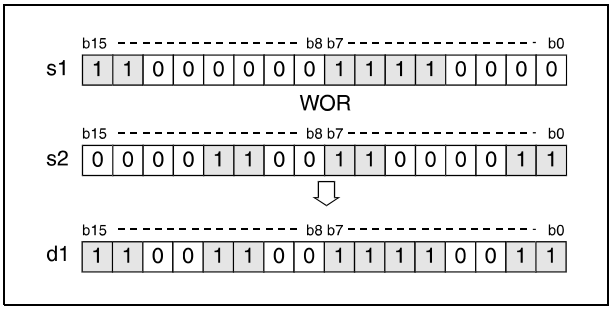
- 1. Variante:

Die in s und d angegebenen 16-Bit-Daten werden bitweise addiert. Das Ergebnis wird an den in d angegebenen Operanden ausgegeben.



- 2. Variante:

Die in s1 und s2 angegebenen 16-Bit-Daten werden bitweise addiert. Das Ergebnis wird an den in d1 angegebenen Operand ausgegeben.

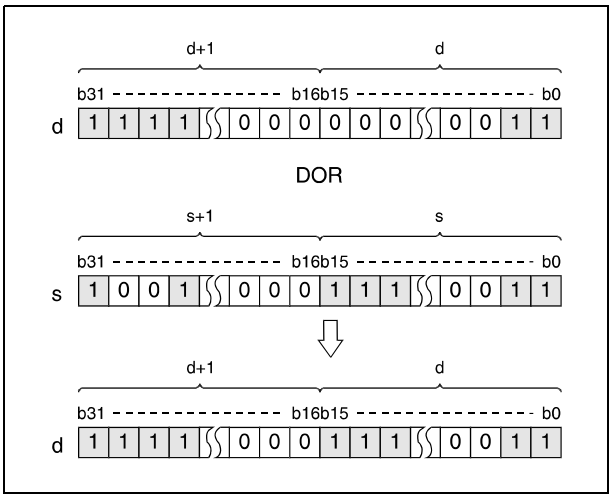


Bits oberhalb der Blocklänge werden auf 0 gesetzt. Ist die Blocklänge beispielsweise mit K2 festgelegt, werden die oberen 8 Bits (b8 bis b15) mit 0 verarbeitet.

DOR 32-Bit-Daten

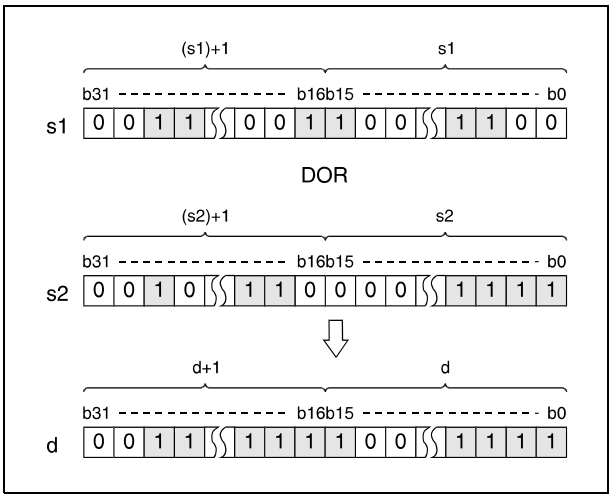
- 1. Variante:

Die in s und d angegebenen 32-Bit-Daten werden bitweise addiert. Das Ergebnis wird an den in d angegebenen Operanden ausgegeben.



- 2. Variante (QnA-Serie/System Q):

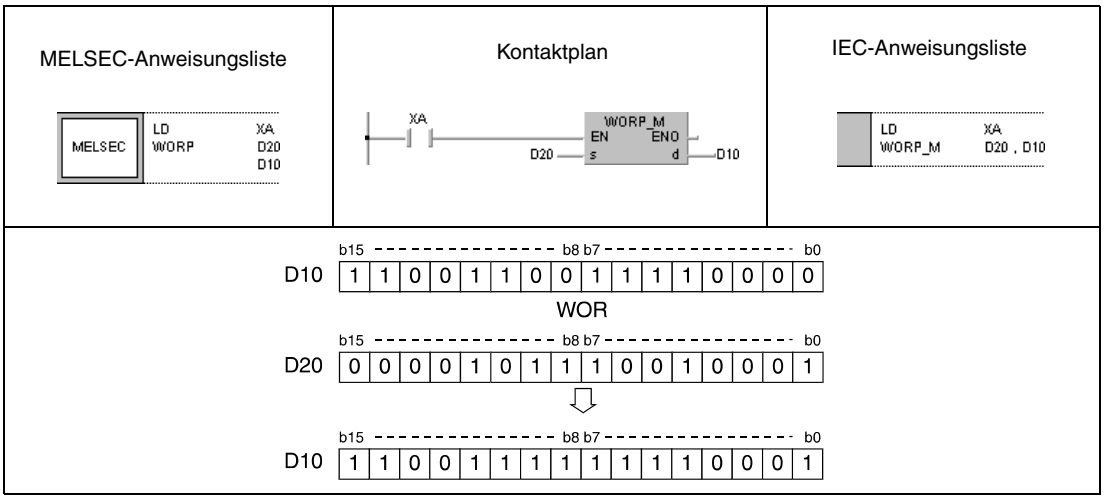
Die in s1 und s2 angegebenen 32-Bit-Daten werden bitweise addiert. Das Ergebnis wird an den in d angegebenen Operand ausgegeben.



Nach Durchführung der Verknüpfung werden alle Bits, die außerhalb des Blockbereiches liegen, auf 0 gesetzt.

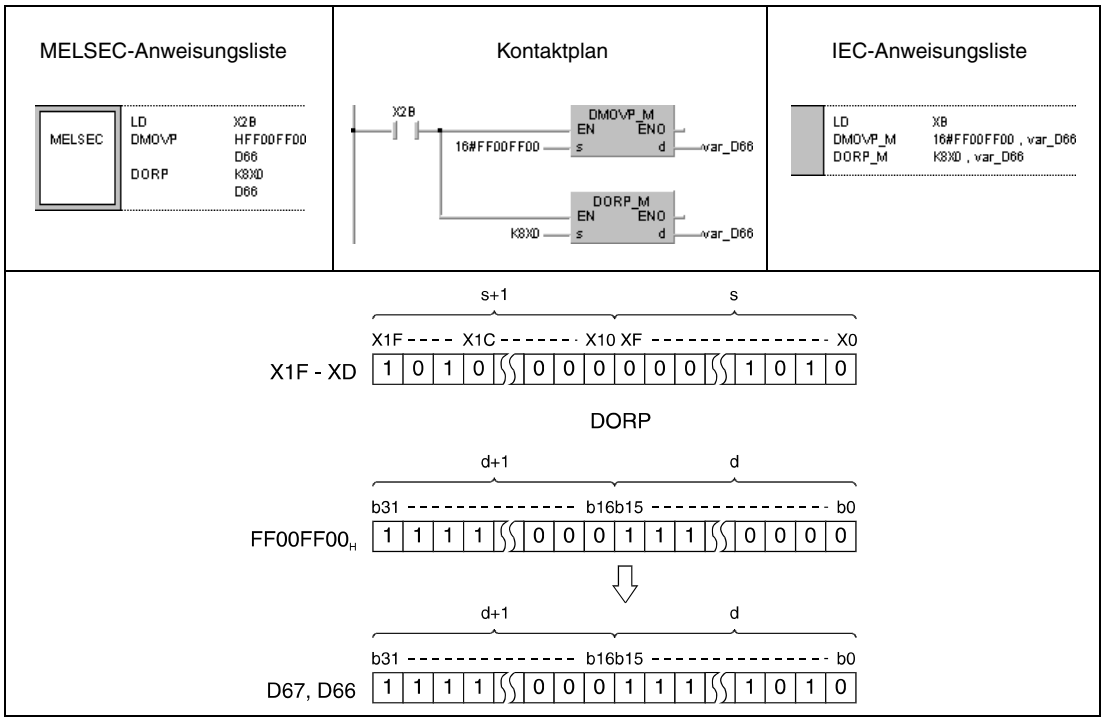
Beispiel 1 WORP (s, d)

Das folgende Programm addiert mit positiver Flanke von XA die Daten aus D10 zu den Daten aus D20. Das Ergebnis wird in D10 gespeichert.



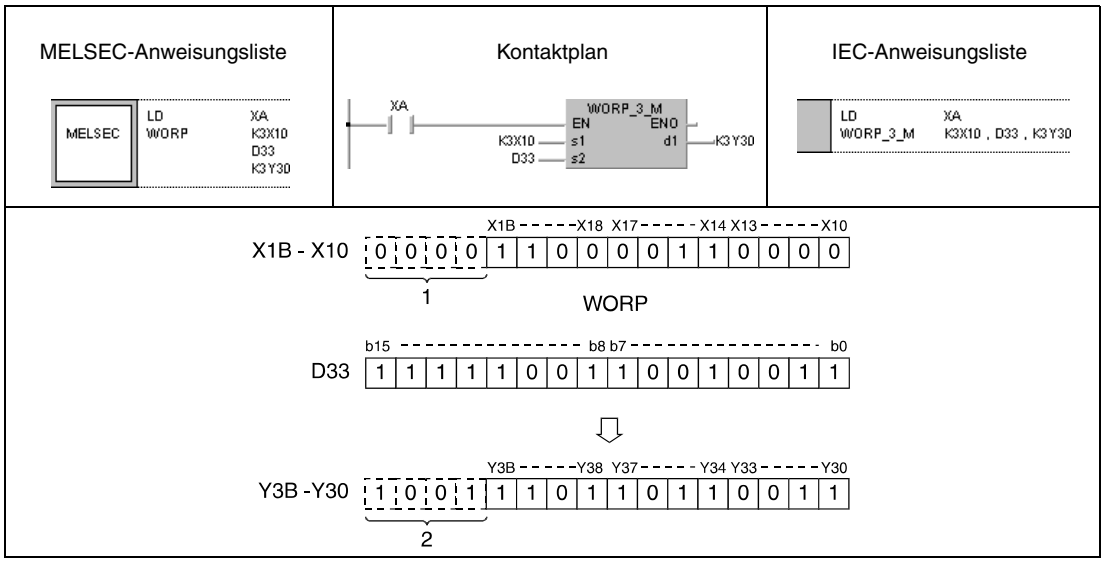
Beispiel 2 DORP (s, d)

Das folgende Programm addiert mit positiver Flanke von X2B die Daten der Eingänge X0 bis X1F zu dem hexadezimalen Wert FF00FF00. Das Ergebnis wird in D66 und D67 gespeichert.



Beispiel 3 WORP (s1, s2, d1)

Das folgende Programm addiert mit positiver Flanke von XA die Daten der Eingänge X10 bis X1B zu den Daten aus D33. Das Ergebnis wird an die Ausgänge Y30 bis Y3B ausgegeben.

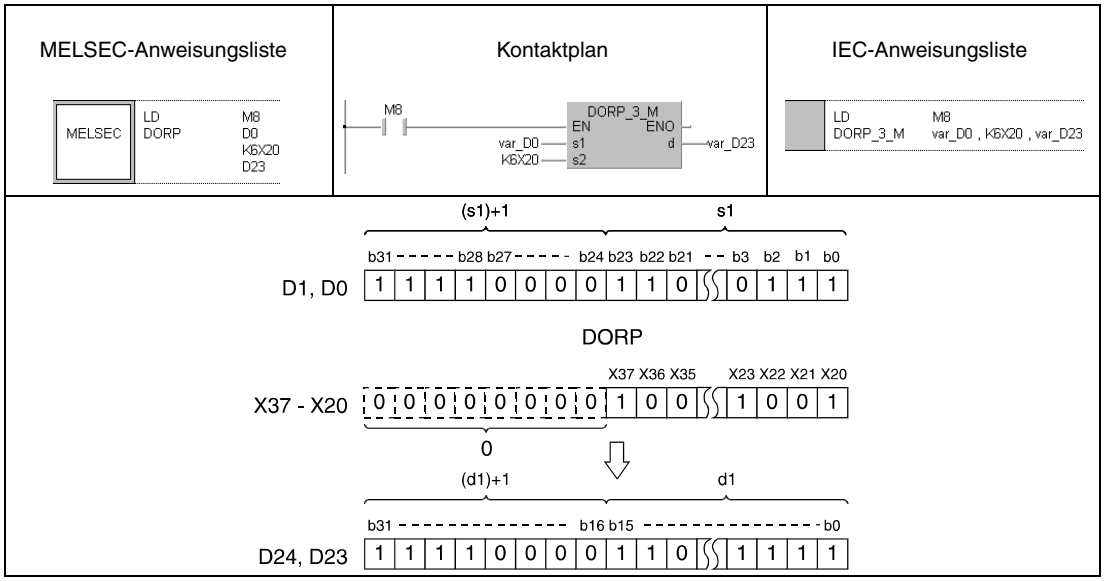


[1] Diese Bits werden als 0 eingelesen

[2] Diese Bits verändern ihren Zustand nicht

Beispiel 4 DORP (s1, s2, d)

Das folgende Programm addiert mit positiver Flanke von M8 die 32-Bit-Daten aus D0 und D1 zu den 24-Bit-Daten der Eingänge X20 bis X37. Das Ergebnis wird in D23 und D24 gespeichert.



HINWEIS *Die Programmbeispiele 2 und 4 sind ohne Variablendefinition im Header der Programm-Organisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.1.4 BKOR, BKORP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

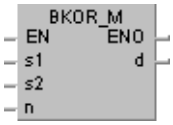
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	—	●	●	—	—	—	—	—	—	SM0	5
s2	—	●	●	—	—	—	—	●	—		
d	—	●	●	—	—	—	—	—	—		
n	●	●	●	●	●	●	●	●	—		

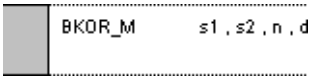
GX IEC Developer

MELSEC-Anweisungsliste

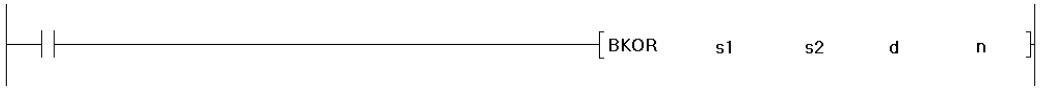
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

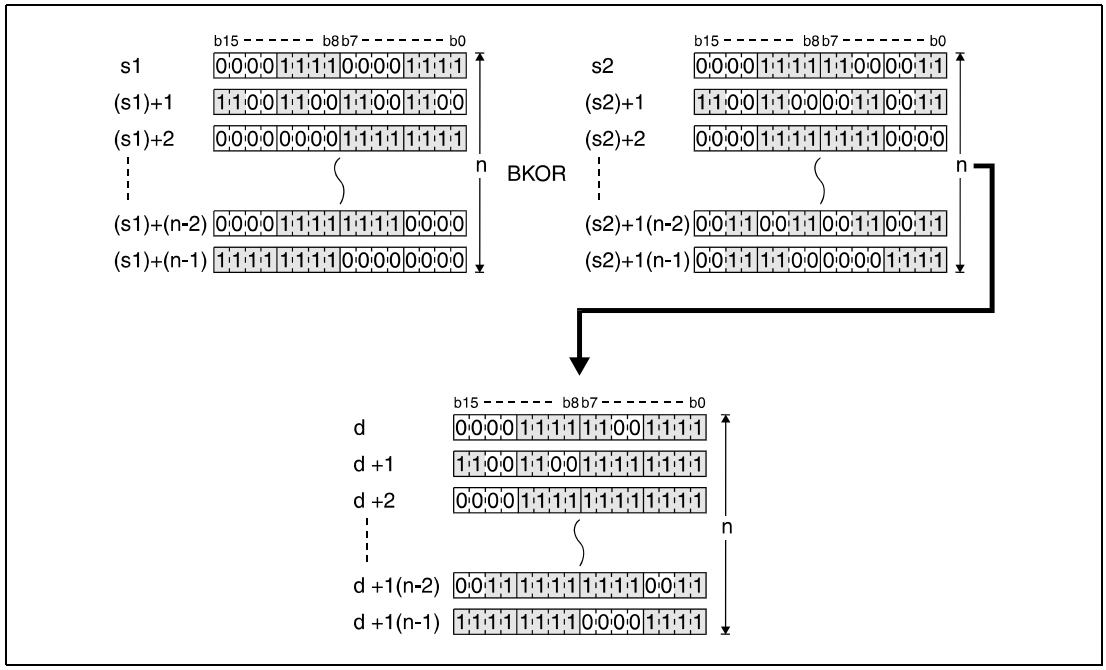
Operand	Befehlswert	Datentyp
s1	Erste Adresse des Operanden, in dem die Daten für die Operation gespeichert sind.	BIN-16-Bit
s2	Erste Adresse der Daten für die Operation oder erste Adresse des Operanden, in dem diese Daten gespeichert sind.	
d	Erste Adresse des Operanden, in dem das Ergebnis gespeichert wird.	
n	Anzahl der Datenblöcke, mit denen die Bildung der logischen Summe vorgenommen wird.	

Funktions-weise

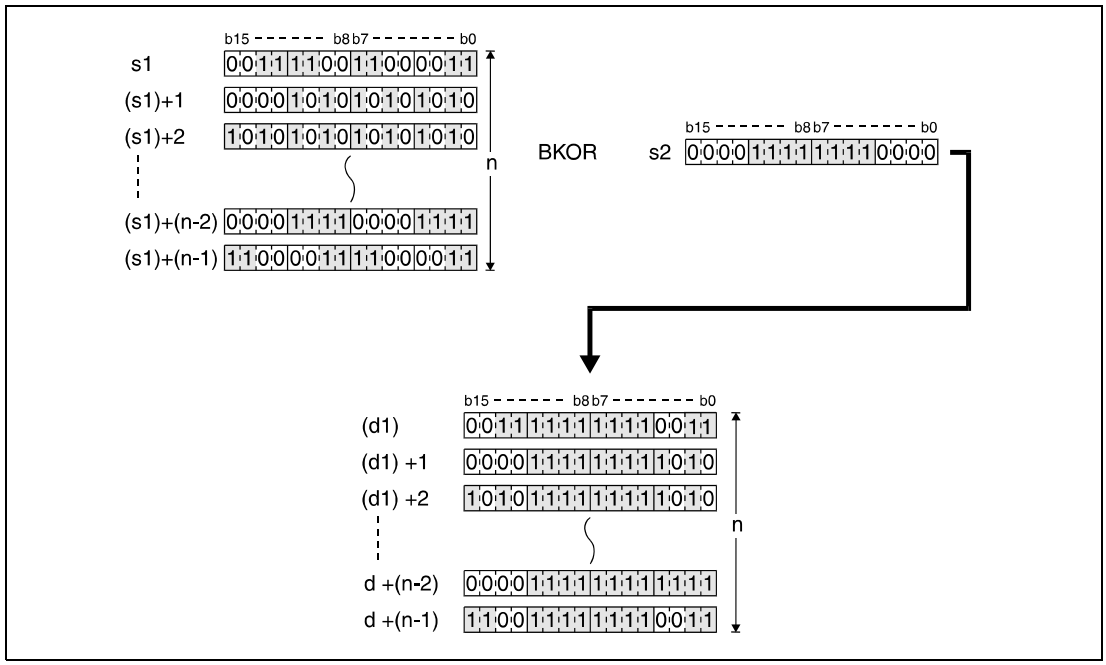
Bildung einer logischen Summe mit 16-Bit-Datenblöcken

BKOR Blockweise Bildung einer logischen Summe

Die BKOR-Anweisung bildet die logische Summe aus den n-ten 16-Bit-Blöcken ab s1 und den n-ten 16-Bit-Blöck ab s2. Der entsprechende 16-Bit-Block des Ergebnisses wird bei dem in d angegebenen Operanden beginnend gespeichert. Die Anzahl der Blöcke, mit denen die Operation durchgeführt wird, ist in n angegeben.



Eine in s2 abgelegte Konstante muss einen Wert zwischen -32768 und 32767 besitzen.



Fehlerquellen

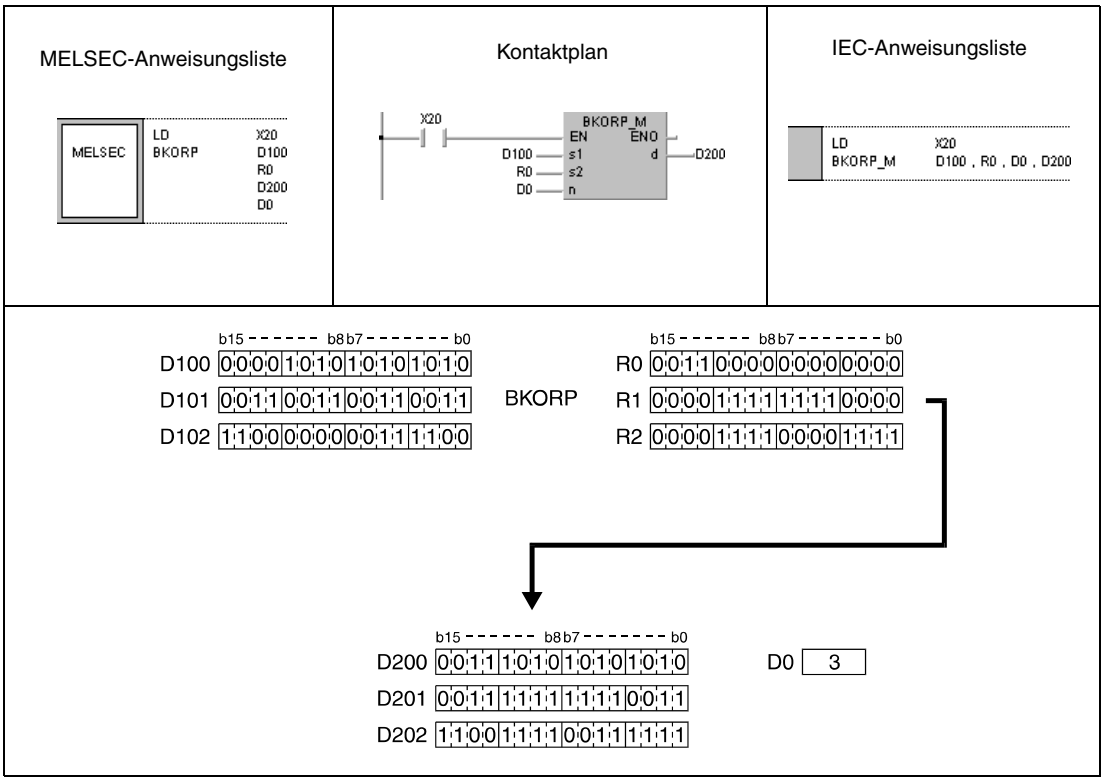
In den folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die in n angegebene Anzahl der Blöcke von s1, s2 oder d liegt außerhalb des für die Speicherung vorgesehenen Bereichs des Operanden (Fehlercode 4101).
- Die für die Speicherung vorgesehenen Bereiche von s1, s2 oder d überlappen (Fehlercode 4101).

Beispiel

BKORP

Das folgende Programm bildet mit der positiven Flanke von X20 die logische Summe aus den Daten in den Registern D100 bis D102 und den Daten in den Registern R0 bis R2. Das Ergebnis wird in den Registern D200 bis D202 gespeichert. Die Anzahl der an der Operation beteiligten 16-Bit-Datenblöcke (3) ist in D0 hinterlegt.



7.1.5 WXOR, WXORP, DXOR, DXORP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
WXOR																										
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				K1 ↓ K4	5 ●[1]	●		●
d		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●										
s1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					7 ●[1]			
s2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●								
d1		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●										
DXOR																										
s	●	●	●	●	●	●	●	●	●	●	●	●	●		●		●	●				K1 ↓ K8	9 ●[1]	●		●
d		●	●	●	●	●	●	●	●	●	●	●	●		●											

[1] Die Anzahl der Schritte bei Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

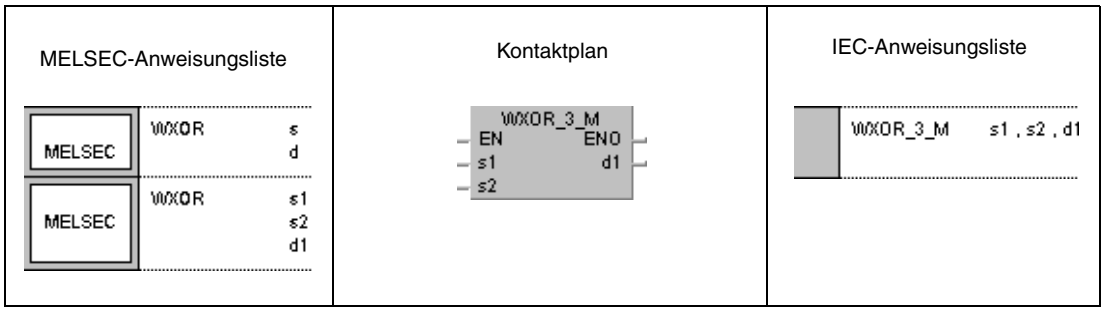
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
WXOR											
s	●	●	●	●	●	●	●	●	—	—	3
d	●	●	●	●	●	●	●	—	—	—	
s1	●	●	●	●	●	●	●	●	—	—	4
s2	●	●	●	●	●	●	●	●	—	—	
d1	●	●	●	●	●	●	●	—	—	—	
DXOR											
	●	●	●	●	●	●	●	●	—	—	4[1)]
	●	●	●	●	●	●	●	—	—	—	
s1	●	●	●	●	●	●	●	●	—	—	4[2)]
s2	●	●	●	●	●	●	●	●	—	—	
d	●	●	●	●	●	●	●	—	—	—	

[1] Die Anzahl der Programmschritte ist abhängig vom Typ der CPU und von den verwendeten Operanden.
Bei Verwendung einer QnA-CPU: 4
Bei einer Single-Prozessor-Q-CPU 3
Bei Multi-Prozessor-Q-CPU, interne Wortoperanden (außer File-Register ZR) oder Konstanten: 6
Bei Multi-Prozessor-Q-CPU und Bit-Operanden, deren Adresse ein Vielfaches von16 ist, die die Bit-Blockbezeichnung 4 haben und die nicht durch Index-Vergabe bearbeitet werden: 6
Bei Verwendung einer Multi-Prozessor-Q-CPU und anderer Operanden als oben aufgeführt: 4

[2] Die Anzahl der Programmschritte ist abhängig vom Typ der CPU und von den verwendeten Operanden.
Bei Verwendung einer QnA-CPU: 4
Bei einer Q-CPU und interne Wortoperanden (außer File-Register ZR) oder Konstanten: 6
Bei einer Q-CPU und Bit-Operanden, deren Adresse ein Vielfaches von16 ist, die die Bit-Blockbezeichnung 4 haben und die nicht durch Index-Vergabe bearbeitet werden: 6
Bei Verwendung einer Q-CPU und anderer Operanden als oben aufgeführt: 4

GX IEC Developer



GX Developer

Variablen

Operand	Befehlswert	Datentyp
s	Daten, mit welchen die Exklusiv-ODER-Operation vorgenommen wird, oder erste Adresse des Operanden, in dem diese Daten gespeichert sind.	BIN-16-/32-Bit
d		
s1	Daten, mit welchen die Exklusiv-ODER-Operation vorgenommen wird, oder erste Adresse des Operanden, in dem diese Daten gespeichert sind.	
s2		
d1 (bei DXOR d)	Erste Adresse des Operanden, in dem das Ergebnis gespeichert wird.	

Funktionsweise

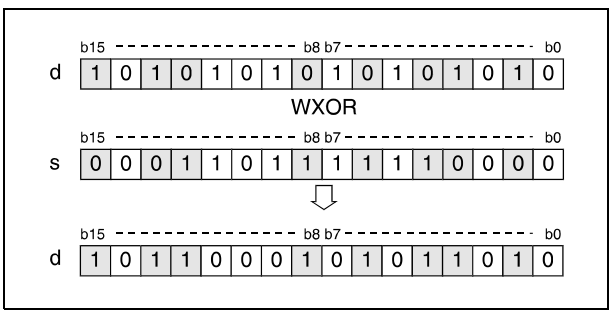
Exklusiv-ODER-Logik

WXOR 16-Bit-Daten

Die Exklusiv-ODER-Logik (englisch: exclusive OR) bildet die logische Summe aus dem Produkt zweier Eingangsvariablen (Y= ($\overline{A}$xB)+(Ax$\overline{B}$)).

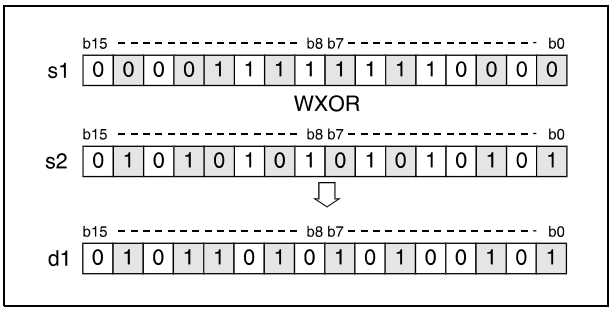
- 1. Variante:

Aus den in s und d angegebenen 16-Bit-Daten wird eine logische Exklusiv-ODER-Verknüpfung gebildet. Das Ergebnis wird an den in d angegebenen Operanden ausgegeben.



- 2. Variante:

Aus den in s1 und s2 angegebenen 16-Bit-Daten wird eine logische Exklusiv-ODER-Verknüpfung gebildet. Das Ergebnis wird an den in d1 angegebenen Operanden ausgegeben.

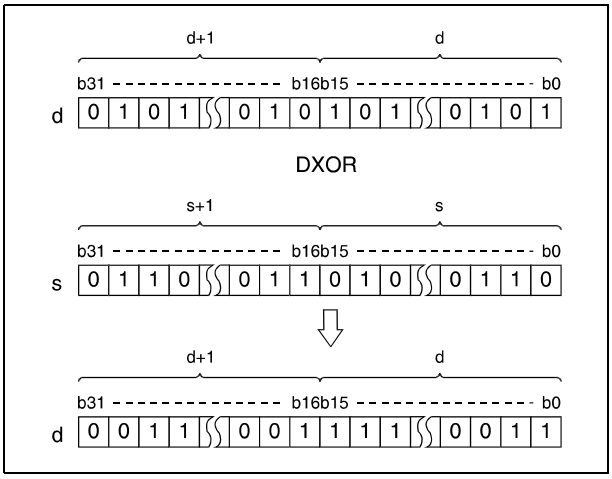


Bits oberhalb der Blocklänge werden auf 0 gesetzt. Ist die Blocklänge beispielsweise mit K2 festgelegt, werden die oberen 8 Bits (b8 bis b15) mit 0 verarbeitet.

DXOR 32-Bit-Daten

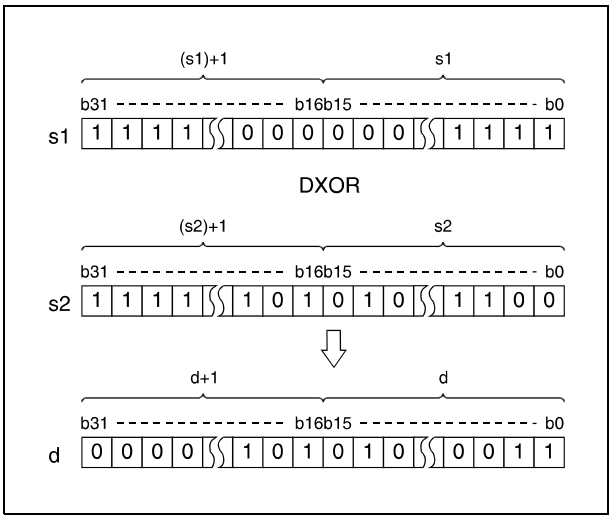
- 1. Variante:

Aus den in s und d angegebenen 32-Bit-Daten wird eine logische Exklusiv-ODER-Verknüpfung gebildet. Das Ergebnis wird an den in d angegebenen Operand ausgegeben.



- 2. Variante (QnA-Serie/System Q):

Aus den in s1 und s2 angegebenen 32-Bit-Daten wird eine logische Exklusiv-ODER-Verknüpfung gebildet. Das Ergebnis wird an den in d angegebenen Operanden ausgegeben.

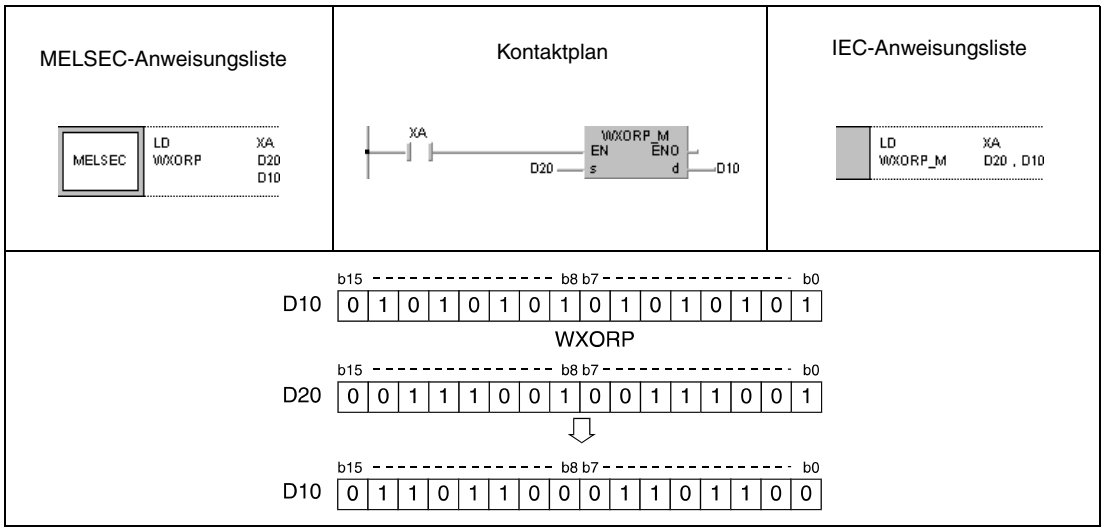


Nach Durchführung der Verknüpfung werden alle Bits, die außerhalb des Blockbereiches liegen, auf 0 gesetzt.

HINWEIS *Es sind keine Fehlerquellen bei der WXOR-, WXORP-, DXOR- und DXORP-Anweisung bei Verwendung der Variante 1 (s, d) bekannt, sofern keine indizierte Adressierung vorgenommen wird.*

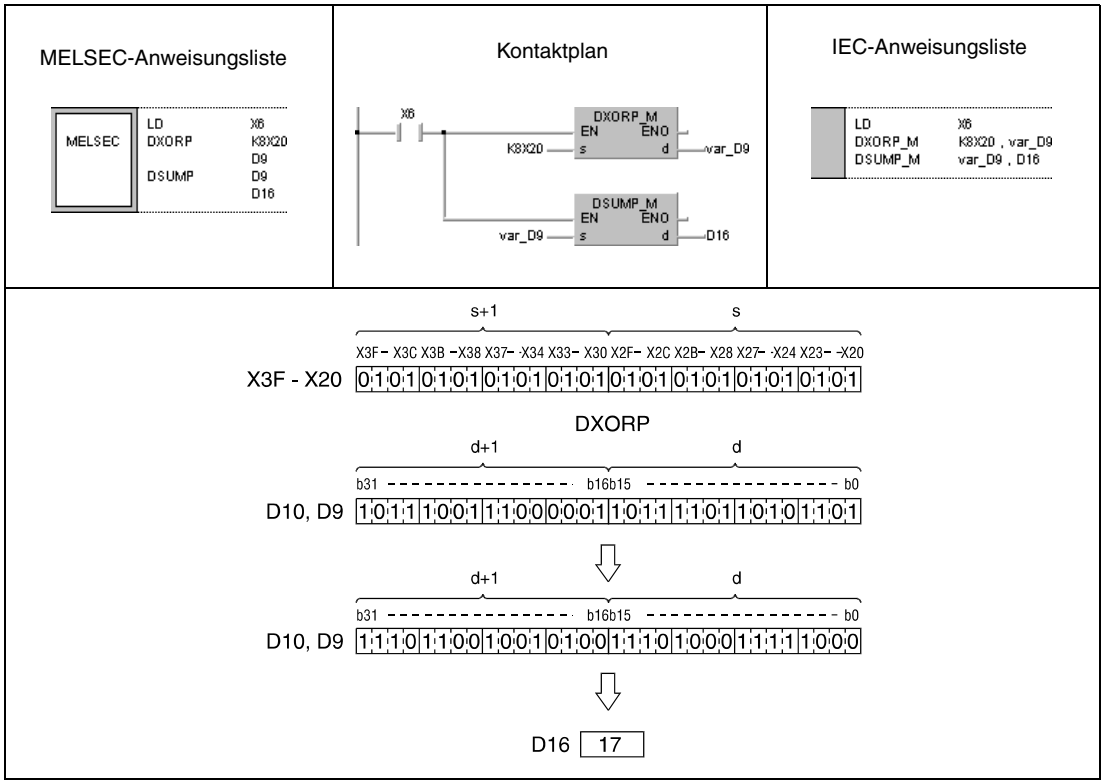
Beispiel 1 WXORP (s, d)

Das folgende Programm verknüpft mit positiver Flanke von XA die Daten aus D10 mit den Daten aus D20 und speichert das Ergebnis wieder in D10.



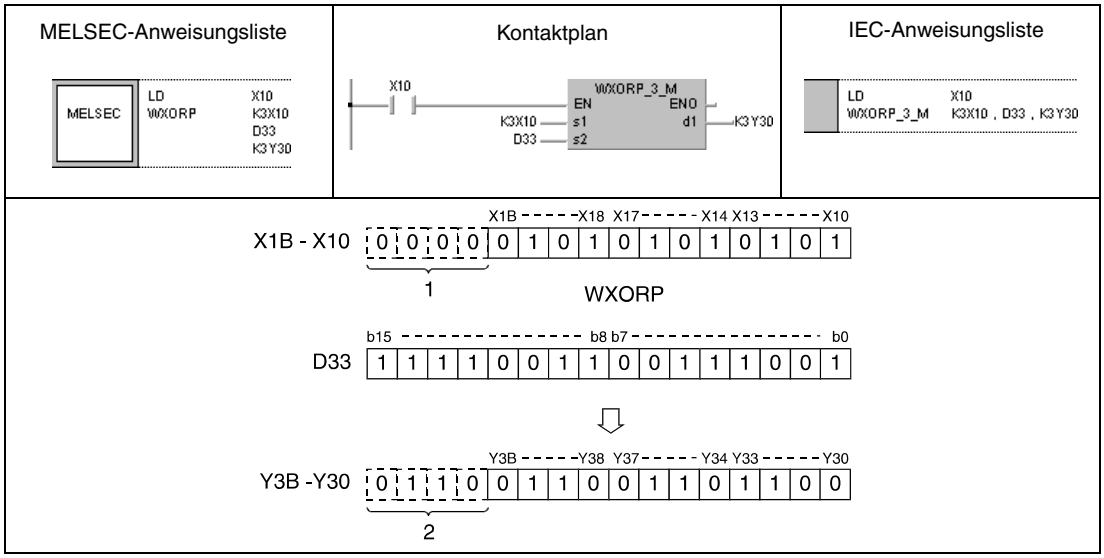
Beispiel 2 DXORP (s, d)

Das folgende Programm vergleicht den 32-Bit-Datenwert der Eingänge X20 bis X3F mit dem Bit-Muster der Datenregister D9 und D10. Das Ergebnis wird wieder in D9 und D10 gespeichert. Die Anzahl der gesetzten Bits in D9 und D10 wird in D16 gespeichert. Die Ausführung der Exklusiv-ODER-Verknüpfung erfolgt mit der positiven Flanke von X6.



Beispiel 3 WXORP (s1, s2, d1)

Das folgende Programm führt mit positiver Flanke von X10 eine Exklusiv-ODER-Verknüpfung der Eingangsdaten X10 bis X1B mit den Daten aus D33 durch. Das Ergebnis wird in D33 gespeichert und an die Ausgänge Y30 bis Y3B ausgegeben.

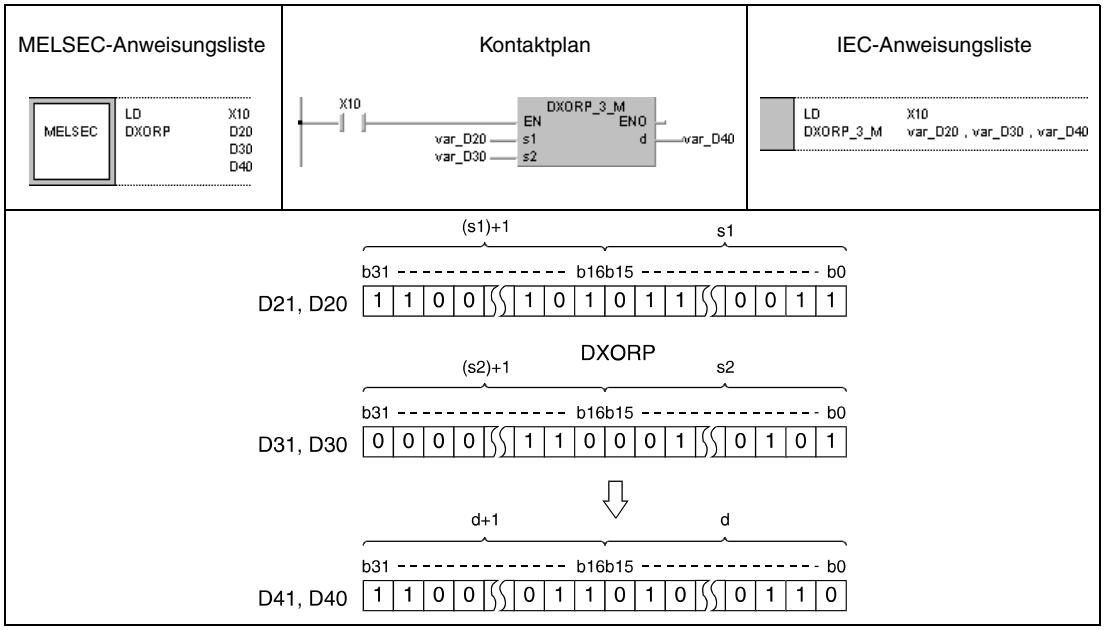


1 Diese Bits werden als 0 eingelesen

2 Diese Bits verändern ihren Zustand nicht

Beispiel 4 DXORP (s1, s2, d)

Das folgende Programm führt mit positiver Flanke von X10 eine Exklusiv-ODER-Operation mit den Daten in D20 und D21 und den Daten in D30 und D31 durch. Das Ergebnis wird in D40 und D41 gespeichert.



HINWEIS *Die Programmbeispiele 2 und 4 sind ohne Variablendefinition im Header der Programm-Organisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.1.6 BKXOR, BKXORP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

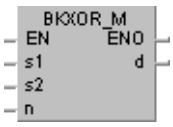
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	—	●	●	—	—	—	—	—	—	SM0	5
s2	—	●	●	—	—	—	—	●	—		
d	—	●	●	—	—	—	—	—	—		
n	●	●	●	●	●	●	●	●	—		

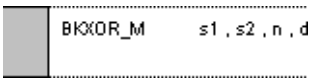
GX IEC Developer

MELSEC-Anweisungsliste

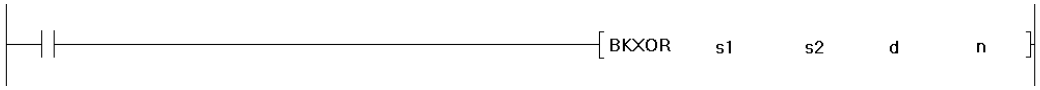
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

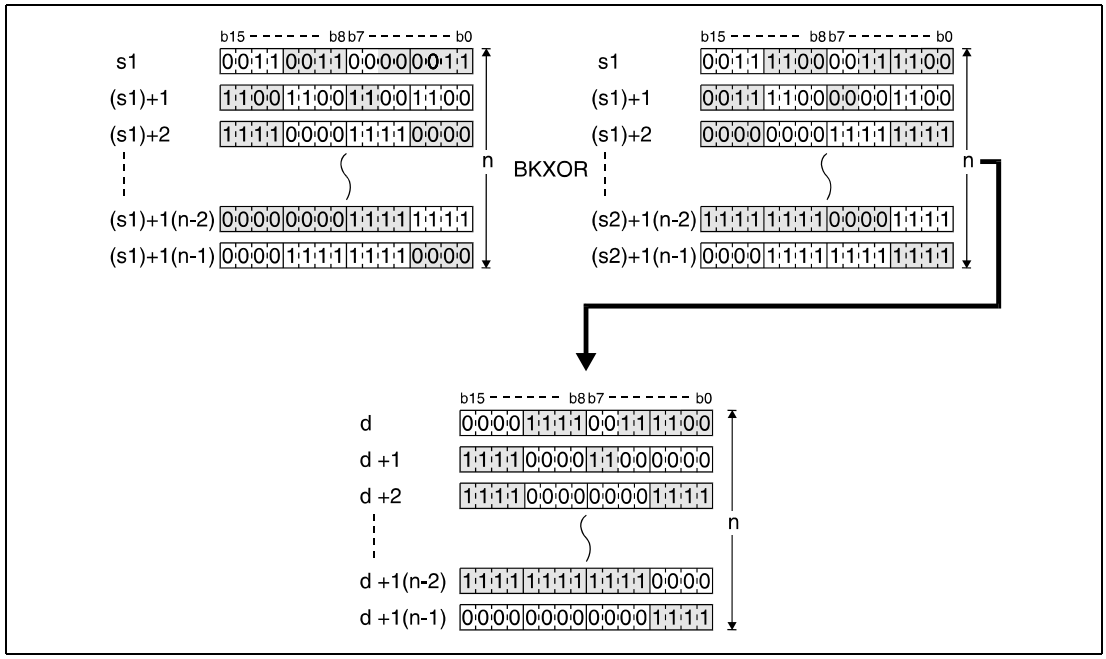
Operand	Befehlswert	Datentyp
s1	Erste Adresse des Operanden, in dem die Daten für die Operation gespeichert sind.	BIN-16-Bit
s2	Erste Adresse der Daten, mit welchen die Operation vorgenommen wird, oder erste Adresse des Operanden, in dem diese Daten gespeichert sind.	
d	Erste Adresse des Operanden, in dem das Ergebnis gespeichert wird.	
n	Anzahl der Datenblöcke, mit der die Exklusiv-ODER-Operation durchgeführt wird.	

Funktions-weise

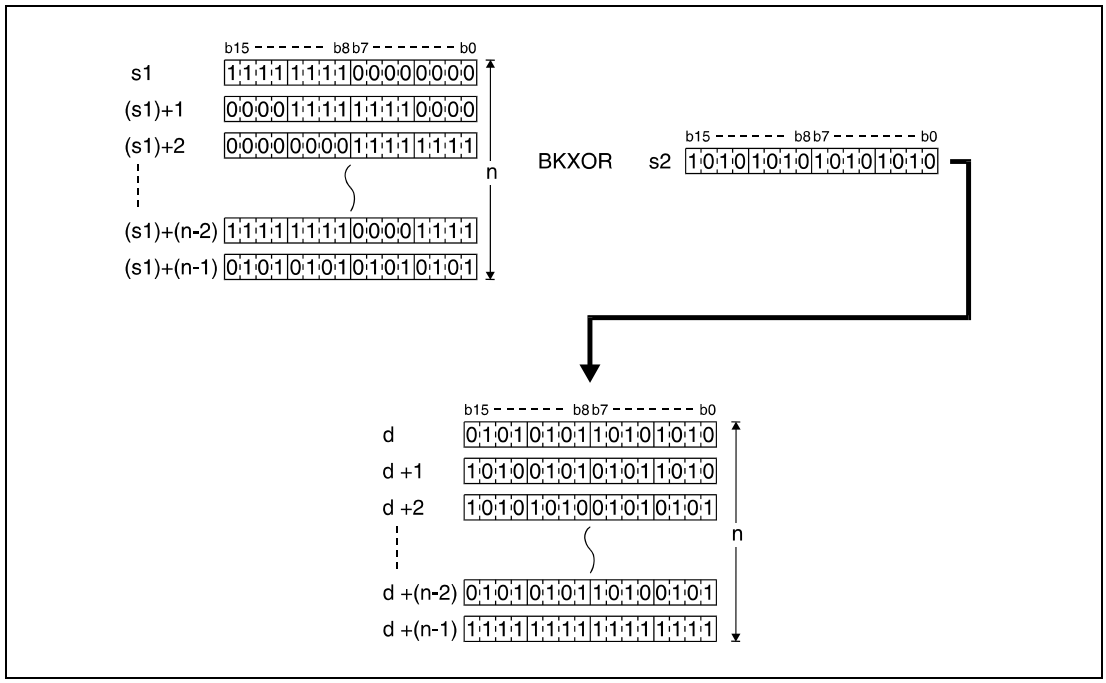
Exklusiv-ODER-Operationen mit 16-Bit-Blöcken

BKXOR Blockweise Exklusiv-ODER-Operation

Die BKXOR-Anweisung führt eine Exklusiv-ODER-Operation mit den n-ten 16-Bit-Blöcken ab s1 und den n-ten 16-Bit-Blöcken ab s2 durch. Der entsprechende 16-Bit-Block des Ergebnisses wird bei dem in d angegebenen Operanden beginnend gespeichert. Die Anzahl der Blöcke mit denen die Operation durchgeführt wird, ist in n angegeben.



Die in s2 abgelegte Konstante muss einen Wert zwischen -32768 und 32767 besitzen.



Fehler-quellen

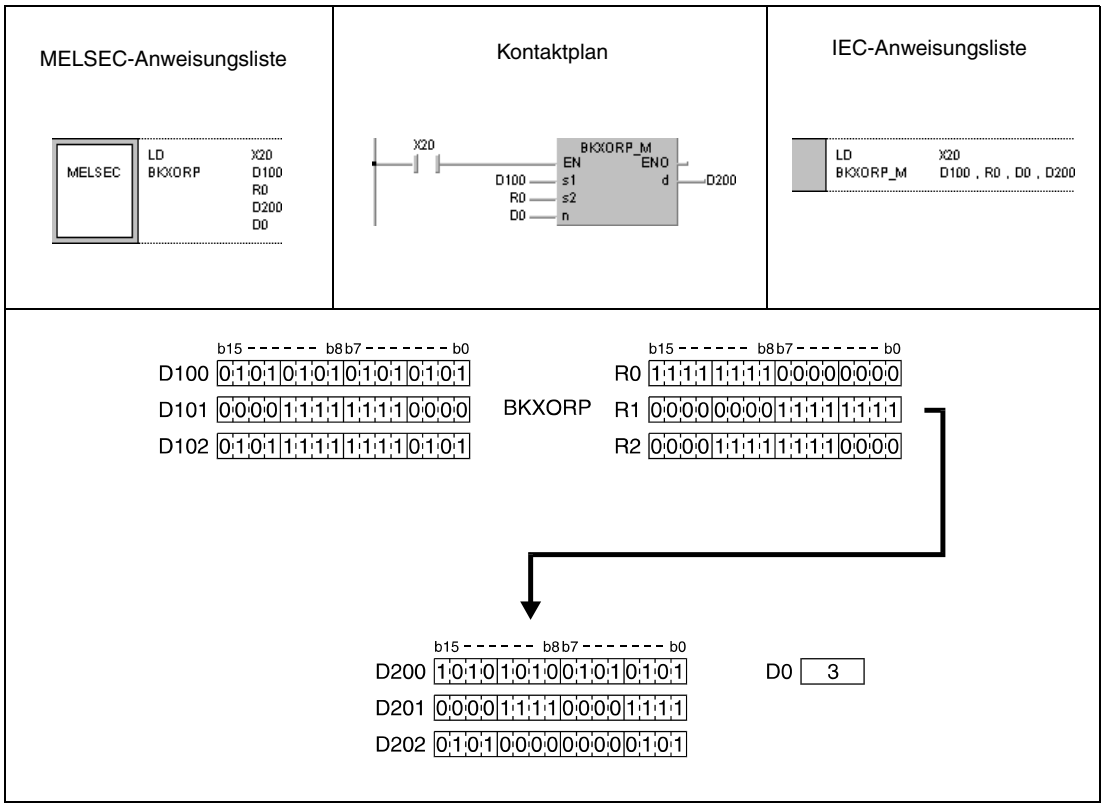
In den folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die in n angegebene Anzahl der Blöcke von s1, s2 oder d liegt außerhalb des für die Speicherung vorgesehenen Bereichs des Operanden (Fehlercode 4101).
- Die für die Speicherung vorgesehenen Bereiche von s1, s2 oder d überlappen (Fehlercode 4101).

Beispiel

BKXORP

Das folgende Programm führt mit der positiven Flanke von X20 eine Exklusiv-ODER-Operation mit den Daten in den Registern D100 bis D102 und den Daten in den Registern R0 bis R2 durch. Das Ergebnis wird in den Registern D200 bis D202 gespeichert. Die Anzahl der an der Operation beteiligten 16-Bit-Datenblöcke (3) ist in D0 hinterlegt.



7.1.7 WXNR, WXNRP, DXNR, DXNRP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
WXNR																										
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				K1 ↓ K4	5 ●[1]	●		●
d		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●										
s1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					7 ●[1]			
s2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●								
d1		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●										
DXNR																										
s	●	●	●	●	●	●	●	●	●	●	●	●	●		●		●	●				K1 ↓ K8	9 ●[1]	●		●
d		●	●	●	●	●	●	●	●	●	●	●	●		●											

[1] Die Anzahl der Schritte bei Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

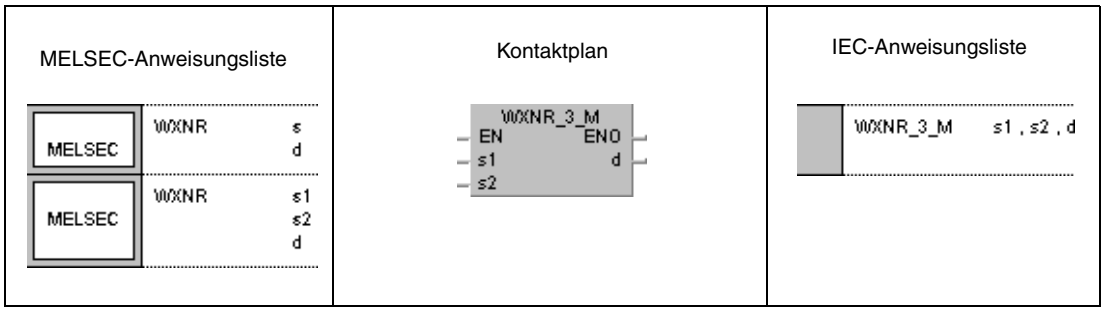
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
WXNR, WXNRP											
s	●	●	●	●	●	●	●	●	—	—	3
d	●	●	●	●	●	●	●	—	—	—	
s1	●	●	●	●	●	●	●	●	—	—	4
s2	●	●	●	●	●	●	●	●	—	—	
d	●	●	●	●	●	●	●	—	—	—	
DXNR, DXNRP											
s	●	●	●	●	●	●	●	●	—	—	4[1)]
d	●	●	●	●	●	●	●	—	—	—	
s1	●	●	●	●	●	●	●	●	—	—	4[2)]
s2	●	●	●	●	●	●	●	●	—	—	
d1	●	●	●	●	●	●	●	—	—	—	

[1] Die Anzahl der Programmschritte ist abhängig vom Typ der CPU und von den verwendeten Operanden.
Bei Verwendung einer QnA-CPU: 4
Bei einer Single-Prozessor-Q-CPU 3
Bei Multi-Prozessor-Q-CPU, interne Wortoperanden (außer File-Register ZR) oder Konstanten: 6
Bei Multi-Prozessor-Q-CPU und Bit-Operanden, deren Adresse ein Vielfaches von16 ist, die die Bit-Blockbezeichnung 4 haben und die nicht durch Index-Vergabe bearbeitet werden: 6
Bei Verwendung einer Multi-Prozessor-Q-CPU und anderer Operanden als oben aufgeführt: 4

[2] Die Anzahl der Programmschritte ist abhängig vom Typ der CPU und von den verwendeten Operanden.
Bei Verwendung einer QnA-CPU: 4
Bei einer Q-CPU und interne Wortoperanden (außer File-Register ZR) oder Konstanten: 6
Bei einer Q-CPU und Bit-Operanden, deren Adresse ein Vielfaches von16 ist, die die Bit-Blockbezeichnung 4 haben und die nicht durch Index-Vergabe bearbeitet werden: 6
Bei Verwendung einer Q-CPU und anderer Operanden als oben aufgeführt: 4

GX IEC Developer



GX Developer

Variablen

Operand	Befehlswert	Datentyp
s	Daten, mit denen die Exklusiv-NOR-Operation durchgeführt wird, oder erste Adresse des Operanden, in dem diese Daten gespeichert sind.	BIN-16-/32-Bit
d		
s1	Daten, mit denen die Exklusiv-NOR-Operation durchgeführt wird, oder erste Adresse des Operanden, in dem diese Daten gespeichert sind.	
s2		
d (bei WXNRP d1)	Erste Adresse des Operanden, in dem das Ergebnis gespeichert wird.	

Funktionsweise

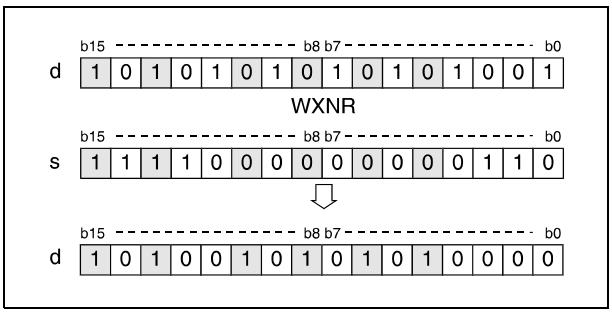
Exklusiv-NOR-Logik

WXNR 16-Bit-Daten

Die Exklusiv-NOR-Logik (englisch: exclusive NOR) bildet das logische Produkt aus der logischen Summe zweier Eingangsvariablen (Y= ($\overline{A}$+B) x (A+$\overline{B}$)).

- 1. Variante:

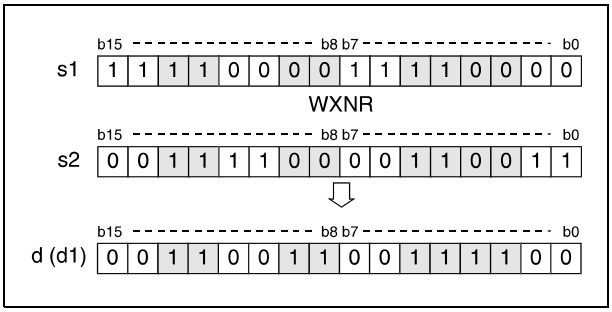
Aus den in s und d angegebenen 16-Bit-Daten wird eine logische Exklusiv-NOR-Verknüpfung gebildet. Das Ergebnis wird an den in d angegebenen Operanden ausgegeben.



- 2. Variante:

Aus den in s1 und s2 angegebenen 16-Bit-Daten wird eine logische Exklusiv-NOR-Verknüpfung gebildet. Das Ergebnis wird an den in d angegebenen Operanden ausgegeben.

Bei Verwendung der WXNRP-Anweisung wird das Ergebnis an den in d1 angegebenen Operanden ausgegeben.

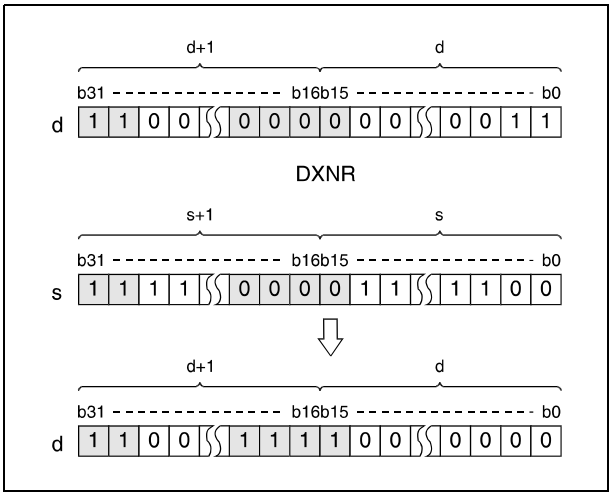


Bits oberhalb der Blocklänge werden auf 0 gesetzt. Ist die Blocklänge beispielsweise mit K2 festgelegt, werden die oberen 8 Bits (b8 bis b15) mit 0 verarbeitet.

DXNR 32-Bit-Daten

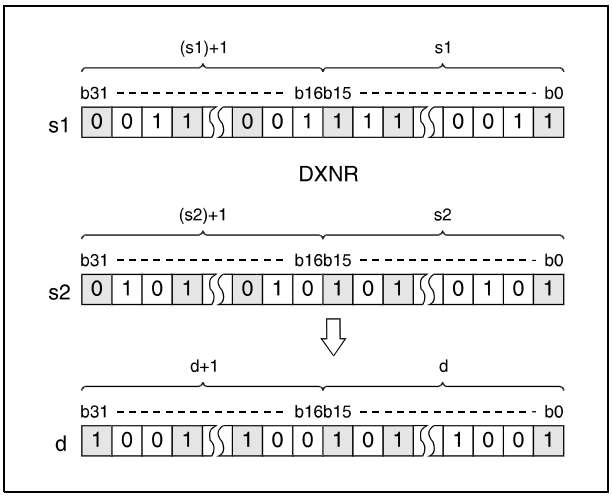
- 1. Variante:

Aus den in s und d angegebenen 32-Bit-Daten wird eine logische Exklusiv-NOR-Verknüpfung gebildet. Das Ergebnis wird an den in d angegebenen Operanden ausgegeben.



- 2. Variante (QnA-Serie/System Q):

Aus den in s1 und s2 angegebenen 32-Bit-Daten wird eine logische Exklusiv-NOR-Verknüpfung gebildet. Das Ergebnis wird an den in d angegebenen Operanden ausgegeben.



Nach Durchführung der Verknüpfung werden alle Bits, die außerhalb des Blockbereiches liegen, auf 0 gesetzt.

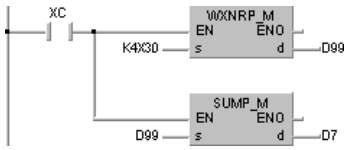
Beispiel 1 WXNRP (s, d)

Im folgenden Programm wird mit positiver Flanke von XC das Bit-Muster des 16-Bit-Datenwertes der Eingänge X30 bis X3F mit dem Datenwert in D99 mittels einer Exklusiv-NOR-Operation verglichen und das Operationsergebnis wieder in D99 gespeichert. Die Anzahl der gesetzten Bits wird in D7 gespeichert.

MELSEC-Anweisungsliste

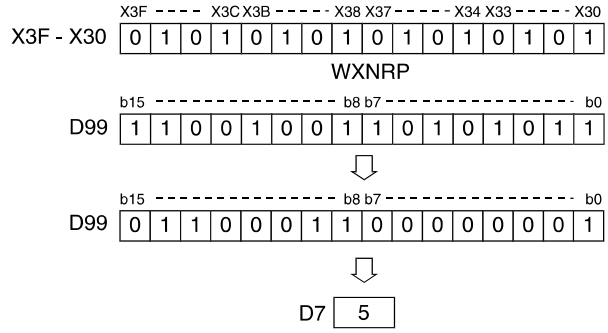
```
MELSEC   LD      XC
         WXNRP   K4X30
                 D99
         SUMP    D99
                 D7
```

Kontaktplan



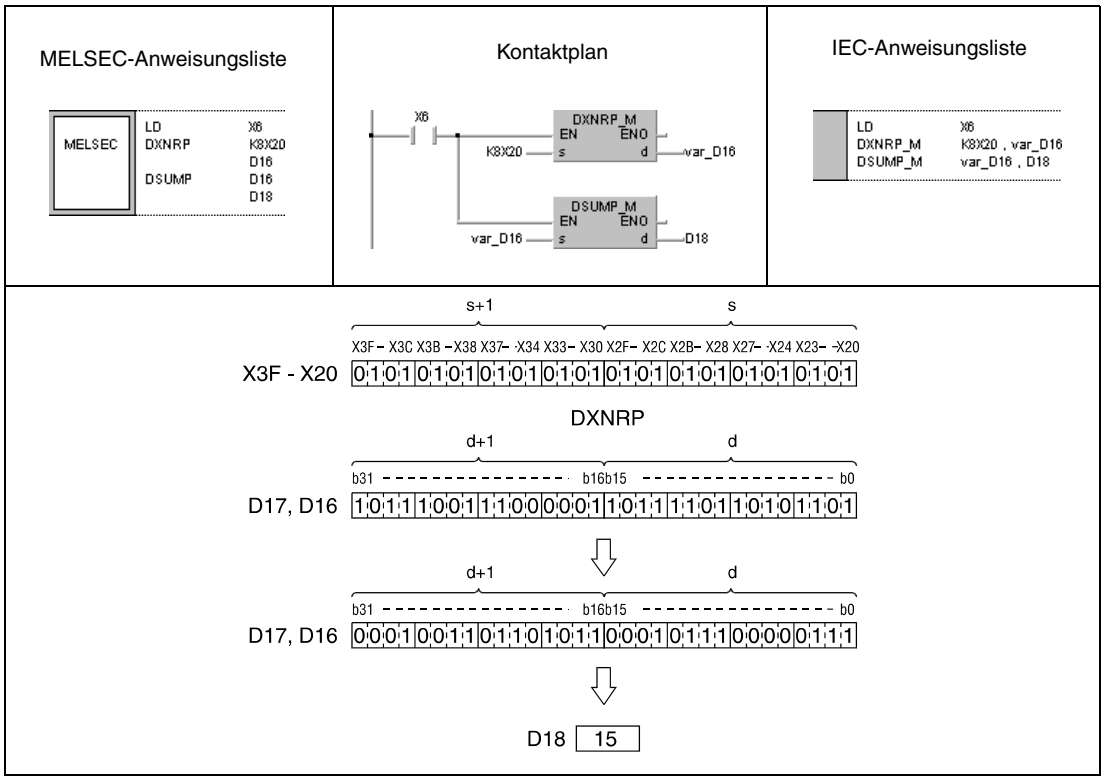
IEC-Anweisungsliste

```
LD      XC
WXNRP_M K4X30 , D99
SUMP_M  D99 , D7
```



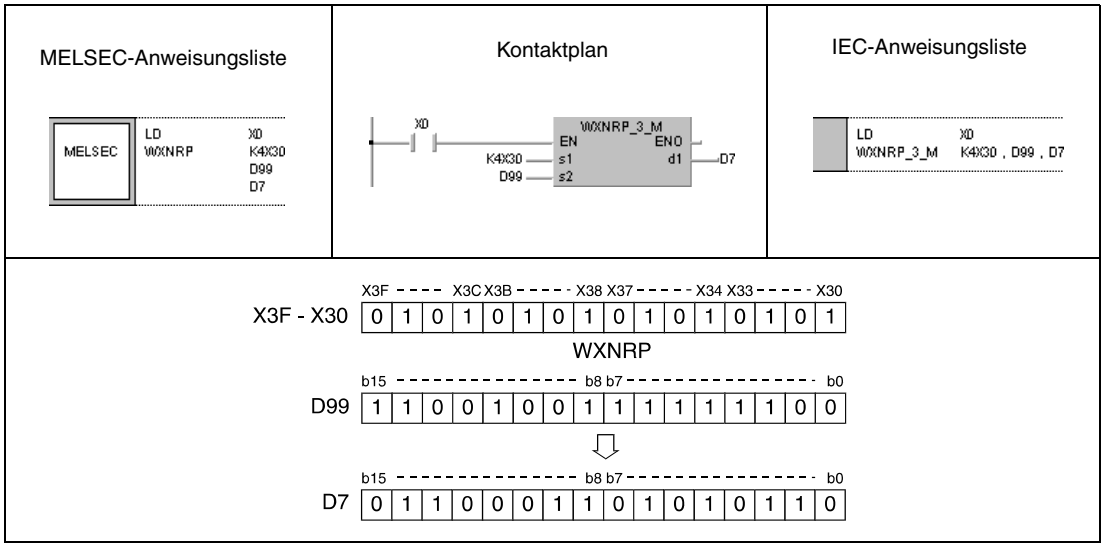
Beispiel 2 DXNRP (s, d)

Im folgenden Programm wird mit positiver Flanke von X6 das Bit-Muster des 32-Bit-Datenwertes der Eingänge X20 bis X3F mit den Daten aus D16 und D17 über eine Exklusiv-NOR-Operation verglichen und das Operationsergebnis wieder in D16 und D17 gespeichert. Die Anzahl der gesetzten Bits wird in D18 gespeichert.



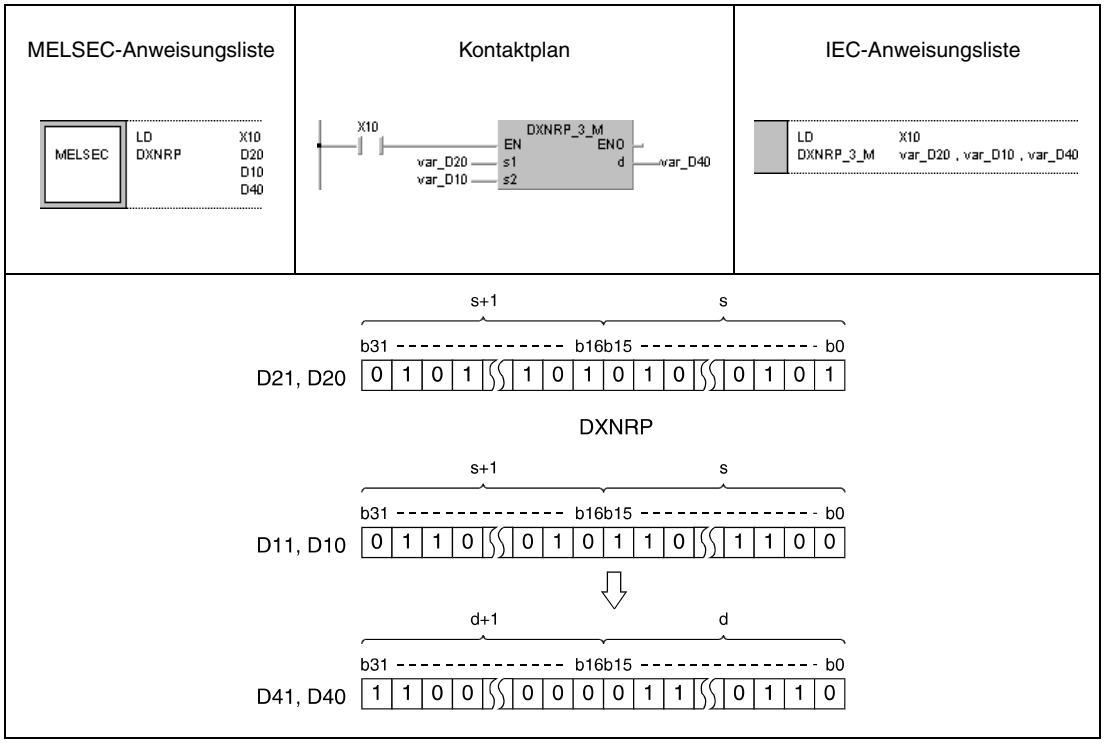
Beispiel 3 WXNRP (s1, s2, d1)

Im folgenden Programm wird mit positiver Flanke von X0 eine Exklusiv-NOR-Operation mit dem 16-Bit-Datenwert der Eingänge X30 bis X3F und dem Datenwert in D99 durchgeführt und das Operationsergebnis in D7 gespeichert.



Beispiel 4 DXNRP (s1, s2, d)

Im folgenden Programm wird bei positiver Flanke von X10 eine Exklusiv-NOR-Operation mit den 32-Bit-Daten in den Registern D20 und D21 und den Daten in D10 und D11 ausgeführt und das Operationsergebnis in D40 und D41 gespeichert.



HINWEIS *Die Programmbeispiele 2 und 4 sind ohne Variablendefinition im Header der Programm-Organisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.1.8 BKXNR, BKXNRP

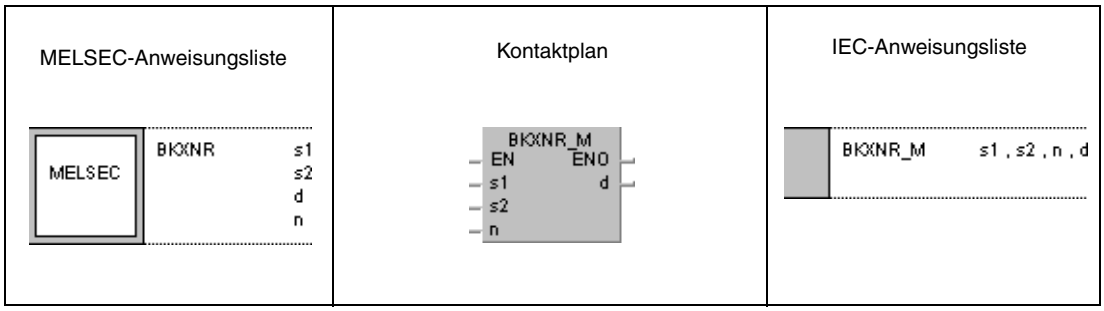
CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

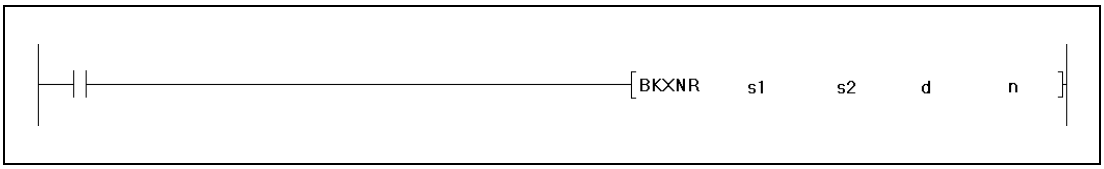
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	—	●	●	—	—	—	—	—	—	SM0	5
s2	—	●	●	—	—	—	—	●	—		
d	—	●	●	—	—	—	—	—	—		
n	●	●	●	●	●	●	●	●	—		

GX IEC Developer



GX Developer



Variablen

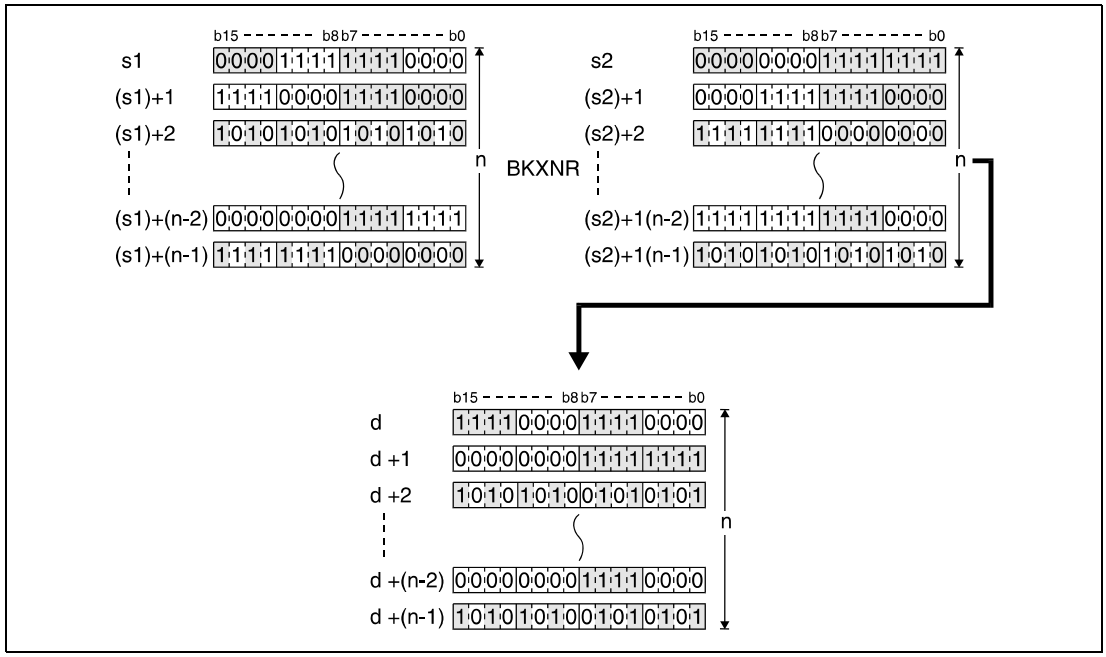
Operand	Befehlswert	Datentyp
s1	Erste Adresse des Operanden, in dem die Daten für die Operation gespeichert sind.	BIN-16-Bit
s2	Erste Adresse der Daten, mit denen die Operation vorgenommen wird, oder erste Adresse des Operanden, in dem diese Daten gespeichert sind.	
d	Erste Adresse des Operanden, in dem das Ergebnis der Operation gespeichert wird.	
n	Anzahl der Datenblöcke, mit denen die Exklusiv-NOR-Operation durchgeführt wird.	

Funktions-weise

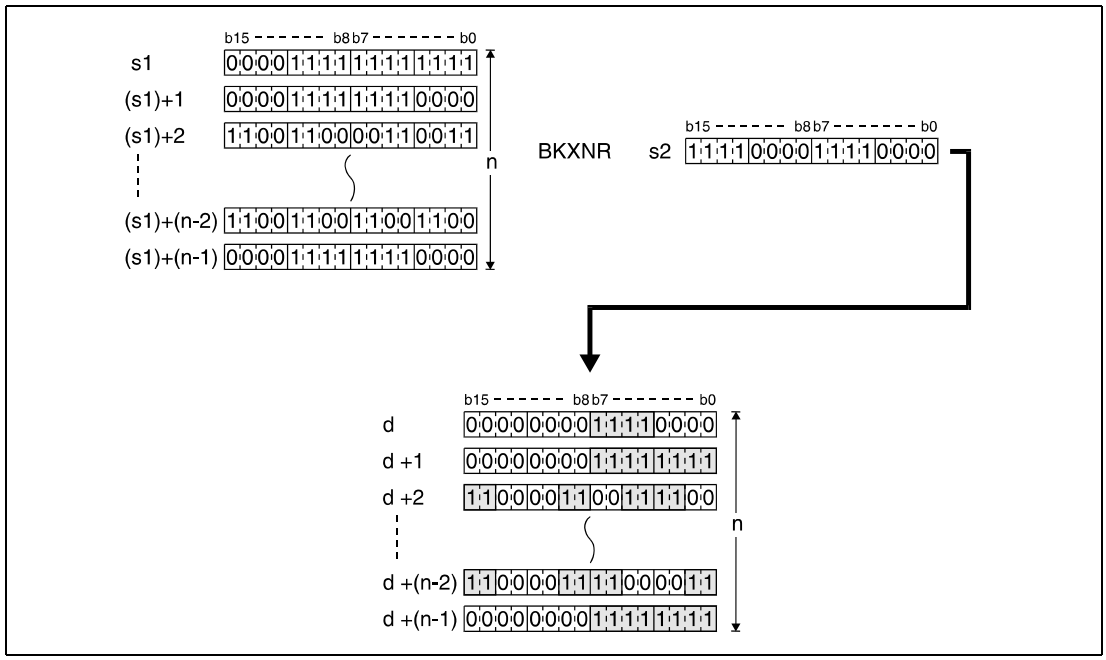
Exklusiv-NOR-Operationen mit 16-Bit-Datenblöcken

BKXNR777 Blockweise Exklusiv-NOR-Operation

Die BKXNR-Anweisung führt eine Exklusiv-NOR-Operation mit den n-ten 16-Bit-Blöcken ab s1 und den n-ten 16-Bit-Blöcken ab s2 durch. Der entsprechende 16-Bit-Block des Ergebnisses wird bei dem in d angegebenen Operanden beginnend gespeichert. Die Anzahl der Blöcke, mit denen die Operation ausgeführt wird, ist in n angegeben.



Eine in s2 abgelegte Konstante muss einen Wert zwischen -32768 und 32767 besitzen.



Fehlerquellen

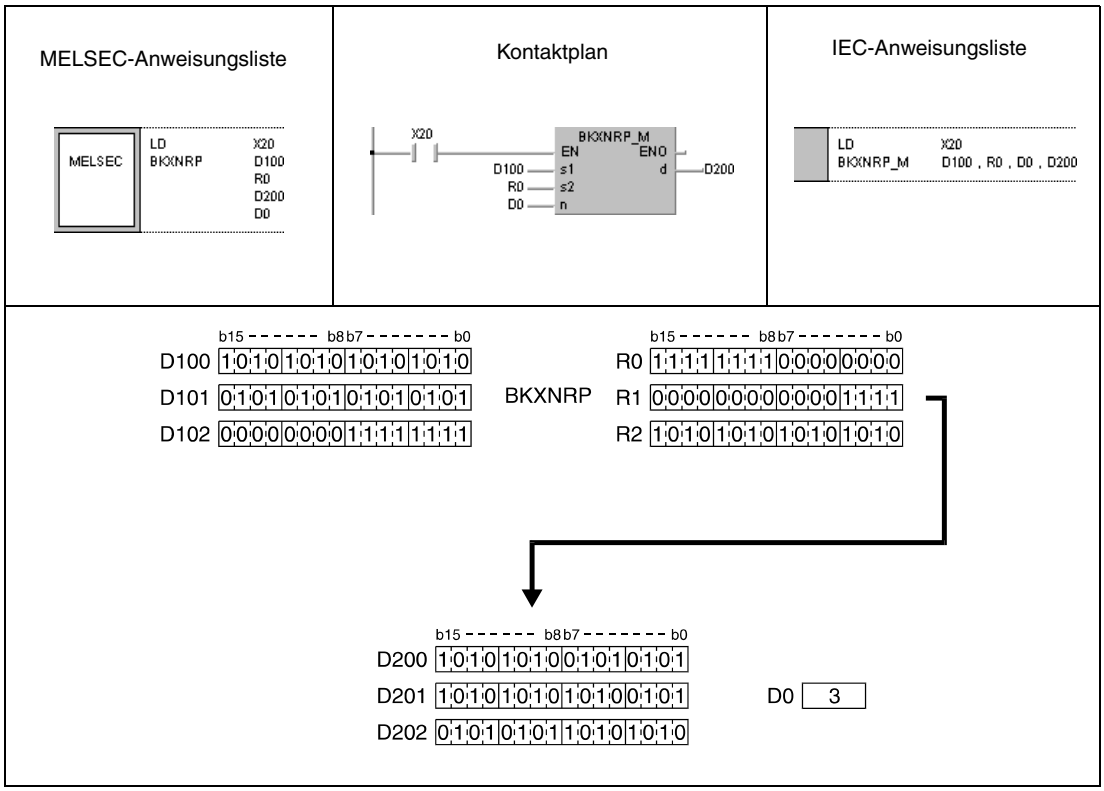
In den folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die in n angegebene Anzahl der Blöcke von s1, s2 oder d liegt außerhalb des für die Speicherung vorgesehenen Bereich des Operanden (Fehlercode 4101).
- Die für die Speicherung vorgesehenen Bereiche von s1, s2 oder d überlappen (Fehlercode 4101).

Beispiel

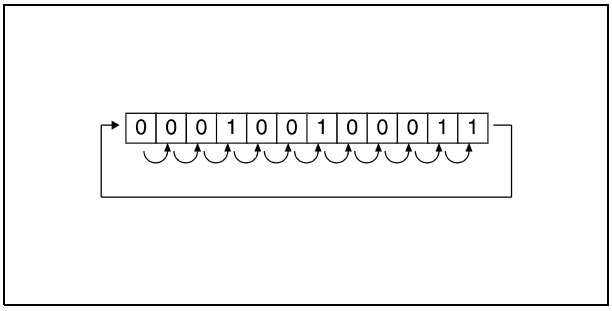
BKXNRP

Das folgende Programm führt mit positiver Flanke von X20 eine Exklusiv-NOR-Operation mit den Daten in den Registern D100 bis D102 und den Daten in den Registern R0 bis R2 durch. Das Ergebnis wird in den Registern D200 bis D202 gespeichert. Die Anzahl der an der Operation beteiligten 16-Bit-Datenblöcke (3) ist in D0 hinterlegt.



7.2 Rotationsanweisungen

Mit den nachfolgend beschriebenen Rotationsanweisungen können die in den Akkumulatoren und Registern zwischengespeicherten Datenwerte bitweise in sich rotiert werden. Die Rotation kann sowohl nach rechts als auch nach links erfolgen.



Rotationsanweisungen können wahlweise mit Übertrag (Carry Flag) eingesetzt werden. Die Anwendung der Anweisung findet sowohl bei 16- als auch bei 32-Bit-Daten Verwendung. Insgesamt stehen 16 verschiedene Rotationsanweisungen zur Verfügung.

Funktion	MELSEC-Anweisung im MELSEC-Editor	IEC-Anweisung im IEC-Editor
Datenrotation rechts (16 Bit)	ROR	ROR_M
	RORP	RORP_M
	RCR	RCR_M
	RCRP	RCRP_M
Datenrotation links (16 Bit)	ROL	ROL_M
	ROLP	ROLP_M
	RCL	RCL_M
	RCLP	RCLP_M
Datenrotation rechts (32 Bit)	DROR	DROR_M
	DRORP	DRORP_M
	DRCR	DRCR_M
	DRCRP	DRCRP_M
Datenrotation links (32 Bit)	DROL	DROL_M
	DROLP	DROLP_M
	DRCL	DRCL_M
	DRCLP	DRCLP_M

HINWEIS *Nutzen Sie in den IEC-Editoren die IEC-Anweisungen.*

7.2.1 ROR, RORP, RCR, RCRP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)								Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
n																	●	●					3 ●[1]	●	●	●

[1] Die Anzahl der Schritte bei Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

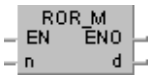
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
d	●	●	●	●	●	●	●	—	—	—	3
n	●	●	●	●	●	●	●	●	—	—	

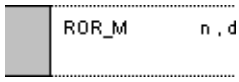
GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan



IEC-Anweisungsliste



GX Developer

HINWEIS *Bei der A-Serie wird immer im Register A0 rotiert. Aus diesem Grund fehlt bei der Programmierung dieses Befehls in der A-Serie der Operand d.*

Variablen

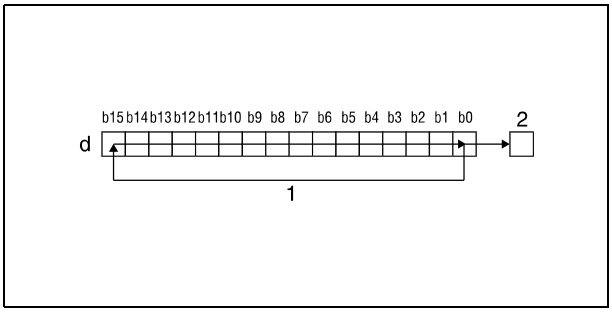
Operand	Befehlswert	Datentyp
d	Startadresse des Operanden, mit dem die Rotationsoperation ausgeführt wird. Dieser Operand ist für die A-Serie immer A0.	BIN-16-Bit
n	Anzahl der Rotationen (0 bis 15).	

Funktionsweise

Datenrotation rechts (16 Bit)

ROR Rotationsanweisung ohne Carry Flag

Die ROR-Anweisung rotiert die Datenbits in dem in d (A0) angegebenen Operanden um n Bits nach rechts. Der Übertrag (Carry Flag) ist hierbei nicht einbezogen. Das Carry Flag (A-Serie = M9012, QnA-Serie/System Q = SM700) nimmt den Wert des zuletzt von b0 nach b15 rotierten Bits an.

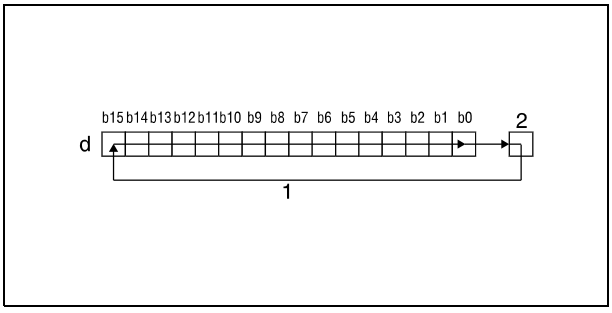


[1] Rotation um n Bits

[2] Carry Flag

RCR Rotationsanweisung mit Carry Flag

Die RCR-Anweisung rotiert die Datenbits in dem in d (A0) angegebenen Operanden um n Bits nach rechts und bezieht dabei das Carry Flag mit ein. Das Carry Flag (A-Serie = M9012,Q-Serie/System Q = SM700) nimmt dabei den Wert des um n Stellen verschobenen Bits an. Der vor der Rotation bestehende Zustand des Carry Flags (0 oder 1) wird in d (A0) von b15 an um n Stellen nach rechts weitergeschoben.



[1] Rotation um n Bits

[2] Carry Flag

HINWEIS

Nur Q-Serie und System Q:

Wurde in d ein Bit-Operand bestimmt, wird die Rotationsoperation mit einem Operanden mit angegebener Datenbreite durchgeführt. Die Anzahl der Stellen, um die die Bits rotiert werden, wird durch den Rest der folgenden Division bestimmt:

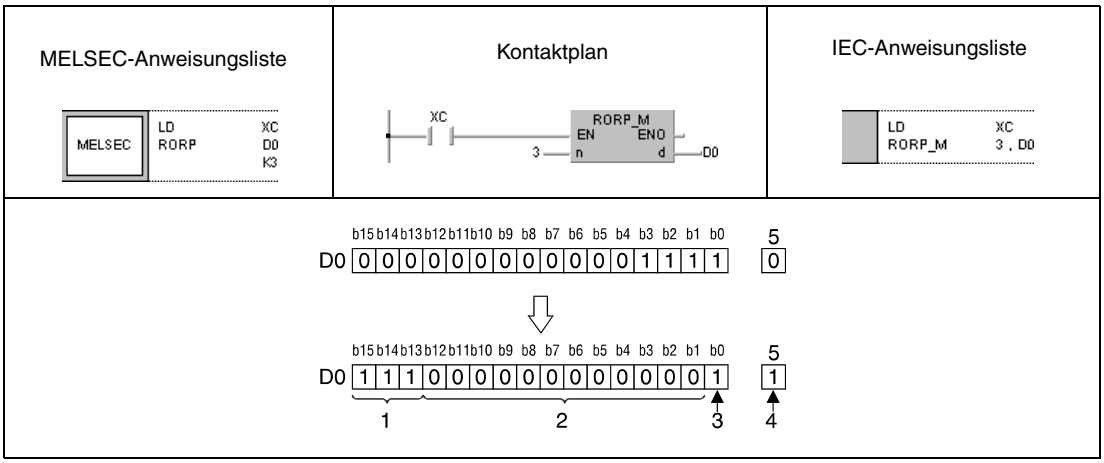
Anzahl der Rotationen n / Anzahl der Bits

Eine 16-fache Rotation von 12 Bits würde beispielsweise einer Rotation um 4 Bits entsprechen, da der Rest der Division 16/12 gleich 4 ist. Der Grund hierfür liegt in der Tatsache, dass das Bit x von 12 Bits nach einer 12-fachen Rotation wieder an der Stelle steht, an der es vor der Rotation gestanden hat.

Geben Sie aus diesem Grund für n einen Wert von 0 bis 15 an.

Beispiel 1 RORP (Q-Serie und System Q)

Das folgende Programm rotiert mit positiver Flanke von XC den Inhalt von D0 um 3 Bits nach rechts.



[1] Inhalt der Bits b0-b2 vor der Rotation

[2] Inhalt der Bits b4-b15 vor der Rotation

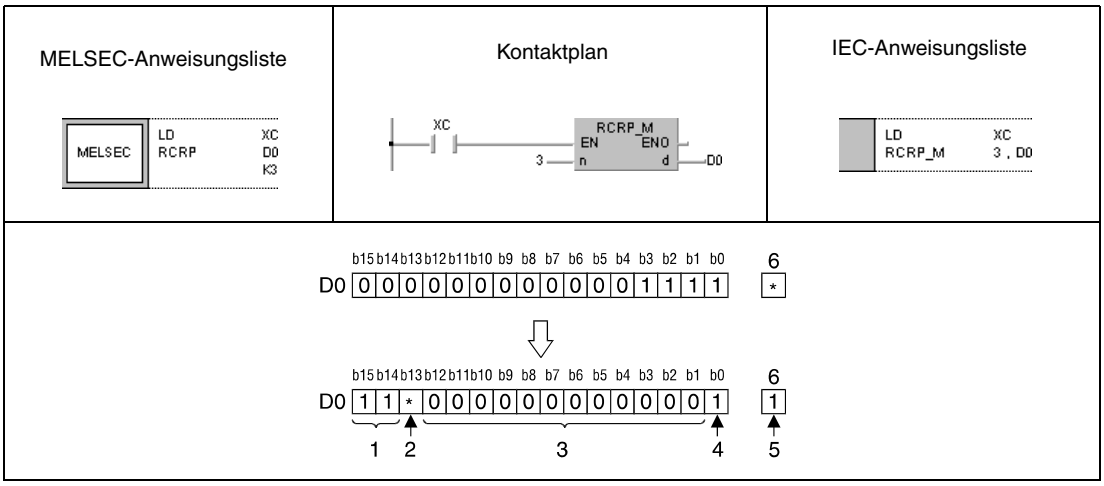
[3] Inhalt des Bits b3 vor der Rotation

[4] Inhalt des Bits b2 vor der Rotation

[5] Carry Flag

Beispiel 2 RCRP (Q-Serie und System Q)

Das folgende Programm rotiert mit positiver Flanke von XC den Inhalt von D0 unter Einbeziehung des Carry Flags SM700 um 3 Bits nach rechts. Der in SM700 vor der Rotation bestehende Zustand (0/1) wird um 3 Stellen nach rechts geschoben.



[1] Inhalt der Bits b1 und b0 vor der Rotation

[2] Inhalt des Carry Flags vor der Rotation

[3] Inhalt der Bits b4-b15 vor der Rotation

[4] Inhalt des Bits b3 vor der Rotation

[5] Inhalt des Bits b2 vor der Rotation

[6] Carry Flag

7.2.2 ROL, ROLP, RCL, RCLP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
n																	●	●					3 ●[1]	●	●	●

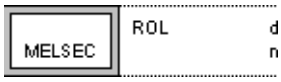
[1] Die Anzahl der Schritte bei Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

Operanden MELSEC Q

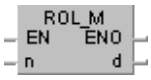
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
d	●	●	●	●	●	●	●	—	—	—	3
n	●	●	●	●	●	●	●	●	—	—	

GX IEC Developer

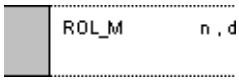
MELSEC-Anweisungsliste



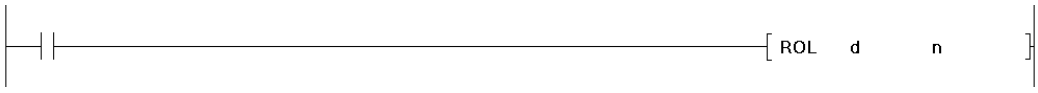
Kontaktplan



IEC-Anweisungsliste



GX Developer



HINWEIS *Bei der A-Serie wird immer im Register A0 rotiert. Aus diesem Grund fehlt bei der Programmierung dieses Befehls in der A-Serie der Operand d.*

Variablen

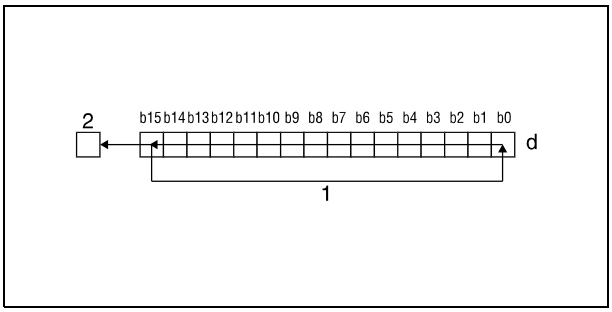
Operand	Befehlswert	Datentyp
d	Startadresse des Operanden, mit dem die Rotationsoperation ausgeführt wird. Dieser Operand ist für die A-Serie immer A0.	BIN-16-Bit
n	Anzahl der Rotationen (0 bis 15).	

Funktions-weise

Datenrotation links (16 Bit)

ROL Rotationsanweisung ohne Carry Flag

Die ROL-Anweisung rotiert die Datenbits in d (A0) um n Bits nach links. Der Übertrag (Carry Flag) ist hierbei nicht einbezogen. Das Carry Flag (A-Serie = M9012, Q-Serie/ System Q = SM700) nimmt den Wert des zuletzt von b15 nach b0 rotierten Bits an.

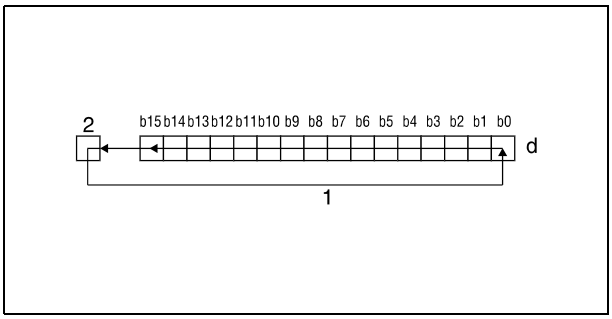


1 Rotation um n Bits

2 Carry Flag

RCL Rotationsanweisung mit Carry Flag

Die RCL-Anweisung rotiert die Datenbits in d (A0) um n Bits nach links und bezieht dabei das Carry Flag mit ein. Das Carry Flag (A-Serie = M9012, Q-Serie/System Q = SM700) nimmt dabei den Wert des um n Stellen verschobenen Bits an. Der vor der Rotation bestehende Zustand des Carry Flags (0 oder 1) wird in d (A0) von b0 an um n Stellen nach links weitergeschoben.



1 Rotation um n Bits

2 Carry Flags

HINWEIS

Nur Q-Serie und System Q:

Wurde in d ein Bit-Operand bestimmt, wird die Rotationsoperation mit einem Operanden mit angegebener Datenbreite durchgeführt. Die Anzahl der Stellen, um die die Bits rotiert werden, wird durch den Rest der folgenden Division bestimmt:

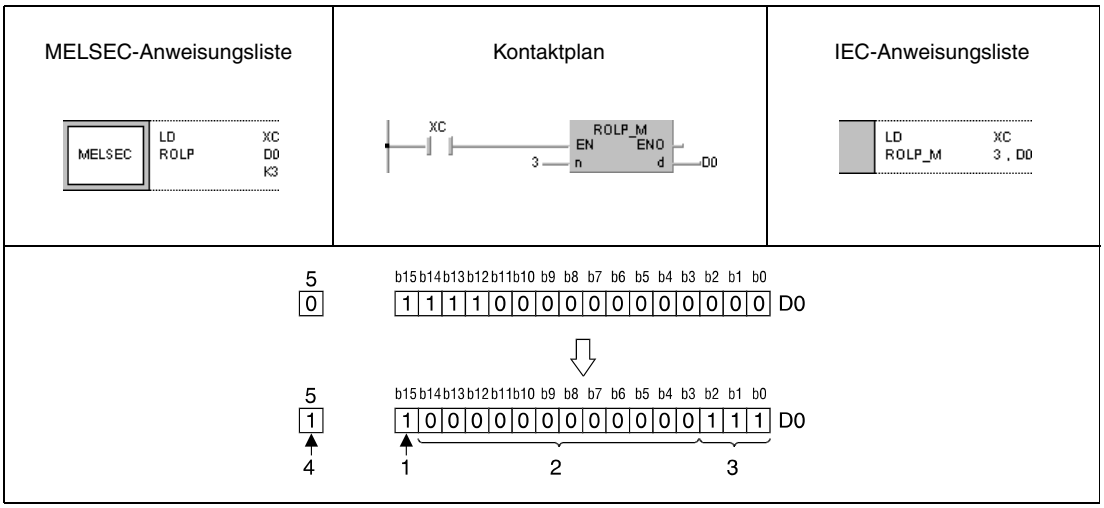
Anzahl der Rotationen n / Anzahl der Bits

Eine 16-fache Rotation von 12 Bits würde beispielsweise einer Rotation um 4 Bits entsprechen, da der Rest der Division 16/12 gleich 4 ist. Der Grund hierfür liegt in der Tatsache, dass das Bit x von 12 Bits nach einer 12-fachen Rotation wieder an der Stelle steht, an der es vor der Rotation gestanden hat.

Geben Sie aus diesem Grund für n einen Wert von 0 bis 15 an.

Beispiel 1 ROLP (Q-Serie und System Q)

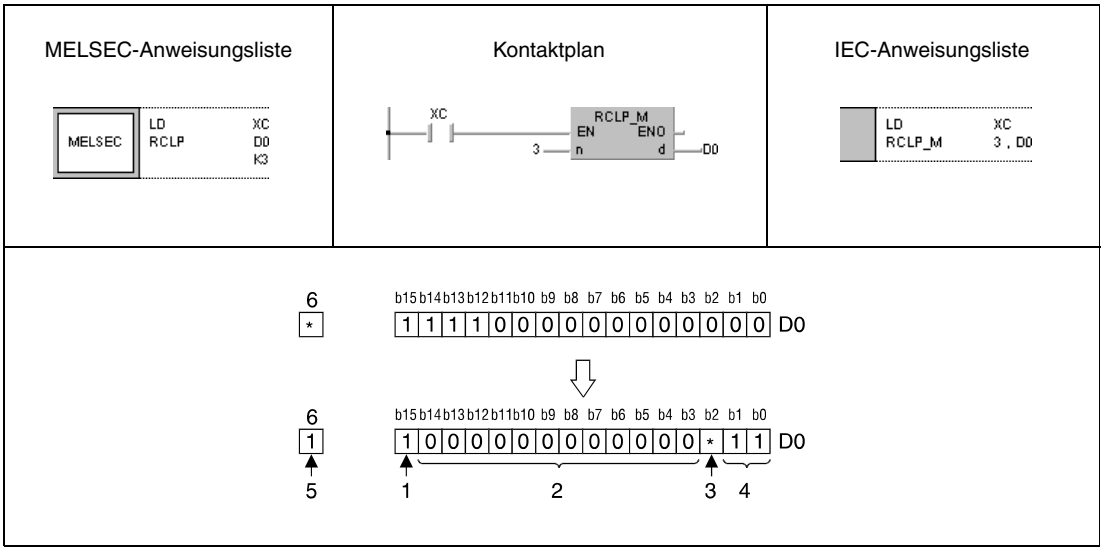
Das folgende Programm rotiert mit der positiven Flanke von XC den Inhalt von D0 um 3 Bits nach links.



[1] Inhalt des Bits b12 vor der Rotation
[2] Inhalt der Bits b11-b0 vor der Rotation
[3] Inhalt der Bits b15-b13 vor der Rotation
[4] Inhalt des Bits b12 vor der Rotation
[5] Carry Flag

Beispiel 2 RCLP (Q-Serie und System Q)

Das folgende Programm rotiert mit der positiven Flanke von XC den Inhalt von D0 unter Einbeziehung des Carry Flags SM700 um 3 Bits nach links. Der in SM700 vor der Rotation bestehende Zustand (0/1) wird um 3 Stellen nach links geschoben.



[1] Inhalt des Bits b12 vor der Rotation
[2] Inhalt der Bits b11-b0 vor der Rotation
[3] Inhalt des Carry Flags
[4] Inhalt der Bits b14 und b15 vor der Rotation
[5] Inhalt des Carry Flags vor der Rotation
[6] Carry Flag

7.2.3 DROR, DRORP, DRCR, DRCRP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)								Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
n																	●	●					3 ●[1]	●	●	●

[1] Die Anzahl der Schritte bei Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

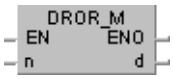
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
d	●	●	●	●	●	●	●	—	—	—	3
n	●	●	●	●	●	●	●	●	—	—	

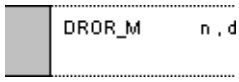
GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan



IEC-Anweisungsliste



GX Developer

HINWEIS *Bei der A-Serie wird immer in den Registern A0 und A1 rotiert. Aus diesem Grund fehlt bei der Programmierung dieses Befehls in der A-Serie der Operand d.*

Variablen

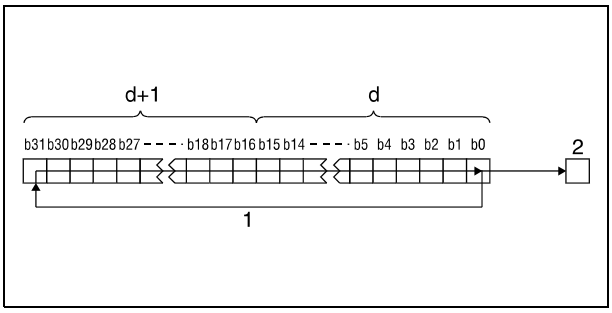
Operand	Befehlswert	Datentyp
d	Erste Adresse des Operanden, mit dem die Rotationsoperation ausgeführt wird. Diese Operanden sind für die A-Serie immer A0 und A1.	BIN-32-Bit
n	Anzahl der Rotationen (0 bis 31).	BIN-16-Bit

Funktionsweise

Datenrotation rechts (32 Bit)

DROR Rotationsanweisung ohne Carry Flag

Die DROR-Anweisung rotiert die Datenbits in d (A0, A1) um n Bits nach rechts. Der Übertrag (Carry Flag) ist hierbei nicht einbezogen. Das Carry Flag (A-Serie = M9012, Q-Serie/System Q = SM700) nimmt den Wert des zuletzt von b0 nach b31 rotierten Bits an.

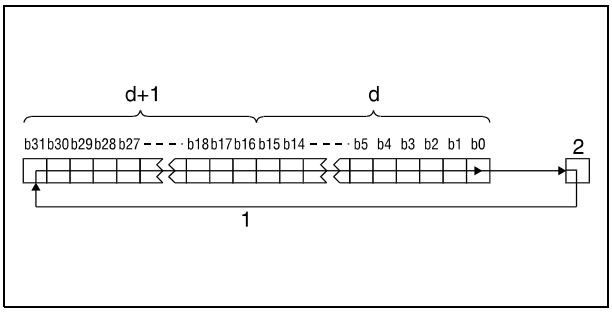


[1] Rotation um n Bits

[2] Carry Flag

DRCR Rotationsanweisung mit Carry Flag

Die DRCR-Anweisung rotiert die Datenbits in d (A0, A1) um n Bits nach rechts und bezieht dabei das Carry Flag mit ein. Das Carry Flag (A-Serie = M9012, Q-Serie/System Q = SM700) nimmt dabei den Wert des um n Stellen verschobenen Bits an. Der vor der Rotation bestehende Zustand des Carry Flags (0 oder 1) wird in d (A0, A1) von b31 an um n Stellen nach rechts weitergeschoben.



[1] Rotation um n Bits

[2] Carry Flag

HINWEIS

Nur Q-Serie und System Q

Wurde in d ein Bit-Operand bestimmt, wird die Rotationsoperation mit einem Operanden mit angegebener Datenbreite durchgeführt. Die Anzahl der Stellen, um die die Bits rotiert werden, wird durch den Rest der folgenden Division bestimmt:

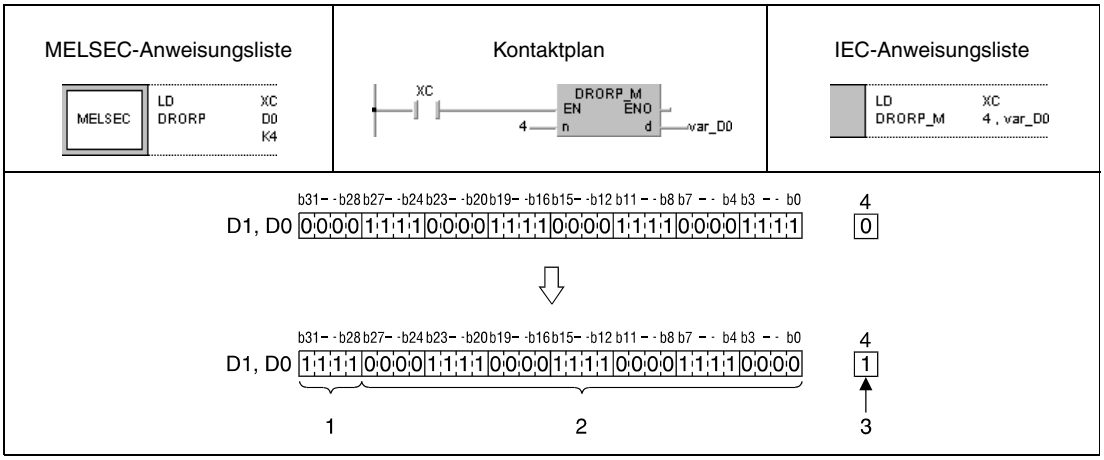
Anzahl der Rotationen n / Anzahl der Bits

Eine 31-fache Rotation von 24 Bits würde beispielsweise einer Rotation um 7 Bits entsprechen, da der Rest der Division 31/24 gleich 7 ist. Der Grund hierfür liegt in der Tatsache, dass das Bit x von 24 Bits nach einer 24-fachen Rotation wieder an der Stelle steht, an der es vor der Rotation gestanden hat.

Geben Sie für n einen Wert von 0 bis 31 an.

Beispiel 1 DRORP (Q-Serie und System Q)

Das folgende Programm rotiert mit der positiven Flanke von XC den Inhalt von D0 und D1 um 4 Bits nach rechts.



[1] Inhalt der Bits b3-b0 vor der Rotation

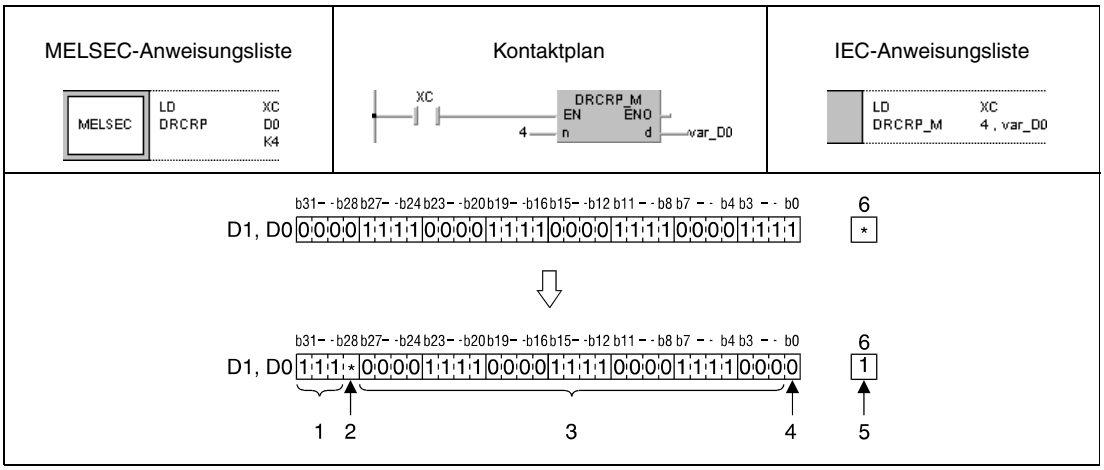
[2] Inhalt der Bits b31-b4 vor der Rotation

[3] Inhalt des Bits b3 vor der Rotation

[4] Carry Flag

Beispiel 2 DRCRP (Q-Serie und System Q)

Das folgende Programm rotiert mit der positiven Flanke von XC den Inhalt von D0 und D1 unter Einbeziehung des Carry Flags SM700 um 4 Bits nach rechts. Der in SM700 vor der Rotation bestehende Zustand (0/1) wird um 4 Stellen nach rechts geschoben.



[1] Inhalt der Bits b2-b0 vor der Rotation

[2] Inhalt des Carry Flags vor der Rotation

[3] Inhalt der Bits b5-b31 vor der Rotation

[4] Inhalt des Bits b4 vor der Rotation

[5] Inhalt des Bits b3 vor der Rotation

[6] Carry Flag

HINWEIS *Diese Programmbeispiele sind ohne Variablendefinition im Header der Programm-Organisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.2.4 DROL, DROLP, DRCL, DRCLP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	

Operanden MELSEC A

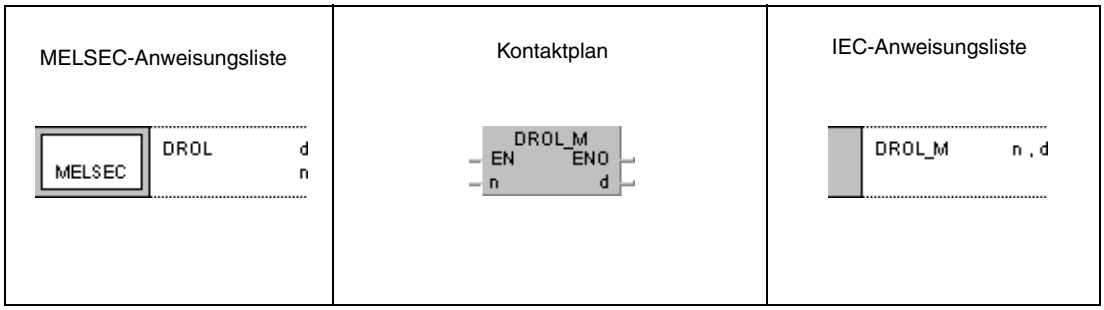
	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
n																	●	●					3 ●[1]	●	●	●

[1] Die Anzahl der Schritte bei Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten E	Andere		
	Bit	Wort		Bit	Wort						
d	●	●	●	●	●	●	●	—	—	—	3
n	●	●	●	●	●	●	●	●	—	—	

GX IEC Developer



GX Developer

HINWEIS *Bei der A-Serie wird immer in den Registern A0 und A1 rotiert. Aus diesem Grund fehlt bei der Programmierung dieses Befehls in der A-Serie der Operand d.*

Variablen

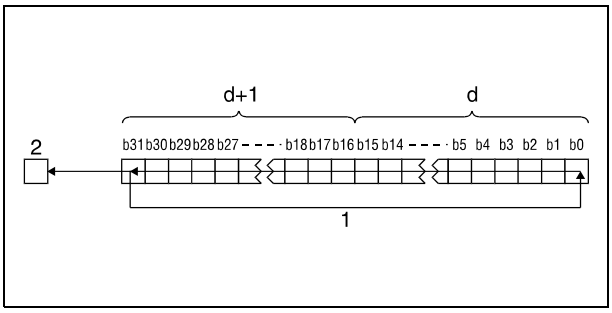
Operand	Befehlswert	Datentyp
d	Erste Adresse des Operanden, mit dem die Rotationsoperation ausgeführt wird. Diese Operanden sind für die A-Serie immer A0 und A1.	BIN-32-Bit
n	Anzahl der Rotationen (0 bis 31).	BIN-16-Bit

Funktions-weise

Datenrotation links (32 Bit)

DROL Rotationsanweisung ohne Carry Flag

Die DROL-Anweisung rotiert die Datenbits in d (A0, A1) um n Bits nach links. Der Übertrag (Carry Flag) ist hierbei nicht einbezogen. Das Carry Flag (A-Serie = M9012, Q-Serie/System Q = SM700) nimmt den Wert des zuletzt von b31 nach b0 rotierten Bits an.

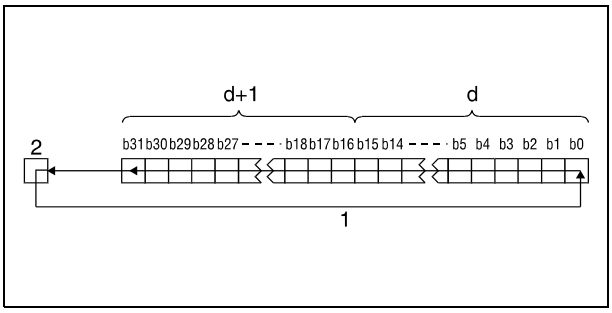


[1] Rotation um n Bits

[2] Carry Flag

DRCL Rotationsanweisung mit Carry Flag

Die DRCL-Anweisung rotiert die Datenbits in d (A0, A1) um n Bits nach links und bezieht dabei das Carry Flag mit ein. Das Carry Flag (A-Serie = M9012, Q-Serie/System Q = SM700) nimmt dabei den Wert des um n Stellen verschobenen Bits an. Der vor der Rotation bestehende Zustand des Carry Flags (0 oder 1) wird in d (A0, A1) von b31 an um n Stellen nach links weitergeschoben.



[1] Rotation um n Bits

[2] Carry Flag

HINWEIS *Nur Q-Serie* und System Q

Wurde in d ein Bit-Operand bestimmt, wird die Rotationsoperation mit einem Operanden mit angegebener Datenbreite durchgeführt. Die Anzahl der Stellen, um die die Bits rotiert werden, wird durch den Rest der folgenden Division bestimmt:

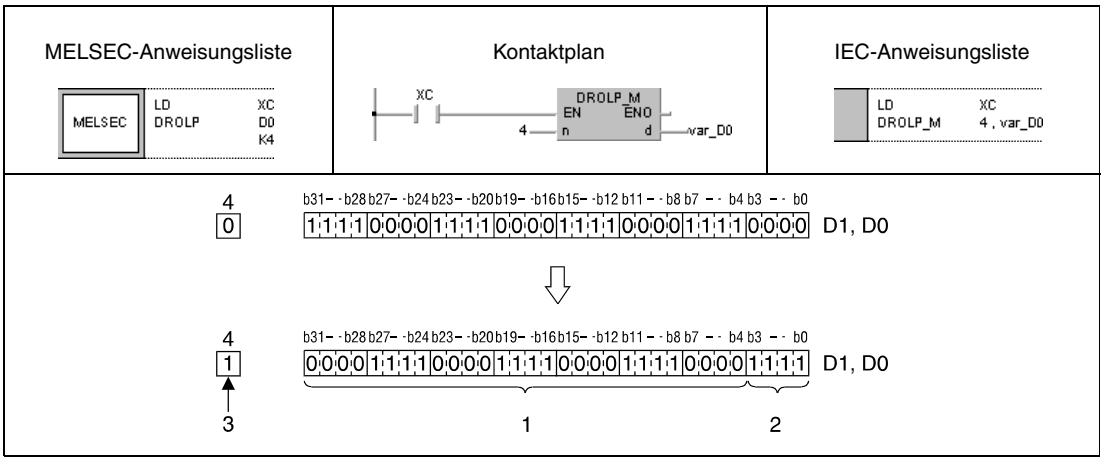
Anzahl der Rotationen n / Anzahl der Bits

Eine 31-fache Rotation von 24 Bits würde beispielsweise einer Rotation um 7 Bits entsprechen, da der Rest der Division 31/24 gleich 7 ist. Der Grund hierfür liegt in der Tatsache, dass das Bit x von 24 Bits nach einer 24-fachen Rotation wieder an der Stelle steht, an der es vor der Rotation gestanden hat.

Geben Sie für n einen Wert von 0 bis 31 an.

Beispiel 1 DROLP (Q-Serie und System Q)

Das folgende Programm rotiert mit der positiven Flanke von XC den Inhalt von D0 und D1 um 4 Bits nach links.



1 Inhalt der Bits b27-b0 vor der Rotation

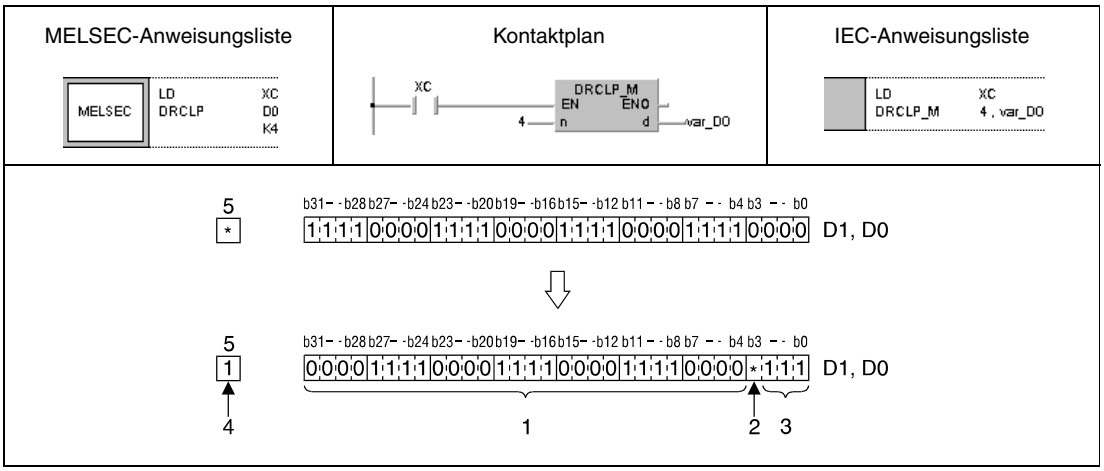
2 Inhalt der Bits b31-b28 vor der Rotation

3 Inhalt des Bits b28 vor der Rotation

4 Carry Flag

Beispiel 2 DRCLP (Q-Serie und System Q)

Das folgende Programm rotiert mit positiver Flanke von XC den Inhalt von D0 und D1 unter Einbeziehung des Carry Flags SM700 um 4 Bits nach links. Der in SM700 vor der Rotation bestehende Zustand (0/1) wird um 4 Stellen nach links geschoben.



1 Inhalt der Bits b27-b0 vor der Rotation

2 Inhalt des Carry Flags vor der Rotation

3 Inhalt der Bits b31-b29 vor der Rotation

4 Inhalt des Bits b28 vor der Rotation

5 Carry Flag

HINWEIS *Diese Programmbeispiele sind ohne Variablendefinition im Header der Programm-Organisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.3 Verschiebeanweisungen

Die Verschiebeanweisungen (SHIFT-Anweisungen) ermöglichen das bit- oder blockweise Verschieben von Daten innerhalb eines Datenwortes. Die Verschiebung kann sowohl nach rechts als auch nach links erfolgen.

Insgesamt stehen 12 Verschiebeanweisungen zur Verfügung.

Funktion	MELSEC-Anweisung im MELSEC-EDITOR	MELSEC-Anweisung im IEC-EDITOR
Verschiebung eines 16-Bit-Datenwortes um n Bit	SFR	SFR_M
	SFRP	SFRP_M
	SFL	SFL_M
	SFLP	SFLP_M
Verschiebung von n Bit-Operanden um 1 Bit	BSFR	BSFR_M
	BSFRP	BSFRP_M
	BSFL	BSFL_M
	BSFLP	BSFLP_M
Verschiebung von n Wort-Operanden um 1 Stelle	DSFR	DSFR_M
	DSFRP	DSFRP_M
	DSFL	DSFL_M
	DSFLP	DSFLP_M

HINWEIS *Nutzen Sie in den IEC-Editoren die IEC-Anweisungen.*

7.3.1 SFR, SFRP, SFL, SFLP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)								Konstante		Pointer		Ebene				
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
d		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●						K1 ↓ K4	3 ●[1]	●	●	●
n																	●	●								

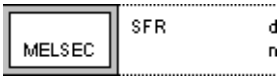
[1] Die Anzahl der Schritte bei der Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

Operanden MELSEC Q

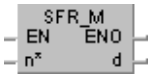
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
d	●	●	●	●	●	●	●	—	—	—	3
n	●	●	●	●	●	●	●	●	—	—	

GX IEC Developer

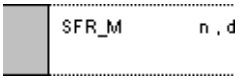
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

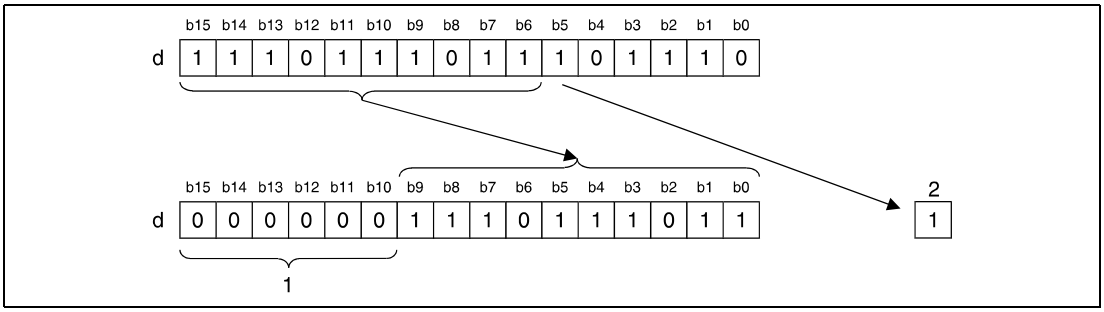
Operand	Befehlswert	Datentyp
d	Erste Adresse des Operanden, in dem die Daten für die Verschiebeoperation gespeichert sind.	BIN-16-Bit
n	Anzahl der Verschiebungen (0 bis 15).	

Funktionsweise

Verschiebung eines 16-Bit-Datenwortes um n Bit

SFR Verschiebung nach rechts

Die SFR-Anweisung verschiebt das in d vorgegebene 16-Bit-Datenwort bitweise um n Bits nach rechts.



[1] Diese Bits werden mit 0 beschrieben

[2] Carry Flag

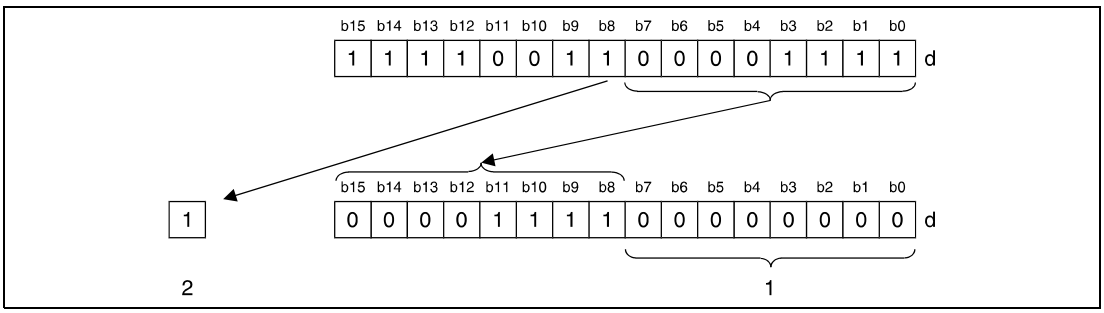
Die höchstwertigen n Bits werden, beginnend mit Bit b15, auf 0 gesetzt. Das n-te zu verschiebende Bit (b(n-1)) wird in das Carry Flag (A-Serie = M9012, Q-Serie/System Q = SM700) geschoben.

Bei Timern oder Countern wird der Istwert (Zählwert) verschoben. Ein Verschieben des Sollwertes (Vorgabewert) ist nicht möglich.

Bei der Verwendung von Bit-Operanden ist das Verschieben innerhalb eines Operanden mit angegebener Datenbreite möglich (siehe Beispiel 1).

SFL Verschiebung nach links

Die SFL-Anweisung verschiebt das in d vorgegebene 16-Bit-Datenwort bitweise um n Bits nach links.



[1] Diese Bits werden mit 0 beschrieben

[2] Carry Flag

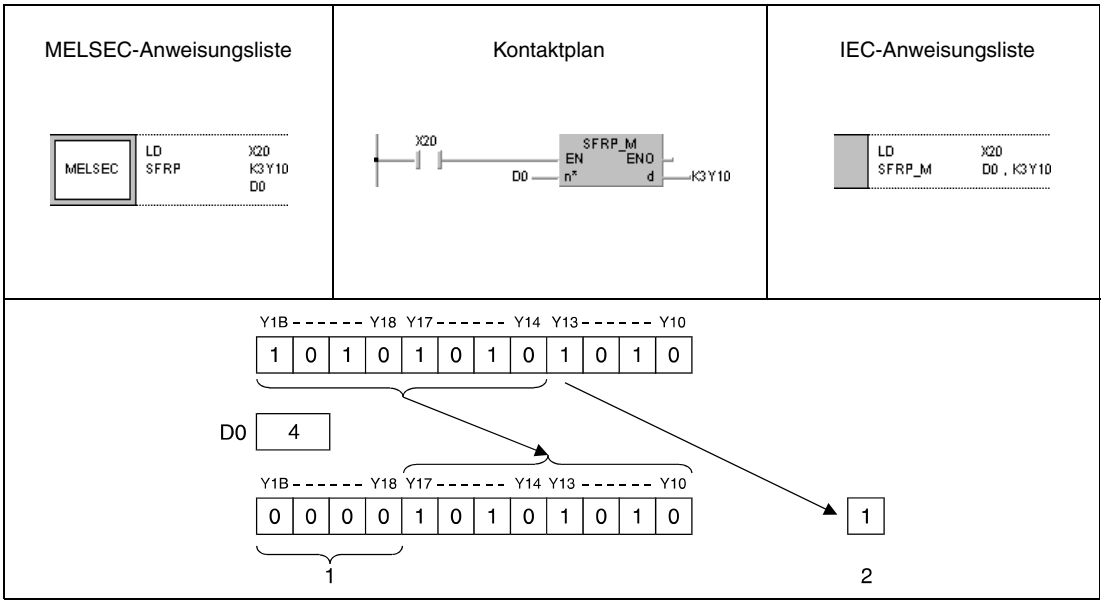
Die niedrigstwertigen n Bits werden, beginnend mit Bit b0, auf 0 gesetzt. Das n-te zu verschiebende Bit (b(15-n)) wird in das Carry Flag (A-Serie = M9012, Q-Serie/System Q = SM700) geschoben.

Bei Timern oder Countern wird der Istwert (Zählwert) verschoben. Ein Verschieben des Sollwertes (Vorgabewert) ist nicht möglich.

Bei der Verwendung von Bit-Operanden ist das Verschieben innerhalb eines Operanden mit angegebener Datenbreite möglich (siehe Beispiel 2).

Beispiel 1 SFRP

Im folgenden Programm wird mit der positiven Flanke von X20 der Inhalt von Y10 bis Y1B um die in D0 angegebene Anzahl Bits nach rechts geschoben. Der Zustand von Bit Y13 wird hierbei im Carry Flag (A-Serie = M9012, Q-Serie/System Q = SM700) gespeichert.

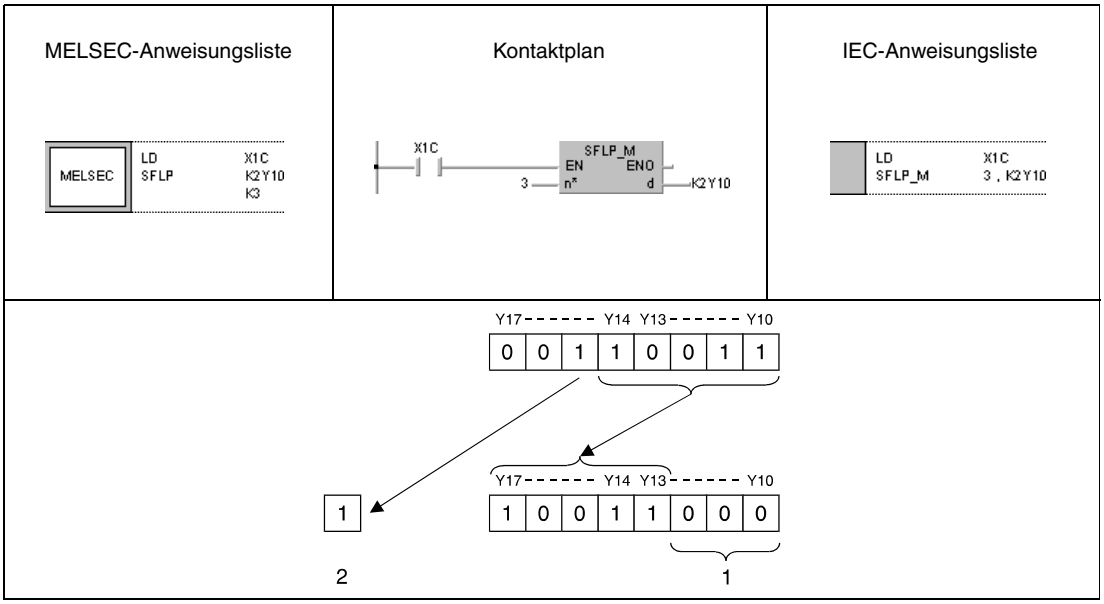


[1] Diese Bits werden mit Null beschrieben

[2] Carry Flag

Beispiel 2 SFLP

Das folgende Programm verschiebt den Inhalt von Y10 bis Y18 mit der positiven Flanke von X1C um 3 Bits nach links. Hierbei wird der Zustand von Y15 im Carry Flag (A-Serie = M9012, Q-Serie/System Q = SM700) gespeichert.



[1] Diese Bits werden mit Null beschrieben

[2] Carry Flag

7.3.2 BSFR, BSFRP, BSFL, BSFLP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)								Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
d		●	●	●	●	●	●																7 ●[1]	●	●	●
n																	●	●								

[1] Die Anzahl der Schritte bei der Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

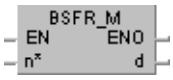
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten E	Andere		
	Bit	Wort		Bit	Wort				U		
d	●	—	—	—	—	—	—	—	—	SM0	3
n	—	●	●	●	●	●	●	●	—		

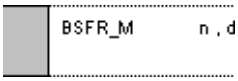
GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

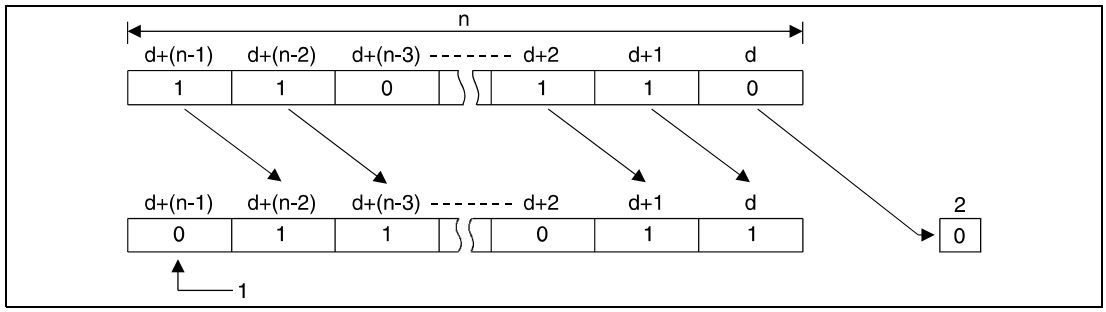
Operand	Befehlswert	Datentyp
d	Erste Adresse des Operanden, der verschoben wird.	Bit
n	Anzahl der zu verschiebenden Operanden.	BIN-16-Bit

Funktions-weise

Verschiebung von n Bit-Operanden um 1 Bit

BSFR Verschiebung nach rechts

Die BSFR-Anweisung verschiebt die Zustände definierter Bit-Operanden um ein Bit nach rechts. Der Schiebevorgang beginnt bei der in d vorgegebenen Operandenadresse und wird für die folgenden n Adressen durchgeführt.

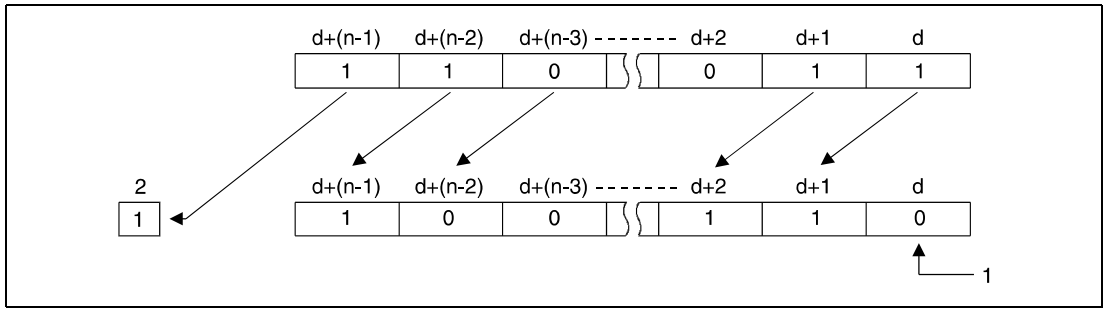


[1] Dieses Bit wird mit 0 beschrieben

[2] Carry Flag

BSFL Verschiebung nach links

Die BSFL-Anweisung verschiebt die Zustände definierter Bit-Operanden um ein Bit nach links. Der Schiebevorgang beginnt bei der in d vorgegebenen Operandenadresse und wird für die folgenden n Adressen durchgeführt.



[1] Dieses Bit wird mit 0 beschrieben

[2] Carry Flag

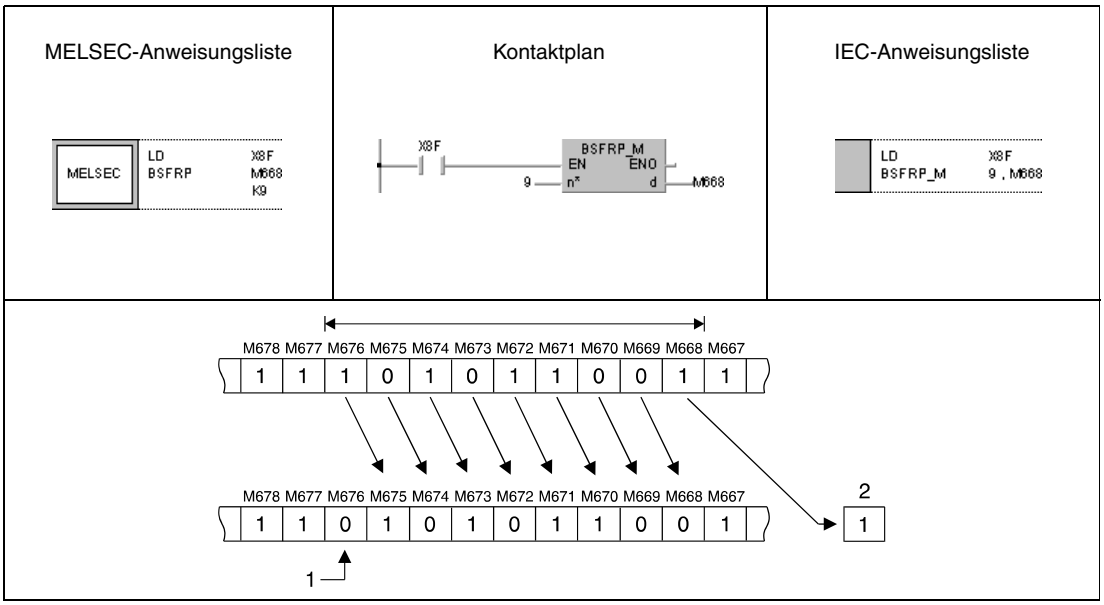
Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Der Wert in n ist negativ.
- Der Wert in n übersteigt die Bitzahl des in d genannten Operanden (Q-Serie/System Q = Fehlercode 4101).

Beispiel 1 BSFRP

Das folgende Programm schiebt mit positiver Flanke von X8F die Daten der Merker M668 bis M676 um ein Bit nach rechts. M668 nimmt den Wert von M669 an, M669 von M670 usw. Der Zustand des ersten Operanden (M668) wird im Carry Flag abgelegt (A-Serie = M9012, Q-Serie/System Q = SM700), und der letzte Operand (M676) nimmt den Wert 0 an.

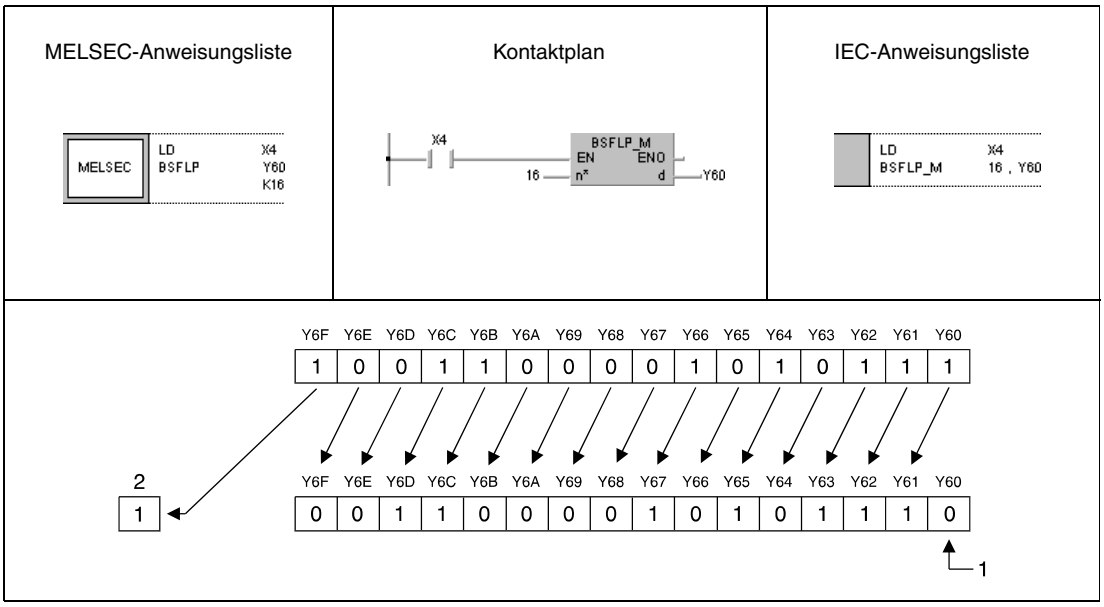


[1] Dieses Bit wird mit 0 beschrieben

[2] Carry Flag

Beispiel 2 BSFLP

Im folgenden Programm werden mit positiver Flanke von X4 die Zustände der Ausgänge Y60 bis Y6F um einen Operanden nach links geschoben. Der Zustand des letzten Ausgangs (Y6F) wird im Carry Flag gespeichert (A-Serie = M9012, Q-Serie/System Q = SM700), und der erste Ausgang (Y60) wird auf 0 gesetzt.



[1] Dieses Bit wird mit 0 beschrieben

[2] Carry Flag

7.3.3 DSFR, DSFRP, DSFL, DSFLP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
d								●	●	●	●	●											7 ●[1]	●		●
n																	●	●								

[1] Die Anzahl der Schritte bei der Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

Operanden MELSEC Q

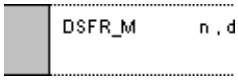
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten E	Andere		
	Bit	Wort		Bit	Wort						
d	—	●	●	—	—	—	—	—	—	SM0	3
n	●	●	●	●	●	●	●	●	—		

GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan

IEC-Anweisungsliste



GX Developer

Variablen

Operand	Befehlswert	Datentyp
d	Erste Adresse des Operanden, der verschoben wird.	BIN-16-Bit
n	Anzahl der zu verschiebenden Operanden.	BIN-16-Bit

Funktionsweise

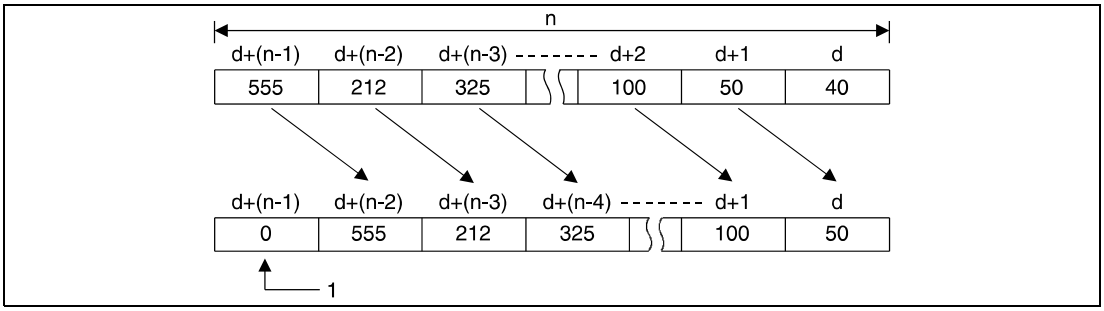
Verschiebung von n Wortoperanden um 1 Adresse

DSFR Verschiebung nach rechts

Die DSFR-Anweisung verschiebt den Inhalt definierter Wortoperanden um eine Adresse nach rechts. Der Schiebevorgang beginnt bei der in d vorgegebenen Adresse und wird für die folgenden n Adressen durchgeführt.

Der Inhalt des höchstwertigen Operanden wird nach der Verschiebung auf 0 gesetzt.

Bei Timern oder Countern wird der Istwert (Zählwert) verschoben. Ein Verschieben des Sollwertes (Vorgabewert) ist nicht möglich.



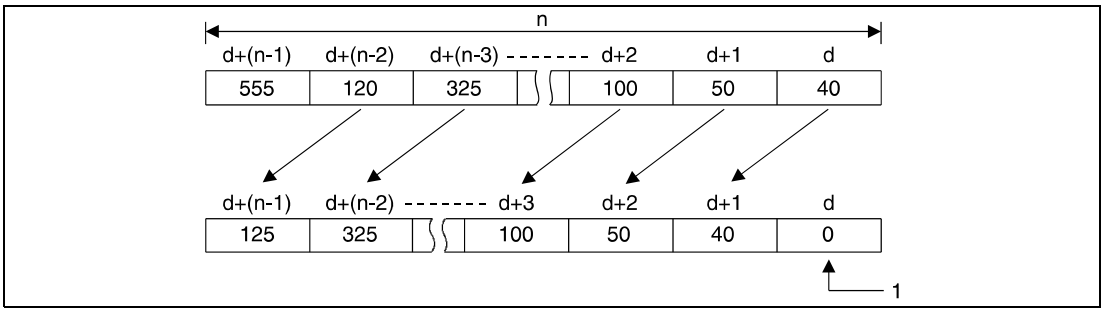
[1] Dieses Bit wird mit 0 beschrieben

DSFL Verschiebung nach links

Die DSFL-Anweisung verschiebt den Inhalt definierter Wortoperanden um eine Adresse nach links. Der Schiebevorgang beginnt bei der in d vorgegebenen Adresse und wird für die folgenden n Adressen durchgeführt.

Der Inhalt des niedrigstwertigen Operanden wird nach der Verschiebung auf 0 gesetzt.

Bei Timern oder Countern wird der Istwert (Zählwert) verschoben. Ein Verschieben des Sollwertes (Vorgabewert) ist nicht möglich.



[1] Dieses Bit wird mit 0 beschrieben

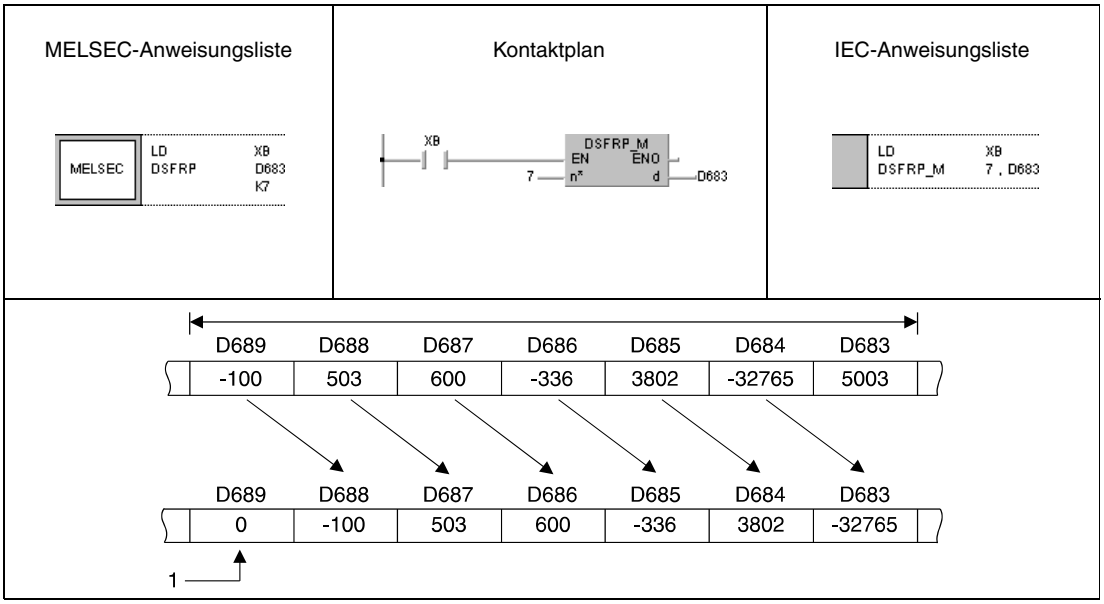
Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Der Wert in n ist negativ.
- Der Wert in n übersteigt die Bitzahl des in d genannten Operanden (Q-Serie/System Q = Fehlercode 4101).

Beispiel 1 DSFRP

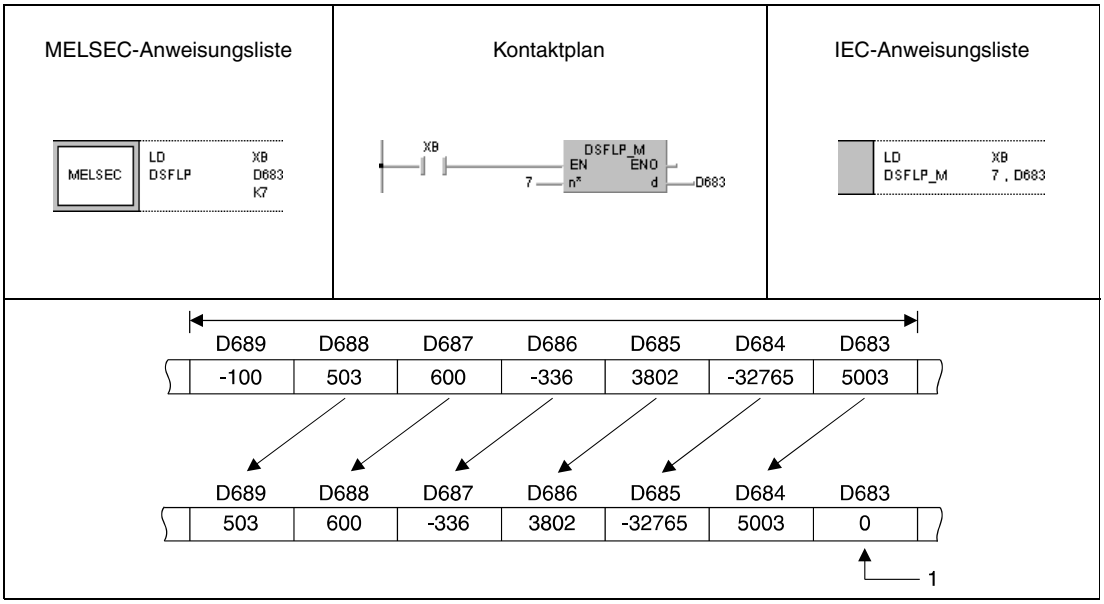
Das folgende Programm schiebt mit der positiven Flanke von XB die Daten der Datenregister D683 bis D689 um einen Adresse nach rechts. D683 nimmt den Wert von D684 an, D684 von D685 usw. Der Inhalt des letzten Datenregisters (D689) nimmt den Wert 0 an.



[1] Dieses Bit wird mit 0 beschrieben

Beispiel 2 DSFLP

Das folgende Programm schiebt mit der positiven Flanke von XB die Daten der Datenregister D683 bis D689 um einen Adresse nach links. D689 nimmt den Wert von D688 an, D688 von D687 usw. Der Inhalt des ersten Datenregisters (D683) nimmt den Wert 0 an.



[1] Dieses Bit wird mit 0 beschrieben

7.4 Bitverarbeitungsanweisungen

Die Bitverarbeitungsanweisungen ermöglichen die Zustandsänderungen (Setzen und Rücksetzen) einzelner Bits oder ganzer Bitbereiche. Auch Zustandsabfragen von Bits in Datenwörtern sind mit den Bitverarbeitungsanweisungen möglich.

Insgesamt stehen 10 Bitverarbeitungsanweisungen zur Verfügung.

Funktion	MELSEC-Anweisung im MELSEC-EDITOR	MELSEC-Anweisung im IEC-EDITOR
Setzen/Rücksetzen einzelner Bits	BSET	BSET_M
	BSETP	BSETP_M
	BRST	BRST_M
	BRSTP	BRSTP_M
Zustandsabfrage einzelner Bits in 16-/32-Bit-Datenwörtern	TEST	TEST_M
	TESTP	TESTP_M
	DTEST	DTEST_M
	DTESTP	DTESTP_M
Zurücksetzen von Bitbereichen	BKRST	BKRST_M
	BKRSTP	BKRSTP_M

7.4.1 BSET, BSETP, BRST, BRSTP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

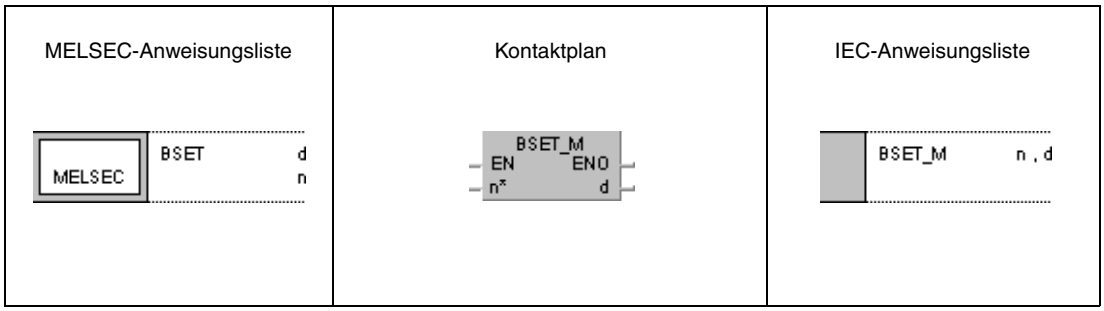
	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)								Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
d								●	●	●	●	●	●	●	●	●							7 ●[1]	●		●
n																	●	●								

[1] Die Anzahl der Schritte bei der Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
d	●	●	●	●	●	●	●	—	—	—	3
n	●	●	●	●	●	●	●	●	—	—	

GX IEC Developer



GX Developer

Variablen

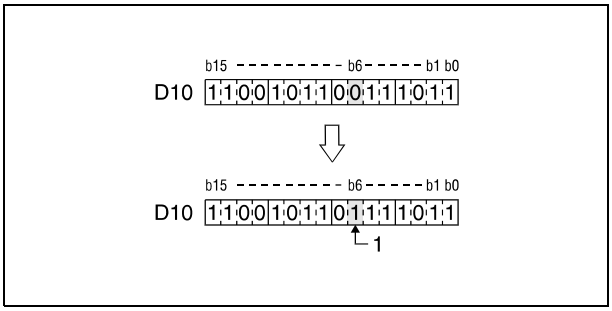
Operand	Befehlswert	Datentyp
d	Operand, in welchem Bits gesetzt oder zurückgesetzt werden.	BIN-16-Bit
n	Adresse des Bits, das gesetzt oder zurückgesetzt wird.	

Funktions-weise

Setzen/Rücksetzen einzelner Bits

BSET Setzen einzelner Bits in einem Wort-Operanden

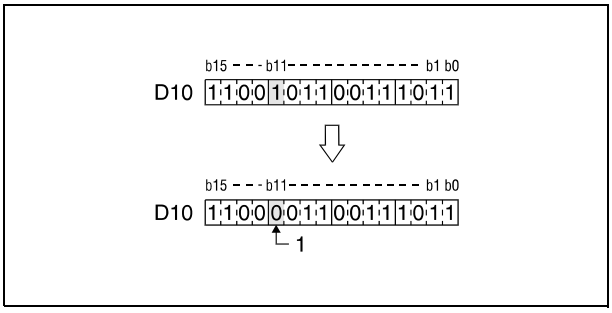
Die BSET-Anweisung setzt das n-te Bit eines Wort-Operanden auf 1. Für n kann ein Wert zwischen 0 und 15 (entsprechend b0 und b15) eingesetzt werden. Der Wortoperand wird in d festgelegt. Ist der Wert in n größer als 15, erfolgt die Ausführung der BSET-Anweisung innerhalb der ersten 4 Bits (b0 bis b3) des Wortoperanden. In der folgenden Darstellung ist für n=6 angegeben, und das Bit b6 wird gesetzt.



[1] Dieses Bit wird gesetzt

BRST Rücksetzen einzelner Bits in einem Wort-Operanden

Die BRST-Anweisung setzt das n-te Bit eines Wort-Operanden auf 0 zurück. Für n kann ein Wert zwischen 0 und 15 (entsprechend b0 und b15) eingesetzt werden. Der Wortoperand wird in d festgelegt. Ist der Wert in n größer als 15, erfolgt die Ausführung der BRST-Anweisung innerhalb der ersten 4 Bit (b0 bis b3) des Wortoperanden. In der folgenden Abbildung für n=11 angegeben und das Bit b11 wird zurückgesetzt.

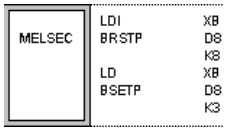


[1] Dieses Bit wird zurückgesetzt

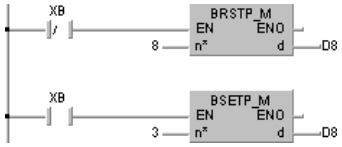
Beispiel BRSTP/BSETP

Das folgende Programm setzt mit positiver Flanke von XB das Bit b3 von D8 auf 1. Das Bit b8 wird mit positiver Flanke des Öffners XB auf 0 zurückgesetzt.

MELSEC-Anweisungsliste

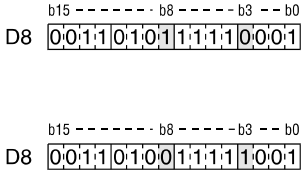


Kontaktplan



IEC-Anweisungsliste

```
LDN     XB
BRSTP_M 8 , D8
LD      XB
BSETP_M 3 , D8
```



HINWEIS *Das Setzen oder Rücksetzen einzelner Bits in Wort-Operanden kann gleichermaßen mit der SET- und RST-Anweisung erfolgen. In diesem Fall müssen die Bits in den Datenwörtern bei der Adressierung der Register angegeben werden. Das Bit b8 in Datenwort D5 wird beispielsweise mit D5.8 adressiert.*

7.4.2 TEST, TESTP, DTEST, DTESTP

CPU

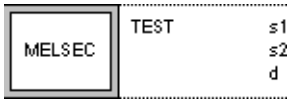
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Operanden MELSEC Q

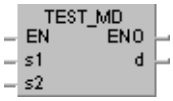
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	—	—	—	—	—	—	—	—	—	—	4
s2	—	—	—	—	—	—	—	—	—	—	
d	●	—	—	●	—	—	—	—	—	—	

GX IEC Developer

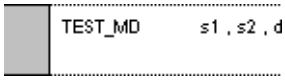
MELSEC-Anweisungsliste



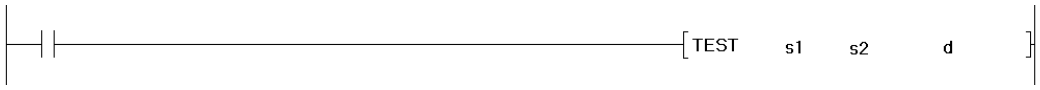
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

Operand	Befehlswert	Datentyp
s1	Operand, in dem sich die abzufragenden Bits befinden.	Wort
s2	Angabe, welches Bit abgefragt wird.	Wort
d	Bit-Operand, in dem das Abfrageergebnis gespeichert wird.	Bit

Funktionsweise

Zustandsabfrage einzelner Bits in 16-/32-Bit-Datenwörtern

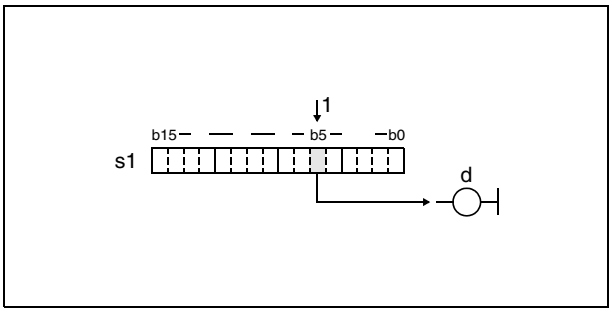
TEST Abfrageanweisung 16-Bit

Die TEST-Anweisung fragt den Zustand des Bits s2 im Wortoperanden s1 ab. Das Abfrageergebnis wird in dem in d angegebenen Bitoperanden gespeichert.

Der in d angegebene Bit-Operand wird gesetzt, wenn das abgefragte Bit den Zustand 1 hat, und zurückgesetzt, wenn das abgefragte Bit den Zustand 0 hat.

Das in s2 angegebene Bit kann ein beliebiges Bit zwischen b0 und b15 in einem 16-Bit Datenwort sein. Wird in s2 ein Wert angegeben, der größer als 15 ist, wird das Bit abgefragt, das als Rest übrigbleibt, wenn der Inhalt von s2 durch 16 geteilt wird. Wenn z.B. der Inhalt von s2=18 ist, wird b2 abgefragt (Bei der Division von 18 durch 16 bleibt 2 als Rest).

In der folgenden Abbildung ist für s2=5 angegeben, der Zustand von Bit b5 in s1 wird abgefragt.



[1] Abgefragtes Bit

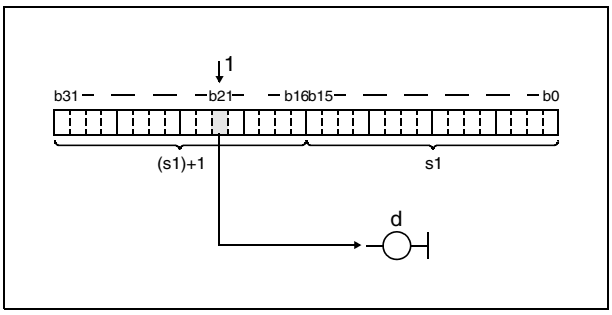
DTEST Abfrageanweisung 32-Bit

Die DTEST-Anweisung fragt den Zustand des Bits s2 im Wortoperanden s1 und (s1)+1 ab. Das Abfrageergebnis wird in dem in d angegebenen Bitoperanden gespeichert.

Der in d angegebene Bit-Operand ist gesetzt, wenn das abgefragte Bit den Zustand 1 hat, und zurückgesetzt, wenn das abgefragte Bit den Zustand 0 hat.

Das in s2 angegebene Bit kann ein beliebiges Bit zwischen b0 und b31 in einem 32-Bit Datenwort sein. Wird in s2 ein Wert angegeben, der größer als 31 ist, wird das Bit abgefragt, das als Rest übrigbleibt, wenn der Inhalt von s2 durch 32 geteilt wird. Wenn z.B. der Inhalt von s2=34 ist, wird b2 abgefragt (Bei der Division von 34 durch 32 bleibt 2 als Rest).

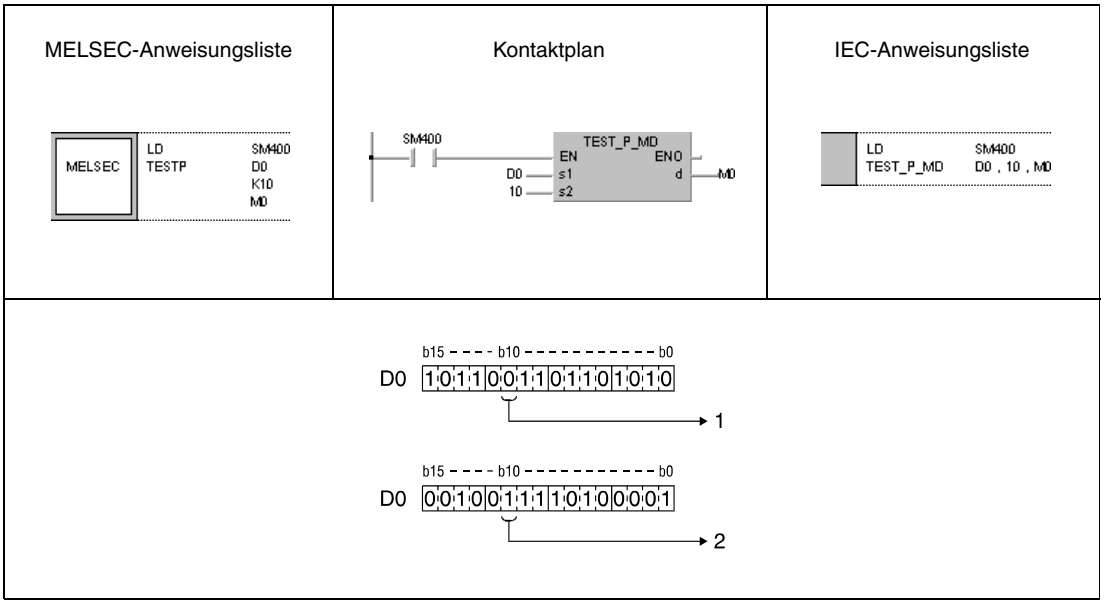
In der folgenden Darstellung ist für s2=21 angegeben, und der Zustand von Bit b21 in s1 wird abgefragt.



[1] Abgefragtes Bit

Beispiel 1 TESTP

Das folgende Programm setzt mit positiver Flanke von SM400 den Merker M0 in Abhängigkeit des Abfrageergebnisses des Bits b10 in dem 16-Bit-Datenwort in D0 oder setzt den Merker M0 zurück.

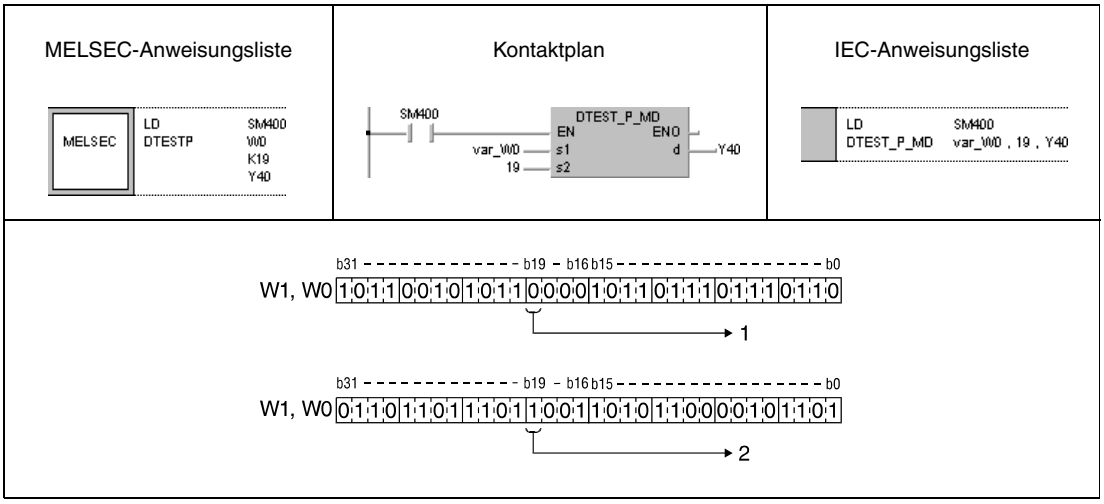


1= Zurücksetzen

2= Setzen

Beispiel 2 DTESTP

Das folgende Programm setzt mit positiver Flanke von SM400 den Ausgang Y40 in Abhängigkeit des Abfrageergebnisses des Bits b19 in dem 32-Bit-Datenwort in W0 und W1 oder setzt den Ausgang Y40 zurück.

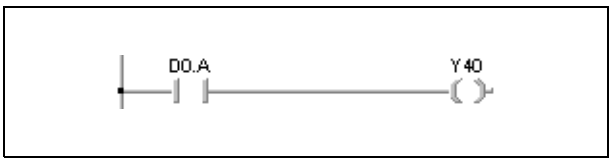


1= Zurücksetzen

2= Setzen

HINWEISE *Das Programmbeispiel 2 ist ohne Variablendefinition im Header der Programm-Organisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

Anstelle der TEST-Anweisung kann auch das abzufragende Bit als Eingangskontakt definiert werden (siehe Abbildung).



7.4.3 BKRST, BKRSTP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Operanden MELSEC Q

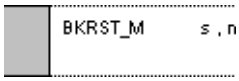
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	●	●	●	—	—	—	—	—	—	SM0	3
n	●	●	●	●	●	●	●	●	—		

GX IEC Developer

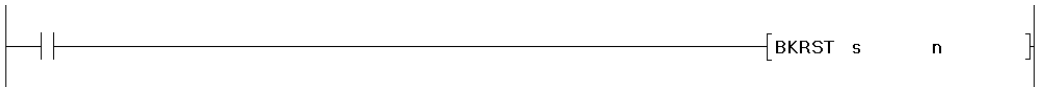
MELSEC-Anweisungsliste

Kontaktplan

IEC-Anweisungsliste



GX Developer



Variablen

Operand	Befehlswert	Datentyp
s	Erste Adresse der Operanden, die zurückgesetzt werden.	Bit
n	Anzahl der zurückzusetzenden Operanden.	BIN-16-Bit

Funktionsweise

Zurücksetzen von Bitbereichen

BKRST Rücksetzanweisung

Die BKRST-Anweisung setzt n Bits des in s angegebenen Operanden zurück.

Bei Fehlermerkern (F) wird die in n angegebene Anzahl der in s gespeicherten Fehlermerker zurückgesetzt und der Inhalt entsprechend der zurückgesetzten Fehlermerker der Register SD64 bis SD79 gelöscht. Die verbleibenden Daten werden vorgeschoben. Ferner wird die Anzahl der Fehlermerkereinträge der Register SD64 bis SD79 im Register SD63 gespeichert.

Bei Timern (T) und Countern (C) werden nach Ausführung der Anweisung die Einstellwerte von n Timern oder Countern auf Null gesetzt und die Spulenkontakte zurückgesetzt.

Bei allen anderen Bit-Operanden wird die n angegebene Anzahl von Spulen oder Kontakten in dem in s angegebenen Operanden zurückgesetzt.

Wenn der entsprechende Operand bereits zurückgesetzt ist, verändert sich sein Zustand nach Ausführung der Anweisung nicht.

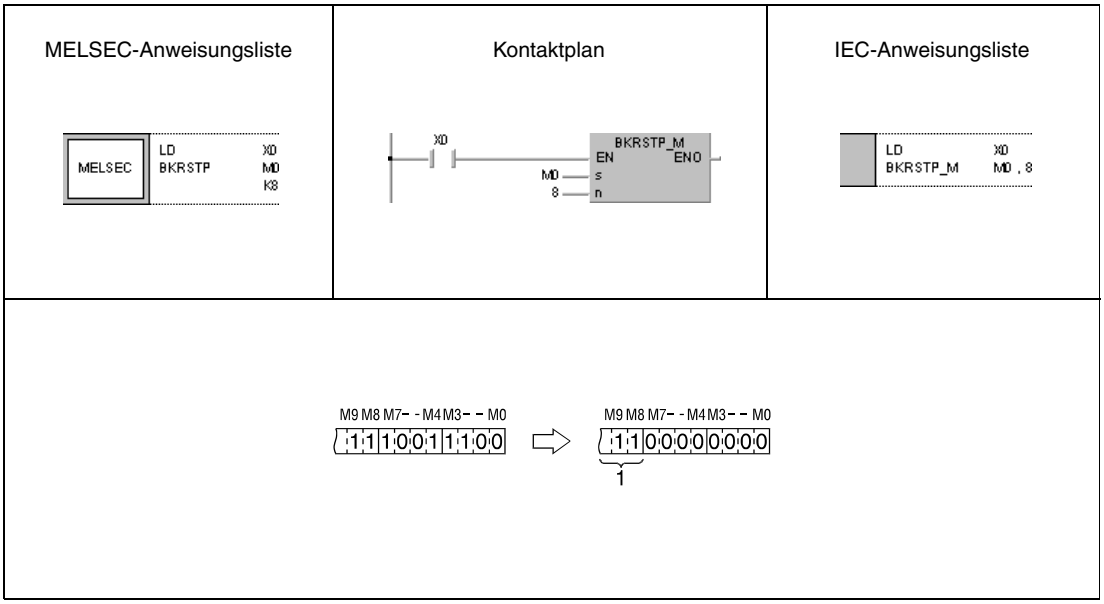
Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Der Wert in n übersteigt die Bitzahl der in s genannten Operanden (Fehlercode 4101).

Beispiel 1 BKRSTP

Das folgende Programm setzt mit positiver Flanke von X0 die Merker M0 bis M7 zurück.



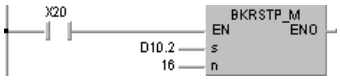
[1] Diese Bits verändern ihren Zustand nicht

Beispiel 2 BKRSTP

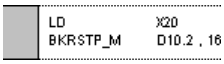
Das folgende Programm setzt mit positiver Flanke von X20 die Bits beginnend mit dem Bit b2 in D10 bis zum Bit b1 in D11 zurück.

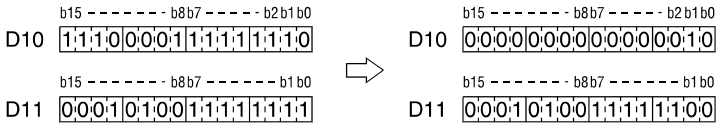
MELSEC-Anweisungsliste

Kontaktplan



IEC-Anweisungsliste





7.5 Datenverarbeitungsanweisungen

Die Datenverarbeitungsanweisungen bieten die Möglichkeit, Daten in definierten Bereichen zu suchen, die Anzahl der gesetzten Bits zu ermitteln, Daten zu codieren und decodieren (auch für 7-Segment-Anzeige), Datenwerte aufzutrennen bzw. zusammenzuführen, Maximal- und Minimalwerte zu suchen, Daten zu sortieren oder die Summe von 16-/32-Bit-Binärdatenblöcken zu bilden.

Insgesamt stehen 41 verschiedene Anweisungen zur Datenverarbeitung zur Verfügung. Die folgende Tabelle enthält eine Übersicht sämtlicher Anweisungen.

Funktion	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor
Suchen von 16-/32-Bit-Daten	SER	SER_M
	SERP	SERP_M
	DSER	DSER_M
	DSERP	DSERP_M
Datenbit-Kontrolle (16-/32-Bit)	SUM	SUM_M
	SUMP	SUMP_M
	DSUM	DSUM_M
	DSUMP	DSUMP_M
Daten codieren, decodieren	DECO	DECO_M
	DECOP	DECOP_M
	ENCO	ENCO_M
	ENCOP	ENCOP_M
7-Segment-Decodierung	SEG	SEG_M
16-Bit-Datenworte auftrennen, zusammenführen (4-Bit-Einheiten)	DIS	DIS_M
	DISP	DISP_M
	UNI	UNI_M
	UNIP	UNIP_M
16-Bit-Datenwerte auftrennen, zusammenführen (Variable Größe der Bit-Einheiten)	NDIS	NDIS_M
	NDISP	NDISP_M
	NUNI	NUNI_M
	NUNIP	NUNIP_M
16-Bit-Datenwerte auftrennen, zusammenführen (Byte-Einheiten)	WTOB	WTOB_MD
		WTOB_K_MD
	WTOBP	WTOB_P_MD
		WTOB_K_P_MD
	BTOW	BTOW_MD
		BTOW_K_MD
	BTOWP	BTOW_P_MD
		BTOW_K_P_MD
Suchen von Maximalwerten in 16-/32-Bit-Daten	MAX	MAX_M
	MAXP	MAXP_M
	DMAX	DMAX_M
	DMAXP	DMAXP_M

Funktion	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor
Suchen von Minimalwerten in 16-/32-Bit-Daten	MIN	MIN_M
	MINP	MINP_M
	DMIN	DMIN_M
	DMINP	DMINP_M
Sortieren von 16-/32-Bit-Daten	SORT	SORT_M
	SORTP	SORTP_M
	DSORT	DSORT_M
	DSORTP	DSORTP_M
Summenbildung von 16-/32-Bit-Binärdatenblöcken	WSUM	WSUM_M
	WSUMP	WSUMP_M
	DWSUM	DWSUM_M
	DWSUMP	DWSUMP_M

7.5.1 SER, SERP, DSER , DSERP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

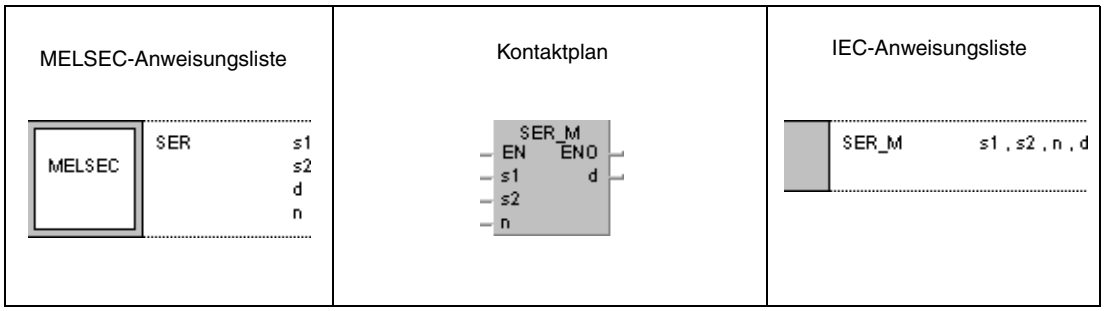
	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
s1								●	●	●	●	●	●	●	●	●	●	●					9 ●[1]	●		●
s2								●	●	●	●	●														
n																	●	●								

[1] Die Anzahl der Schritte bei der Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

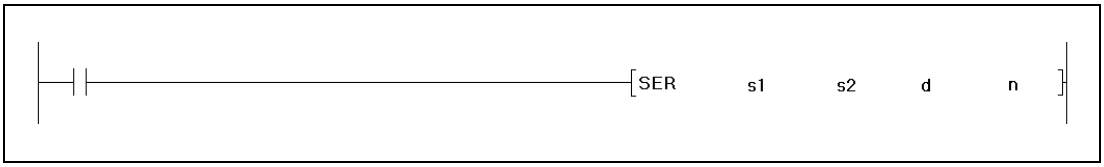
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort				U		
s1	●	●	●	●	●	●	●	●	—	SM0	5
s2	—	●	●	—	—	—	—	—	—		
d	—	●	●	—	●	●	●	—	—		
n	●	●	●	●	●	●	●	●	—		

GX IEC Developer



GX Developer



HINWEIS *Bei der A-Serie wird das Ergebnis des Suchvorgangs immer in den Registern A0 und A1 gespeichert. Aus diesem Grund fehlt bei der Programmierung dieses Befehls in der A-Serie der Operand d.*

Variablen

Operand	Befehlswert	Datentyp	
		MELSEC	IEC
s1	Datenwert, der gesucht wird, oder erste Adresse des Operanden, in dem dieser Wert gespeichert ist.	Wort	ANY16
s2	Daten, die nach dem Wert durchsucht werden, oder erste Adresse des Operanden, in dem diese Daten gespeichert sind.		ANY16/ANY32
d	Erste Adresse des Operanden, in dem das Suchergebnis gespeichert wird. Diese Operanden sind bei der A-Serie immer A0 und A1.		Array [1..2] of ANY16/ANY32
n	Anzahl der zu durchsuchenden Adressen.		ANY16

Funktionsweise

Suchen von Daten

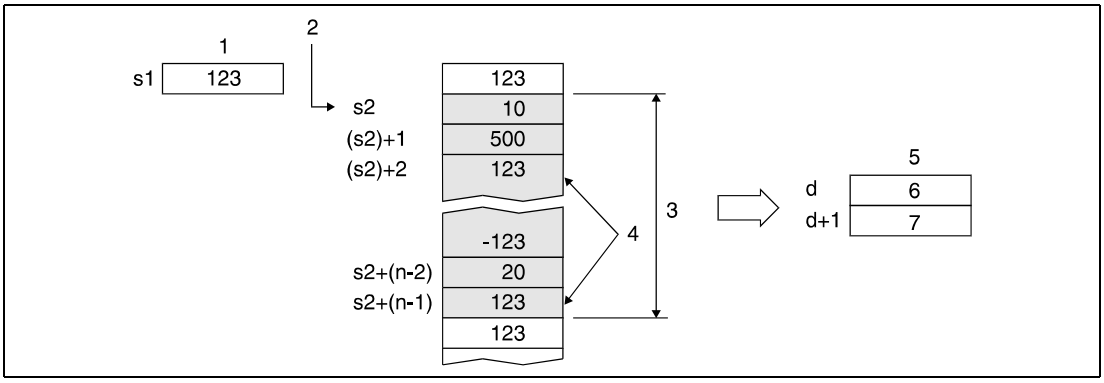
SER (A- und Q-Serie/System Q) / SERP (Q-Serie/System Q)Suchen von 16-Bit-Daten

Die SER-Anweisung ermöglicht das Suchen bestimmter Daten innerhalb eines definierten Bereiches. Der Suchvorgang beginnt ab der in s2 angegebenen Startadresse. Der gesuchte Datenwert wird in s1 vorgegeben. s1 definiert somit den Suchbegriff. Die Länge des Suchbereiches, d.h. die Anzahl der Adressen wird in n festgelegt.

Bei der Q-Serie bzw. dem System Q wird das Suchergebnis in (Array_d[1]) und (Array_d[2]) gespeichert.

Nach Ausführung des Suchvorgangs wird die Position der ersten Adresse, an der sich der gesuchte Datenwert befindet, in dem in (Array_d[1]) abgespeichert. (Array_d[2]) enthält die Anzahl der gefundenen Daten, die mit dem Suchbegriff identisch sind.

Bei der A-Serie wird die Position der ersten Adresse des gefundenen Datenwertes in dem Register A0 gespeichert. Die Anzahl der gefundenen Daten wird in dem Register A1 abgelegt.



[1] Suchbegriff

[2] Startadresse der Suche

[3] Suchbereich

[4] Übereinstimmende Werte

[5] Suchergebnis

[6] Adresse des zuerst gefundenen Wertes

[7] Anzahl der gefundenen Werte

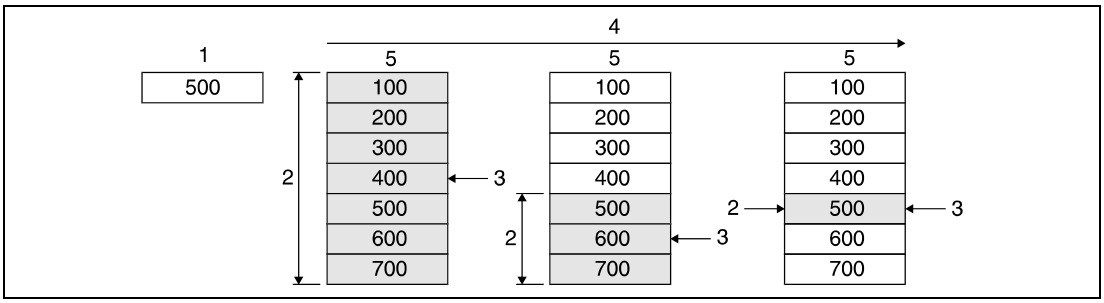
Ist der Wert in n negativ oder gleich 0, wird die Datensuche nicht ausgeführt. Werden bei der Suchoperation keine übereinstimmenden Daten gefunden, wird in (Array_[d1]) und (Array_d[2]) bzw. in den Registern A0 und A1 „0“ eingetragen.

HINWEIS *Q-Serie und System Q*

Wenn die zu durchsuchenden Daten in aufsteigender Reihenfolge abgelegt sind, kann die Verarbeitungszeit durch den Einsatz der Binärsuche bei eingeschaltetem Diagnosemerker SM702 verkürzt werden.

SM702 EIN:

Der zu durchsuchende Datenbereich wird halbiert und anhand der Größe des Suchbegriffs wird entschieden, in welcher Hälfte sich der Begriff befindet. Diese Häfte des Suchbereichs wird erneut halbiert, und es wird wieder eine Hälfte zur Weitersuche ausgewählt. Dieser Vorgang wird fortgesetzt, bis der Suchbegriff gefunden wird.



[1] Suchbegriff
[2] Suchbereich
[3] Vergleich mit Suchbegriff
[4] Verarbeitungsreihenfolge
[5] Zu durchsuchende Daten

SM702 AUS:

Die Datensuche, bei der der Suchbegriff mit jedem Datenwert verglichen wird, beginnt am Anfang des zu durchsuchenden Datenbereichs.

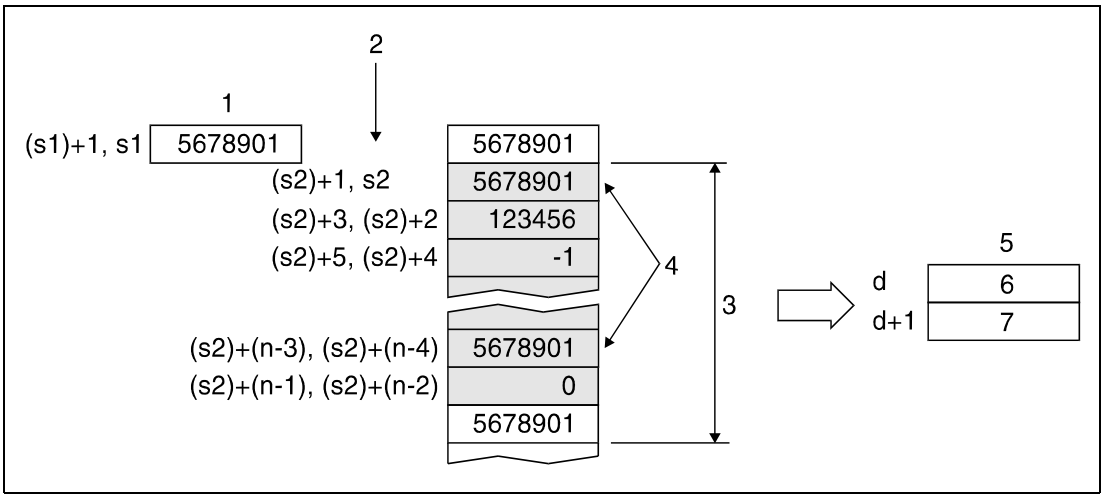
Ist der zu durchsuchende Datenbereich nicht der Größe nach sortiert, kann bei der Suche mit gesetztem Diagnosemerker SM702 kein korrektes Suchergebnis erzielt werden.

DSER/ DSERP (Q-Serie und System Q) Suchen von 32-Bit-Daten

Die DSER-Anweisung ermöglicht das Suchen bestimmter 32-Bit-Daten innerhalb eines definierten Bereiches. Der Suchvorgang beginnt ab der in s2 angegebenen Startadresse (2 x n Adressen). Der gesuchte Datenwert wird in s1 und (s1)+1 vorgegeben. s1 und (s1)+1 definieren somit den Suchbegriff. Die Länge des Suchbereiches, d.h. die Anzahl der Adressen wird in n festgelegt.

Das Suchergebnis in d und d1 wird als Array [1..2] of ANY16 gespeichert.

Nach Ausführung des Suchvorgangs wird die Position der ersten Adresse, an der sich der gesuchte Datenwert befindet, in d (Array_d[1])abgespeichert. (Array_d[2]) enthält die Anzahl der gefundenen Daten, die mit dem Suchbegriff identisch sind.



[1] Suchbegriff

[2] Startadresse der Suche

[3] Suchbereich

[4] Übereinstimmende Werte

[5] Suchergebnis

[6] Adresse des zuerst gefundenen Ergebnisses

[7] Anzahl der gefundenen Werte

Ist der Wert in n negativ oder gleich 0, wird die Datensuche nicht ausgeführt.

Werden bei der Suchoperation keine übereinstimmenden Daten gefunden, lautet der in (Array_d[1]) und (Array_d[2]) gespeicherte Inhalt 0.

Fehlerquellen

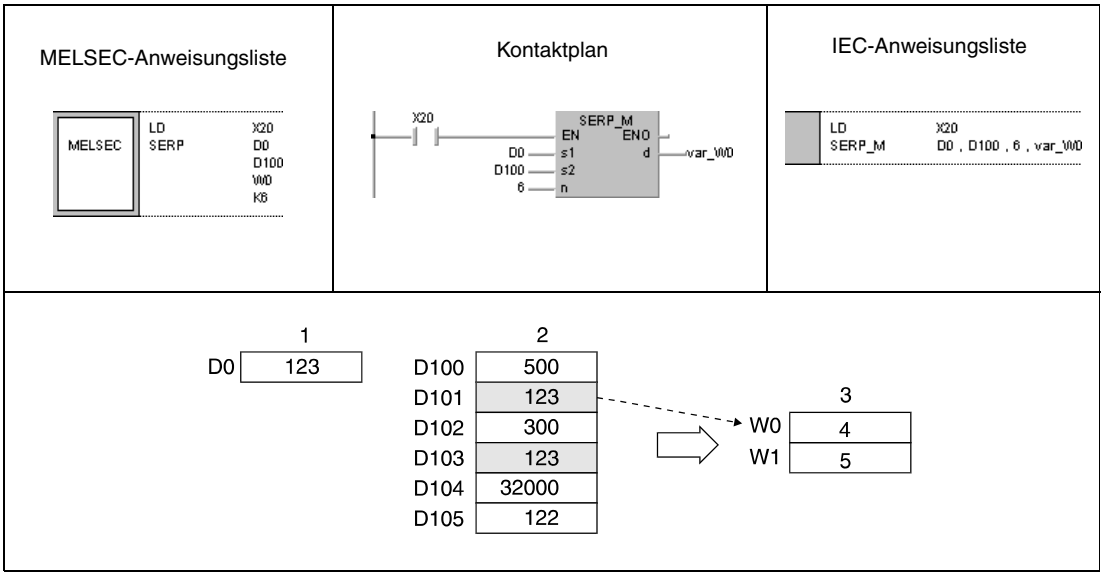
In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Der in n angegebene Adressenbereich, beginnend mit s2, liegt außerhalb des für die Speicherung vorgesehenen Bereichs des Operanden (Fehlercode 4101).

Informationen zur Verwendung der indizierten Adressierung enthält Kap. 3.6.

Beispiel 1 SERP (Q-Serie und System Q)

Das Beispiel zeigt ein Programm, das mit positiver Flanke von X20 die Daten aus D100 bis D105 mit dem Datenwert in D0 vergleicht. Die erste übereinstimmende Position wird in W0 und die Anzahl der Positionen in W1 gespeichert.



[1] Suchbegriff

[2] Zu durchsuchende Daten

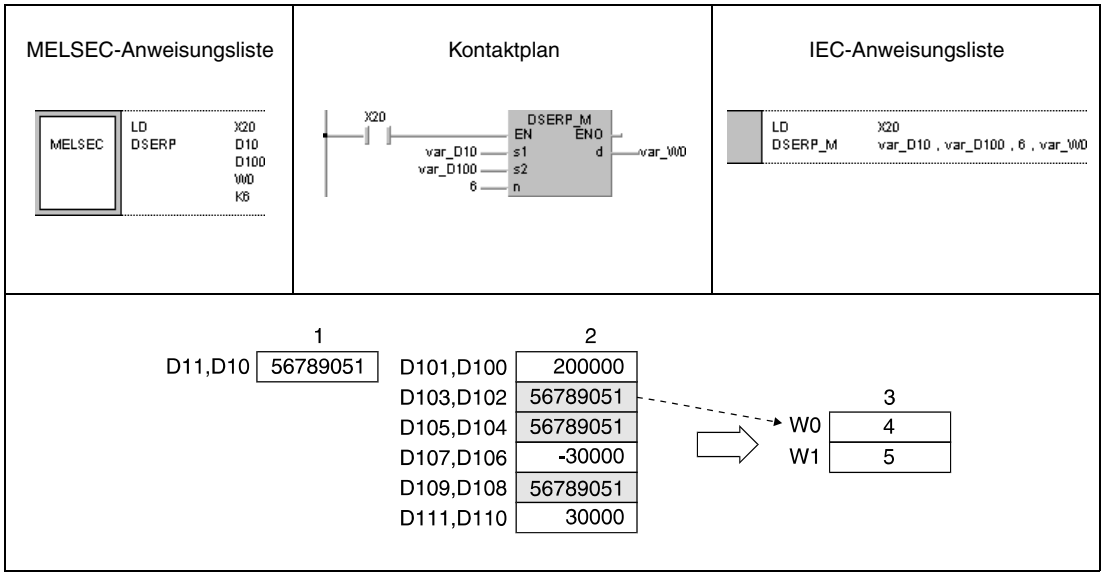
[3] Suchergebnis

[4] Adresse des zuerst gefundenen Ergebnisses

[5] Anzahl der gefundenen Werte

Beispiel 2 DSERP (Q-Serie und System Q)

Das Beispiel zeigt ein Programm, das mit positiver Flanke von X20 die Daten aus D100 bis D111 mit dem Datenwert in D11 und D10 vergleicht. Die erste übereinstimmende Position wird in W0 und die Anzahl der Positionen in W1 gespeichert.



[1] Suchbegriff

[2] Zu durchsuchende Daten

[3] Suchergebnis

[4] Adresse des zuerst gefundenen Ergebnisses

[5] Anzahl der gefundenen Werte

HINWEIS *Diese Programmbeispiele sind ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.5.2 SUM, SUMP, DSUM, DSUMP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)								Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
SUM																									
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●						K1 ↓ K4	3 ●[1]	●		●
DSUM																									
s	●	●	●	●	●	●	●	●	●	●	●	●	●		●							K1 ↓ K4	3 ●[1]	●		●

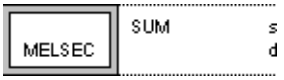
[1] Die Anzahl der Schritte bei der Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

Operanden MELSEC Q

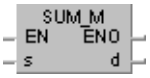
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sonder-module U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	●	●	●	●	●	●	●	●	—	—	3
d	●	●	●	●	●	●	●	—	—	—	

GX IEC Developer

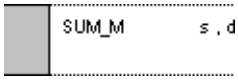
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



GX Developer

HINWEIS *Bei der A-Serie wird die Anzahl der gesetzten Bits immer in dem Register A0 gespeichert. Aus diesem Grund fehlt bei der Programmierung dieses Befehls in der A-Serie der Operand d.*

Variablen

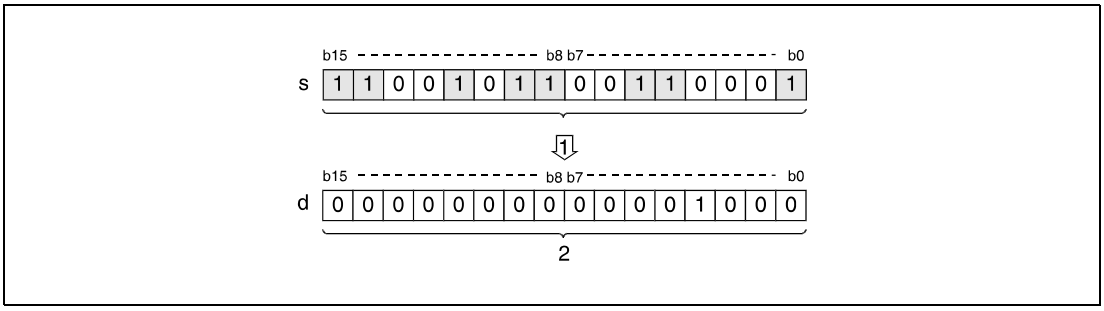
Operand	Befehlswert	Datentyp
s	Erste Adresse des Operanden, in dem die gesetzten Bits gezählt werden.	BIN-16-/32-Bit
d	Erste Adresse des Operanden, in dem die Anzahl der gesetzten Bits gespeichert wird. Dieser Operand ist bei der A-Serie immer A0.	

Funktionsweise

Datenbit-Kontrolle

SUM 16 Bit

Mit Hilfe der SUM-Anweisung kann die Anzahl der Bits, die in einem 16-Bit-Datenwort 1 gesetzt sind, festgestellt werden. Der Adressenbereich, der kontrolliert werden soll, wird in s vorgegeben. Die Summe der gesetzten Bits wird nach Ausführung der Anweisung in d (A0) abgelegt.

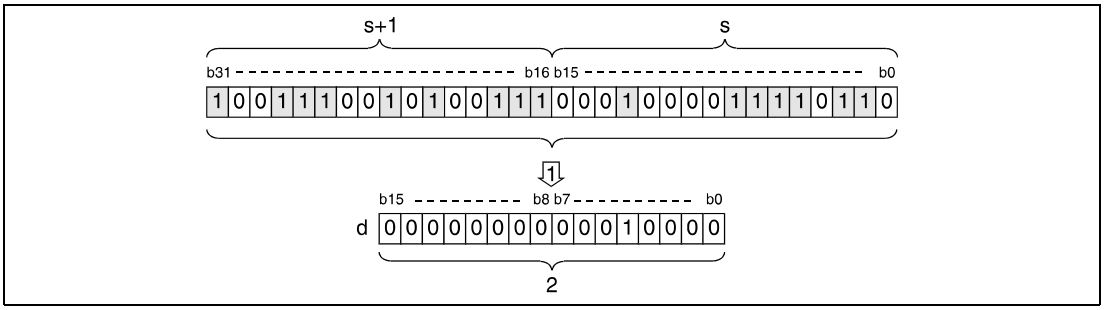


[1] Zählen der gesetzten Bits

[2] Binärcodierte Anzahl der gesetzten Bits

DSUM 32 Bit

Mit Hilfe der DSUM-Anweisung kann festgestellt werden, wieviele Bits in einem 32-Bit-Datenwort gesetzt sind. Der zu prüfende Adressenbereich wird in s vorgegeben. Die Summe der Bits mit dem Wert 1 wird nach Ausführung der Anweisung in d (A0) abgelegt.

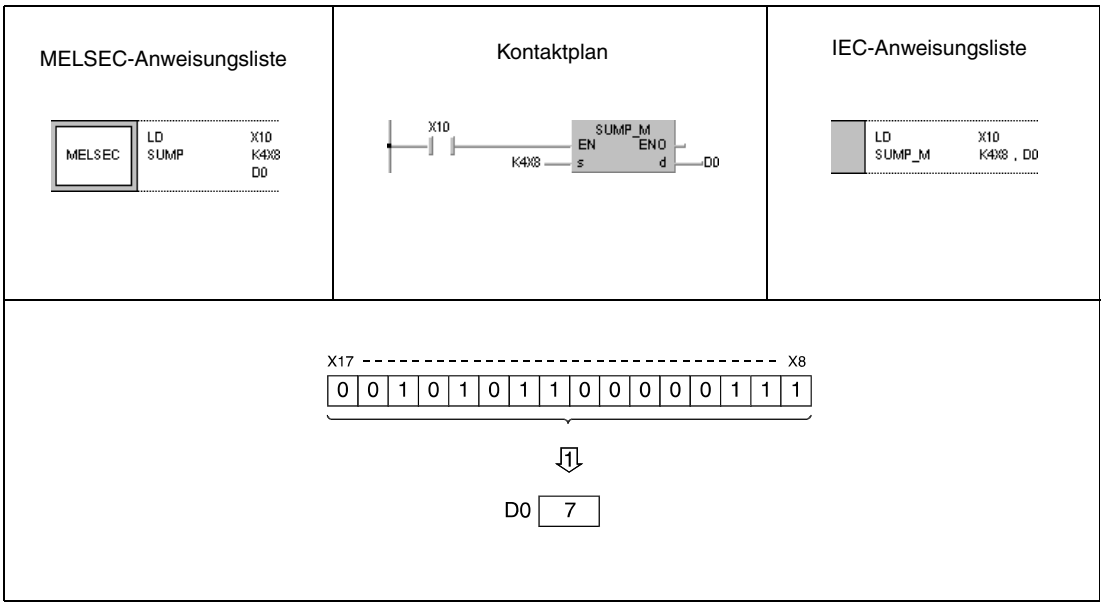


[1] Zählen der gesetzten Bits

[2] Binärcodierte Anzahl der gesetzten Bits

Beispiel 1 SUMP (Q-Serie und System Q)

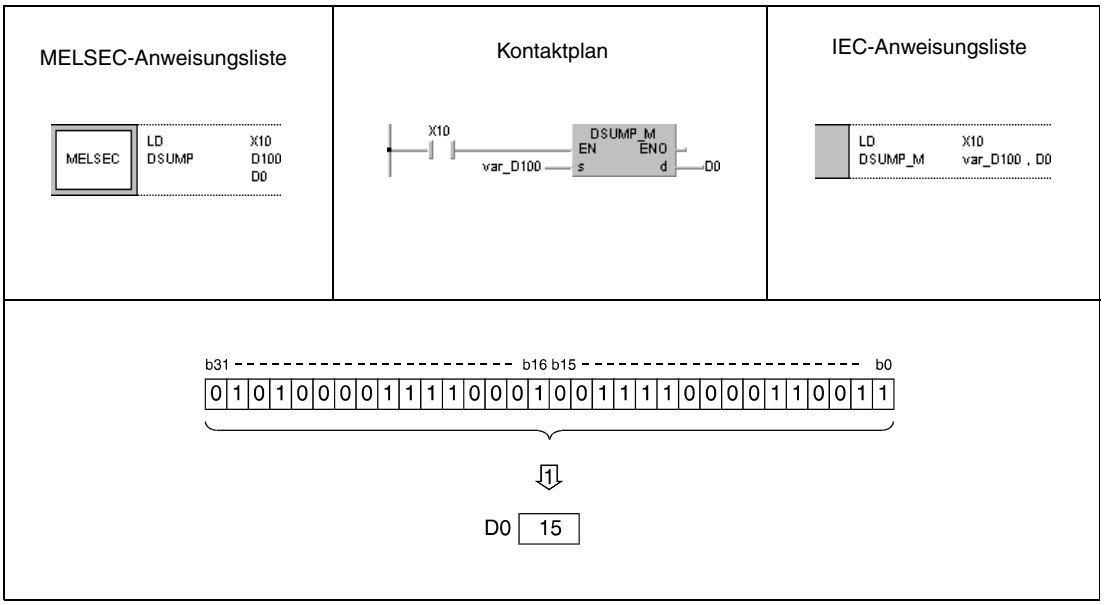
Im nachfolgenden Programm wird mit positiver Flanke von X10 die Summe der eingeschalteten (auf 1 gesetzten) Eingänge zwischen X8 und X17 ermittelt und in D0 abgelegt.



[1] Speicherung der gesetzten Bits in D0

Beispiel 2 DSUMP (Q-Serie und System Q)

Im nachfolgenden Programm wird mit positiver Flanke von X10 die Summe der gesetzten Bits in D100 und D101 ermittelt und in D0 abgelegt.



[1] Speicherung der gesetzten Bits in D0

HINWEIS *Das Programmbeispiel 2 ist ohne Variablendefinition im Header der Proggrammorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.5.3 DECO, DECOP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

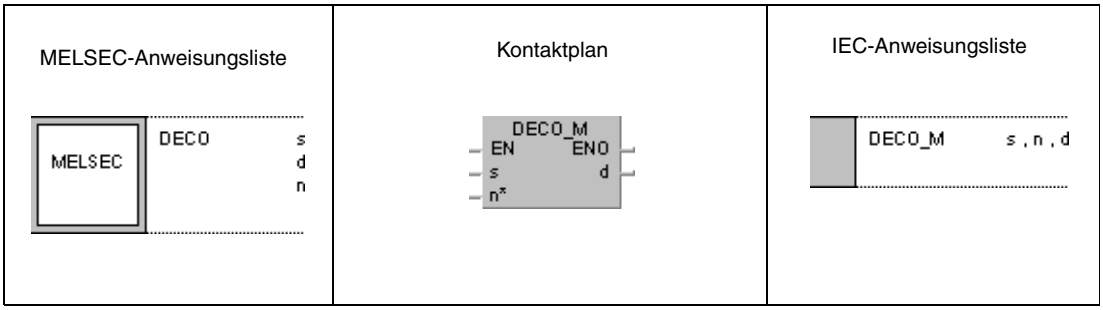
	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●								
d		●	●	●	●	●	●	●	●	●	●	●											9 ●[1]	●		●
n																	●	●								

[1] Die Anzahl der Schritte bei der Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	●	●	●	●	●	●	●	●	—	SM0	4
d	●	●	●	—	—	—	—	—	—		
n	●	●	●	●	●	●	●	●	—		

GX IEC Developer



GX Developer

Variablen

Operand	Befehlswert	Datentyp
s	Kodierte Daten oder Operand, in dem diese Daten gespeichert sind.	BIN-16-Bit
d	Erste Adresse des Operanden, in dem der decodierte Wert gespeichert wird.	Adresse
n	Anzahl der Bits, die die kodierten Daten enthalten.	BIN-16-Bit

Funktions-weise

Decodieren von Daten von 8 nach 256 Bit

DECO Daten decodieren

Die DECO-Anweisung decodiert die Daten des in s angegebenen Operanden. Der darin enthaltene binärcodierte Wert wird in eine Dezimalzahl decodiert. Diese Dezimalzahl (max. 256) gibt das Bit x (bx), entsprechend dem 2^x-ten Bit eines in d angegebenen Operanden an, das gesetzt wird. Mit n wird die Anzahl der Adressen in s definiert, die die kodierten Daten enthalten.

In n muss ein Wert zwischen 1 und 8 eingesetzt werden.

Ist n = 0, erfolgt keine Verarbeitung der Anweisung, und der vorgegebene Adressenbereich bleibt unverändert.

Ein Bit-Operand wird als einzelnes Bit und ein Wortoperand als 16-Bit-Datenwert verarbeitet.

Fehler-quellen

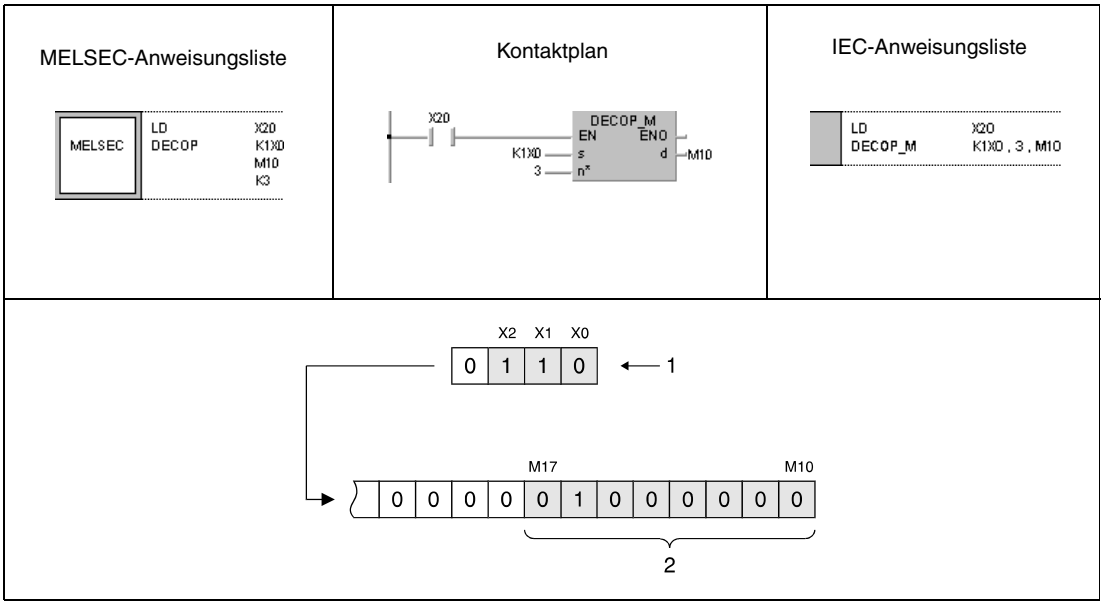
In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Der Wert in n liegt nicht zwischen 1 und 8 (Q-Serie/System Q = Fehlercode 4100).
- Das Bit x von d liegt außerhalb des für die Speicherung vorgesehenen Bereichs des Operanden (Q-Serie/System Q = Fehlercode 4101).

Beispiel

DECOP

Das folgende Programm dekodiert mit positiver Flanke von X20 die Daten aus X0 bis X2 und speichert das Ergebnis in M10 bis M17 ab. Da der binärcodierte Wert 6 in X0 bis X2 enthalten ist , wird das Bit b6 (M16) in M10 bis M17 gesetzt.



[1] Binärcodierter Wert 6

[2] Wenn der binärcodierte Wert mit 4 Bits angegeben ist, werden für die Darstellung 8 Bits benötigt

HINWEIS

Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.

7.5.4 ENCO, ENCOP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)								Konstante		Pointer		Ebene				
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●										
d								●	●	●	●	●	●	●	●	●							9 ●[1]	●		●
n																	●	●								

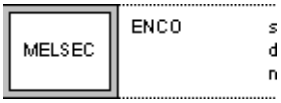
[1] Die Anzahl der Schritte bei der Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

Operanden MELSEC Q

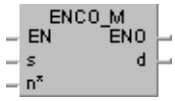
	Operanden								Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere	
	Bit	Wort		Bit	Wort					
s	●	●	●	—	—	—	—	●	—	SM0	4
d	●	●	●	●	●	●	●	—	—		
n	●	●	●	●	●	●	●	●	—		

GX IEC Developer

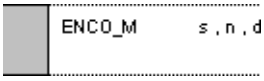
MELSEC-Anweisungsliste



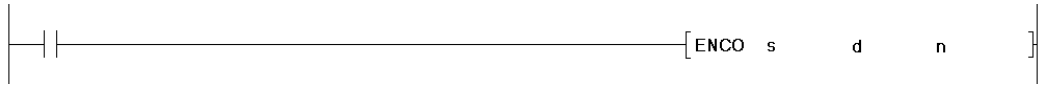
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

Operand	Befehlswert	Datentyp
s	Dekodierte Daten oder Operand, in dem diese Daten gespeichert sind.	BIN-16-Bit
d	Operand, in dem die kodierten Daten gespeichert werden.	
n	Anzahl der Bits, in denen der kodierte Wert gespeichert wird.	

Funktionsweise

Codierung von Daten von 256 nach 8 Bit

ENCO Daten codieren

Die ENCO-Anweisung codiert die Daten eines bis zu 256 Bit großen Datensatzes in eine binäre 8-Bit-Datenfolge. Die Startadresse der Operanden, deren Daten codiert werden sollen, wird in s festgelegt. Das in s gesetzte Bit x gibt mit x den Dezimalwert an, der binärcodiert in d gespeichert wird. Die Anzahl der Bits des in d angegebenen Operanden, in dem das codierte Ergebnis abgespeichert wird, ist in n angegeben.

In n muss ein Wert zwischen 0 und 8 eingesetzt werden.

Ist n = 0, erfolgt keine Verarbeitung der Anweisung, und der vorgegebene Adressenbereich bleibt unverändert.

Ein Bit-Operand wird als einzelnes Bit und ein Wortoperand als 16-Bit-Datenwert verarbeitet.

Haben mehr als ein Bit den Wert 1, beginnt die Verarbeitung mit dem höchsten Bit.

Fehlerquellen

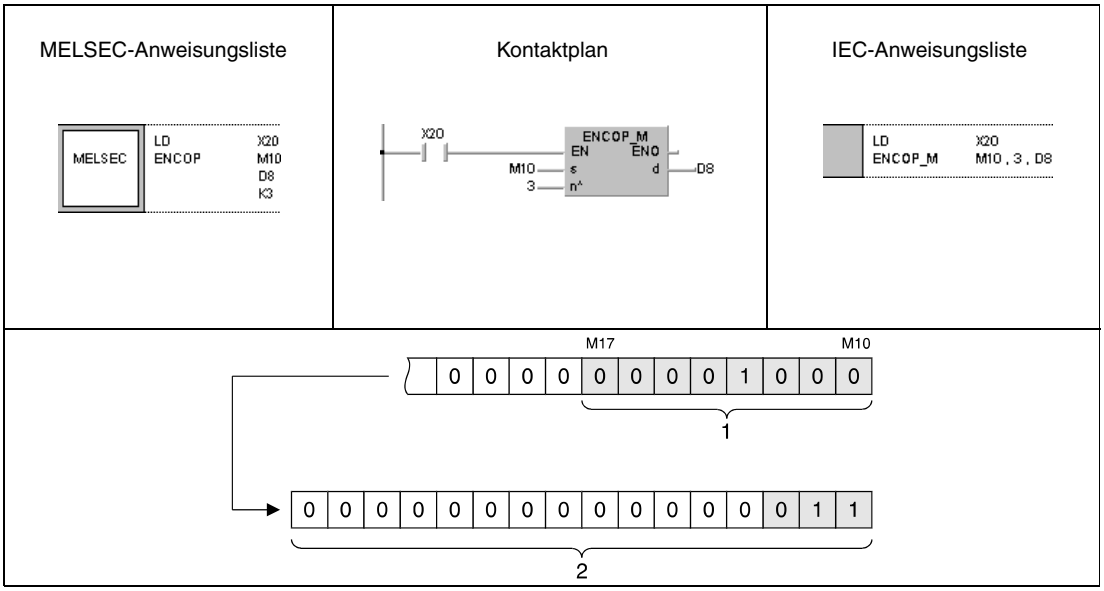
In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Der Wert in n liegt nicht zwischen 0 und 8 (Q-Serie/System Q = Fehlercode 4100).
- Sämtliche Bits in s sind bis Bit x bei Ausführung der ENCO-Anweisung gleich 0.
- Der Wert x des gesetzten Bits x in s liegt außerhalb des Bereichs, der mit 0 bis 8 Bit binär dargestellt werden kann (Q-Serie/System Q = Fehlercode 4101).
- Sämtliche Bits in s sind bis Bit x identisch mit d (Q-Serie/System Q = Fehlercode 4100).

Beispiel

ENCOP

Im nachfolgenden Programm werden mit positiver Flanke von X20 die Daten aus M10 bis M17 gelesen, codiert und als Binärwert in D8 gespeichert.



[1] Wenn das gesetzte Bit mit 4 Bits binärcodiert wird, kann ein Bereich von 8 Bits dargestellt werden

[2] Binärcodierte 3 für gesetztes Bit 3 (M13)

HINWEIS *Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.5.5 SEG, SEGP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●[1]	●[1]	●[1]	●[1]	●	●

[1] Die Funktion der SEG-Anweisung als 7-Segment-Decodierung ist nur dann möglich, wenn der Sondermerker M9052 nicht gesetzt ist. Ist der Sondermerker M9052 gesetzt, hat die SEG-Anweisung die Funktion der Teilaktualisierung.

Operanden MELSEC A

	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				K1	7	●		
d		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●						●[1]	●[2]			

[1] Die Blocklänge kann bei A3H, A3M oder AnN CPU zwischen K1 und K4 gewählt werden. Bei allen anderen CPUs wird die vorgegebene Blocklänge übergangen und automatisch K2 (entsprechend 8 Bit) verarbeitet.

[2] Die Anzahl der Schritte bei der Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

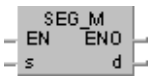
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	●	●	●	●	●	●	●	●	—	—	3
d	●	●	●	●	●	●	●	—	—	—	

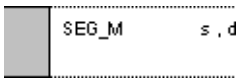
GX IEC Developer

MELSEC-Anweisungsliste

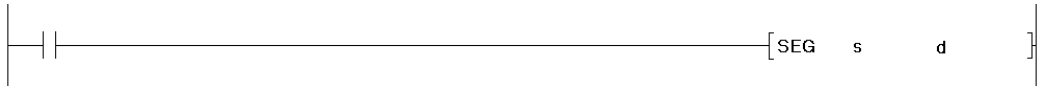
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

Operand	Befehlswert	Datentyp
s	Daten, die in 7-Segment-Daten umgewandelt werden sollen, oder erste Adresse des Operanden, in dem diese Daten gespeichert sind.	BIN-16-Bit
d	Erste Adresse des Operanden, in dem die 7-Segment-Daten gespeichert werden.	

Funktions-weise

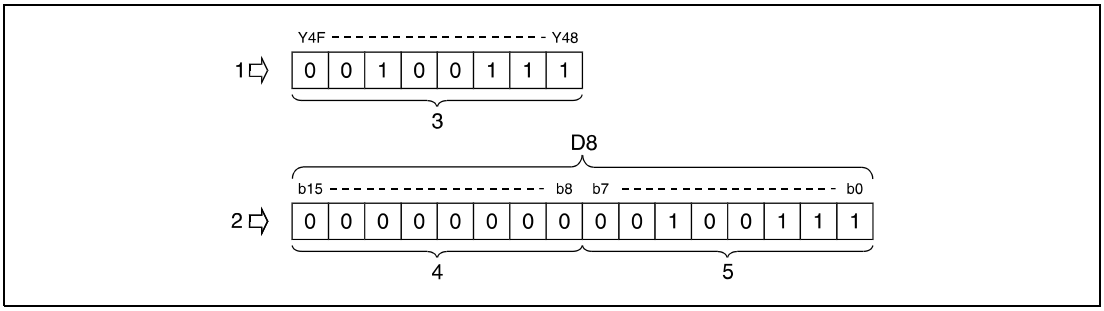
7-Segment-Decodierung

SEG (A- und Q-Serie/System Q) / SEGP (Q-Serie/System Q)Umwandlung eines 4-stelligen Binärwerts

Die SEG-Anweisung wandelt einen 4-stelligen Binärwert in einen 7-Segment-Code zur Anzeige der Werte 0 bis F um. Der Datenwert oder die Startadresse der Daten, die codiert werden sollen, wird in s vorgegeben. Die 7-Segment-Daten werden in d gespeichert.

Sollen die codierten 7-Segment-Daten an Bit-Operanden ausgegeben werden, ist in d die Startadresse und die Größe des Adressenbereiches anzugeben, in dem die Daten gespeichert werden sollen. Handelt es sich in d um einen Wortoperanden, ist nur die Adresse des Operanden anzugeben.

Die Speicherung der Daten in mehreren Bit-Operanden oder einem Wortoperand erfolgt entsprechend dem folgenden Schema.



[1] Bit-Operand

[2] Wortoperand

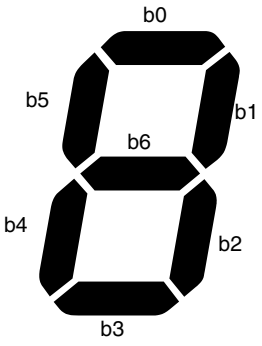
[3] 8 Bits

[4] Diese Bits haben immer den Wert 0

[5] 7-Segment-Daten

7-Segment-Daten

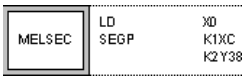
Die folgende Tabelle enthält eine Übersicht der 7-Segment-Daten bezogen auf das Bit-Muster der Quelldaten. Das erste Bit (b0) der 7-Segment-Daten repräsentiert den Status des ersten Operanden bei einer Speicherung der Daten in Bit-Operanden oder dem Status des niedrigstwertigen Bit in einem Wortoperanden.

s		Anordnung der Segmente	d								Anzeige
HEX	Bit-Muster		B7	B6	B5	B4	B3	B2	B1	B0	
0	0000		0	0	1	1	1	1	1	1	0
1	0001		0	0	0	0	0	1	1	0	1
2	0010		0	1	0	1	1	0	1	1	2
3	0011		0	1	0	0	1	1	1	1	3
4	0100		0	1	1	0	0	1	1	0	4
5	0101		0	1	1	0	1	1	0	1	5
6	0110		0	1	1	1	1	1	0	1	6
7	0111		0	0	1	0	0	1	1	1	7
8	1000		0	1	1	1	1	1	1	1	8
9	1001		0	1	1	0	1	1	1	1	9
A	1010		0	1	1	1	0	1	1	1	A
B	1011		0	1	1	1	1	1	0	0	B
C	1100		0	0	1	1	1	0	0	1	C
D	1101		0	1	0	1	1	1	1	0	D
E	1110		0	1	1	1	1	0	0	1	E
F	1111		0	1	1	1	0	0	0	1	F

Beispiel SEGP (Q-Serie und System Q)

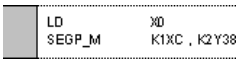
Im folgenden Programm werden nach dem Einschalten von X0 die Zustände der Eingänge XC bis XF als 7-Segment-Code an die Ausgänge Y38 bis Y3F ausgegeben. Die Zustände der Ausgänge Y38 bis Y3F bleiben so lange erhalten, bis sie durch neue Daten überschrieben werden.

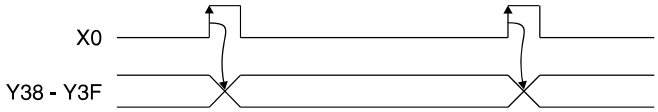
MELSEC-Anweisungsliste



Kontaktplan

IEC-Anweisungsliste





7.5.6 DIS, DISP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
s								●	●	●	●	●	●	●	●	●	●	●								
d								●	●	●	●	●										K1 ↓ K4	9 ●[1]	●		●
n																	●	●								

[1] Die Anzahl der Schritte bei der Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

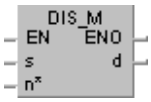
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	●	●	●	●	●	●	●	●	—	SM0	4
d	—	●	●	—	—	—	—	—	—		
n	●	●	●	●	●	●	●	●	—		

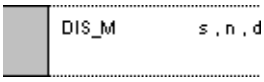
GX IEC Developer

MELSEC-Anweisungsliste

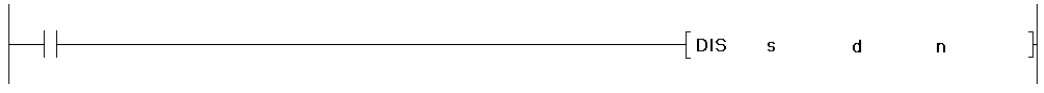
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

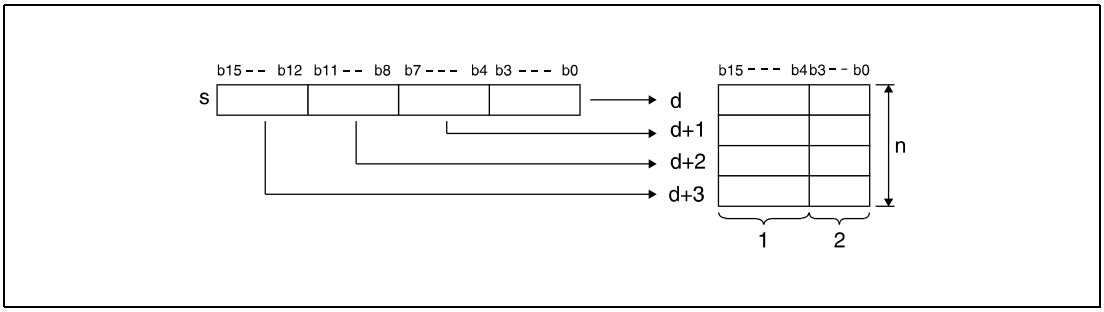
Operand	Befehlswert	Datentyp
s	Erste Adresse des Operanden, in dem die aufzutrennenden Daten gespeichert sind.	BIN-16-Bit
d	Erste Adresse des Operanden, in dem die aufgetrennten Daten gespeichert werden.	
n	Anzahl der aufzutrennenden 4-Bit-Gruppen. Bei n = 0 erfolgt keine Verarbeitung.	

Funktions-weise

Trennen von 16-Bit-Daten

DIS Auftrennen von 16-Bit-Datenwerten

Die DIS-Anweisung trennt einen 16-Bit-Datenwert in Gruppen zu 4 Bits auf und speichert die Zustände der Reihe nach in bis zu 4 Zieloperanden ab. In der Anweisung wird der aufzutrennende Datenwert in s, die Anzahl der 4-Bit-Gruppen in n und die erste Zieladresse in d festgelegt. Weitere 4-Bit-Gruppen werden in d+n abgelegt.



[1] Diese Bits werden mit 0 beschrieben

[2] Speicherbereich

Die höchstwertigen 12 Bits der Zieloperanden, beginnend mit der Adresse in d, werden auf 0 gesetzt.

Für n kann ein Wert zwischen 1 und 4 (entsprechend 4 bis 16 Bit) gesetzt werden.

Ist n = 0, erfolgt keine Verarbeitung, und die vorgegebene Operandenadresse bleibt unverändert.

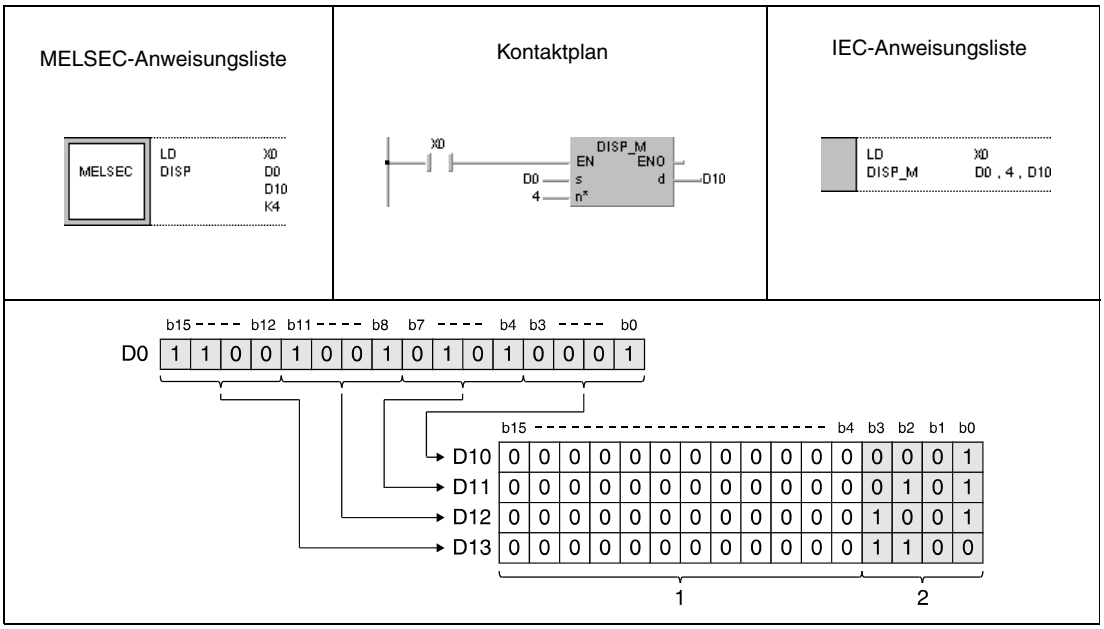
Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Der Wert in n liegt nicht zwischen 0 und 4 (Q-Serie/System Q = Fehlercode 4100).
- Der mit n angegebene Bereich in d überschreitet den für die Speicherung vorgesehenen Bereich des Operanden (Q-Serie/System Q = Fehlercode 4101).

Beispiel DISP

Das folgende Programm trennt mit positiver Flanke von X0 den 16-Bit-Datenwert aus D0 auf und speichert das Bit-Muster in Gruppen zu 4 Bits der Reihe nach in D10 bis D13.



[1] Diese Bits werden mit 0 beschrieben

[2] Speicherbereich

7.5.7 UNI, UNIP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)								Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
s								●	●	●	●														
d								●	●	●	●	●	●	●	●	●						K1 ↓ K4	9 ●[1]	●		●
n																	●	●								

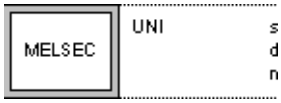
[1] Die Anzahl der Schritte bei der Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

Operanden MELSEC Q

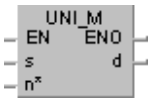
	Operanden								Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	●	●	●	—	—	—	—	—	—	SM0	4
d	—	●	●	●	●	●	●	—	—		
n	●	●	●	●	●	●	●	●	—		

GX IEC Developer

MELSEC-Anweisungsliste

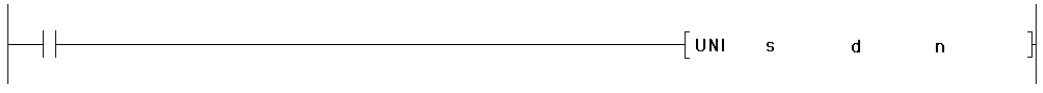


Kontaktplan



IEC-Anweisungsliste

GX Developer



Variablen

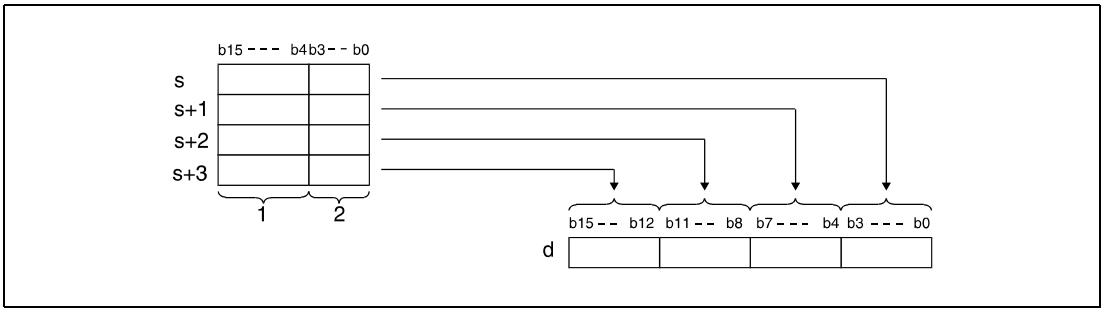
Operand	Befehlswert	Datentyp
s	Erste Adresse des Operanden, in dem die zusammenzuführenden Daten gespeichert sind.	BIN-16-Bit
d	Erste Adresse des Operanden, in dem die zusammengeführten Daten gespeichert werden.	
n	Anzahl der zusammenzuführenden 4-Bit-Gruppen. Bei n = 0 erfolgt keine Verarbeitung.	

Funktionsweise

Gruppieren von 16-Bit-Daten

UNI Zusammenführen von 16-Bit-Datenwerten

Die UNI-Anweisung trennt die jeweils 4 niedrigstwertigen Bits von bis zu vier 16-Bit-Datenwerten auf und speichert die Zustände zusammen in einem 16-Bit-Datenwert ab. In der Anweisung wird die Startadresse der zusammenzuführenden Datenwerte in s, die Anzahl der Operanden in Folge in n und die Zieladresse in d festgelegt.



[1] Diese Bits werden bei der Verarbeitung nicht berücksichtigt

[2] In d zu speichernde 4-Bit-Gruppen

Die niedrigstwertigen 4 Bits der Startoperanden, beginnend mit der Adresse in d, werden auf 0 gesetzt.

Für n kann ein Wert zwischen 1 und 4 gesetzt werden.

Ist n = 0, erfolgt keine Verarbeitung, und die vorgegebene Operandenadresse bleibt unverändert.

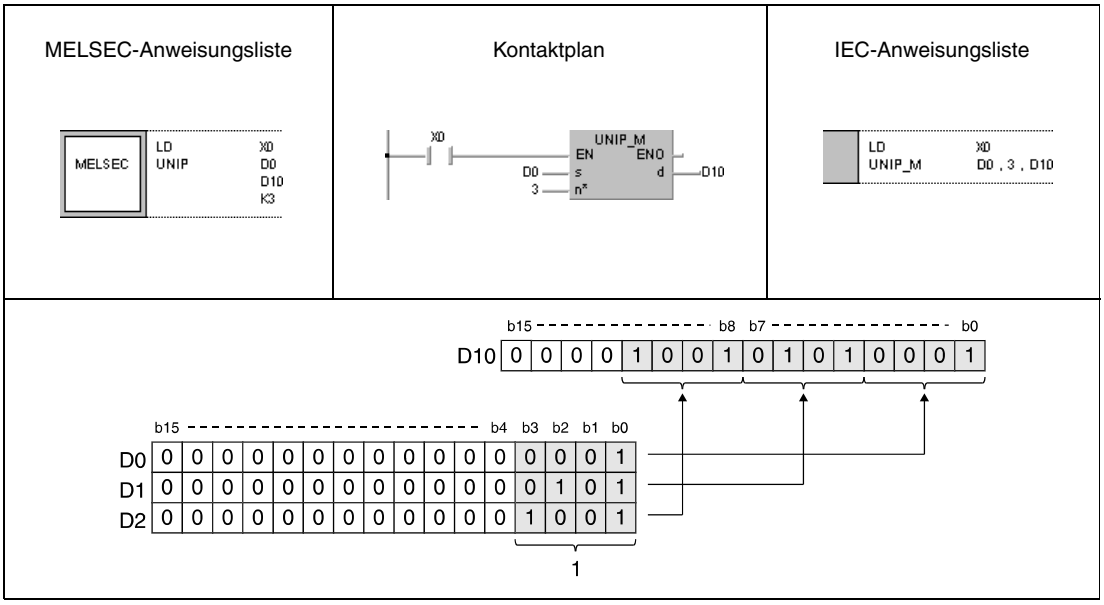
Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Der Wert in n liegt nicht zwischen 0 und 4 (Q-Serie/System Q = Fehlercode 4100).
- Der mit n angegebene Bereich in s überschreitet den für die Speicherung vorgesehenen Bereich des Operanden (Q-Serie/System Q = Fehlercode 4101).

Beispiel UNIP

Im folgenden Programm werden mit positiver Flanke von X0 die jeweils ersten 4 Bits (b0 bis b3) der Datenregister D0 bis D2 der Reihe nach zu einem 16-Bit-Datenwert (die letzten 4 Stellen sind "0") in D10 zusammengefügt.



[1] In D10 zu speichernden 4-Bit-Gruppen

7.5.8 NDIS, NDISP, NUNI, NUNIP

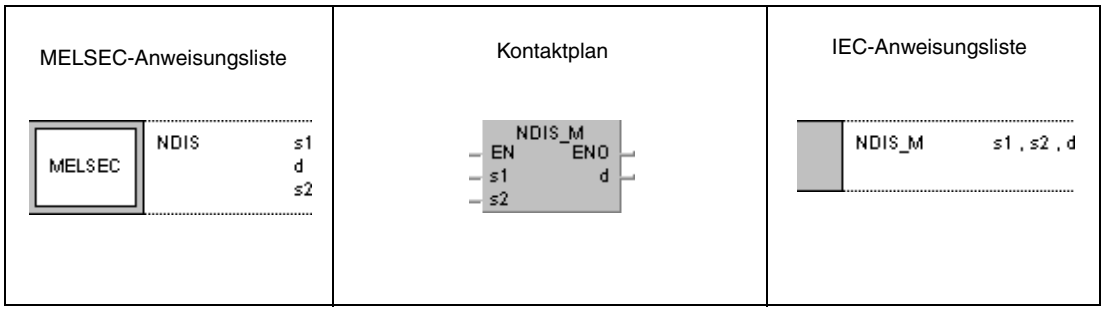
CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	—	●	●	—	—	—	—	—	—	SM0	4
d	—	●	●	—	—	—	—	—	—		
s2	—	●	●	—	—	—	—	—	—		

GX IEC Developer



GX Developer

Variablen

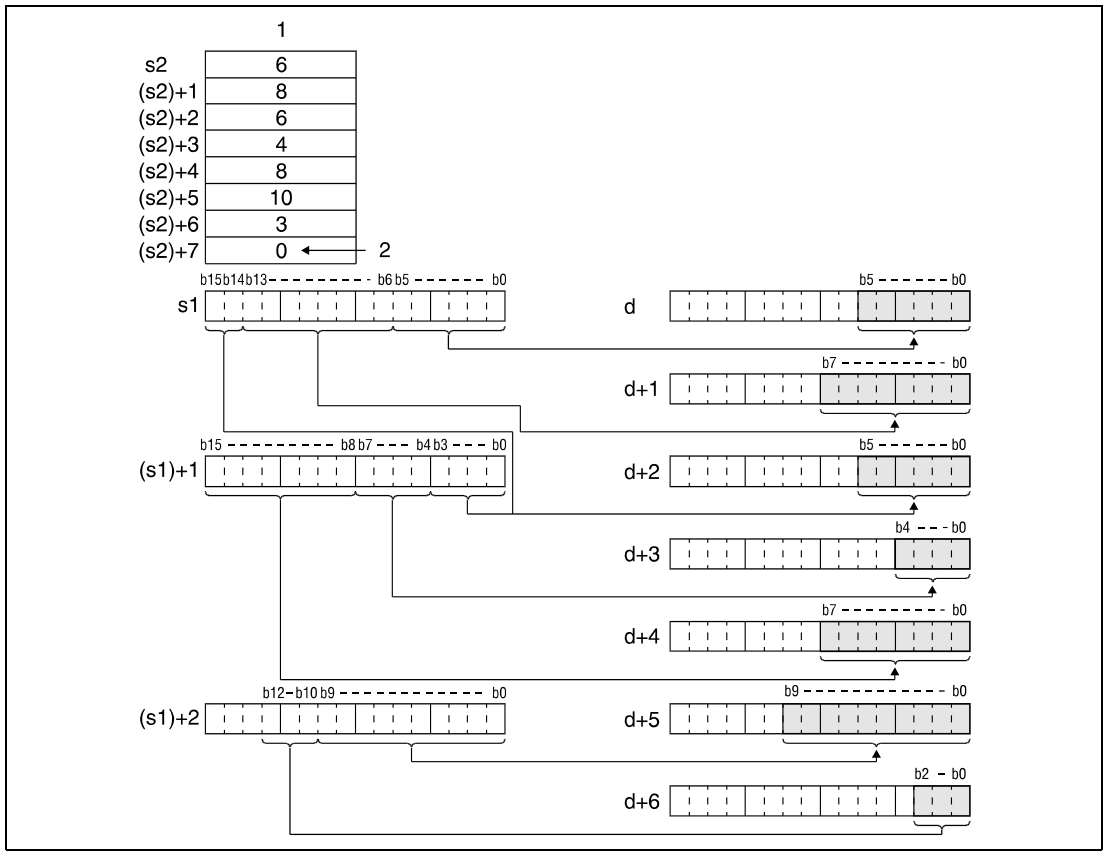
Operand	Befehlswert	Datentyp
s1	Erste Adresse des Operanden, in dem die aufzutrennenden/ zusammenzuführenden Daten gespeichert sind.	BIN-16-Bit
d	Erste Adresse des Operanden, in dem die aufgetrennten/zusammengeführten Daten gespeichert sind.	
s2	Größe der aufzutrennenden/zusammenzuführenden Bit-Gruppen.	

Funktions-weise

Trennen und Gruppieren von Daten in Bit-Gruppen variabler Größe

NDIS Auftrennen von Daten

Die NDIS-Anweisung trennt die Daten der ab s1 angegebenen Operanden in Bit-Gruppen mit den in s2 angegebenen Größen auf. Die aufgetrennten Bit-Gruppen werden einzeln ab dem in d angegebenen Operanden gespeichert.



[1] Größe der Bit-Gruppen

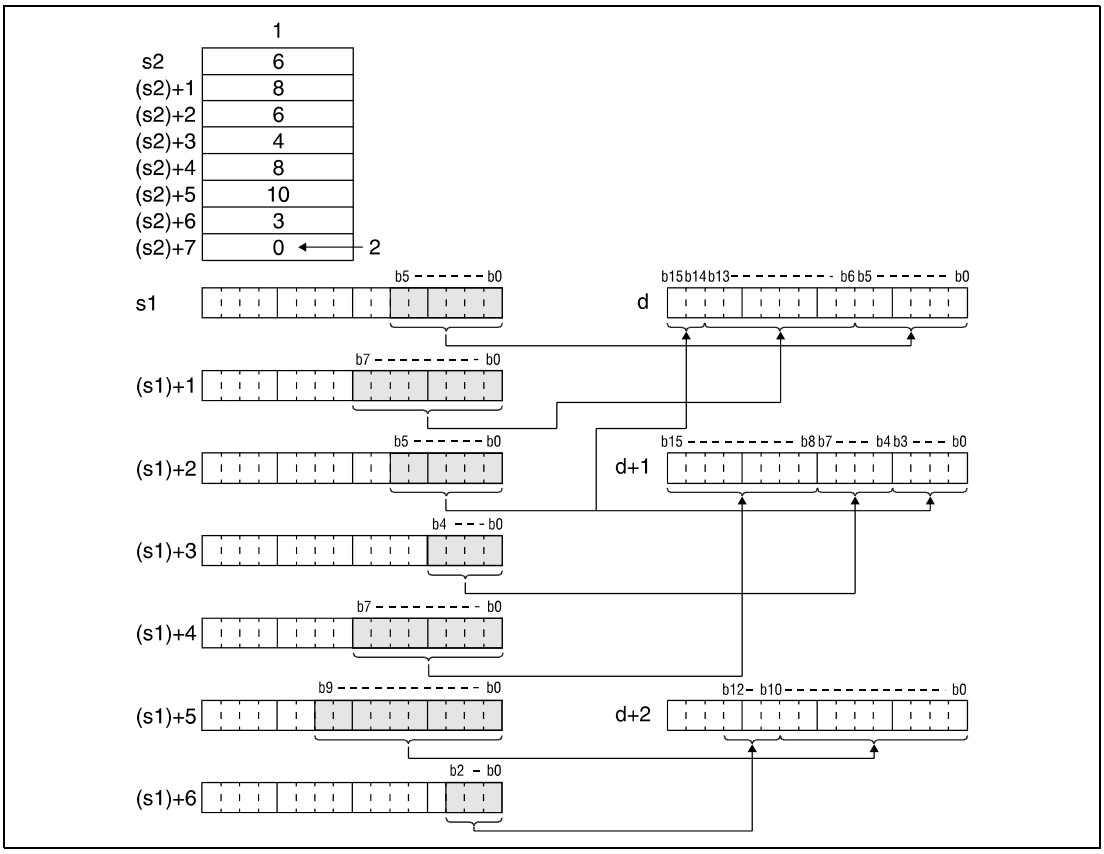
[2] Die 0 kennzeichnet das Ende der Verarbeitung

Die in s2 angegebene Größe der Bit-Gruppen kann einen Wert zwischen 1 und 16 Bit besitzen.

Es werden die in s2 eingetragenen Werte verarbeitet, die zwischen der ersten Adresse des in s2 angegebenen Operanden und der Adresse mit dem Eintrag 0 liegen.

NUNI Zusammenführen von Daten

Die NUNI-Anweisung trennt die ab s2 angegebenen Größen der Bit-Gruppen aus den ab s1 gespeicherten Operanden heraus und führt diese Bit-Gruppen in einem Datenwert zusammen. Die Bit-Gruppen werden aufeinanderfolgend ab dem in d angegebenen Operanden gespeichert.



[1] Größe der Bit-Gruppen

[2] Die 0 kennzeichnet das Ende der Verarbeitung

Die in s2 angegebene Größe der Bit-Gruppen kann einen Wert zwischen 1 und 16 Bit besitzen.

Es werden die in s2 eingetragenen Werte verarbeitet, die zwischen der ersten Adresse des in s2 angegebenen Operanden und der Adresse mit dem Eintrag 0 liegen.

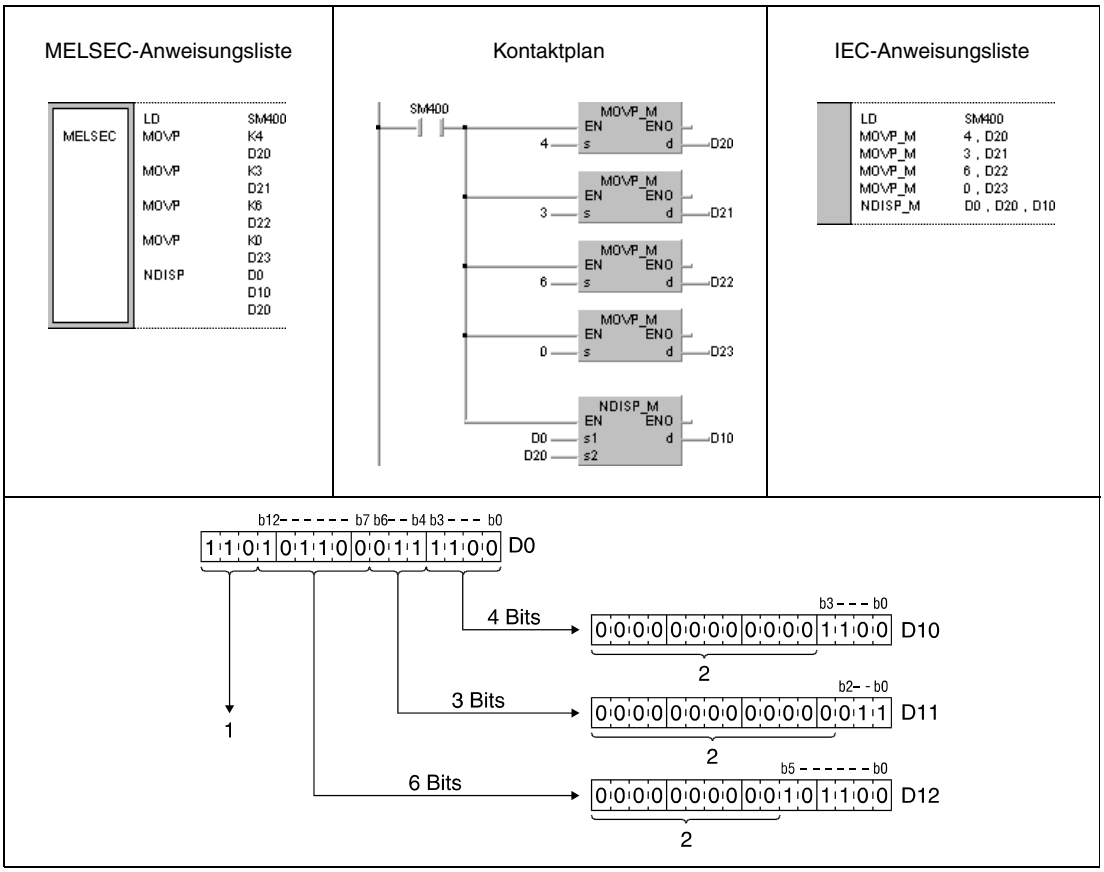
Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die Bit-Gruppen mit den in s2 angegebenen Größen in den in s1 oder d angegebenen Operanden liegen nicht innerhalb der für die Speicherung vorgesehenen Bereiche der Operanden (Fehlercode 4101).
- Die in s2 angegebenen Größen der Bit-Gruppen liegen nicht in dem Bereich zwischen 1 und 16 Bits (Fehlercode 4100).

Beispiel 1 NDISP

Das folgende Programm trennt mit positiver Flanke von SM400 die Bit-Gruppen b0 - b3 (4), b4 - b6 (3) und b 7- b12 (6) aus D0 heraus und speichert jede einzelne Bit-Gruppe beginnend mit der Gruppe b0 - b3 in D10-D12 ab. Die Werte in Klammern geben die in D20 bis D22 angegebenen Bit-Gruppen-Größen an. In D23 muss eine Null eingetragen sein (siehe Funktionsweise).

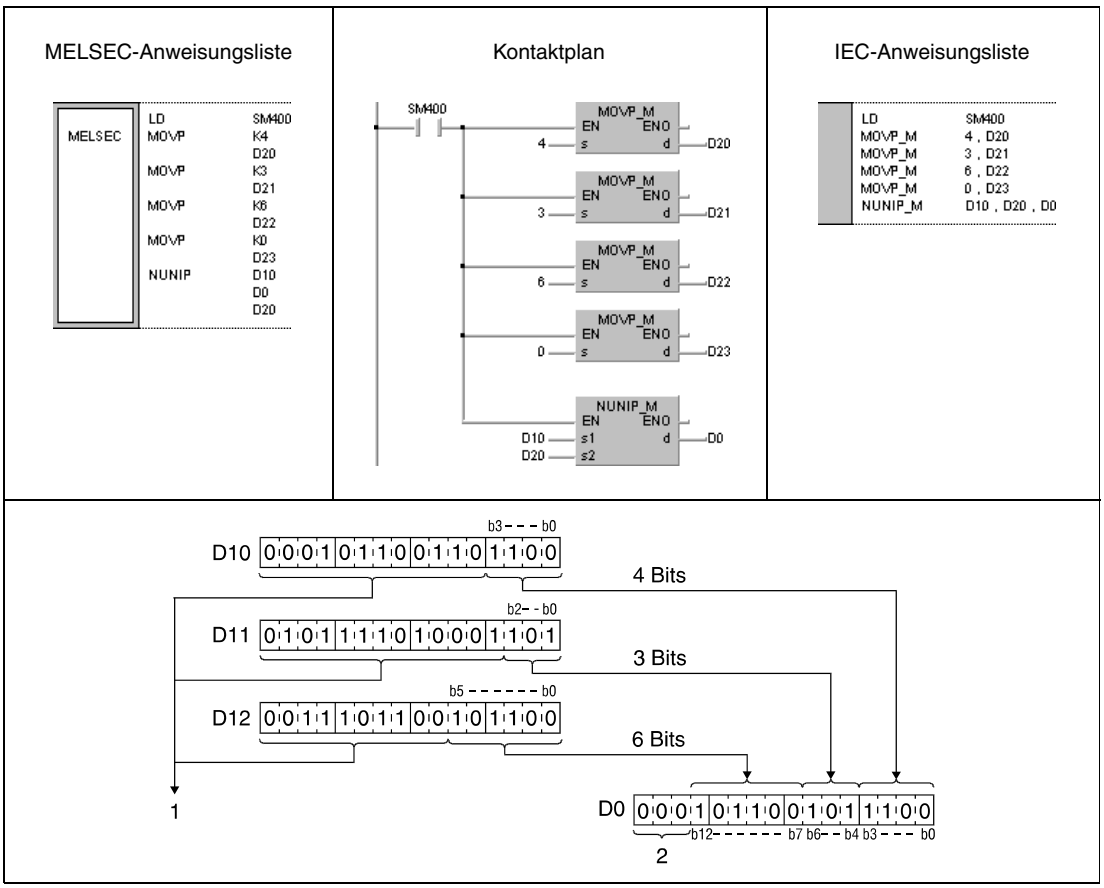


[1] Diese Bits werden bei der Verarbeitung nicht berücksichtigt

[2] Diese Bits werden mit 0 beschrieben

Beispiel 2 NUNIP

Das folgende Programm trennt mit positiver Flanke von SM400 die Bit-Gruppen b0 - b3 (4), b0 - b2 (3) und b0 - b5 (6) aus D10 bis D12 heraus und speichert die Bit-Gruppen beginnend mit der Gruppe b0 - b3 aufeinanderfolgend in D0 ab. Die Werte in Klammern geben die in D20 bis D22 angegebenen Bit-Gruppen-Größen an. In D23 muss eine Null eingetragen sein (siehe Funktionsweise).



[1] Diese Bits werden bei der Verarbeitung nicht berücksichtigt

[2] Diese Bits werden mit 0 beschrieben

7.5.9 WTOB, WTOBP, BTOW, BTOWP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●

[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

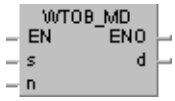
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	—	—	—	—	—	SM0	4
d	—	●	●	—	—	—	—	—	—		
n	●	●	●	●	●	●	●	●	—		

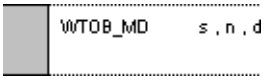
GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

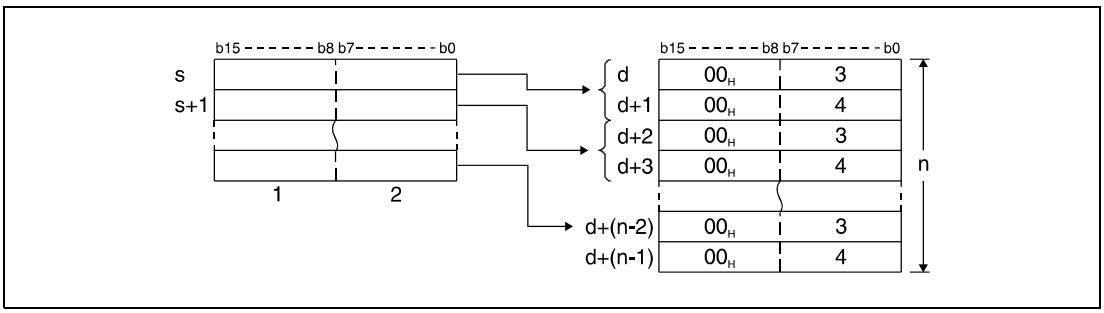
Operand	Befehlswert	Datentyp
s	Erste Adresse des Operanden, in dem die aufzutrennenden/ zusammenzuführenden Bytes gespeichert sind.	BIN-16-Bit
d	Erste Adresse des Operanden, in dem die aufgetrennten/zusammengeführten Bytes gespeichert sind.	
n	Anzahl der aufzutrennenden/zusammenzuführenden Byte-Gruppen.	

Funktionsweise

Trennen und Gruppieren von Daten in Byte-Gruppen

WTOB Auftrennen von Daten

Die WTOB-Anweisung trennt einen 16-Bit-Datenwert in Byte-Gruppen auf und speichert die Zustände der Reihe nach in den Zieloperanden ab. In der Anweisung werden die aufzutrennenden Datenwerte in s, die Anzahl der Byte-Gruppen in n und die erste Zieladresse in d festgelegt. Weitere Byte-Gruppen werden in d+n abgelegt. Zur Speicherung werden nur die niedrigstwertigen Bytes der in d angegebenen Operanden verwendet.



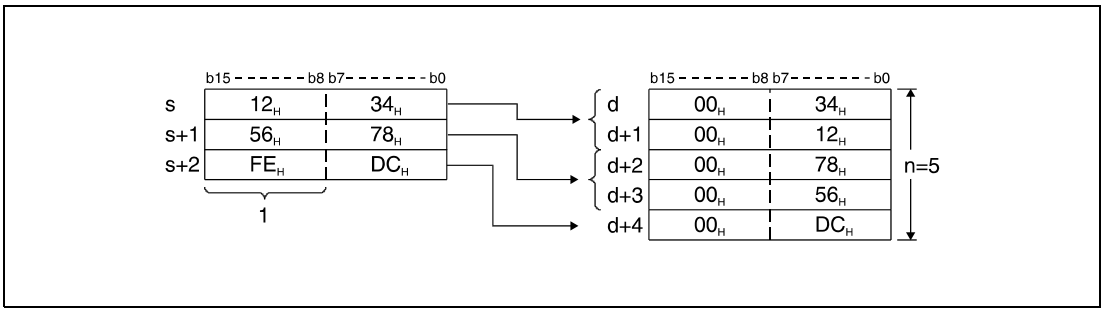
[1] Höchstwertige Bytes

[2] Niedrigswertige Bytes

[3] Daten des entsprechenden niedrigstwertigen Bytes

[4] Daten des entsprechenden höchstwertigen Bytes

Ist beispielsweise n = 5, werden 5 Bytes aus den in s bis s+2 angegebenen Operanden herausgetrennt und aufeinanderfolgend in den niedrigstwertigen Bytes von den in d bis d+4 angegebenen Operanden gespeichert.

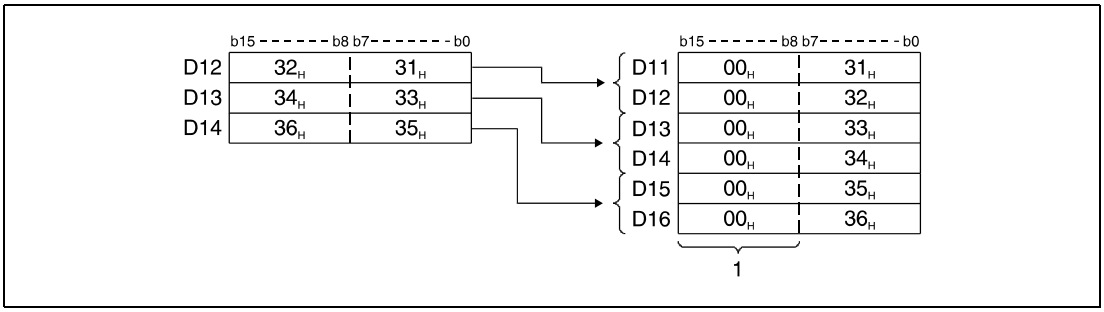


[1] Diese Bytes werden bei der Verarbeitung nicht berücksichtigt

Die in n angegebene Anzahl der Byte-Gruppen bestimmt automatisch den Bereich der 16-Bit-Daten in s und den Speicherbereich der Byte-Gruppen in d.

Ist n = 0, erfolgt keine Verarbeitung, und die vorgegebenen Operandenadressen bleiben unverändert.

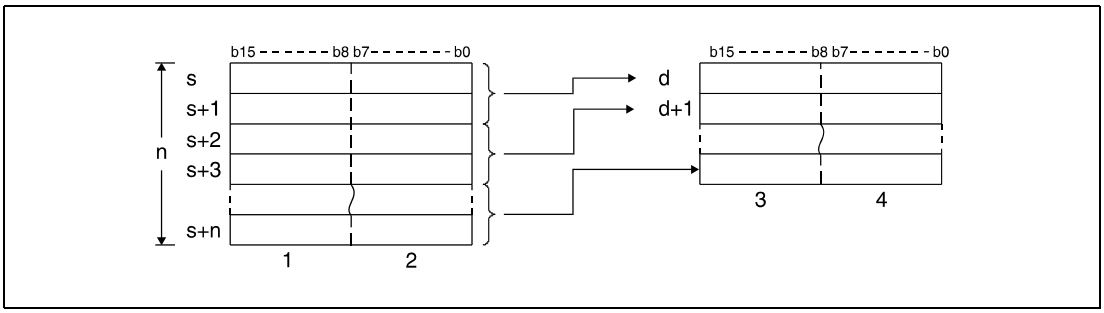
Die höchstwertigen Bytes in den in d angegebenen Operanden werden mit dem Wert "00H" beschrieben.



1 Diese Bytes werden mit dem Wert "00H" beschrieben

BTOW Zusammenführen von Daten

Die BTOW-Anweisung trennt die jeweils niedrigstwertigen Bytes von 16-Bit-Datenwerten auf und speichert die Zustände zusammen in 16-Bit-Datenwerten ab. In der Anweisung wird die Startadresse der zusammenzuführenden Datenwerte in s, die Anzahl der Byte-Gruppen in Folge in n und die Zieladresse in d festgelegt.



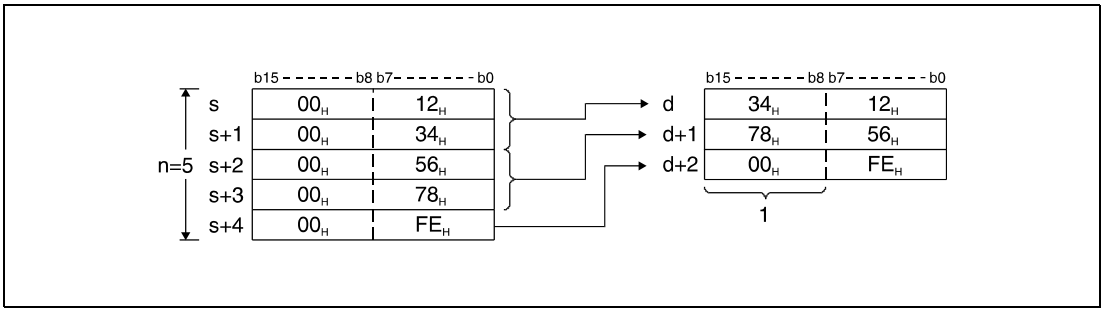
[1] Diese Bytes werden bei der Verarbeitung nicht berücksichtigt

[2] 1. bis n-tes Byte der Daten

[3] Daten des 2., 4. und n-ten Bytes

[4] Daten des 1., 3. und (n-1)-ten Bytes

Ist beispielsweise n = 5, werden die niedrigstwertigen 5 Bytes aus den in s bis s+4 angegebenen Operanden herausgetrennt und aufeinanderfolgend in den in d bis d+2 angegebenen Operanden gespeichert.



[1] Dieses Byte wird mit dem Wert "00H" beschrieben

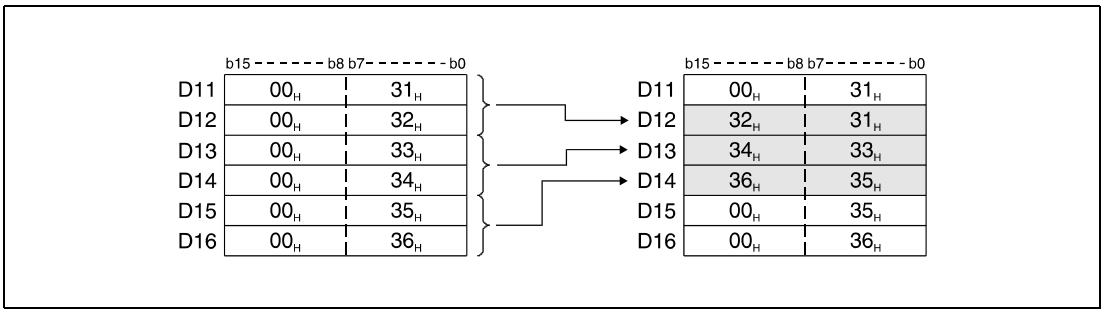
Die in n angegebene Anzahl der Byte-Gruppen bestimmt automatisch den Bereich der Byte-Daten in s und den Speicherbereich der Byte-Daten in d.

Ist n = 0, erfolgt keine Verarbeitung, und die vorgegebene Operandenadresse bleibt unverändert.

Die höchstwertigen Bytes in den in s angegebenen Operanden werden bei der Verarbeitung nicht berücksichtigt.

Eine fehlerfreie Verarbeitung erfolgt auch in Fällen, in denen die Speicherbereiche von s bis s+n mit denen von d bis d+n überlappen.

Die folgende Darstellung zeigt die Speicherung für den Fall, dass die niedrigstwertigen Bytes aus D11 bis D16 herausgetrennt werden und in D12 bis D14 aufeinanderfolgend wieder abgelegt werden.



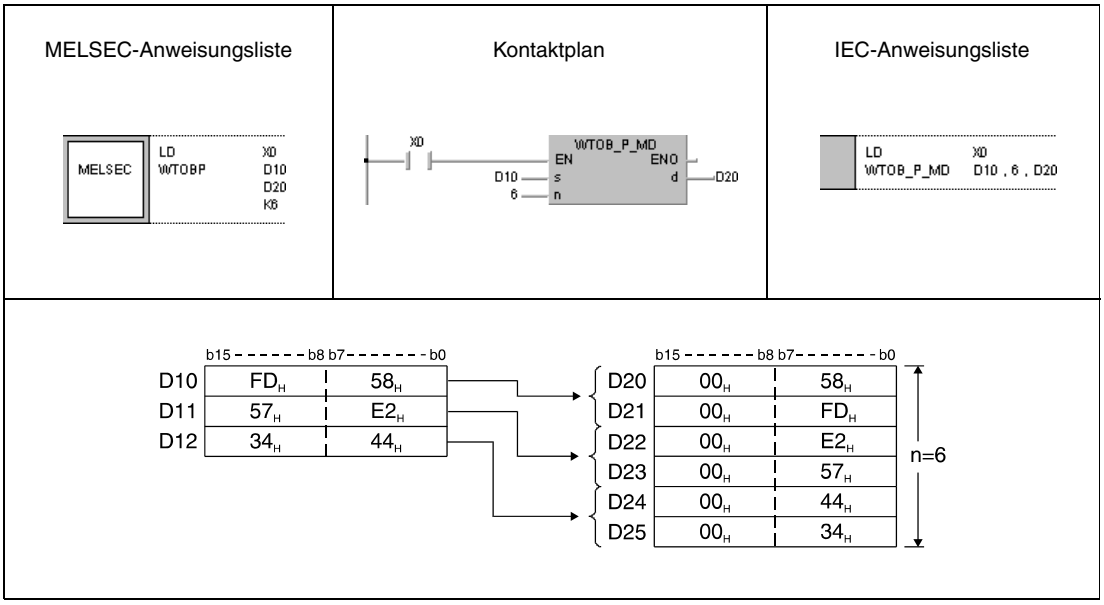
Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die mit n angegebene Anzahl von Byte-Gruppen, die ab dem in s angegebenen Operanden gespeichert sind, liegt außerhalb des für die Speicherung vorgesehenen Bereichs. (Fehlercode 4101).
- Die mit n angegebene Anzahl von Byte-Gruppen, die ab dem in d angegebenen Operanden gespeichert sind, liegt außerhalb des für die Speicherung vorgesehenen Bereichs. (Fehlercode 4101).

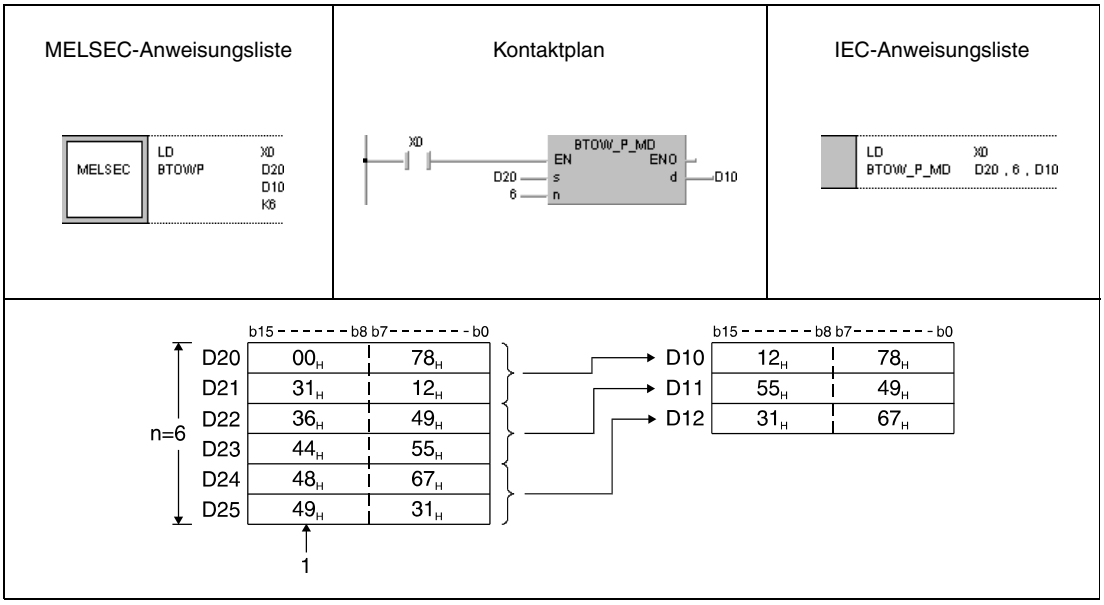
Beispiel 1 WTOBP

Das folgende Programm trennt mit positiver Flanke von X0 die 6 Bytes in D10 bis D12 der Reihe nach heraus und speichert diese Bytes einzeln in den niedrigstwertigen Bytes in D20 bis D25 ab.



Beispiel 2 BTOWP

Das folgende Programm trennt mit positiver Flanke von X0 die 6 niedrigstwertigen Bytes der Register D20 bis D25 heraus und führt diese Bytes aufeinanderfolgend in D10 bis D12 zusammen.



[1] Diese Bytes werden bei der Verarbeitung nicht berücksichtigt

7.5.10 MAX, MAXP, DMAX, DMAXP

CPU

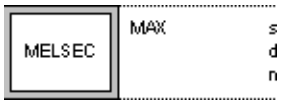
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	—	—	—	—	—	SM0	4
d	—	●	●	—	—	—	—	—	—		
n	●	●	●	●	●	●	●	●	—		

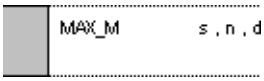
GX IEC Developer

MELSEC-Anweisungsliste



Kontaktplan

IEC-Anweisungsliste



GX Developer

Variablen

Operand	Befehlswert	Datentyp
s	Erste Adresse des Operanden, in dem die zu durchsuchenden Daten gespeichert sind.	BIN-16-/32-Bit
d	Erste Adresse des Operanden, in dem das Suchergebnis des Maximalwertes gespeichert wird.	
n	Anzahl der zu durchsuchenden Datenblöcke.	BIN-16-Bit

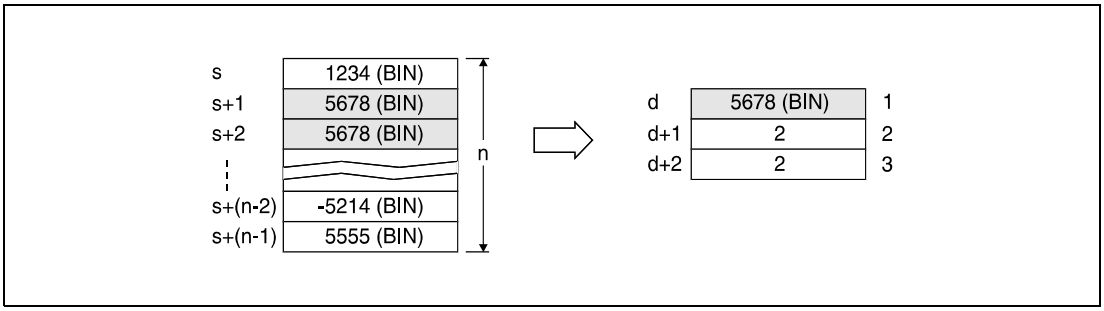
Funktionsweise

Suchen von Maximalwerten in 16-/32-Bit-Daten

MAX Suchen von Maximalwerten in 16-Bit-Daten

Die MAX-Anweisung sucht in 16-Bit-Datenblöcken nach dem größten Wert. Die Anzahl der zu durchsuchenden Datenblöcke wird in n angegeben. Der größte in s bis s+(n-1) gefundene Wert wird in d abgespeichert.

Die Stelle in s bis s+(n-1), an der der Maximalwert das erste Mal auftritt, wird mit s = 1 beginnend gezählt, und in d+1 abgespeichert. Die Anzahl der in s bis s+(n-1) vorhandenen, identischen Maximalwerte wird in d+2 abgelegt.



[1] Gefundener Maximalwert

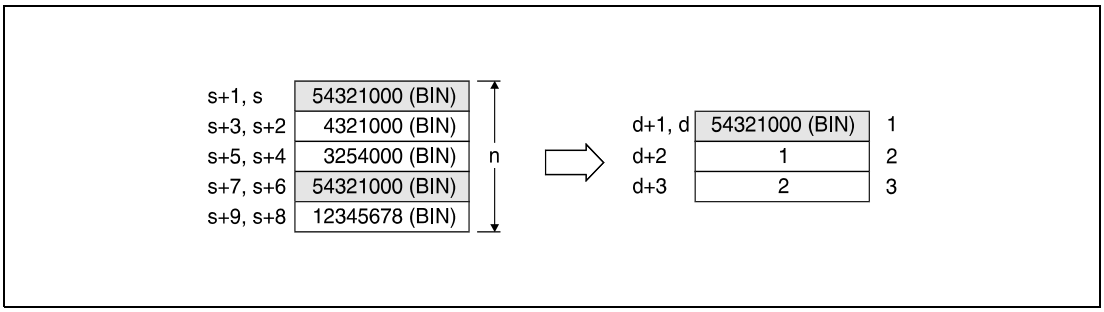
[2] Stelle, an der der Wert das 1. Mal gefunden wird

[3] Anzahl der identischen Maximalwerte

DMAX Suchen von Maximalwerten in 32-Bit-Daten

Die DMAX-Anweisung sucht in 32-Bit-Datenblöcken nach dem größten Wert. Die Anzahl der zu durchsuchenden Datenblöcke wird in n angegeben. Der größte in s bis s+(n-1) gefundene Wert wird in d und d+1 abgespeichert.

Die Stelle in s bis s+(n-1), an der der Maximalwert das erste Mal auftritt, wird in d+2 abgespeichert. Die Anzahl der in s bis s+(n-1) vorhandenen, identischen Maximalwerte wird in d+3 abgelegt.



[1] Gefundener Maximalwert

[2] Stelle, an der der Wert das 1.Mal gefunden wird

[3] Anzahl der identischen Maximalwerte

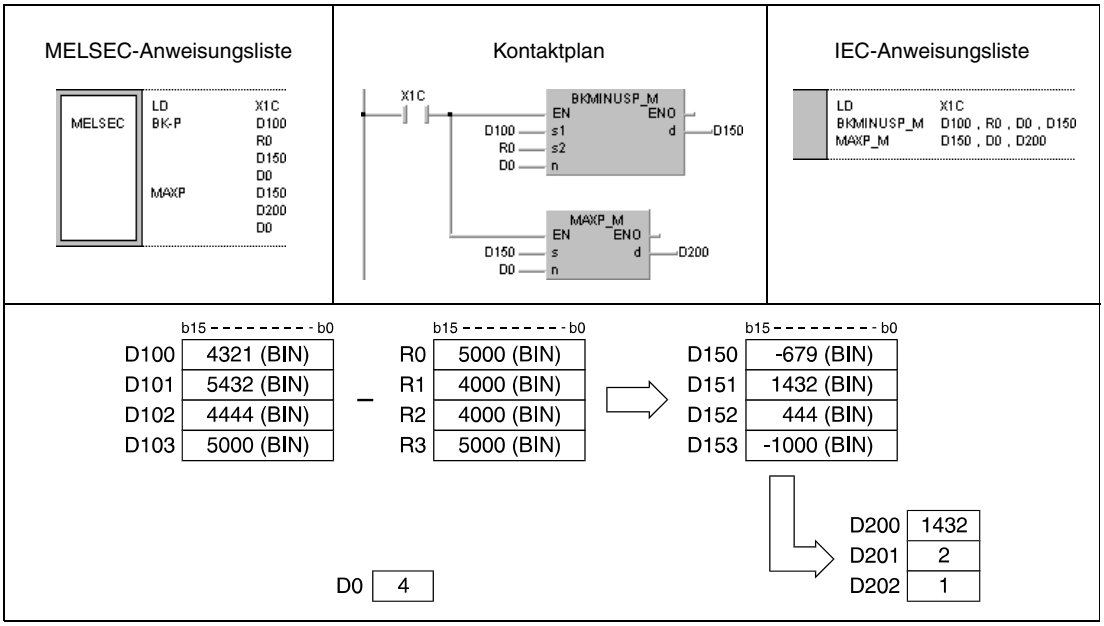
Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die mit n angegebene Anzahl der Datenblöcke, die in den in s angegebenen Operanden gespeichert sind, liegt außerhalb des für die Speicherung vorgesehenen Bereichs (Fehlercode 4101).

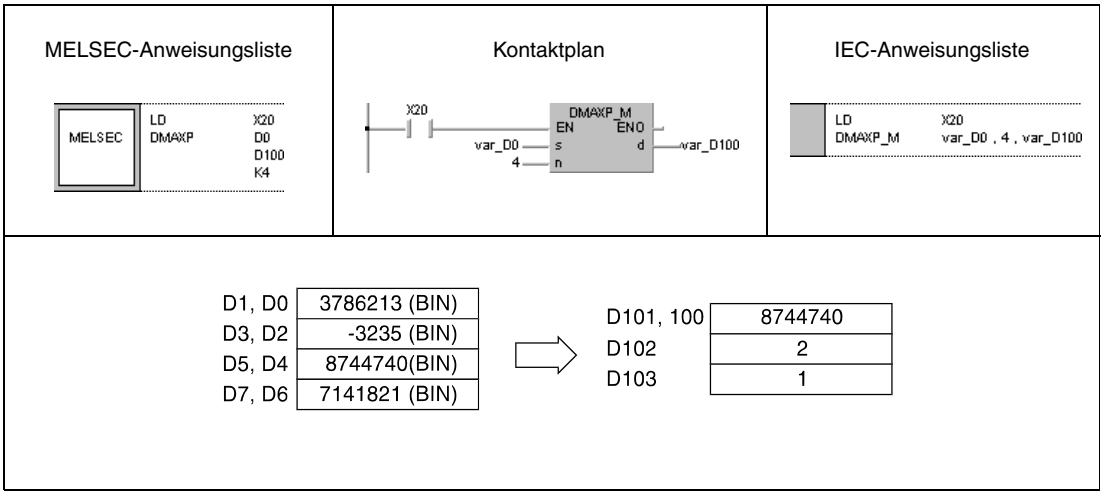
Beispiel 1 MAXP

Das folgende Programm subtrahiert mit positiver Flanke von X1C die Daten in R0 bis R3 von den Daten in D100 bis D103 und speichert das Ergebnis in D150 bis D153 ab. Die Anzahl der 16-Bit-Datenblöcke (4) ist in D0 angegeben. Im folgenden Schritt werden ebenfalls mit positiver Flanke von X1C die Register D150 bis D153 nach dem Maximalwert durchsucht. Der gefundene Wert wird in D200, die Angabe der Stelle in D201 und die Anzahl der identischen Maximalwerte in D202 gespeichert.



Beispiel 2 DMAXP

Das folgende Programm sucht mit positiver Flanke von X20 den Maximalwert der 32-Bit-Daten in D0 bis D7, und speichert den gefundenen Wert in D100 und D101 ab. Die Stelle des Wertes wird in D102 und die Anzahl der identischen Maximalwerte in D103 abgespeichert.



HINWEIS *Das Programmbeispiel 2 ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.5.11 MIN, MINP, DMIN, DMINP

CPU

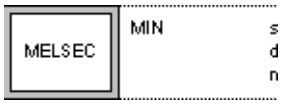
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Operanden MELSEC Q

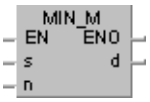
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	—	—	—	—	—	SM0	4
d	—	●	●	—	—	—	—	—	—		
n	●	●	●	●	●	●	●	●	—		

GX IEC Developer

MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste

GX Developer

Variablen

Operand	Befehlswert	Datentyp
s	Erste Adresse des Operanden, in dem die zu durchsuchenden Daten gespeichert sind.	BIN-16-/32-Bit
d	Erste Adresse des Operanden, in dem das Suchergebnis des Minimalwertes gespeichert wird.	
n	Anzahl der zu durchsuchenden Datenblöcke.	BIN-16-Bit

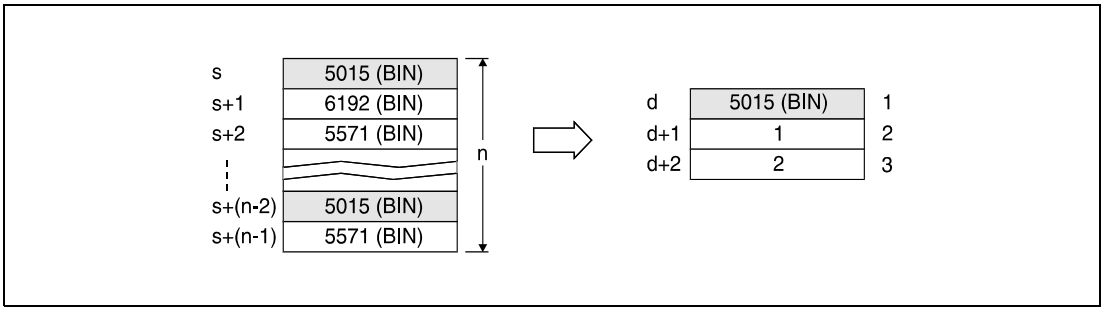
Funktionsweise

Suchen von Minimalwerten in 16-/32-Bit-Daten

MIN Suchen von Minimalwerten in 16-Bit-Daten

Die MIN-Anweisung sucht in 16-Bit-Datenblöcken nach dem kleinsten Wert. Die Anzahl der zu durchsuchenden Datenblöcke wird in n angegeben. Der kleinste in s bis s+(n-1) gefundene Wert wird in d abgespeichert.

Die Stelle in s bis s+(n-1), an der der Minimalwert das erste Mal auftritt, wird mit s = 1 beginnend gezählt, und in d+1 abgespeichert. Die Anzahl der in s bis s+(n-1) vorhandenen, identischen Minimalwerte wird in d+2 abgelegt.



[1] Gefundener Minimalwert

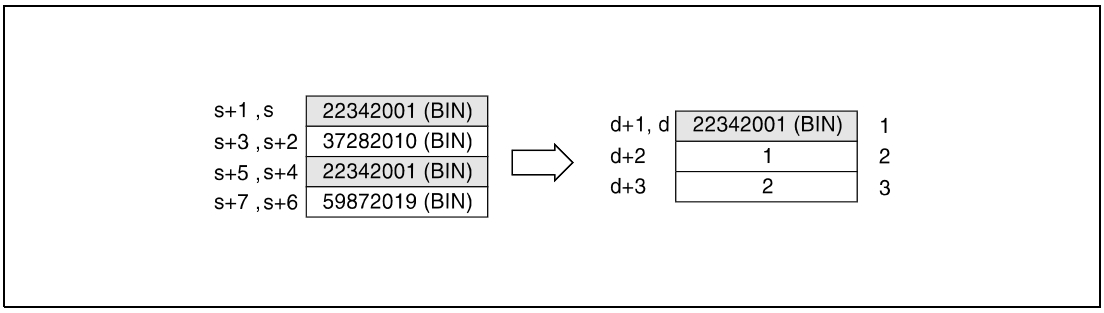
[2] Stelle, an der der Wert das 1.Mal auftritt

[3] Anzahl der identischen Minimalwerte

DMIN Suchen von Minimalwerten in 32-Bit-Daten

Die DMIN-Anweisung sucht in 32-Bit-Datenblöcken nach dem kleinsten Wert. Die Anzahl der zu durchsuchenden Datenblöcke wird in n angegeben. Der kleinste in s bis s+(n-1) gefundene Wert wird in d und d+1 abgespeichert.

Die Stelle in s bis s+(n-1), an der der Minimalwert das erste Mal auftritt, wird in d+2 abgespeichert. Die Anzahl der in s bis s+(n-1) vorhandenen, identischen Minimalwerte wird in d+3 abgelegt.



[1] Gefundener Minimalwert

[2] Stelle, an der der Wert das 1. mal auftritt

[3] Anzahl der identischen Minimalwerte

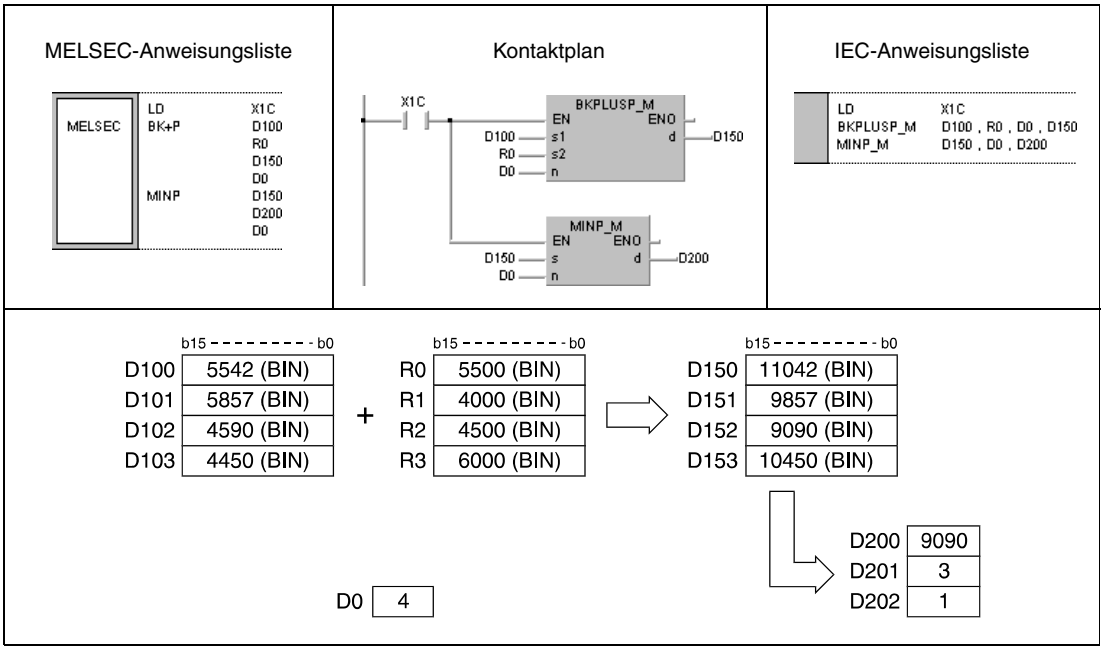
Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die mit n angegebene Anzahl der 32-Bit-Datenblöcke, die ab den in s und s+1 angegebenen Operanden gespeichert sind, liegt außerhalb des für die Speicherung vorgesehenen Bereichs (Fehlercode 4101).

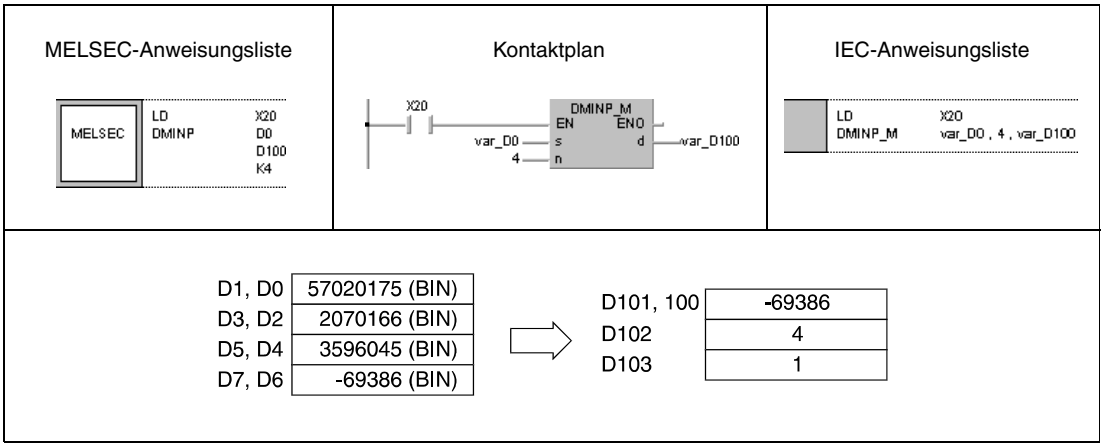
Beispiel 1 MINP

Das folgende Programm addiert mit positiver Flanke von X1C die Daten in D100 bis D103 zu den Daten in R0 bis R3 und speichert das Ergebnis in D150 bis D153 ab. Die Anzahl der 16-Bit-Datenblöcke (4) ist in D0 angegeben. Im folgenden Schritt werden ebenfalls mit positiver Flanke von X1C die Register D150 bis D153 nach dem Minimalwert durchsucht. Der gefundene Wert wird in D200, die Angabe der Stelle in D201 und die Anzahl der identischen Minimalwerte in D202 gespeichert.



Beispiel 2 DMINP

Das folgende Programm sucht mit positiver Flanke von X20 den Minimalwert der 32-Bit-Daten in D0 bis D7 und speichert den gefundenen Wert in D100 und D101 ab. Die Stelle des Wertes wird in D102 und die Anzahl der identischen Minimalwerte in D103 abgespeichert.



HINWEIS *Das Programmbeispiel 2 ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.5.12 SORT, SORTP, DSORT, DSORTP

CPU

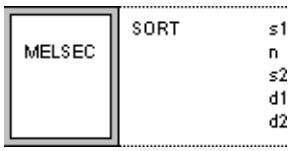
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Operanden MELSEC Q

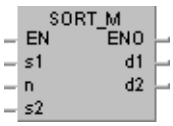
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	—	●	●	—	—	—	—	—	—	SM0	6
n	●	●	●	—	—	—	—	—	—		
s2	●	●	●	●	●	●	●	●	—		
d1	●	—	—	●	●	●	●	●	—		
d2	—	●	●	—	—	—	—	—	—		

GX IEC Developer

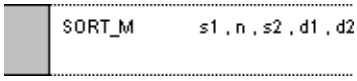
MELSEC-Anweisungsliste



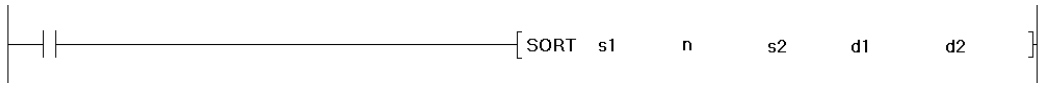
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

Operand	Befehlswert	Datentyp
s1	Erste Adresse des Operanden, in dem die zu sortierenden Daten gespeichert sind.	BIN-16-/32-Bit
n	Anzahl der zu sortierenden Datenblöcke.	BIN-16-Bit
s2	Anzahl der Datenblöcke, die während einer Sortieroperation verglichen werden.	BIN-16-Bit
d1	Adresse des Bits, das nach Abschluss der Sortieroperation gesetzt wird.	Bit
d2	Operand für systeminterne Verwendung.	BIN-16-Bit

Funktionsweise

Sortieren von 16-/32-Bit-Daten

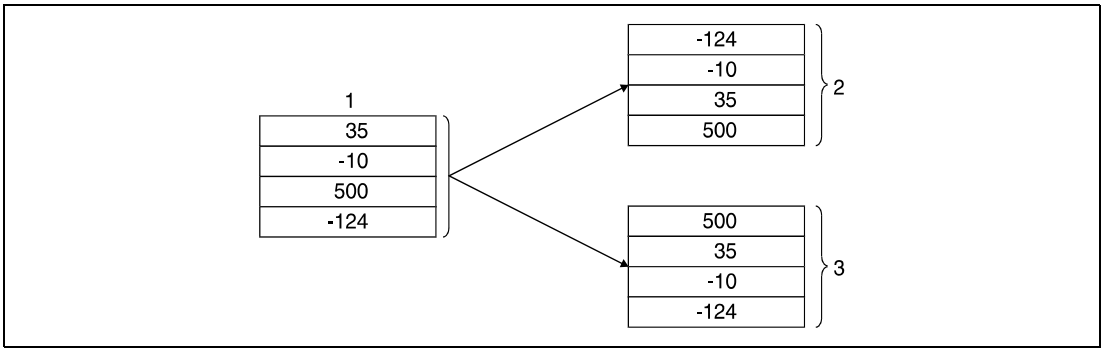
SORT Sortieren von 16-Bit-Daten

Die SORT-Anweisung sortiert die mit n angegebene Anzahl der in s1 angegebenen 16-Bit-Daten in steigender oder fallender Reihenfolge.

Die Sortierrichtung wird mit dem Diagnosemerker SM703 festgelegt.

SM703 AUS: Steigende Sortierrichtung

SM703 EIN: Fallende Sortierrichtung



[1] Zu sortierende Daten

[2] In steigender Reihenfolge sortierte Daten (SM703 = AUS)

[3] In fallender Reihenfolge sortierte Daten (SM703 = Ein)

Für die Ausführung der SORT-Anweisung sind mehrere Sortierdurchgänge erforderlich. Die Anzahl der erforderlichen Sortierdurchgänge wird aus der Division der maximalen Anzahl der Sortierdurchgänge durch die in s2 angegebene Anzahl der 16-Bit-Daten, die in einem Sortierdurchgang verglichen werden, gebildet (gebrochen rationale Zahlen werden aufgerundet). Eine Erhöhung der in s2 angegebenen Anzahl von 16-Bit-Daten senkt die Zahl der erforderlichen Sortierdurchgänge bei gleichzeitiger Verlängerung der Verarbeitungszeit.

Die erforderliche Anzahl der Sortierdurchgänge bis zum Abschluss der Sortieroperation wird gemäß nachstehender Gleichung berechnet:

Erforderliche Anzahl der Sortierdurchgänge = ((n) x (n-1)) / (2 x (s2))

Werden beispielsweise n = 10 und s2 = 1 in die Gleichung eingesetzt, ergeben sich 45 Sortierdurchgänge bis zum Abschluss der Sortieroperation.

Das Ergebnis bei n = 10 und s2 = 2 lautet 22,5 Sortierdurchgänge. Durch das Aufrunden des Ergebnisses ergeben sich dann 23 Sortierdurchgänge.

Das in d1 angegebene Bit ist während der Sortieroperation zurückgesetzt und wird erst gesetzt, wenn die Sortieroperation abgeschlossen ist. Dieses Bit bleibt gesetzt und muss durch entsprechende Programmierung zurückgesetzt werden.

Die in d2+0 und d2+1 angegebenen Operanden werden für die systeminterne Verarbeitung während der Sortieroperation genutzt. Diese Operanden sollten daher durch die Programmierung nicht verändert werden.

Wird der Wert in n während der Sortieroperation verändert, erfolgt die Verarbeitung mit der aktuell angegebenen Anzahl von 16-Bit-Daten.

Die Verarbeitung wird mit Rücksetzen der Ausführungsbedingung abgebrochen. Bei erneutem Setzen der Ausführungsbedingung wird mit der Sortieroperation von neuem begonnen.

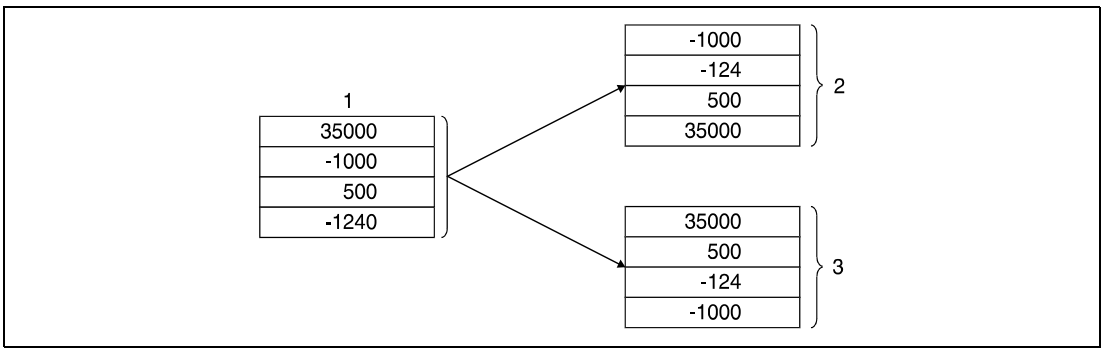
DSORT Sortieren von 32-Bit-Daten

Die SORT-Anweisung sortiert die in n angegebene Anzahl der in s1 angegebenen 32-Bit-Daten in steigender oder fallender Reihenfolge.

Die Sortierrichtung wird mit dem Diagnosemerker SM703 festgelegt.

SM703 AUS: Steigende Sortierrichtung

SM703 EIN: Fallende Sortierrichtung



[1] Zu sortierende Daten

[2] In steigender Reihenfolge sortierte Daten (SM703 = AUS)

[3] In fallender Reihenfolge sortierte Daten (SM703 = Ein)

Für die Ausführung der DSORT-Anweisung sind mehrere Sortierdurchgänge erforderlich. Die Anzahl der erforderlichen Sortierdurchgänge wird aus der Division der maximalen Anzahl der Sortierdurchgänge durch die in s2 angegebene Anzahl der 32-Bit-Daten, die in einem Sortierdurchgang verglichen werden, gebildet (gebrochen rationale Zahlen werden aufgerundet). Eine Erhöhung der in s2 angegebenen Anzahl von 32-Bit-Daten senkt die Zahl der erforderlichen Sortierdurchgänge bei gleichzeitiger Verlängerung der Verarbeitungszeit.

Die erforderliche Anzahl der Sortierdurchgänge bis zum Abschluss der Sortieroperation wird gemäß nachstehender Gleichung berechnet:

Erforderliche Anzahl der Sortierdurchgänge = ((n) x (n-1)) / (2 x (s2))

Werden beispielsweise n = 10 und s2 = 1 in die Gleichung eingesetzt, ergeben sich 45 Sortierdurchgänge bis zum Abschluss der Sortieroperation.

Das Ergebnis bei n = 10 und s2 = 2 lautet 22,5 Sortierdurchgänge. Durch das Aufrunden des Ergebnisses ergeben sich dann 23 Sortierdurchgänge.

Das in d1 angegebene Bit ist während der Sortieroperation zurückgesetzt und wird erst gesetzt, wenn die Sortieroperation abgeschlossen ist. Dieses Bit bleibt gesetzt und muss durch entsprechende Programmierung zurückgesetzt werden.

Die in (d2)+0 und (d2)+1 angegebenen Operanden werden für die systeminterne Verarbeitung während der Sortieroperation genutzt. Diese Operanden sollten daher durch die Programmierung nicht verändert werden.

Wird der Wert in n während der Sortieroperation verändert, erfolgt die Verarbeitung mit der aktuell angegebenen Anzahl von 32-Bit-Daten.

Die Verarbeitung wird mit Rücksetzen der Ausführungsbedingung abgebrochen. Bei erneutem Setzen der Ausführungsbedingung wird mit der Sortieroperation von neuem begonnen.

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Der mit n bzw. 2 x n angegebene Bereich in dem in s1 angegebenen Operanden liegt außerhalb des für die Speicherung vorgesehenen Bereich (Fehlercode 4101).
- Der in s2 angegebene Wert ist 0 oder negativ (Fehlercode 4100).
- Der Wert in d2+0 ist größer als der Wert in n (Fehlercode 4101).
- Der Wert in d2+1 ist größer als der Wert in d2+0 (Fehlercode 4101).

Beispiel

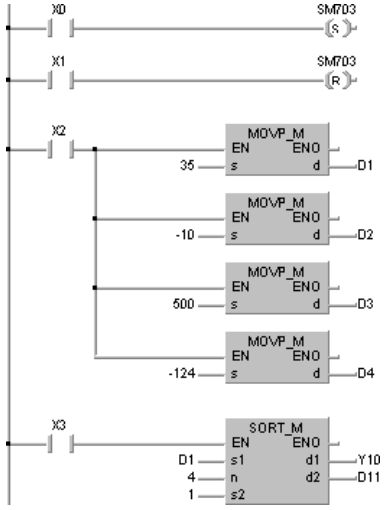
SORT

Das folgende Programm sortiert für die Einschaltdauer von X3 die 16-Bit-Daten in D1 bis D4. Im ersten Schritt werden mit positiver Flanke von X2 die Werte 35, -10, 500 und -124 in die Register D1 bis D4 geladen. Anschließend erfolgt die Sortierung. Die Sortierrichtung wird über X0 (SM703 setzen) und X1 (SM703 rücksetzen) festgelegt. Wenn die Sortieroperation abgeschlossen ist, wird der Ausgang Y10 gesetzt.

MELSEC-Anweisungsliste

```
MELSEC
        LD      X0
        SET     SM703
        LD      X1
        RST     SM703
        LD      X2
        MOVP    K35
                D1
        MOVP    K-10
                D2
        MOVP    K500
                D3
        MOVP    K-124
                D4
        LD      X3
        SORT    D1
                K4
                K1
                Y10
                D11
```

Kontaktplan



IEC-Anweisungsliste

```
        LD      X0
        S       SM703
        LD      X1
        R       SM703
        LD      X2
        MOVP_M  35 , D1
        MOVP_M  -10 , D2
        MOVP_M  500 , D3
        MOVP_M  -124 , D4
        LD      X3
        SORT_M  D1 , 4 , 1 , Y10 , D11
```

7.5.13 WSUM, WSUMP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

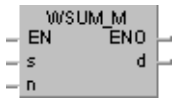
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	—	—	—	—	—	SM0	4
d	●	●	●	●	●	●	●	—	—		
n	●	●	●	●	●	●	●	●	—		

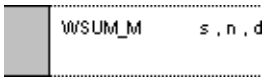
GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

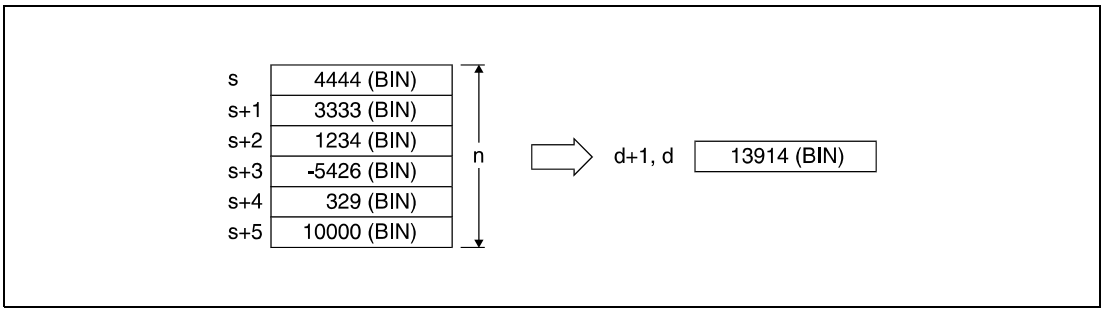
Operand	Befehlswert	Datentyp
s	Erste Adresse des Operanden, in dem die zu addierenden Daten gespeichert sind.	BIN-16-Bit
d	Erste Adresse des Operanden, in dem das Ergebnis gespeichert wird.	BIN-32-Bit
n	Anzahl der zu addierenden Datenblöcke.	BIN-16-Bit

Funktionsweise

Summenbildung mit 16-Bit-Binärdaten

WSUM Anweisung zur Summenbildung

Die WSUM-Anweisung bildet die Summe der mit n angegebenen Anzahl von 16-Bit-Binärdatenblöcken in dem in s angegebenen Operanden. Das Ergebnis wird in dem in d angegebenen Operanden gespeichert.



Fehlerquellen

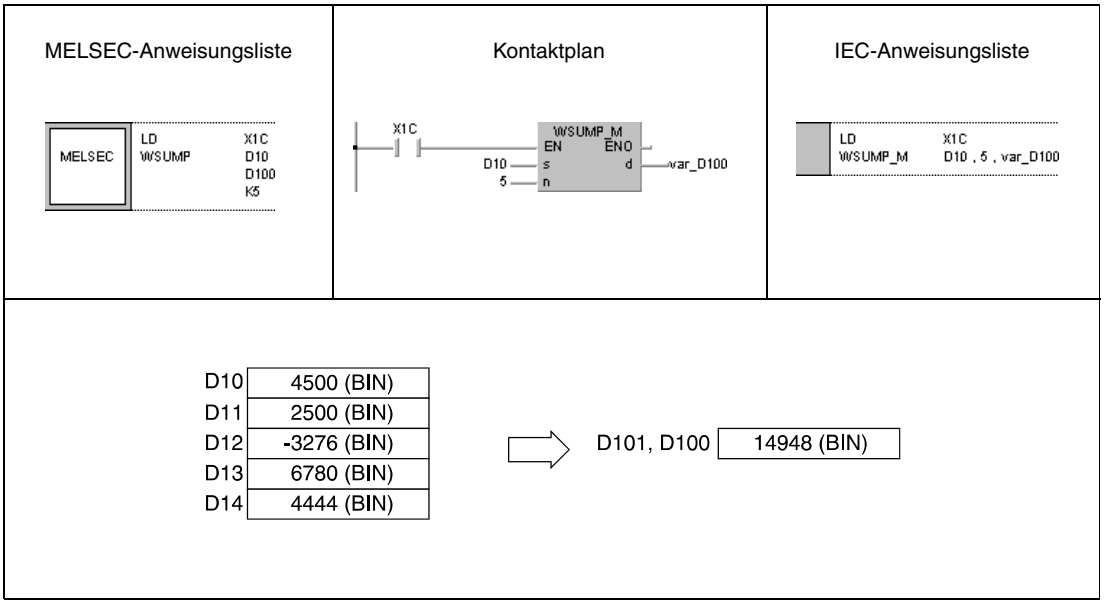
In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Der mit n angegebene Bereich in dem in s angegebenen Operanden liegt außerhalb des für die Speicherung vorgesehenen Bereichs (Fehlercode 4101).

Beispiel

WSUMP

Das folgende Programm addiert mit positiver Flanke von X1C die 16-Bit-Binärdatenblöcke in D10 bis D14 und speichert das Ergebnis in D100 und D101 ab.



7.5.14 DWSUM, DWSUMP

CPU

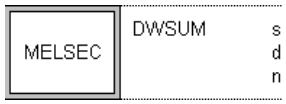
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Operanden MELSEC Q

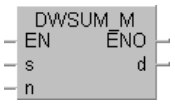
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	—	—	—	—	—	SM0	4
d	●	●	●	—	—	—	—	—	—		
n	●	●	●	●	●	●	●	●	—		

GX IEC Developer

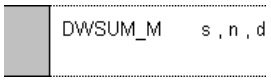
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

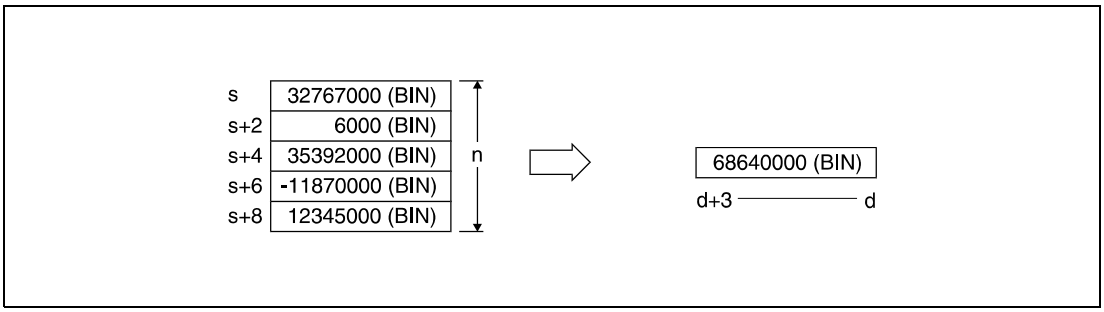
Operand	Befehlswert	Datentyp	
		MELSEC	IEC
s	Erste Adresse des Operanden, in dem die zu addierenden Daten gespeichert sind.	BIN-32-Bit	ANY32
d	Erste Adresse des Operanden, in dem das Ergebnis gespeichert wird.	BIN-64-Bit	Array [1..4] of ANY16
n	Anzahl der zu addierenden Datenblöcke.	BIN-16-Bit	ANY16

Funktionsweise

Summenbildung mit 32-Bit-Binärdaten

DWSUM Anweisung zur Summenbildung

Die DWSUM-Anweisung bildet die Summe der mit n angegebenen Anzahl von 32-Bit-Binärdatenblöcken in dem in s angegebenen Operanden. Das Ergebnis wird in d (Array_d[1]) bis d+3 (Array_d[4]) angegebenen Operanden gespeichert.



Fehlerquellen

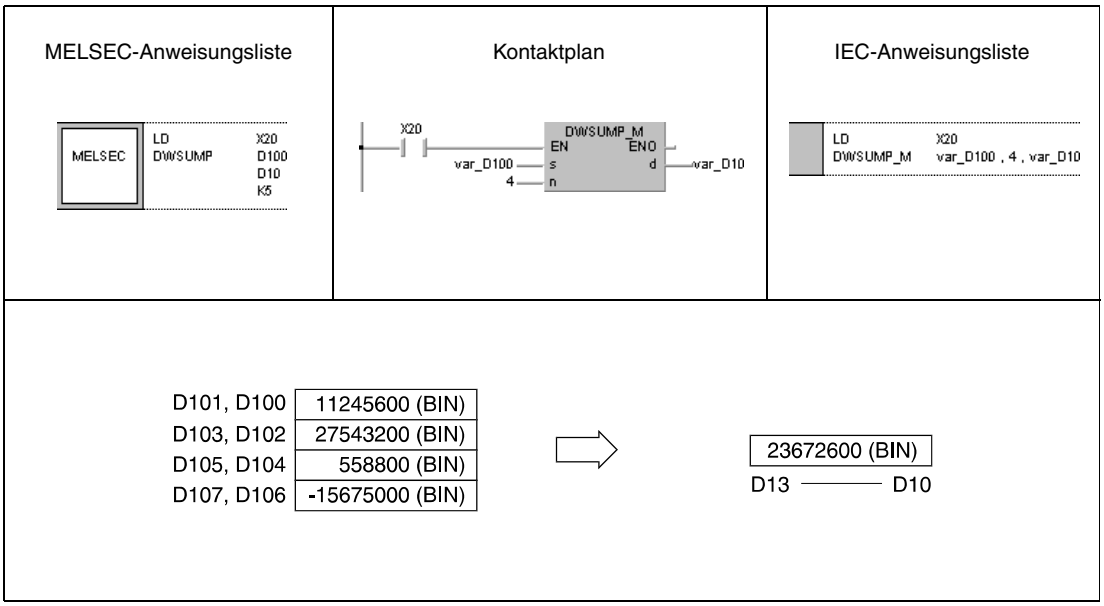
In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Der mit n angegebene Bereich in dem in s angegebenen Operanden liegt außerhalb des für die Speicherung vorgesehenen Bereichs (Fehlercode 4101).

Beispiel

DWSUMP

Das folgende Programm addiert mit positiver Flanke von X20 die 32-Bit-Binärdatenblöcke in D100 bis D107 und speichert das Ergebnis in D10 bis D13 ab.



HINWEIS

Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung beim GX IEC Developer“ dieser Programmieranleitung zu entnehmen.

7.6 Strukturierte Programmanweisungen

Die strukturierten Programmanweisungen bieten die Möglichkeit, Programme und Programmbereiche aufzurufen oder zwischen ihnen zu wechseln. Ein weiterer Bestandteil der Programmanweisungen sind die Anweisungen zur indizierten Adressierung und Wiederholungsanweisungen.

Die folgende Tabelle enthält eine Übersicht sämtlicher Anweisungen.

Funktion	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor
Wiederholungsanweisungen	FOR	FOR_M
	NEXT	NEXT_M
	BREAK	BREAK_MD
	BREAKP	BREAK_P_MD
Unterprogrammaufruf	CALL	CALL_M
	CALLP	CALLP_M
	RET	RET_M
	FCALL	FCALL_MD
	FCALLP	FCALL_P_MD
Unterprogrammaufruf zwischen Programmdateien (nur möglich im GX Developer)	ECALL	ECALL_M
	ECALLP	ECALLP_M
	EFCALL	EFCALL_M
	EFCALLP	EFCALLP_M
Umschaltung zwischen Main- und Sub-Programmbereich	CHG	CHG_M
Mikrocomputer-Programmaufruf	SUB	SUB_M
	SUBP	SUBP_M
Indizierte Adressierung eines gesamten Programmteils	IX	IX_MD
	IXEND	IXEND_MD
Speicherung indizierter Operandenadressen in einer Index-Liste	IXDEV	IXDEV_M
	IXSET	IXSET_M

7.6.1 FOR, NEXT

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
n								●	●	●	●	●	●	●	●	●	●	●					3/1 ●1			●

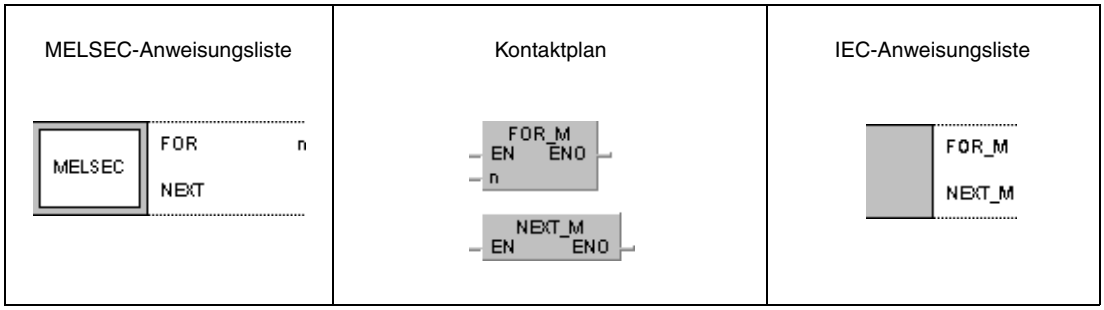
[1] Die FOR-Anweisung benötigt drei Schritte und die NEXT-Anweisung einen Schritt.
Die Anzahl der Schritte bei der Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
n	●	●	●	●	●	●	●	●	—	SM0	2/1 ●1

[1] Die FOR-Anweisung benötigt zwei Schritte und die NEXT-Anweisung einen Schritt.

GX IEC Developer



GX Developer

Variablen

Operand	Befehlswert	Datentyp
n	Anzahl der Durchläufe der FOR-NEXT-Schleife (zwischen 1 und 32767)	BIN-16-Bit

Funktions-weise

Anweisungsschleife FOR bis NEXT

FOR, NEXT Anweisungskombination

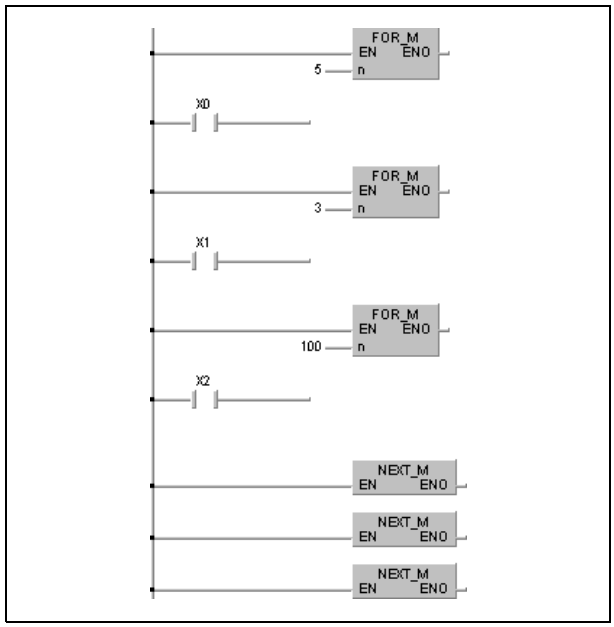
Die FOR-NEXT-Anweisungskombination ermöglicht es, einzelne Programmsequenzen ohne Vorgabe einer Eingangsbedingung zu wiederholen. Wiederholt wird dabei der zwischen FOR und NEXT befindliche Programmteil. Die Anzahl der Wiederholungen wird in n festgelegt.

Nach n-facher Ausführung der FOR-NEXT- Anweisungskombination wird der nächste Schritt nach der NEXT-Anweisung ausgeführt.

Für n kann ein Wert zwischen 1 und 32767 vorgegeben werden. Ist in n eine 0 oder ein negativer Wert vorgegeben, wird n gleich 1. Damit wird die Schleife mindestens einmal durchlaufen.

Soll die Programmsequenz zwischen FOR und NEXT nicht zur Ausführung gebracht werden, kann die FOR-NEXT-Schleife mit einer Sprunganweisung (CJ oder SCJ) übersprungen werden.

Insgesamt können bis zu 16 Ebenen (A-Serie = 5 Ebenen) der FOR-Anweisung ineinander verschachtelt werden. Die folgende Abbildung stellt das Prinzip der Verschachtelung dar.



Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt :

- Die END-/FEND-Anweisung wird nach Ausführung der FOR-Anweisung, aber vor Verarbeitung der NEXT-Anweisung ausgeführt (Q-Serie/System Q = Fehlercode 4200).
- Die NEXT-Anweisung wird vor der FOR-Anweisung ausgeführt (Q-Serie/System Q = Fehlercode 4201).
- Die Anzahl von FOR-Anweisungen stimmt nicht mit der Anzahl von NEXT-Anweisungen überein.
- Eine JMP-Anweisung wird innerhalb der FOR-NEXT-Schleife ausgeführt, deren Sprungziel außerhalb der FOR-NEXT-Schleife liegt.
- Eine STOP-Anweisung liegt innerhalb der FOR-NEXT-Schleife (Q-Serie/System Q = Fehlercode 4200).
- Die zulässige Verschachtelungstiefe wird überschritten (Q-Serie/System Q = Fehlercode 4202).

HINWEISE *Um die Ausführung der Anweisungsschleife FOR bis NEXT vor dem Beenden der Schleife abzubrechen, müssen Sie eine BREAK-Anweisung programmieren (Nur Q-Serie und System Q).*

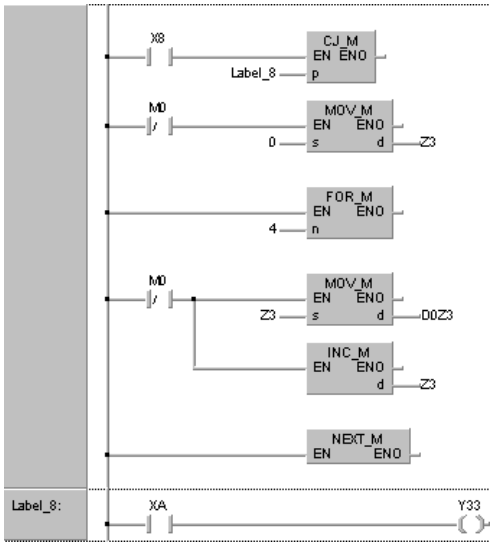
Verwenden Sie die EGP/EGF-Anweisung, um die FOR-NEXT-Anweisung mit einer Schaltbedingung zu verknüpfen (Nur Q-Serie).

Beispiel Das folgende Programm führt den Programmteil zwischen FOR und NEXT 4 mal hintereinander aus, wenn X8 nicht gesetzt ist. Die FOR-NEXT-Schleife wird übersprungen, wenn X8 gesetzt ist.

MELSEC-Anweisungsliste

```
MELSEC   LD    X8
         CJ    Label_8
         LDI   M0
         MOV   K0
               Z3
         FOR   K4
         LDI   M0
         MOV   Z3
               D0Z3
         INC   Z3
         NEXT
Label_8: LD    XA
MELSEC   OUT   Y33
```

Kontaktplan



IEC-Anweisungsliste

```
         LD    X8
         JMPC  Label_8
         LDN   M0
         MOV_M 0 , Z3
         FOR_M 4
         LDN   M0
         MOV_M Z3 , D0Z3
         INC_M Z3
         NEXT_M
Label_8: LD    XA
         ST    Y33
```

7.6.2 BREAK, BREAKP

CPU

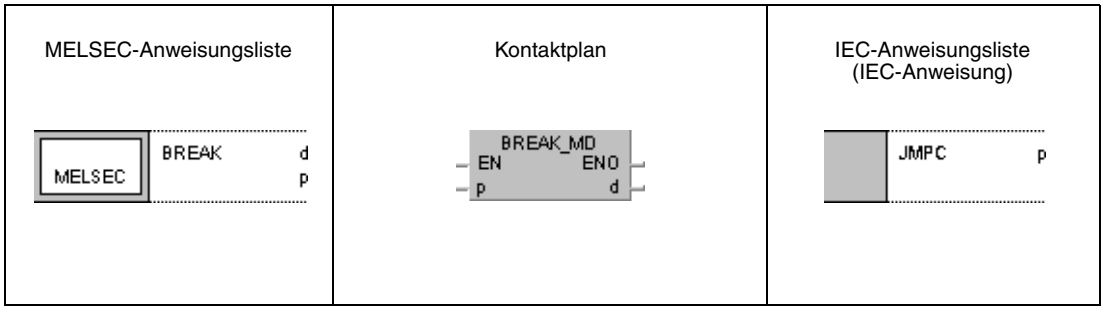
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●

[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sonder-module U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort				P		
d	●	●	●	●	●	●	●	—	—	SM0	3
p	●	●	●	●	●	●	●	—	●		

GX IEC Developer



GX Developer

Variablen

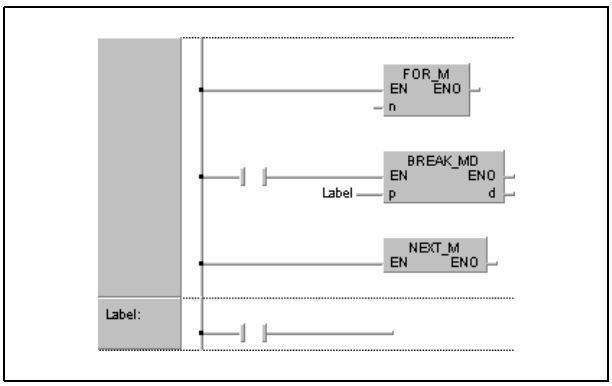
Operand	Befehlswert	Datentyp
d	Operand, in dem die verbleibende Anzahl von FOR-NEXT-Schleifendurchläufen gespeichert wird.	BIN-16-Bit
p	Adresse (Pointer/Label), zu der nach Ausführung der Anweisung gesprungen wird.	Pointer/Label

Funktions-weise

Beenden der FOR-NEXT-Schleife während der Ausführung

BREAK Beenden der FOR-NEXT-Ausführung

Die BREAK-Anweisung unterbricht die FOR-NEXT-Schleife während der Ausführung und springt zu dem in p angegebenen Pointer/Label.



Die Anzahl der noch auszuführenden FOR-NEXT-Schleifen wird in dem in d angegebenen Operanden gespeichert. Bei dem in d gespeicherten Wert wird die Anzahl der Ausführungen der BREAK-Anweisungen berücksichtigt.

Die BREAK-Anweisung kann nur während der Ausführung einer FOR-NEXT-Schleife verwendet werden.

Die BREAK-Anweisung kann nur bei einer Verschachtelungsebene verwendet werden. Bei mehreren Verschachtelungsebenen ist eine äquivalente Anzahl von BREAK-Anweisung auszuführen.

Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die BREAK-Anweisung wurde ohne FOR-NEXT-Schleife verwendet (Fehlercode 4203).
- An dem angegebenen Pointer/Label ist keine Unterprogrammroutine vorhanden. (Fehlercode 4210).

Beispiel BREAKP

Das folgende Programm bricht bei der 30-sten FORT-NEXT-Schleife die Ausführung ab, und springt zu dem mit Label_0 bezeichneten Programmteil. Die Anzahl der noch verbleibenden Ausführungen der FOR-NEXT-Schleifen (70) wird in D1 gespeichert.

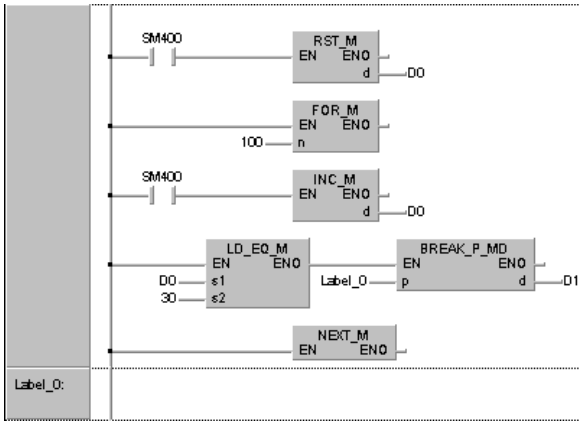
MELSEC-Anweisungsliste

```
MELSEC    LD      SM400
          RST     D0
          FOR     K100
          LD      SM400
          INC     D0
          LD=     D0
                  K30
          BREAKP  D1
                  Label_0
          NEXT
Label_0:
MELSEC
```

IEC-Anweisungsliste

```
          LD      SM400
          RST_M   D0
          FOR_M   100

          LD      SM400
          INC_M   D0
          LD_EQ_M D0 , 30
          BREAK_P_MD Label_0 , D1
          NEXT_M
Label_0:
```

Kontaktplan



7.6.3 CALL, CALLP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
p																			●				3 ●[1]			●

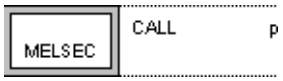
[1] Die Anzahl der Schritte bei der Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

Operanden MELSEC Q

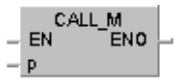
	Operanden								Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□N□		Sondermodule U□G□	Index-Register Zn	Konstanten	Andere	
	Bit	Wort		Bit	Wort					
p	—	●	●	—	—	—	—	—	—	SM0	2

GX IEC Developer

MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste

GX Developer

Variablen

Operand	Befehlswert	Datentyp
pn	Adresse (Pointer/Label) des Unterprogramms.	Pointer/Label

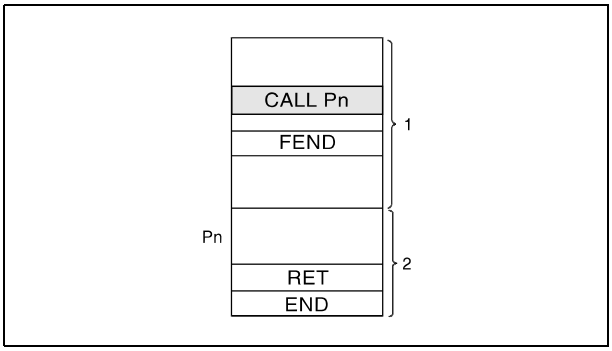
HINWEIS *Die CALL-Anweisung sollte nicht in den IEC-Editoren verwendet werden, da die Unterprogrammstruktur vom GX IEC Developer erzeugt wird.*

Funktionsweise

Aufruf einer Unterprogrammroutine

CALL Unterprogrammroutinenaufruf

Der Aufruf einer Unterprogrammroutine mit Hilfe einer CALL-Anweisung erfolgt durch Angabe der Pointer-Adresse Pxx der Unterprogrammroutine in MELSEC MEDOC bzw. durch Angabe des Labels der Unterprogrammroutine in GX IEC Developer. Die Pointer-Adresse (Label) kann bei der A-Serie zwischen P(Label)0 und P(Label)255 und bei der Q-Serie bzw. dem System Q zwischen P(Label)0 und P(Label)4095 liegen. Die Programmierung der Pointer-Adresse (Label) erfolgt entsprechend den Hinweisen zu den Sprunganweisungen (siehe CJ, SCJ, JMP).



[1] Hauptprogrammroutine

[2] Unterprogrammroutine

Die CALL-Anweisung ruft die mittels Pointer-Adresse (Label) angegebene Unterprogrammroutine auf. Insgesamt können bei Programmierung der CALL-Anweisung bis zu 5 Unterprogrammverschachtelungen bei der A-Serie bzw. 16 Unterprogrammverschachtelungen bei der Q-Serie und dem System Q vorgenommen werden.

Operanden, die während der Ausführung einer Unterprogrammroutine gesetzt worden sind, bleiben auch dann gesetzt, wenn diese Routine nicht mehr ausgeführt wird. Um diese Operanden zurückzusetzen, ist die FCALL-Anweisung zu verwenden.

Fehlerquellen

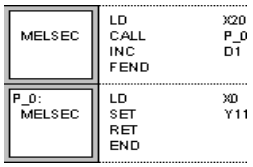
In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Nach der Ausführung einer CALL-Anweisung wird eine END-, FEND-, GOEND- oder STOP-Anweisung ausgeführt, ohne dass zuvor eine RET-Anweisung ausgeführt wird (Q-Serie/System Q = Fehlercode 4211).
- Eine RET-Anweisung wird vor einer CALL-Anweisung ausgeführt (Q-Serie/System Q = Fehlercode 4212).
- Es werden mehr als 5 Verschachtelungen (A-Serie) bzw. mehr als 16 Verschachtelungen (Q-Serie/System Q) ausgeführt (Q-Serie/System Q = Fehlercode 4213).
- An dem angegebenen Pointer/Label ist keine Unterprogrammroutine vorhanden (Q-Serie/System Q = Fehlercode 4210).
- In einer CALL-Anweisung wird eine Pointer-Adresse (Label) oberhalb P(Label)255 angegeben (A-Serie).
- Die Unterroutine wird mit einer JMP-Anweisung vor Ausführung der RET-Anweisung verlassen (A-Serie).

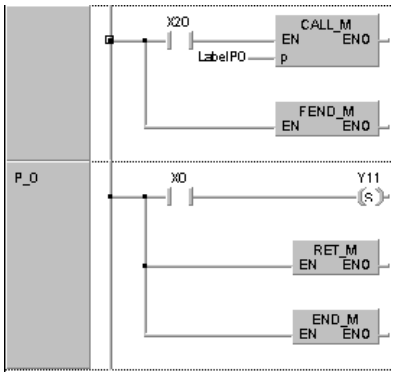
Beispiel CALL

Im folgenden Beispiel wird für die Einschaltdauer von X20 die Unterprogrammroutine an dem Pointer/Label P_0 ausgeführt.

MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste

HINWEISE *Im MELSEC-Modus müssen die FEND-, END- und RET-Anweisungen vom Benutzer programmiert werden. Nachdem die entsprechende Programm-Organisationseinheit abgearbeitet wurde, wird keine weitere mehr ausgeführt, da sie sich nach der FEND-, END- oder RET-Anweisung befindet.*

Eine Alternative zu dieser Programmierung ist die Programmierung im IEC-Editor. In diesem Fall werden die FEND-, END- und RET-Anweisungen von GX IEC Developer automatisch eingesetzt.

7.6.4 RET

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
																							1 ●[1]			●

[1] Die Anzahl der Schritte bei der Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

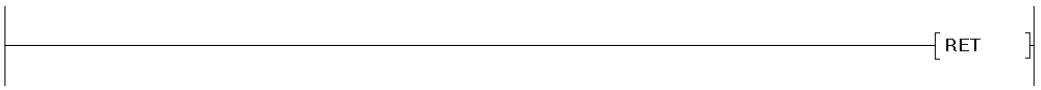
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sonder-module U□\G□	Index-Register Zn	Konstanten	Andere		
	Bit	Wort		Bit	Wort						
—	—	—	—	—	—	—	—	—	—	SM0	1

GX IEC Developer

MELSEC-Anweisungsliste	Kontaktplan	IEC-Anweisungsliste
		

GX Developer



Variablen

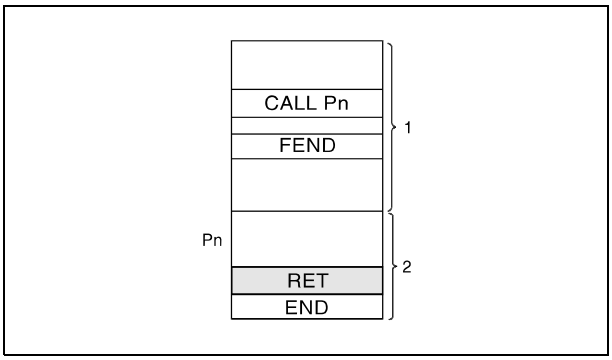
Operand	Befehlswert	Datentyp
—	—	—

Funktions-weise

Ende einer Unterprogrammroutine

RET Rücksprung zum Hauptprogramm

Mit Hilfe einer RET-Anweisung wird das Ende einer Unterprogrammroutine gekennzeichnet. Der Rücksprung in das Hauptprogramm erfolgt zu dem Programmschritt, der der CALL-, FCALL-, ECALL- oder EFCALL-Anweisung folgt.



[1] Hauptprogrammroutine

[2] Unterprogrammroutine

HINWEISE

Zwischen der RET-Anweisung im Unterprogramm und der END-Anweisung im Hauptprogramm muss immer eine NOP-Anweisung programmiert werden, da die CPU das Programm andernfalls nicht korrekt abarbeitet (nur A-Serie).

In der MELSEC-Anweisungsliste müssen die FEND-, END- und RET-Anweisungen vom Benutzer programmiert werden. Nachdem die entsprechende Programm-Organisationseinheit abgearbeitet wurde, wird keine weitere mehr ausgeführt, da sie sich nach der FEND-, END- oder RET-Anweisung befindet.

Eine Alternative zu dieser Programmierung ist die Programmierung im IEC-Editor. In diesem Fall werden die FEND-, END- und RET-Anweisungen vom GX IEC Developer automatisch eingesetzt.

Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Nach Verarbeitung einer CALL-Anweisung wird eine END-, FEND-, GOEND- oder STOP-Anweisung vor Ausführung der RET-Anweisung ausgeführt (Q-Serie/System Q = Fehlercode 4211).
- Eine RET-Anweisung wird vor einer CALL-Anweisung ausgeführt (Q-Serie/System Q = Fehlercode 4212).

7.6.5 FCALL, FCALLP

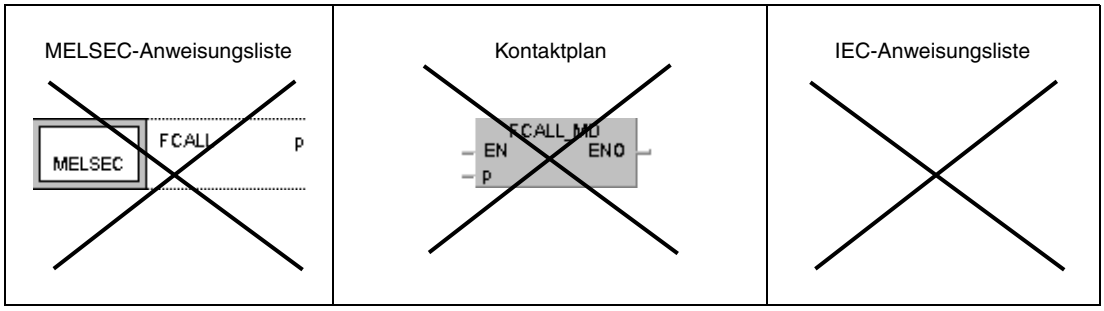
CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●	●	●	●

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten	Andere		
	Bit	Wort		Bit	Wort						
p	—	●	●	—	—	—	—	—	—	SM0	2

GX IEC Developer



GX Developer

Variablen

Operand	Befehlswert	Datentyp
pn	Adresse (Pointer/Label) des Unterprogramms	Pointer/Label

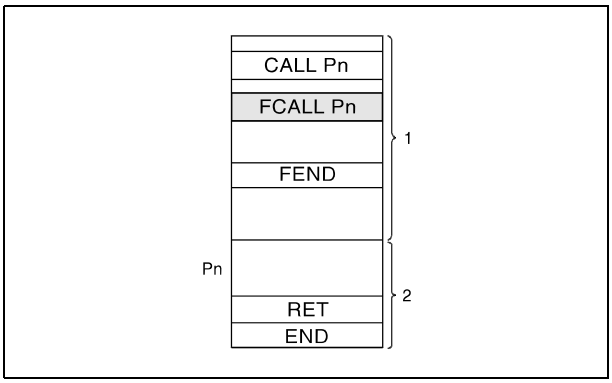
HINWEIS *Diese Anweisungen können nicht im GX IEC Developer programmiert werden.*

Funktions-weise

Rücksetzen der Ausgänge in Unterprogrammroutinen

FCALL Rücksetzen der Ausgänge (Verwendung mit der CALL-Anweisung)

Mit dem Rücksetzen der Ausführungsbedingung der FCALL-Anweisung werden die Kontakte und Spulen der in p (Pointer/Label) angegebenen Unterprogrammroutine in den Zustand versetzt, als ob die entsprechende Ausführungsbedingung der jeweiligen Anweisung, die einen Kontakt oder eine Spule anspricht, nicht gesetzt ist.



[1] Hauptprogrammroutine

[2] Unterprogrammroutine

Der Zustand der Spulen und Kontakte nach Ausführung der FCALL-Anweisung bzw. der Zustand, den die Spulen und Kontakte annehmen, wenn die Ausführungsbedingung der entsprechenden Anweisung nicht gesetzt ist, wird nachfolgend dargestellt:

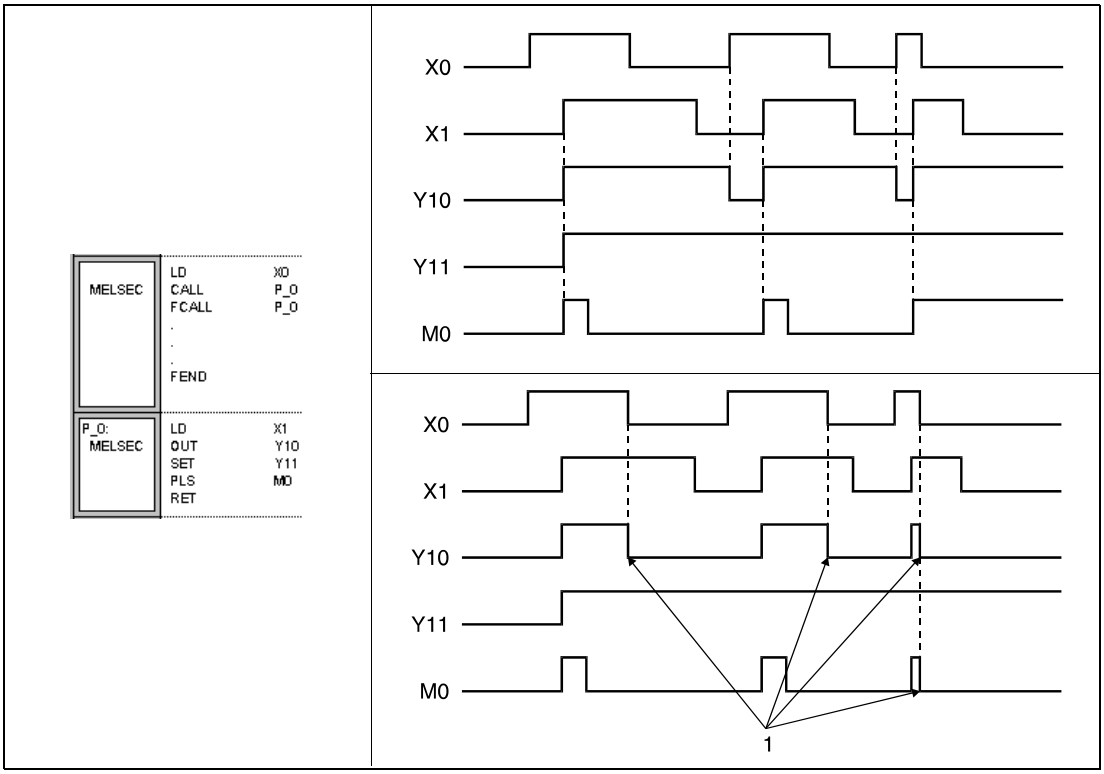
Anweisungen	Zustand der Kontakte und Spulen
OUT-Anweisung	Alle von der OUT-Anweisung angesprochenen Kontakte und Spulen werden zurückgesetzt.
SET-Anweisung	Die von den Anweisungen angesprochenen Kontakte und Spulen behalten ihren Zustand.
RST-Anweisung	
SFT-Anweisung	
Applikationsanweisungen Teil 1	
Applikationsanweisungen Teil 2	
PLS-Anweisung	Die von diesen Anweisungen angesprochenen Kontakte und Spulen nehmen den Zustand an, als ob die Ausführungsbedingungen der Anweisungen nicht gesetzt wären.
Anweisungen mit Erzeugung eines Ausgangsimpulses	
Einstellwerte langsamer und schneller Timer	Die Einstellwerte werden auf 0 gesetzt.
Einstellwerte remanenter Timer	Die Einstellwerte bleiben erhalten.
Einstellwerte von Countern	

Die FCALL-Anweisung wird in Verbindung mit der CALL-Anweisung verwendet.

In der folgenden Darstellung wird ein Programm abgebildet, das die CALL- und FCALL-Anweisung verwendet. Die Abbildungen auf der rechten Seite zeigen die Signalzustandsdiagramme verschiedener Kontakte, die von unterschiedlichen Anweisungen angesprochen werden. Die Abbildung oben rechts zeigt den Zustand dieser Kontakte ohne Verwendung der FCALL-Anweisung und die Abbildung unten rechts mit Verwendung der FCALL-Anweisung.

Bei alleiniger Verwendung der CALL-Anweisung werden die Zustände der in der Unterprogrammroutine angesprochenen Kontakte und Spulen nach Rücksetzen der Ausführungsbedingung der CALL-Anweisung beibehalten (siehe Abbildung oben rechts).

Durch die Verwendung der FCALL-Anweisung werden die Zustände der in der Unterprogrammroutine angesprochenen Kontakte und Spulen nach Rücksetzen der Ausführungsbedingung der FCALL-Anweisung zurückgesetzt (siehe Abbildung unten rechts). Das gilt auch für Spulen und Kontakte, die durch eine OUT-, PLS-Anweisung oder eine Anweisung zur Erzeugung von Impulsen angesprochen werden.



[1] An diesen Punkten werden die Kontakte zurückgesetzt

Die FCALL-Anweisung ruft die mittels Pointer-Adresse (Label) angegebene Unterprogrammroutine auf. Insgesamt können bei Programmierung der FCALL-Anweisung bis zu 16 Unterprogrammverschachtelungen vorgenommen werden.

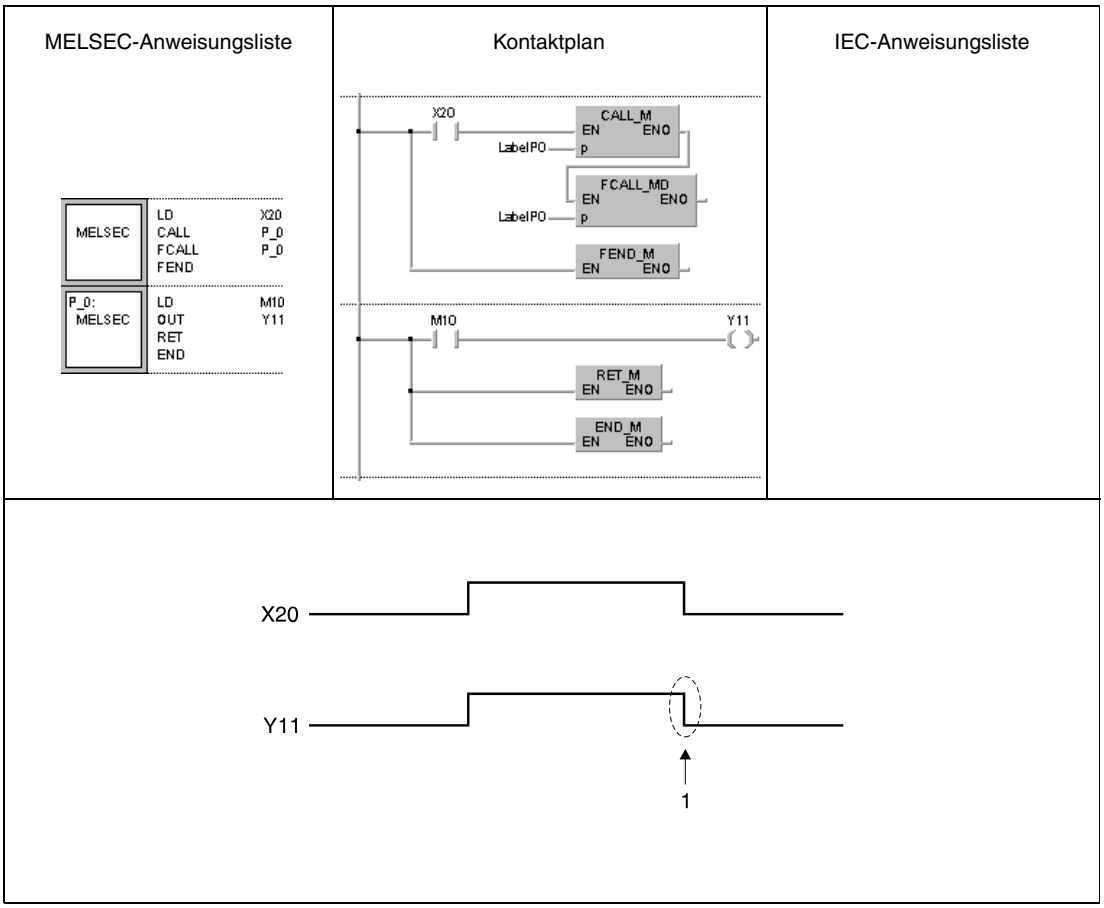
Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Nach der Ausführung einer FCALL-Anweisung wird eine END-, FEND-, GOEND- oder STOP-Anweisung ausgeführt, ohne daß zuvor eine RET-Anweisung ausgeführt wird (Fehlercode 4211).
- Eine RET-Anweisung wird vor einer FCALL-Anweisung ausgeführt (Fehlercode 4212).
- Es werden mehr als 16 Verschachtelungen ausgeführt (Fehlercode 4213).
- An dem angegebenen Pointer/Label ist keine Unterprogrammroutine vorhanden (Fehlercode 4210).

Beispiel FCALL

Im folgenden Beispiel wird für die Einschaltdauer von X20 die Unterprogrammroutine an der Pointer-Adresse (Label) P_0 ausgeführt. Durch Verwendung der FCALL-Anweisung wird mit Zurücksetzen von X20 auch der Ausgang Y11 zurückgesetzt (1).



7.6.6 ECALL, ECALLP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●[1]

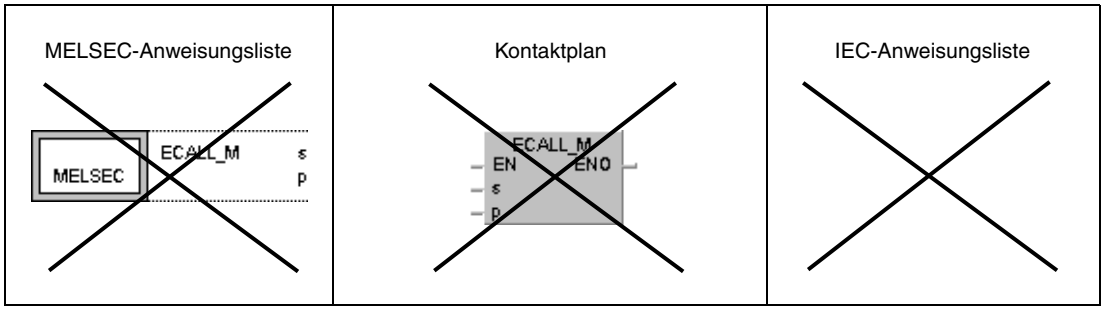
[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□N□		Sondermodule U□\G□	Index-Register Zn	Konstanten	Andere		
	Bit	Wort		Bit	Wort						
p	—	—	—	—	—	—	—	—	●	SM0	3
s1	●[1]	●	●	●	●	●	●	●	—		
s2	●[1]	●	●	●	●	●	●	●	—		
s3	●[1]	●	●	●	●	●	●	●	—		
s4	●[1]	●	●	●	●	●	●	●	—		
s5	●[1]	●	●	●	●	●	●	●	—		

[1] Fehlermerker (F) können nicht verwendet werden

GX IEC Developer



GX Developer

Variablen

Operand	Befehlswert	Datentyp
Dateiname	Name der Unterprogrammdatei, in der sich die aufzurufende Unterprogrammroutine befindet.	Zeichenfolge
pn	Erste Adresse (Pointer/Label) der Unterprogrammroutine.	Pointer/Label
s1 bis s5	Operanden, die dem Unterprogramm übergeben werden	Bit BIN-16-Bit BIN-32-Bit

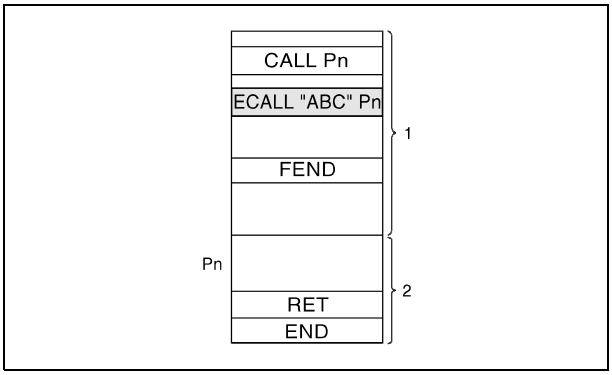
HINWEIS *Diese Anweisungen können nicht im GX IEC Developer programmiert werden.*

Funktionsweise

Aufruf einer Unterprogrammroutine in einer Programmdatei

ECALL Unterprogrammroutinenaufruf

Der Aufruf einer Unterprogrammroutine in einer Programmdatei mit Hilfe einer ECALL-Anweisung erfolgt durch Angabe des Dateinamens der Programmdatei und durch Angabe des Labels der Unterprogrammroutine. Die Pointer-Adresse (Label) kann zwischen P(Label)0 und P(Label)4095 liegen. Die Programmierung der Pointer-Adresse (Label) erfolgt entsprechend den Hinweisen zu den Sprunganweisungen (siehe CJ, SCJ, JMP).



[1] Hauptprogrammroutine in der Programmdatei: "Haupt"

[2] Unterprogrammroutine in der Programmdatei: "ABC"

Als Dateinamen können nur Namen von Programmdateien angegeben werden, die im internen Speicher (Laufwerk 0) abgelegt sind.

Beim Aufruf der Programmdateien mit Dateinamen brauchen keine Endungen angegeben werden.

Die ECALL-Anweisung ruft die mittels Pointer-Adresse (Label) angegebene Unterprogrammroutine auf. Insgesamt können bei Programmierung der ECALL-Anweisung bis zu 16 Unterprogrammverschachtelungen vorgenommen werden.

Operanden, die während der Ausführung einer Unterprogrammroutine gesetzt worden sind, bleiben auch dann gesetzt, wenn diese Routine nicht mehr ausgeführt wird. Um diese Operanden zurückzusetzen, ist die EFCALL-Anweisung zu verwenden.

Falls von dem Unterprogramm Funktionseingänge, -ausgänge oder -register (FX, FY und FD) benutzt werden, können mit den Variablen s1 bis s5 Operanden übergeben werden. Vor der Ausführung des Unterprogrammes werden Bit-Operanden an Funktionseingänge und Wort-Operanden an Funktionsregister übertragen. Nach der Bearbeitung des Unterprogrammes werden die Zustände bzw. Inhalte von FY und FD an die angegebenen Operanden übertragen.

Die Daten, die ein Funktionsregister aufnehmen kann, hängt von den in s1 bis s5 angegebenen Operanden ab: Bei Konstanten, Indexregistern und blockweise adressierten Bit-Operanden können max. 2 Worte, bei Wortoperanden können max. 4 Worte aufgenommen werden. Wird z.B. in s2 der Operand D0 angegeben, werden D0, D1, D2 und D3 in FD1 gespeichert.

Die Anzahl der Funktionseingänge, -ausgänge und -register, die vom Unterprogramm verwendet werden, muss mit den beim Aufruf der ECALL-Anweisung in s1 bis s5 übergebenen Operanden übereinstimmen.

Die Funktionsoperanden müssen mit den Typen der von der ECALL-Anweisung übergebenen Operanden identisch sein.

Die in s1 bis s5 angegebenen Operanden dürfen sich nicht überlappen.

Fehler-quellen

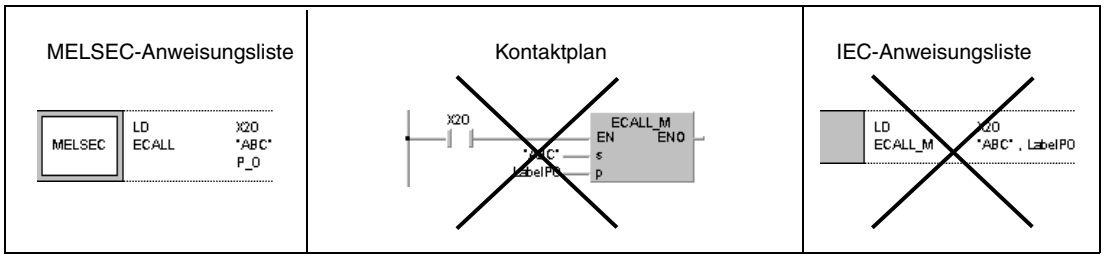
In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Nach der Ausführung einer ECALL-Anweisung wird eine END-, FEND-, GOEND- oder STOP-Anweisung ausgeführt, ohne dass zuvor eine RET-Anweisung ausgeführt wird (Fehlercode 4211).
- Eine RET-Anweisung wird vor einer ECALL-Anweisung ausgeführt (Fehlercode 4212).
- Es werden mehr als 16 Verschachtelungen ausgeführt (Fehlercode 4213).
- An dem angegebenen Pointer/Label ist keine Unterprogrammroutine vorhanden (Fehlercode 4210).
- In s1 bis s5 wurde ein Funktionseingang (FX), Funktionsausgang (FY) oder Funktionsregister angegeben (Fehlercode 4101).
- Die angegebene Programmdatei existiert nicht (Fehlercode 4210).
- Die angegebene Programmdatei kann nicht ausgeführt werden (Fehlercode 2411).

Beispiel

ECALL

Im folgenden Beispiel wird für Einschaltdauer von X20 die Unterprogrammroutine an dem Pointer/Label P_0 in der Programmdatei "ABC" ausgeführt.



7.6.7 EFCALL, EFCALLP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●[1]

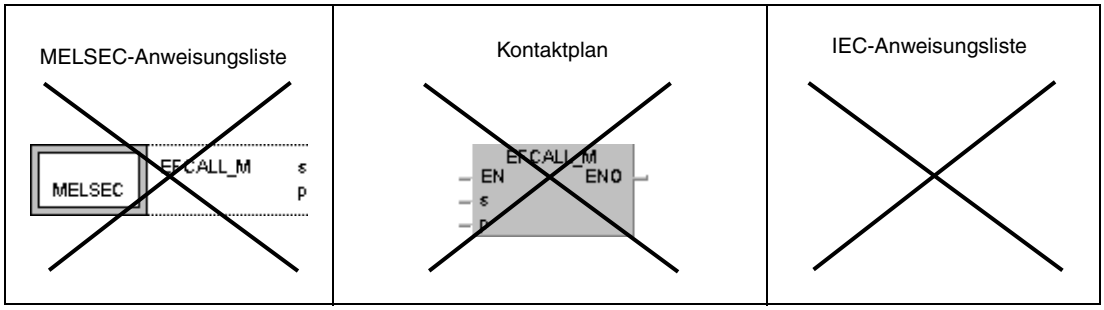
[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten	Andere		
	Bit	Wort		Bit	Wort						
p	—	—	—	—	—	—	—	—	●	SM0	3
s1	●[1]	●	●	●	●	●	●	●	—		
s2	●[1]	●	●	●	●	●	●	●	—		
s3	●[1]	●	●	●	●	●	●	●	—		
s4	●[1]	●	●	●	●	●	●	●	—		
s5	●[1]	●	●	●	●	●	●	●	—		

[1] Fehlermerker (F) können nicht verwendet werden

GX IEC Developer



GX Developer

Variablen

Operand	Befehlswert	Datentyp
Dateiname	Name der Unterprogrammdatei, in der sich die aufzurufende Unterprogrammroutine befindet.	Zeichenfolge
pn	Adresse (Pointer/Label) der Unterprogrammroutine.	Pointer/Label
s1 bis s5	Operanden, die dem Unterprogramm übergeben werden	Bit BIN-16-Bit BIN-32-Bit

HINWEIS *Diese Anweisungen können nicht im GX IEC Developer programmiert werden.*

Funktionsweise

Rücksetzen der Ausgänge in Unterprogrammroutinen in Programmdateien

EFCALL **Rücksetzen der Ausgänge (Verwendung mit der ECALL-Anweisung)**

Mit dem Rücksetzen der Ausführungsbedingung der EFCALL-Anweisung werden die Kontakte und Spulen der in p (Pointer/Label) angegebenen Unterprogrammroutine in der mit dem Dateinamen angegebenen Programmdatei in den Zustand versetzt, als ob die entsprechende Ausführungsbedingung der jeweiligen Anweisung, die einen Kontakt oder eine Spule anspricht, nicht gesetzt ist.

Die EFCALL-Anweisung führt Unterprogrammroutinen aus, die sich innerhalb einer anderen Programmdatei als der befinden, aus der heraus sie aufgerufen werden.

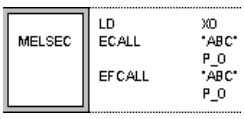
Der Zustand der Spulen und Kontakte nach Ausführung der EFCALL-Anweisung bzw. der Zustand, den die Spulen und Kontakte annehmen, wenn die Ausführungsbedingung der entsprechenden Anweisung nicht gesetzt ist, wird nachfolgend dargestellt:

Anweisungen	Zustand der Kontakte und Spulen
OUT-Anweisung	Alle von der OUT-Anweisung angesprochenen Kontakte und Spulen werden zurückgesetzt.
SET-Anweisung	Die von den Anweisungen angesprochenen Kontakte und Spulen behalten ihren Zustand.
RST-Anweisung	
SFT-Anweisung	
Applikationsanweisungen Teil 1	
Applikationsanweisungen Teil 2	
PLS-Anweisung	Die von diesen Anweisungen angesprochenen Kontakte und Spulen nehmen den Zustand an, als ob die Ausführungsbedingungen der Anweisungen nicht gesetzt wären.
Anweisungen mit Erzeugung eines Ausgangsimpulses	
Einstellwerte langsamer und schneller Timer	Die Einstellwerte werden auf 0 gesetzt.
Einstellwerte remanenter Timer	Die Einstellwerte bleiben erhalten.
Einstellwerte von Countern	

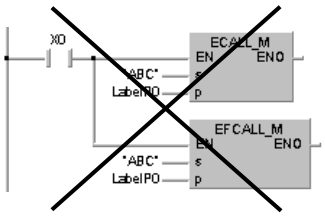
Die EFCALL-Anweisung wird in Verbindung mit der ECALL-Anweisung verwendet.

In der folgenden Darstellung wird ein Programm abgebildet, das die ECALL- und EFCALL-Anweisung verwendet.

MELSEC-Anweisungsliste



Kontaktplan



Falls von dem Unterprogramm Funktionseingänge, -ausgänge oder -register (FX, FY und FD) benutzt werden, können mit den Variablen s1 bis s5 Operanden übergeben werden. Vor der Ausführung des Unterprogrammes werden Bit-Operanden an Funktionseingänge und Wort-Operanden an Funktionsregister übertragen. Nach der Bearbeitung des Unterprogrammes werden die Zustände bzw. Inhalte von FY und FD an die angegebenen Operanden übertragen.

Die Daten, die ein Funktionsregister aufnehmen kann, hängt von den in s1 bis s5 angegebenen Operanden ab: Bei Konstanten, Indexregistern und blockweise adressierten Bit-Operanden können max. 2 Worte, bei Wortoperanden können max. 4 Worte aufgenommen werden. Wird z. B. in s2 der Operand D0 angegeben, werden D0, D1, D2 und D3 in FD1 gespeichert.

Die Anzahl der Funktionseingänge, -ausgänge und -register, die vom Unterprogramm verwendet werden, muss mit den beim Aufruf der EFCALL-Anweisung in s1 bis s5 übergebenen Operanden übereinstimmen.

Die Funktionsoperanden müssen mit den Typen der von der EFCALL-Anweisung übergebenen Operanden identisch sein.

Die in s1 bis s5 angegebenen Operanden dürfen sich nicht überlappen.

Die EFCALL-Anweisung ruft die mittels Pointer-Adresse (Label) angegebene Unterprogrammroutine auf. Insgesamt können bei Programmierung der EFCALL-Anweisung bis zu 16 Unterprogrammverschachtelungen vorgenommen werden.

Fehlerquellen

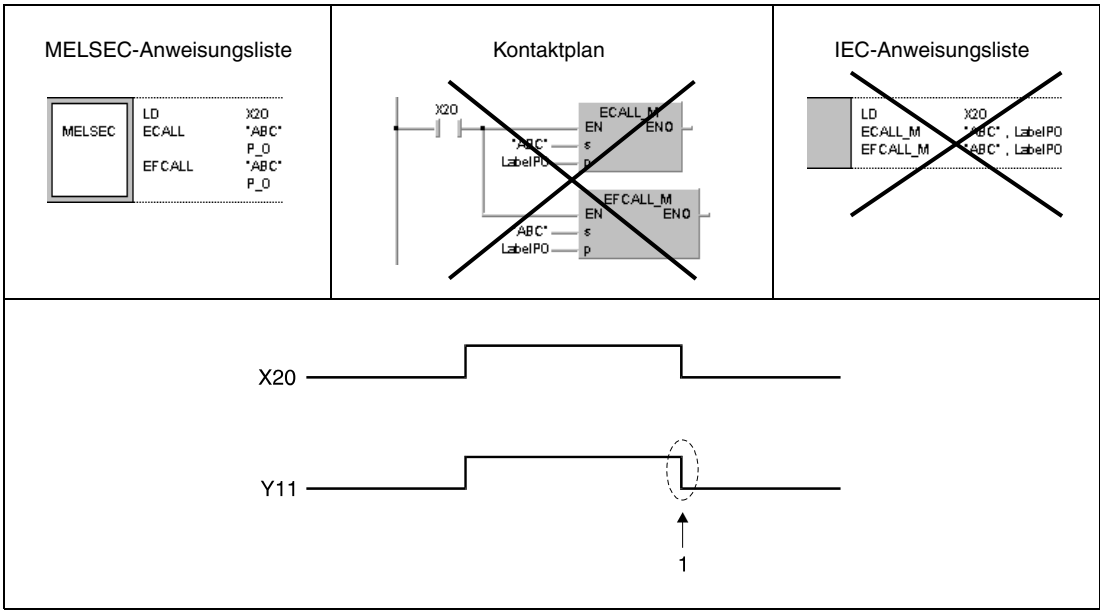
In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Nach der Ausführung einer EFCALL-Anweisung wird eine END-, FEND-, GOEND- oder STOP-Anweisung ausgeführt, ohne dass zuvor eine RET-Anweisung ausgeführt wird (Fehlercode 4211).
- Eine RET-Anweisung wird vor einer EFCALL-Anweisung ausgeführt (Fehlercode 4212).
- Es werden mehr als 16 Verschachtelungen ausgeführt (Fehlercode 4213).
- An dem angegebenen Pointer/Label ist keine Unterprogrammroutine vorhanden (Fehlercode 4210).
- In s1, s2, s3, s4 oder s5 wurde ein Fehlermerker (F) angegeben (Fehlercode 4101).
- Die angegebene Programmdatei existiert nicht (Fehlercode 4210).
- Die angegebene Programmdatei kann nicht ausgeführt werden (Fehlercode 2411).

Beispiel

EFCALL

Im folgenden Beispiel wird die Einschaltdauer von X20 die Unterprogrammroutine an der Pointer-Adresse (Label) P_0 in der Programmdatei "ABC" ausgeführt. Durch Verwendung der EFCALL-Anweisung wird mit Zurücksetzen von X20 auch der Ausgang Y11 zurückgesetzt (1).



7.6.8 CHG

CPU

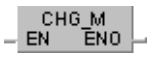
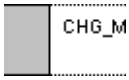
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
	●[1]	●[2]	●[3]		

[1] Nur A3N CPUs
[2] Nur A3A CPUs
[3] Nur A3U CPUs

Operanden MELSEC A

	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
																							1			

GX IEC Developer

MELSEC-Anweisungsliste	Kontaktplan	IEC-Anweisungsliste
		

Variablen

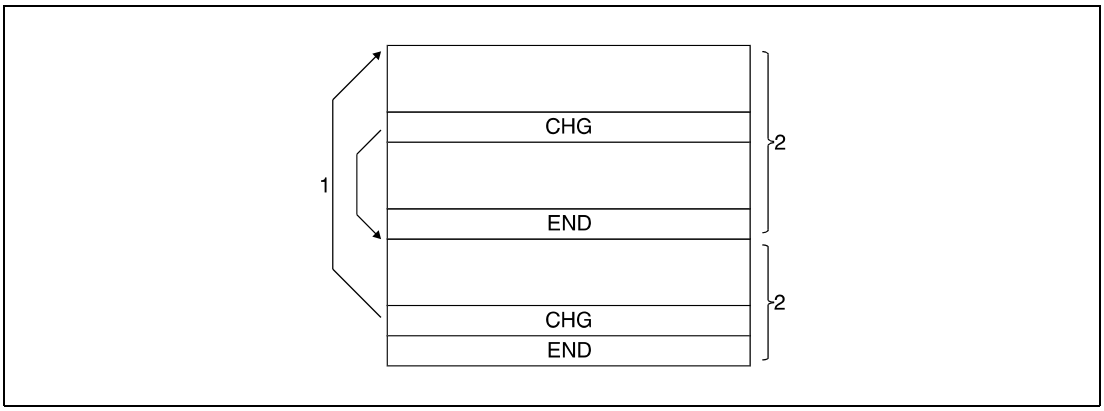
Operand	Befehlswert	Datentyp
—	—	—

Funktions-weise

Umschaltung zwischen MAIN- und SUB-Programmbereich

CHG Umschaltanweisung

Die CHG-Anweisung erlaubt bei gegebener Eingangsbedingung ein Umschalten zwischen dem MAIN- und SUB-Programmbereich. Die Umschaltung erfolgt nach Verarbeitung der Timer und Counter und einer Selbstdiagnose.



[1] Timer-, Counter-Verarbeitung, Selbstdiagnose

[2] Ablaufprogramm

Umschaltung zwischen MAIN- und SUB-Programmbereich

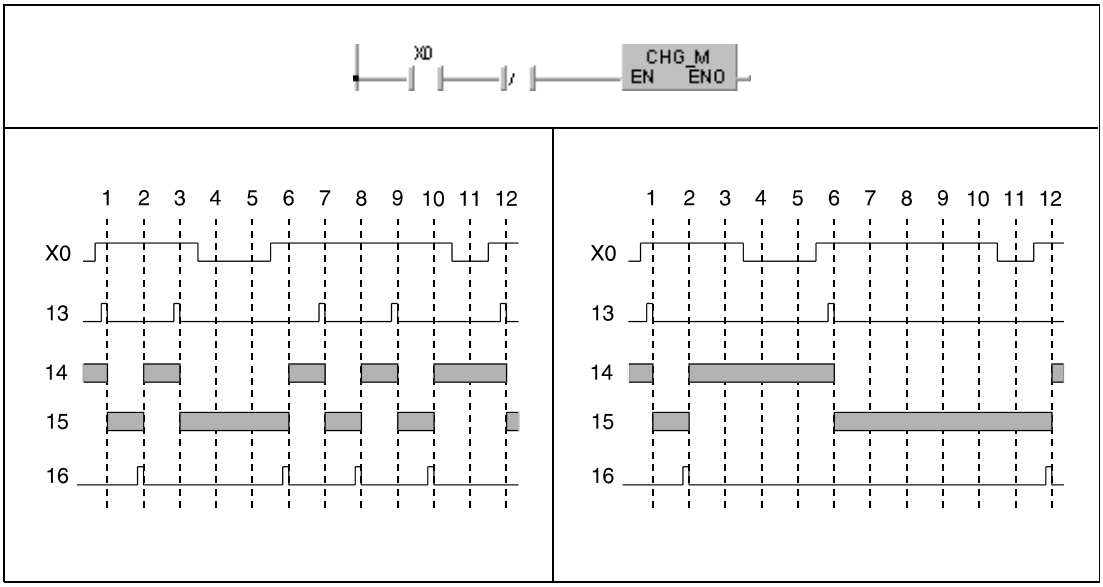
CHG Verwendung einer A3□CPU

Bei Verwendung einer A3□CPU wird die CHG-Anweisung nur bei ansteigender Flanke der Eingangsbedingung ausgeführt. Das Verarbeitungsergebnis der Eingangsbedingung ist vom Zustand des Sondermerkers M9050 abhängig. Die Bedeutung der CHG-Anweisung ändert sich somit mit dem Zustand von M9050.

Der Sondermerker M9050 ist in Verbindung mit einer A3 NCPU nicht einsetzbar. Die A3 NCPU verhält sich so, als sei M9050 gesetzt.

Die folgende Darstellung zeigt im oberen Teil eine programmierte CHG-Anweisung. Dieser Programmteil liegt vor einer END- oder FEND-Anweisung im MAIN- oder SUB-Speicherbereich.

Im unteren Teil sind die entsprechenden Signalverläufe abgebildet. Der linke Signalverlauf gilt bei nicht gesetztem Merker M9050 und der rechte Signalverlauf bei gesetztem Merker M9050. Die auf die Darstellung folgende Tabelle erläutert die Verarbeitung in Abhängigkeit des Einschaltzustandes von X0.



Die Ausführung der CHG-Anweisung im MAIN-Bereich ist mit 13, der MAIN-Bereich mit 14, der SUB-Bereich mit 15 und die Ausführung der CHG-Anweisung mit 16 gekennzeichnet.

Zustand von X0	Zustand von M9050	
	NICHT GESETZT	GESETZT
0	Keine Umschaltung zwischen MAIN- und SUB-Speicherbereich (4, 5, 11).	Keine Umschaltung zwischen MAIN- und SUB-Speicherbereich (4, 5, 11).
1	Die CHG-Anweisung wird mit jedem Programmzyklus ausgeführt und schaltet zwischen MAIN- und SUB-Speicherbereich um (2, 3, 7, 8, 9, 10).	Es wird nur dann vom MAIN- in den SUB-Speicherbereich und wieder zurück in den MAIN-Speicherbereich umgeschaltet, wenn die Eingangsbedingung (positive Flanke von X0) gegeben ist (2).
0 → 1	Umschaltung zwischen MAIN- und SUB-Speicherbereich (1, 6, 12).	Umschaltung zwischen MAIN- und SUB-Speicherbereich (1, 6, 12).

Nach Ausführung der CHG-Anweisung erfolgt die END-Verarbeitung im aktuellen Programm. Die Verarbeitung beginnt mit Schritt 0 des anderen Programms. Der GX IEC Developer schaltet automatisch am Ende des MAIN- und SUB- Speicherbereichs um.

CHG-Anweisung in Verbindung mit einer PLS-Anweisung

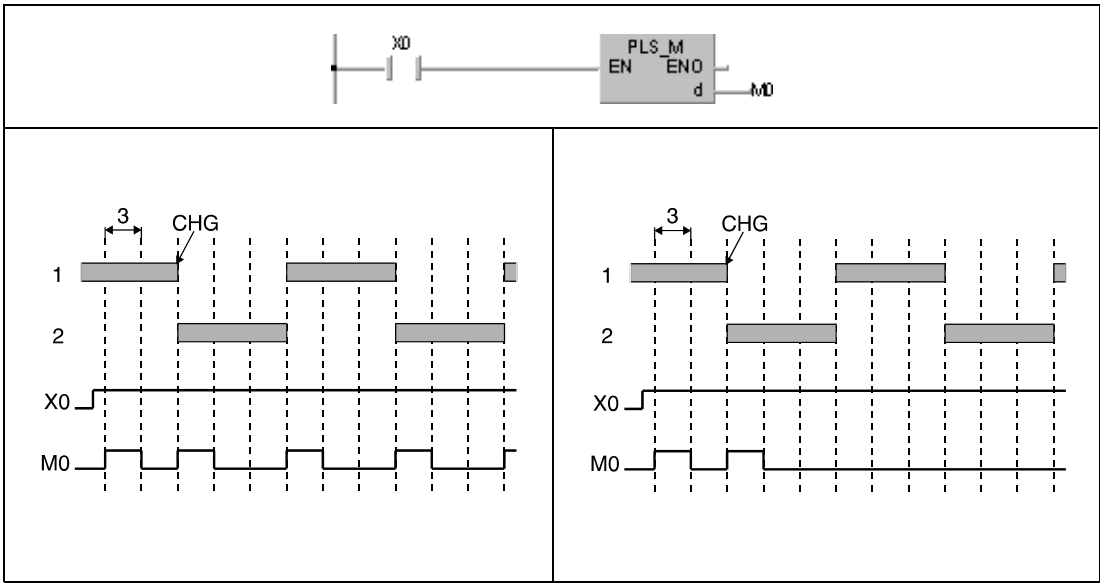
CHG Verwendung einer A3□CPU

Bei Verwendung einer A3□CPU ist die Ausführungsweise der PLS-Anweisung vom Zustand des Sondermerkers M9050 abhängig.

Der Sondermerker M9050 ist in Verbindung mit einer A3 NCPU nicht einsetzbar. Die A3NCPU verhält sich so, als sei M9050 gesetzt.

Die folgende Darstellung zeigt im oberen Teil eine programmierte PLS-Anweisung. Dieser Programmteil liegt am Programmanfang (Schritt 0) des MAIN- oder SUB-Speicherbereichs.

Im unteren Teil sind die entsprechenden Signalverläufe abgebildet. Der linke Signalverlauf gilt bei nicht gesetztem Merker M9050 und der rechte Signalverlauf bei gesetztem Merker M9050. Die auf die Darstellung folgende Tabelle erläutert die Verarbeitung in Abhängigkeit des Einschaltzustandes von X0.



Die Verarbeitung des MAIN-Bereichs ist mit 1, die Verarbeitung des SUB-Bereichs mit 2 und die Dauer eines Zyklus mit 3 gekennzeichnet.

Zustand von X0	Zustand von M9050	
	NICHT GESETZT	GESETZT
0	Kein Setzen von M0.	Kein Setzen von M0.
1	M0 wird nur während des ersten Zyklus nach der Programmumschaltung mittels CHG-Anweisung gesetzt.	M0 wird nach dem Einschalten von X0 nur während des ersten Zyklus des Programms im SUB-Bereich, in das mittels CHG-Anweisung umgeschaltet wurde, gesetzt.
$0 \rightarrow 1$	M0 wird nur für einen Programmzyklus gesetzt.	M0 wird nur für einen Programmzyklus gesetzt.

CHG-Anweisung in Verbindung mit einer gepulsten Anweisung

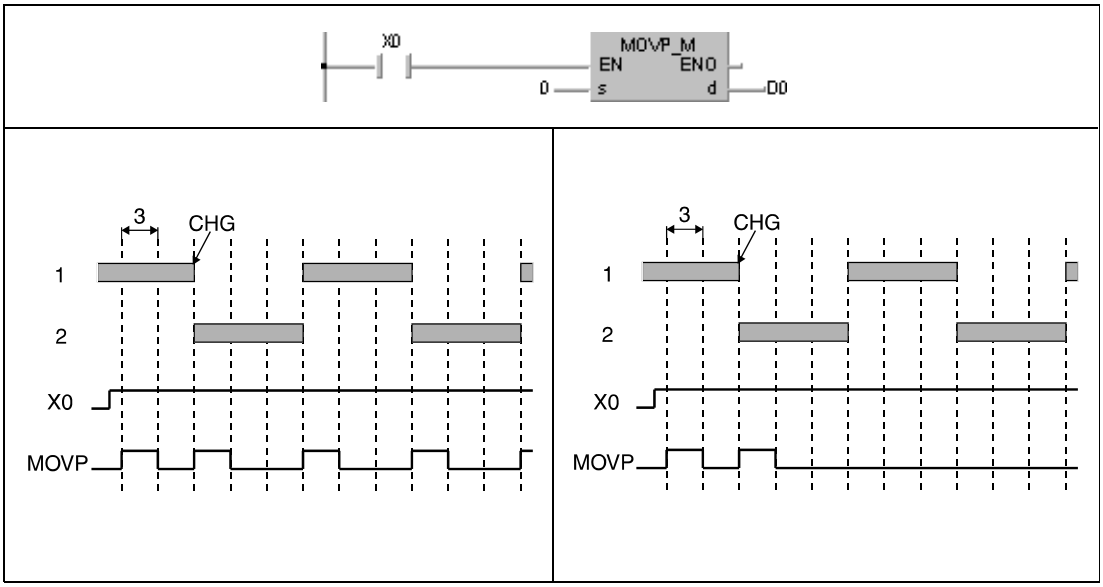
CHG Verwendung einer A3□CPU

Bei Verwendung einer A3□CPU ist die Ausführungsweise der gepulsten Anweisung vom Zustand des Sondermerkers M9050 abhängig.

Der Sondermerker M9050 ist in Verbindung mit einer A3 NCPU nicht einsetzbar. Die A3 NCPU verhält sich so, als sei M9050 gesetzt.

Die folgende Darstellung zeigt im oberen Teil eine programmierte, gepulste Anweisung. Dieser Programmteil liegt am Programmanfang (Schritt 0) des MAIN- oder SUB-Speicherbereichs.

Im unteren Teil sind die entsprechenden Signalverläufe abgebildet. Der linke Signalverlauf gilt bei nicht gesetztem Merker M9050 und der rechte Signalverlauf bei gesetztem Merker M9050. Die auf die Darstellung folgende Tabelle erläutert die Verarbeitung in Abhängigkeit des Einschaltzustandes von X0.



Die Verarbeitung des MAIN-Bereichs ist mit 1, die Verarbeitung des SUB-Bereichs mit 2 und die Dauer eines Zyklus mit 3 gekennzeichnet.

Zustand von X0	Zustand von M9050	
	NICHT GESETZT	GESETZT
0	Die MOVP-Anweisung wird nicht ausgeführt.	Die MOVP-Anweisung wird nicht ausgeführt.
1	Die MOVP-Anweisung wird nur während des ersten Zyklus nach der Programmumschaltung mittels CHG-Anweisung ausgeführt.	Die MOVP-Anweisung wird nach dem Einschalten von X0 nur während des ersten Zyklus des Programms im SUB-Bereich, in das mittels CHG-Anweisung umgeschaltet wurde, ausgeführt.
$0 \rightarrow 1$	Die MOVP-Anweisung wird einmalig ausgeführt.	Die MOVP-Anweisung wird einmalig ausgeführt.

CHG-Anweisung und Zählweise von Countern

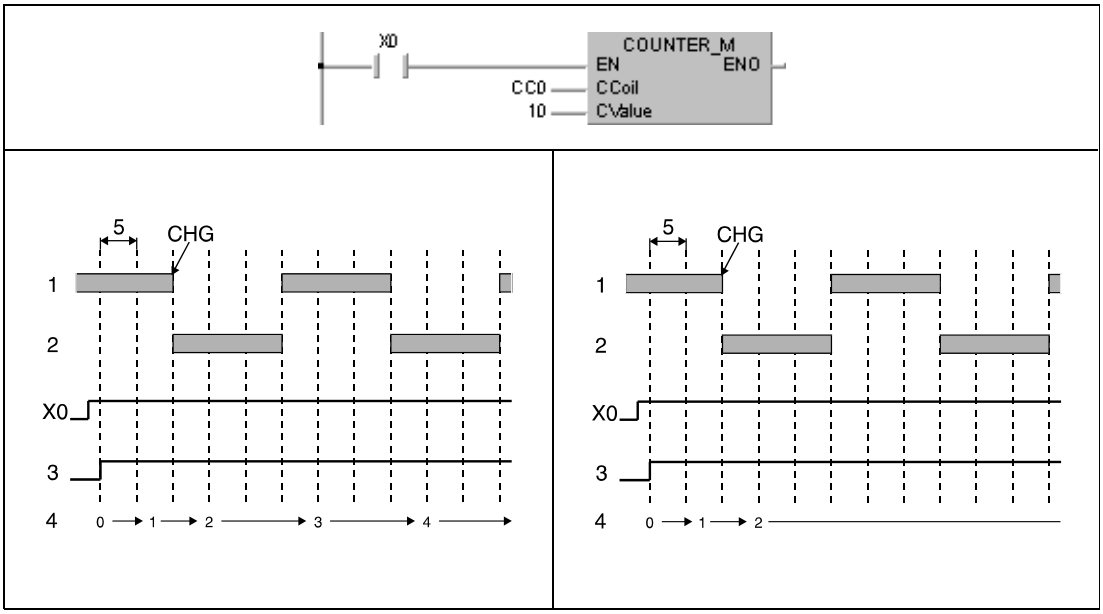
CHG Verwendung einer A3□CPU

Bei Verwendung einer A3□CPU ist die Zählweise eines Counters vom Zustand des Sondermerkers M9050 abhängig, soweit alle übrigen Eingangsbedingungen unverändert bleiben.

Der Sondermerker M9050 ist in Verbindung mit einer A3 NCPU nicht einsetzbar. Die A3 NCPU verhält sich so, als sei M9050 gesetzt.

Die folgende Darstellung zeigt im oberen Teil eine programmierte Counter-Anweisung. Dieser Programmteil liegt am Programmanfang (Schritt 0) des MAIN- oder SUB-Speicherbereichs.

Im unteren Teil sind die entsprechenden Signalverläufe abgebildet. Der linke Signalverlauf gilt bei nicht gesetztem Merker M9050 und der rechte Signalverlauf bei gesetztem Merker M9050. Die auf die Darstellung folgende Tabelle erläutert die Verarbeitung in Abhängigkeit des Einschaltzustandes von X0.



Die Verarbeitung des MAIN-Bereichs ist mit 1, die Verarbeitung des SUB-Bereichs mit 2, der Kontakt von C0 mit 3, der Istwert von C0 mit 4 und die Dauer eines Zyklus mit 5 gekennzeichnet.

Zustand von X0	Zustand von M9050	
	NICHT GESETZT	GESETZT
0	Der Istwert des Counters ändert sich nicht.	Der Istwert des Counters ändert sich nicht.
1	Der Istwert des Counters wird nach END (FEND, CHG) während des ersten Zyklus nach der Programmumschaltung mittels CHG-Anweisung um 1 erhöht.	Nach dem Einschalten von X0 wird der Istwert des Counters nach END (FEND, CHG) während des ersten Zyklus des Programms im SUB-Bereich, in das mittels CHG-Anweisung umgeschaltet wurde, um 1 erhöht.
$0 \rightarrow 1$	Der Istwert des Counters wird nach Ausführung einer END-Anweisung (FEND, CHG) um 1 erhöht.	Der Istwert des Counters wird nach Ausführung einer END-Anweisung (FEND, CHG) um 1 erhöht.

CHG-Anweisung und der Zeitverlauf von Timern

Alle CPUs, die eine CHG-Anweisung verarbeiten können, verfügen über zwei Speicherbereiche für Timer-Sollwerte. Dabei ist ein Speicherbereich für den MAIN-Bereich und einer für den SUB-Bereich reserviert. Das heißt, dass die Verarbeitung der Timer nur entsprechend dem aktuell verarbeiteten Programmteil (MAIN- oder SUB-Speicherbereich) erfolgt.

Die Sollwerte von Timern, die augenblicklich nicht verarbeitet werden, sind in den entsprechenden Speicherbereichen auf "0" gesetzt. Ein Sollwert von "0" entspricht einem unendlich großem Wert, so dass der Timer nicht abläuft.

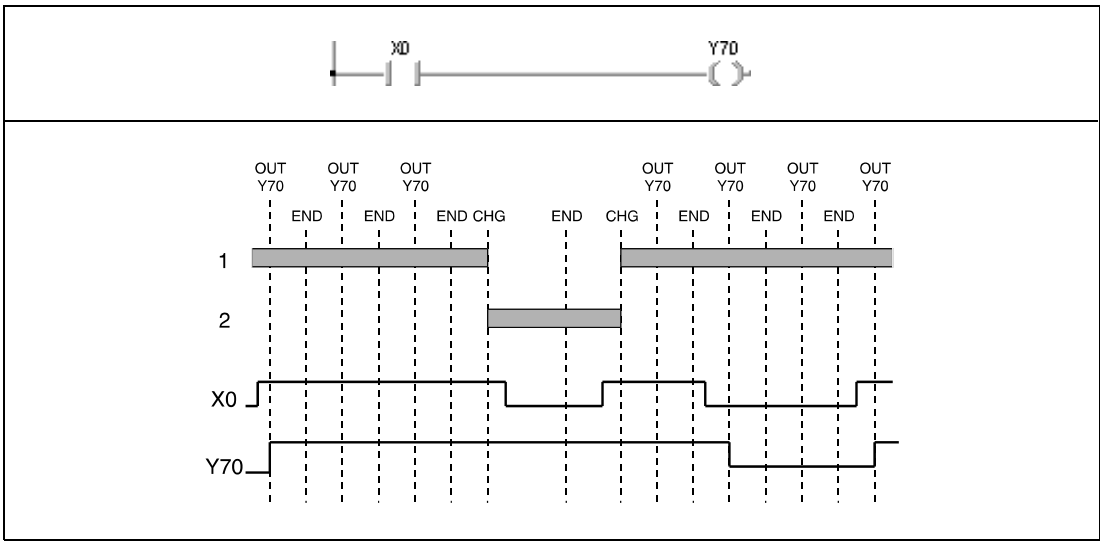
Wird nach dem Starten eines Timers im aktuellen Programm die CHG-Anweisung zum Umschalten in den MAIN-/SUB-Speicherbereich ausgeführt, wird der Timer im umgeschalteteten Programmteil nicht weiter verarbeitet. Dies liegt daran, dass der Timer im aktuellen Programm programmiert wurde und der Sollwert im umgeschalteten Programm als "0" betrachtet wird. Nach dem Wiederumschalten in das aktuelle Programm wird die Verarbeitung des Timers fortgesetzt. Der Timer läuft ab, wenn der Istwert größer als der Sollwert oder kleiner als 0 ist. Bei Ablauf des Timers wird der Timer-Kontakt eingeschaltet.

CHG-Anweisung und die Verarbeitung von OUT-Anweisungen

CPUs, die eine CHG-Anweisung verarbeiten können, schalten die Ausgangskontakte in Abhängigkeit des aktuell verarbeiteten Programmteils.

Ausgangskontakte behalten ihren Zustand nach dem Umschalten vom aktuellen Programmteil in einen anderen Programmteil (MAIN-/SUB-Bereich) bei. Der Zustand bleibt auch dann beibehalten, wenn sich die Eingangsbedingung ändert.

Die folgende Darstellung zeigt im oberen Teil eine programmierte OUT-Anweisung. Dieser Programmteil liegt im MAIN-Speicherbereich. Der Ausgang Y70 wird im SUB-Speicherbereich nicht verwendet.



Im unteren Teil sind die entsprechenden Signalverläufe abgebildet. Die Verarbeitung des MAIN-Bereichs ist mit 1 und die Verarbeitung des SUB-Bereichs mit 2 gekennzeichnet.

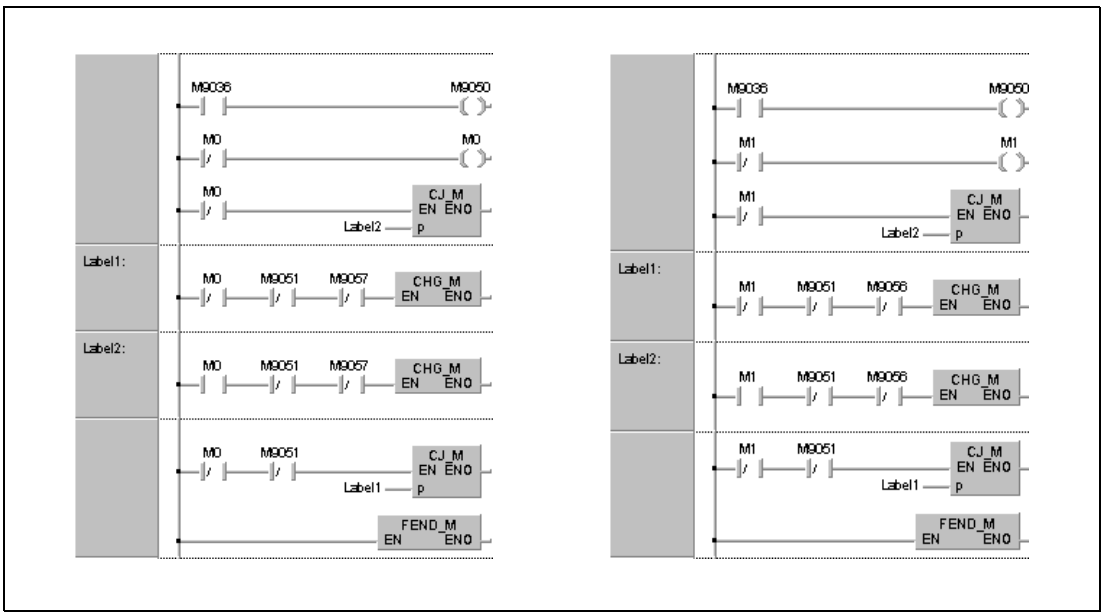
Der Ausgang Y70 wird während der Abarbeitung des MAIN-Bereiches in Abhängigkeit der Eingangsbedingung X0 ein- bzw. ausgeschaltet. Während der Abarbeitung des SUB-Bereiches bleibt der Zustand von Y70 auch dann unverändert, wenn sich die Eingangsbedingung ändert.

Beispiel 1 CHG (A3□CPU)

Für eine einwandfreie Verarbeitung der CHG-Anweisung muss das Verarbeitungsergebnis eines Programmzyklus mit dem des vorangegangenen Zyklus verglichen werden. Aus diesem Grund muss Sondermerker M9050 stets vor Ausführung der CHG-Anweisung gesetzt sein, um das Verarbeitungsergebnis des vorangegangenen Zyklus aus dem Zwischenspeicher in den Arbeitsspeicher zu laden.

Da die CHG-Anweisung in Verbindung mit einer A3□CPU nur bei gegebener Eingangsbedingung ausgeführt wird, müssen Programme entsprechend dem folgenden Schema geschrieben werden. Das linke Programm befindet sich im Main-Speicherbereich und das rechte Programm im Sub-Speicherbereich.

Der Sondermerker M9036 ist immer gesetzt.

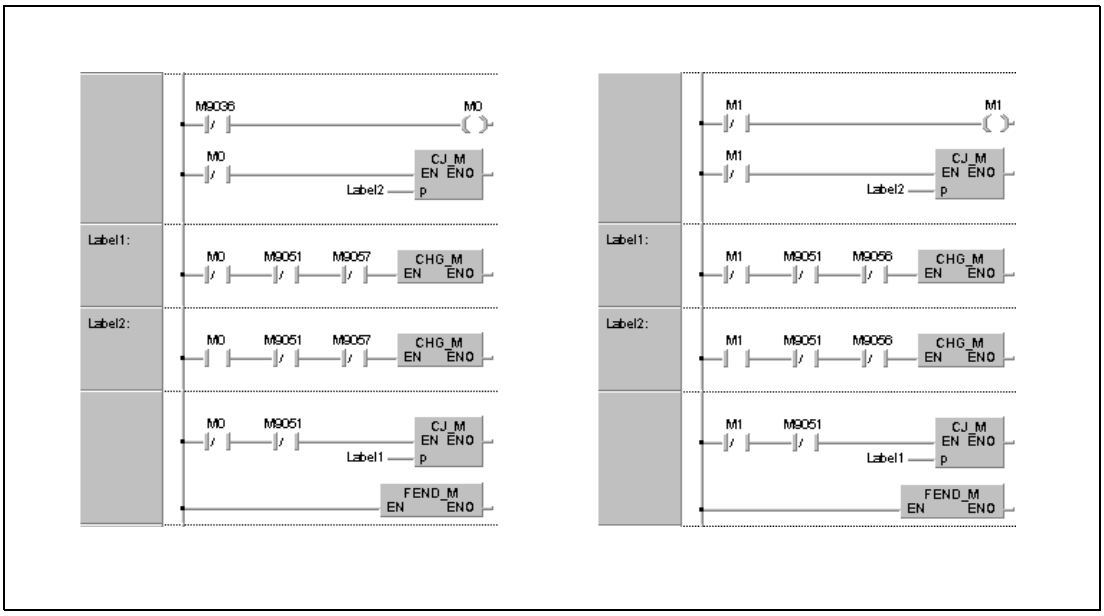


HINWEIS *Das Ändern eines Programms im SUB-Bereich während der Verarbeitung des Programms im MAIN-Bereich (oder umgekehrt) setzt voraus, dass die Sondermerker M9051, M9056 und M9057 eingesetzt werden, damit eine Ausführung der CHG-Anweisung und somit eine Programmumschaltung verhindert wird.*

Dies bedeutet, dass während eines Online-Changes im SUB-Bereich der MAIN-Bereich nicht abgearbeitet wird. Im GX Developer und *GX IEC Developer erfolgt dies automatisch.*

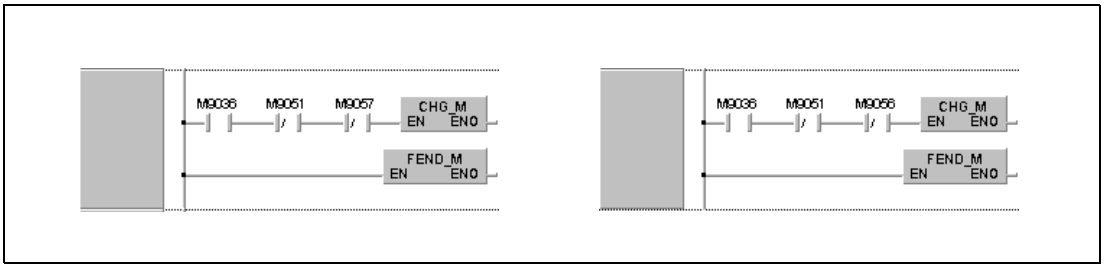
Beispiel 2 CHG (A3N CPU)

Die CHG-Anweisung wird in Verbindung mit einer A3N CPU nur bei gegebener Eingangsbedingung ausgeführt. Programme müssen entsprechend dem Schema in diesem Programmbeispiel aufgebaut sein. Das linke Programm befindet sich im Main-Speicherbereich und das rechte Programm im Sub-Speicherbereich.



Beispiel 3 CHG (A3H CPU)

Programme müssen entsprechend dem Schema in diesem Programmbeispiel aufgebaut sein. Das linke Programm befindet sich im Main-Speicherbereich und das rechte Programm im Sub-Speicherbereich.



7.6.9 SUB, SUBP

CPU

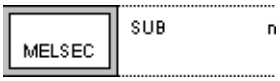
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●				

Operanden MELSEC A

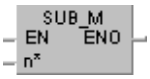
	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)								Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
n								●	●	●	●	●	●	●	●	●	●	●					3	●		●

GX IEC Developer

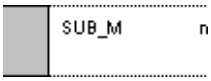
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



Variablen

Operand	Befehlswert	Datentyp
n	Adresse des aufzurufenden Mikrocomputer-Programms.	Adresse

Funktions-weise

Mikrocomputer-Programmaufruf

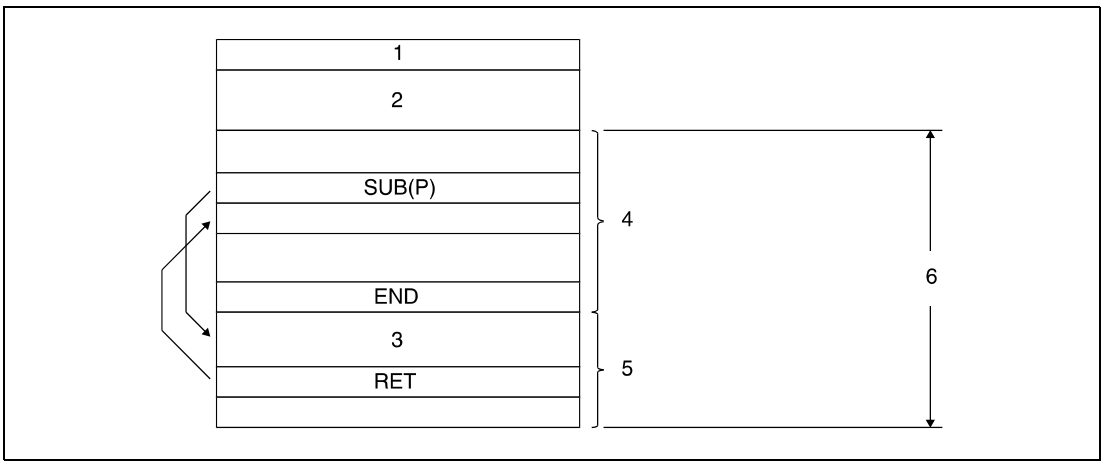
SUB Programmaufruf eines Mikrocomputer-Programms

Der Aufruf eines vom Anwender erstellten Mikrocomputer-Programms erfolgt mit Hilfe der SUB/SUBP-Anweisung.

Bei gegebener Eingangsbedingung ruft die SUB-Anweisung das an der Adresse "n" befindliche Mikrocomputer-Programm auf.

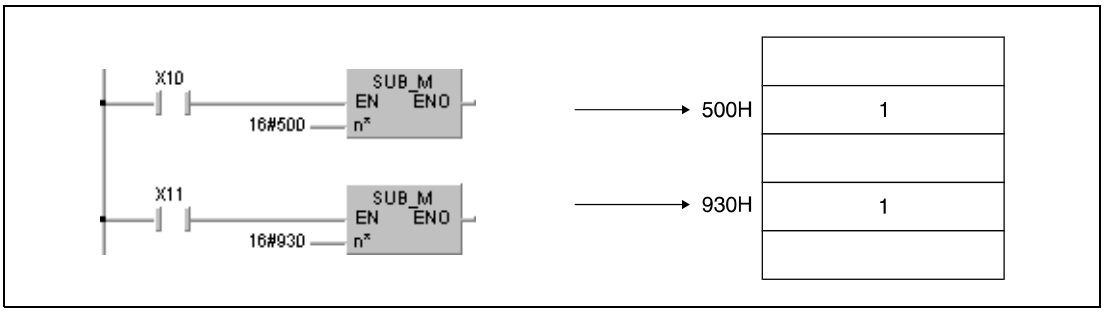
Nach Abarbeitung des Microcomputer-Programms wird das Ablaufprogramm nach dem Programmschritt wieder aufgenommen, der der SUB/SUBP-Anweisung folgt.

Die SUB/SUBP-Anweisung kann im Ablaufprogramm des MAIN- und SUB-Speicherbereiches programmiert werden.



[1] Parameter
[2] Sollwerte für Timer und Counter
[3] Mikrocomputer-Programm
[4] Bereich des Ablaufprogramms
[5] Bereich des Mikrocomputer-Programms
[6] MAIN- oder SUB-Speicherbereich

Innerhalb eines Mikrocomputer-Programmbereiches können mehrere Programme erzeugt werden.



[1] Mikrocomputer-Programm

HINWEIS *Die SUB-Anweisung hat bei den erweiterten Applikationsanweisungen der AnA, AnAS und AnU CPUs (Dedicated Instructions) die Funktion, eine 16-Bit-Konstante im Anweisungsblock festzulegen.*

Weitere Einzelheiten über Mikrocomputer-Programme sind Kapitel 10 dieser Programmieranleitung zu entnehmen.

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die Maximalkapazität des Mikrocomputer-Programms wird überschritten.
- Die in n angegebene Adresse liegt außerhalb des erlaubten Bereiches.

HINWEIS *Die Verarbeitungszeit eines Mikrocomputer-Programms, das über eine SUB(P)-Anweisung aufgerufen wurde, darf 5 ms nicht übersteigen. Dauert die Verarbeitung länger als 5 ms, kommt es zum Konflikt mit dem Ablaufprogramm, so dass die SPS nicht mehr einwandfrei arbeiten kann.*

Soll ein Mikrocomputer-Programm ausgeführt werden, das länger als 5 ms dauert, muss dieses Programm in einzelne Blöcke aufgeteilt, und diese Blöcke müssen nacheinander abgearbeitet werden. Auf diese Weise kann die Verarbeitungszeit des Mikrocomputer-Programms verkürzt werden.

7.6.10 IX, IXEND

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●

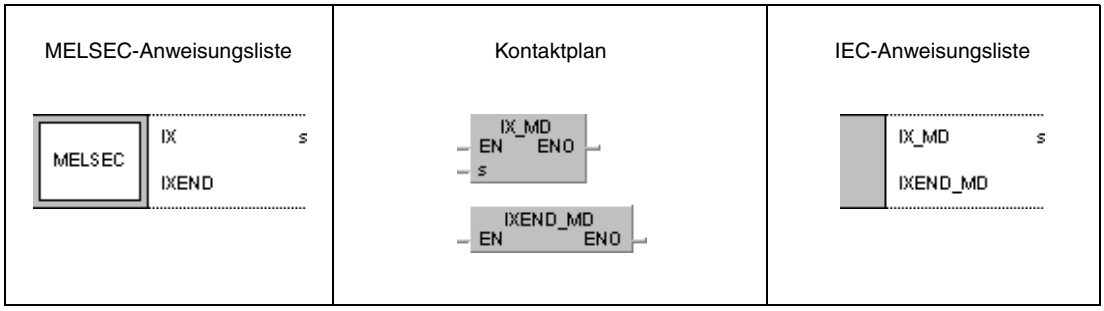
[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	—	—	—	—	—	SM0	2/1 ●[1]

[1] Die IX-Anweisung benötigt zwei Schritte und die IXEND-Anweisung einen Schritt.

GX IEC Developer



GX Developer

Variablen

Operand	Befehlswert	Datentyp
s	Erste Adresse des Operanden, in dem die Daten für die indizierte Adressierung gespeichert sind.	BIN-16-Bit

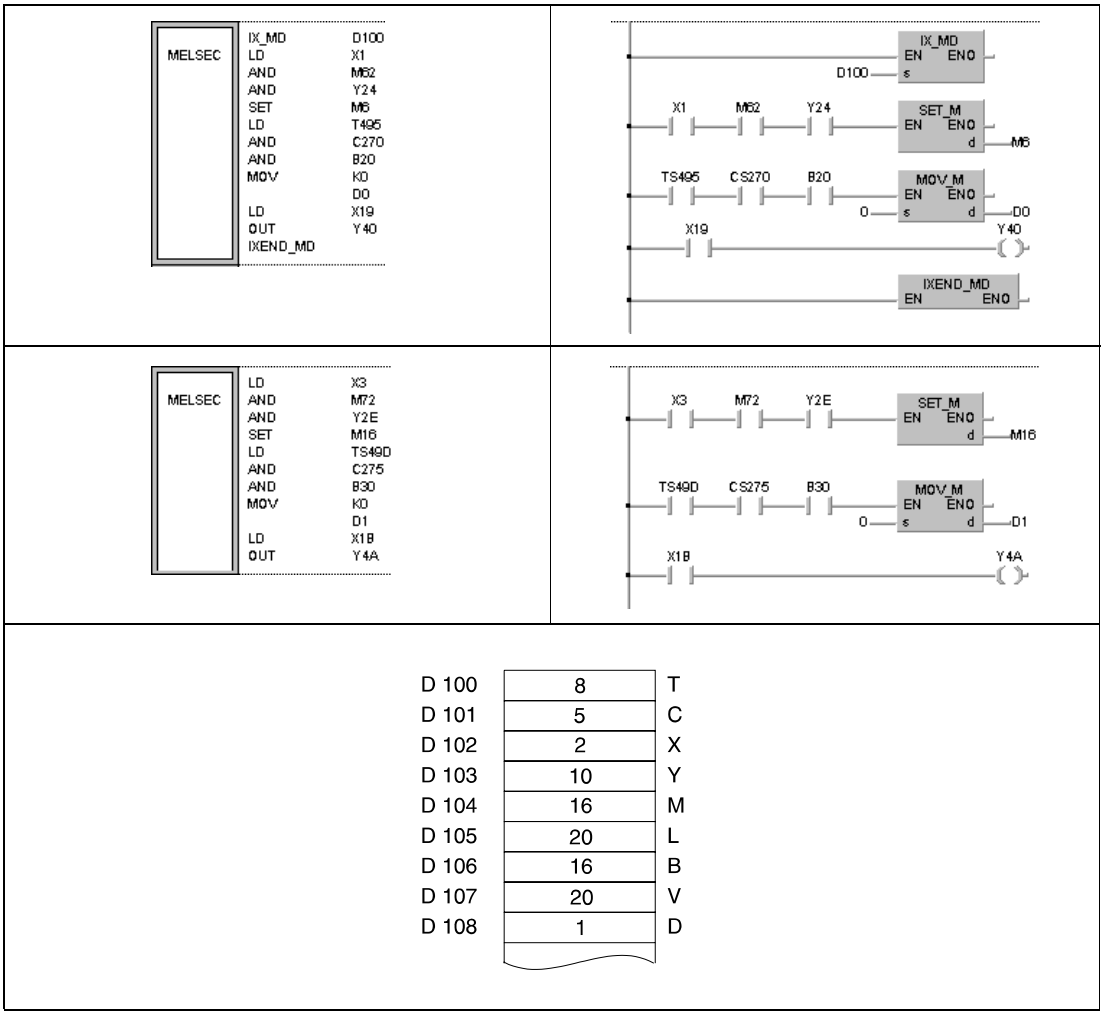
Funktionsweise

Indizierte Adressierung eines gesamten Programmteils

IX, IXEND Adressierungsanweisung

Die Anweisungen IX und IXEND können nur im MELSEC-Modus des GX IEC Developers eingesetzt werden. Die IX- und IXEND-Anweisungen nehmen eine indizierte Adressierung der Operandenadressen in dem Programmteil vor, der zwischen der IX- und IXEND-Anweisung programmiert wird.

Bei diesem Vorgang werden die dezimalen Zahlenwerte der Index-Liste (s) zu den Operandenadressen addiert. Die neue Adresse in hexadezimaler Darstellung ist bei der darauffolgenden Programmabarbeitung die gültige Adresse. Jedem der in s angegebenen Operanden ist ein bestimmter Operandentyp zugeordnet, auf dessen Adresse die Addition angewendet wird. Die nachfolgende Abbildung verdeutlicht diesen Zusammenhang.



Der Wert in D100 (8) wird zu der Timeradresse TS495 addiert. Die neue Adresse lautet TS49D.

Der Wert in D101 (5) wird zu der Counteradresse CS270 addiert. Die neue Adresse lautet CS275.

Der Wert in D102 (2) wird zu den Adressen der Eingänge X1 und X19 addiert. Die neuen Adressen lauten X3 und X1B.

Der Wert in D103 (10) wird zu den Adressen der Ausgänge Y24 und Y40 addiert. Die neuen Adressen lauten Y2E und Y4A.

Der Wert in D104 (16) wird zu den Adressen der Merker M6 und M62 addiert. Die neuen Adressen lauten M16 und M72.

Der Wert in D106 (16) wird zu der Link-Merker-Adresse B20 addiert. Die neue Adresse lautet B30.

Der Wert in D108 (1) wird zu der Register-Adresse D0 addiert. Die neue Adresse lautet D1.

PLS-, PLF- und gepulste Anweisungen, die nur einmal bei gegebener Eingangsbedigung ausgeführt werden, können nicht über die IX-/IXEND-Anweisungsschleife indiziert adressiert werden.

In Fällen, in denen die durch Addition entstandene neue Adresse den zulässigen Adressbereich verläßt, ist eine einwandfreie Verarbeitung der Anweisung nicht mehr gewährleistet.

Wenn die IX- und IXEND-Anweisung während des Wechsels zwischen Programmsequenzen im Online-Modus (Schreiben während des RUN-Modus) ausgeführt wird, ist eine einwandfreie Verarbeitung ebenfalls nicht gewährleistet.

Die Werte, die zu den Adressen von Wortoperanden, bei denen auf jedes Bit zugegriffen werden kann, addiert werden, sind als Binärdaten gespeichert. Die Startadresse der Operanden, für die diese Werte angegeben werden, ist in s gespeichert.

In einem Programm kann zwischen der IX- und IXEND-Anweisung keine indizierte Adressierung ausgeführt werden.

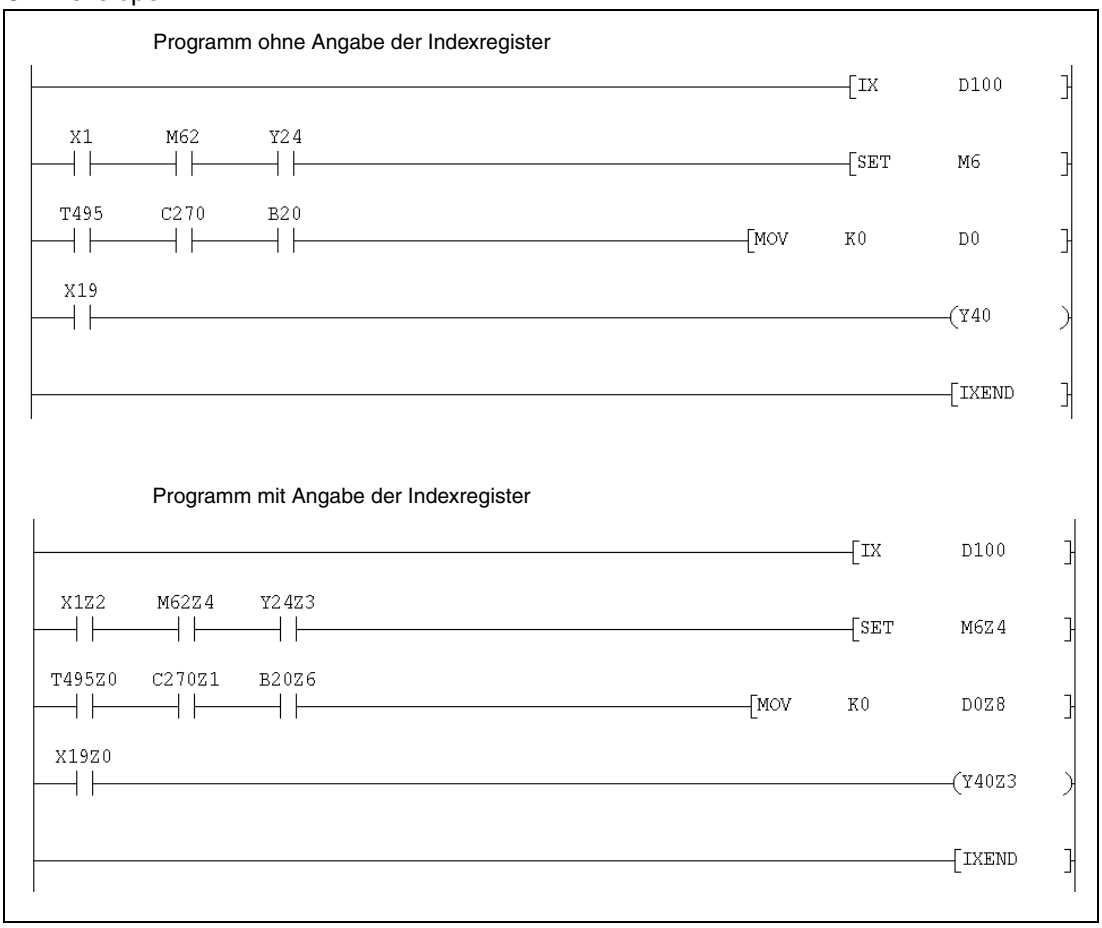
Die zwischen IX- und IXEND-Anweisung indizierten Adressen der Operanden eines Programmteils werden während einer Programmerweiterung in Adressen umgewandelt, die die Index-Register (Zn) verwenden. Die Zuordnung der indizierten Adressen zu den entsprechenden Index-Registern stellt die folgende Tabelle dar.

s	Adresse	Index-Register	s	Adresse	Index-Register
s	Additionswerte für Timeradressen (T)	Z0	s+8	Additionswerte für Datenregister-Adressen (D)	Z8
s+1	Additionswerte für Counteradressen (C)	Z1	s+9	Additionswerte für Link-Register-Adressen (W)	Z9
s+2	Additionswerte für Eingangsadressen (X)	Z2	s+10	Additionswerte für File-Register-Adressen (R)	Z10
s+3	Additionswerte für Ausgangsadressen (Y)	Z3	s+11	Additionswerte für Puffer-Register-Ein-/Ausgangsadressen (U)	Z11
s+4	Additionswerte für Merkeradressen	Z4	s+12	Additionswerte für Puffer-Register-Adressen (G)	Z12
s+5	Additionswerte für Latch-Merker-Adressen (L)	Z5	s+13	Additionswerte für Netzwerkadressen von Link-Operanden mit Direktzugriff (J)	Z13
s+6	Additionswerte für Link-Merker-Adressen (B)	Z6	s+14	Additionswerte für lokale File-Registeradressen (ZR)	Z14
s+7	Additionswerte für Flankenmerker-Adressen (V)	Z7	s+15	Additionswerte für Pointer (Label)-Adressen	Z15

Bei einer Q00J-, Q00- oder Q01CPU stehen die Indexregister Z10 bis Z15 nicht zur Verfügung.

Je nach verwendeter Programmier-Software kann es erforderlich sein, dass im Programmteil zwischen der IX- und der IXEND-Anweisung die Indexregister angeben werden müssen.

Beispiel GX Developer



Die zwischen der IX- und der IXEND-Anweisung verwendeten Indexregister (Z0 bis Z15) beeinflussen nicht die anderen Anweisungen im Programm, die über eine Index-Vergabe adressiert sind.

HINWEIS *Für die indizierte Adressierung der Operanden eines Programmteils ist es erforderlich, die Peripheriegeräte im Standard-Modus zu starten und eine Programmerweiterung auszuführen (nur für QnA-Serie).*

Beim Starten der Peripheriegeräte durch eine Q2A, Q2A-S1, Q3A oder Q4A CPU bei gleichzeitiger Verwendung der indizierten Adressierung der Operanden eines Programmteils zwischen IX- und IXEND-Anweisung ist eine einwandfreie Verarbeitung nicht gewährleistet.

Sehen Sie eine Verriegelung vor, die die gleichzeitige Ausführung von mehreren IX- und IXEND-Anweisungen verhindert, wenn die IX- und IXEND-Anweisung im normalen und im Interrupt-Programm verwendet wird. Sperren Sie im normalen Programm Interrupts zwischen der IX- und der IXEND-Anweisung.

Fehlerquellen In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die IX- und die IXEND-Anweisung sind nicht im Zusammenhang programmiert (Fehlercode 4231).
- Nach der Ausführung einer IX-Anweisung wird eine END-, FEND-, GOEND- oder STOP-Anweisung ausgeführt, ohne dass zuvor eine IXEND-Anweisung ausgeführt wird (Fehlercode 4231).

Beispiel IX, IXEND

Das folgende Programm durchläuft die Programmschleife zwischen der IX- und IXEND-Anweisung insgesamt 10 mal. Bei jedem Durchlauf werden die Adressen der in diesem Bereich programmierten Operanden um den Wert 1 erhöht. In der auf die Abbildung folgenden Tabelle sind die Register, in welchen die zu addierenden Werte der entsprechenden Operanden enthalten sind, aufgelistet. Ferner sind die Veränderungen der Operandenadressen für den 1., 2., 3. und 10. Durchlauf der Programmschleife dargestellt.

MELSEC-Anweisungsliste

```
MELSEC
        LD      SM400
        FMOV    K0
                D100
                K7
        FOR     K10
        IX      D100
        LD      B0
        OR      Y30
        ANI     X30
        OUT     Y30
        SET     M0
        +       D0
                D10
        LD      TS3
        AND     CS4
        MOV     K1
                D40
        IXEND
        LD      SM400
        BK+     D100
                K1
                D100
                K7
        NEXT
```

D	Additionswerte	Operandenadressenänderung / Durchläufe			
		1.	2.	3.	10.
D100	Additionswerte für Timeradressen (T)	T3	T4	T5	TC
D101	Additionswerte für Counteradr. (C)	C4	C5	C6	CD
D102	Additionswerte für Eingangsadressen (X)	X10	X11	X12	X19
D103	Additionswerte für Ausgangsadressen (Y)	Y30	Y31	Y32	Y39
D104	Additionswerte für Merkeradressen (M)	M0	M1	M2	M9
D106	Additionswerte für Link-Merker-Adressen (B)	B0	B1	B2	B9
D108	Additionswerte für Datenregister-Adressen (D)	D0	D1	D2	D9
		D10	D11	D12	D19
		D40	D41	D42	D49

7.6.11 IXDEV, IXSET

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

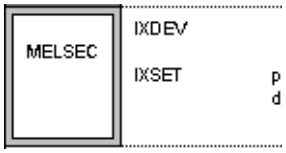
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort				P		
p	—	—	—	—	—	—	—	—	●	SM0	1/3
d	—	●	●	—	—	—	—	—	—		●[1]

[1] Die IXDEV-Anweisung benötigt einen Schritt und die IXSET-Anweisung drei Schritte.

GX IEC Developer

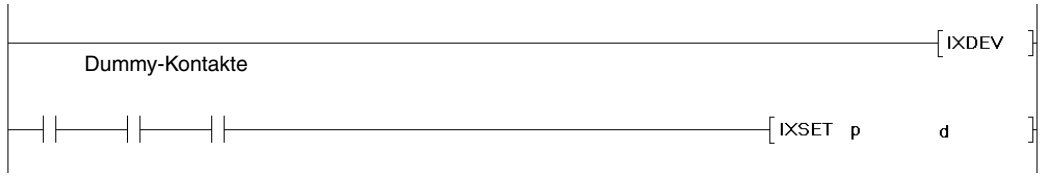
MELSEC-Anweisungsliste



Kontaktplan

IEC-Anweisungsliste

GX Developer



Variablen

Operand	Befehlswert	Datentyp
p	Erste Adresse des Operanden (nur Pointer/Label), in dem die Daten für die indizierte Adressierung gespeichert sind.	Pointer/Label
d	Erste Adresse des Operanden, in welchem die indizierten Adressen der Operanden gespeichert werden.	BIN-16-Bit

Funktions-weise

Speicherung indizierter Operandenadressen in einer Index-Liste

IXDEV/IXSET Anweisung zum Schreiben in eine Index-Liste

Die Anweisungen IXDEV und IXSET können im GX IEC Developer nur im MELSEC-Modus oder im GX Developer eingesetzt werden.

Die IXDEV- und IXSET-Anweisung lesen die Adressen der im Offset-Bereich befindlichen Operanden aus und schreiben diese Offsetwerte als Index-Liste in den in d angegebenen Operanden.

Die Zuordnung der Operandentypen zu den entsprechenden Registern ist der IX- und IXEND-Anweisung zu entnehmen.

Liegt der entsprechende Operandentyp im Offset-Bereich nicht vor, lautet der entsprechende Eintrag in der Index-Liste gleich 0.

Die einzelnen Bits der Wortoperanden werden als Dummy-Kontakt behandelt, d.h. es kann immer nur die Adresse eines einzelnen Bits ausgelesen und in die Index-Liste eingetragen werden. Bei der Adressierung des Dummys wird dann das entsprechende Bit angegeben. Das Bit 0 (b0) im Datenregister D0 wird z.B. mit D0.0 angegeben. Für die Bitangabe in einem 16-Bit-Datenwort werden die Hexadezimalwerte 0 bis F verwendet.

Das Auslesen der Offset-Werte erfolgt wie nachfolgend beschrieben.

- Auslesen der Operanden: T□, C□, X□, Y□, M□, L□, V□, B□

  Der mit □ gekennzeichnete Offset-Wert wird ausgelesen und in die entsprechenden Register geschrieben.

- Auslesen der Operanden: D□.XX, W□.XX, R□.XX[1], U□\G□.XX[1], ZR□.XX[1]

  Der mit □ gekennzeichnete Offset-Wert wird ausgelesen und in die entsprechenden Register geschrieben.

  Der mit XX gekennzeichnete Wert steht als Variable für die Bit-Bestimmung.

  [1]Nicht für Q00JCPU, Q00CPU und Q01CPU

- Auslesen des Operanden: J□\B□[1] ,J□\W□[1], J□\X□[1], J□\Y□[1]

  Der mit □ gekennzeichnete Offset-Wert wird ausgelesen und in die entsprechenden Register geschrieben.

  Wenn für den auf J□\ folgenden Operanden kein Offset eingetragen werden soll, ist dieser Wert auf 0 zu setzen.

  [1]Nicht für Q00JCPU, Q00CPU und Q01CPU

- Der Offset-Wert des Operanden P□ wird bei der Programmierung der IXSET-Anweisung als Adresse (Pointer/Label) direkt angegeben.

Wenn in dem Offset-Bereich zwei indentische Operandentypen angegeben werden, ist der Offset-Wert des am letzten in der Kontaktkette liegenden Operanden gültig.

Die IXDEV- und IXSET-Anweisung müssen zusammen verwendet werden.

Der Offset-Wert des Operanden ZR□.XX kann einen Wert zwischen 0 und 32767 besitzen. Das resultiert aus der Tatsache, dass der Rest des Quotienten der Division der Operandenadresse durch den Wert 32767 als Offset-Wert in das entsprechende Register geschrieben wird.

Als Dummy-Kontakte innerhalb des Offset-Bereiches sind nur LD- und AND-Anweisungen zulässig. Alle anderen Anweisungen werden ignoriert.

Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die IXDEV und die IXSET-Anweisung sind nicht im Zusammenhang programmiert (Fehlercode 4231).

Beispiel

IXDEV, IXSET

Das folgende Programm schreibt die Adressen (Offset-Werte) der Dummykontakte des Offset-Bereiches in die entsprechenden Register. Der Offsetwert des Pointers/Labels ist in der IXSET-Anweisung angegeben. Die Zuordnung der Operandentypen zu den entsprechenden Registern ist dem Abschnitt IX- und IXEND-Anweisung zu entnehmen.

MELSEC-Anweisungsliste

```
MELSEC   IXDEV
         LD      X5
         AND     Y13
         AND     TS4
         AND     D8.3
         IXSET   P5
                 D0
```

D0	4	T
	0	C
	5	X
	3	Y
	0	M
	0	L
	0	B
	0	V
	8	D
	0 ⋮ 0	
D15	5	

7.7 Verarbeitungsanweisungen für Datenlisten

Die Verarbeitungsanweisungen für Datenlisten schreiben und lesen Daten in bzw. aus einer Datenliste. Aktuelle Daten werden in die Liste geschrieben und für eine weitere Verarbeitung in unterschiedlicher Reihenfolge wieder ausgelesen. Ferner geben diese Anweisungen dem Anwender die Möglichkeit, bestimmte Datenblöcke in der Datenliste zu löschen oder einzufügen.

Funktion	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor
Daten in eine Datenliste schreiben	FIFW	FIFW_M
	FIFWP	FIFWP_M
Lesen zuerst eingegebener Daten aus der Datenliste	FIFR	FIFR_M
	FIFRP	FIFRP_M
Lesen zuletzt eingegebener Daten aus der Datenliste	FPOP	FPOP_M
	FPOPP	FPOPP_M
Löschen bestimmter Datenblöcke in der Datenliste	FDEL	FDEL_M
	FDELP	FDELP_M
Einfügen bestimmter Datenblöcke in die Datenliste	FINS	FINS_M
	FINSP	FINSP_M

7.7.1 FIFW, FIFWP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				K1 ↓ K4	7 [1]			●
d								●	●	●	●	●														

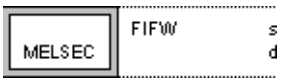
[1] Die Anzahl der Schritte bei der Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

Operanden MELSEC Q

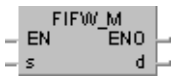
	Operanden								Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten	Andere	
	Bit	Wort		Bit	Wort					
s	●	●	●	●	●	●	●	—	—	SM0	3
d	—	●	●	—	—	—	—	—	—		

GX IEC Developer

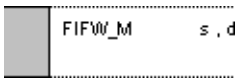
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

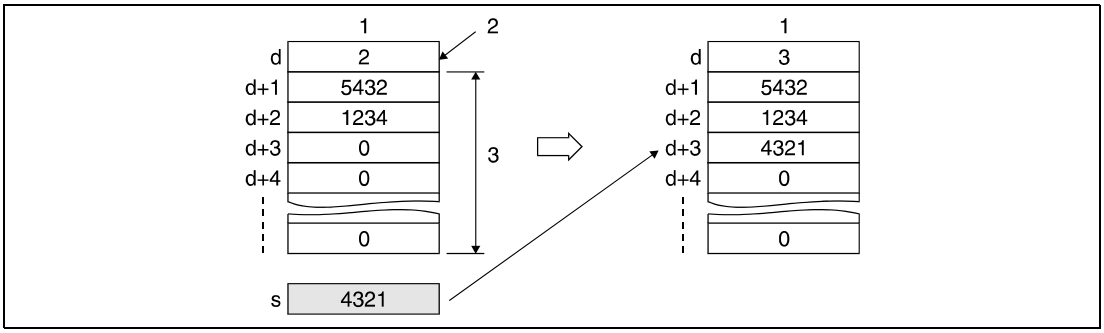
Operand	Befehlswert	Datentyp
s	Daten, die in die Datenliste geschrieben werden oder Operanden, in dem diese Daten gespeichert sind.	BIN-16-Bit
d	Erste Adresse der Datenliste.	

Funktions-weise

Daten in eine Datenliste schreiben

FIFW Anweisung zum Dateneintrag

Die FIFW-Anweisung schreibt die in s bestimmte Datenfolge in eine Datenliste. Diese Liste wird durch den Adressenbereich in d vorgegeben und verwaltet die Daten in der Reihenfolge ihres Eingangs. In der ersten Adresse des Datenbereiches in d wird die Anzahl der Datensätze gespeichert, die insgesamt in der Datenliste enthalten sind. Der Wert in dieser Adresse ist somit der Positionszeiger für Daten, die neu in die Liste aufgenommen werden. Mit jeder Ausführung der FIFW-Anweisung wird der Wert um 1 erhöht. Die Daten werden dann ab der Adresse d+1 abgelegt.



[1] Datenliste

[2] Positionszeiger

[3] Adressbereich der Datenliste

Vor der ersten Ausführung der FIFW-Anweisung müssen die Inhalte des in d angegebenen Operanden gelöscht werden.

Die Anzahl der zu schreibenden Datensätze und der damit verbundene Adressbereich der Datenliste ist bei der Programmierung zu kontrollieren.

Zur Verwaltung mehrerer Datensätze in unterschiedlichen Datenlisten ist ein Anwenderprogramm einzusetzen.

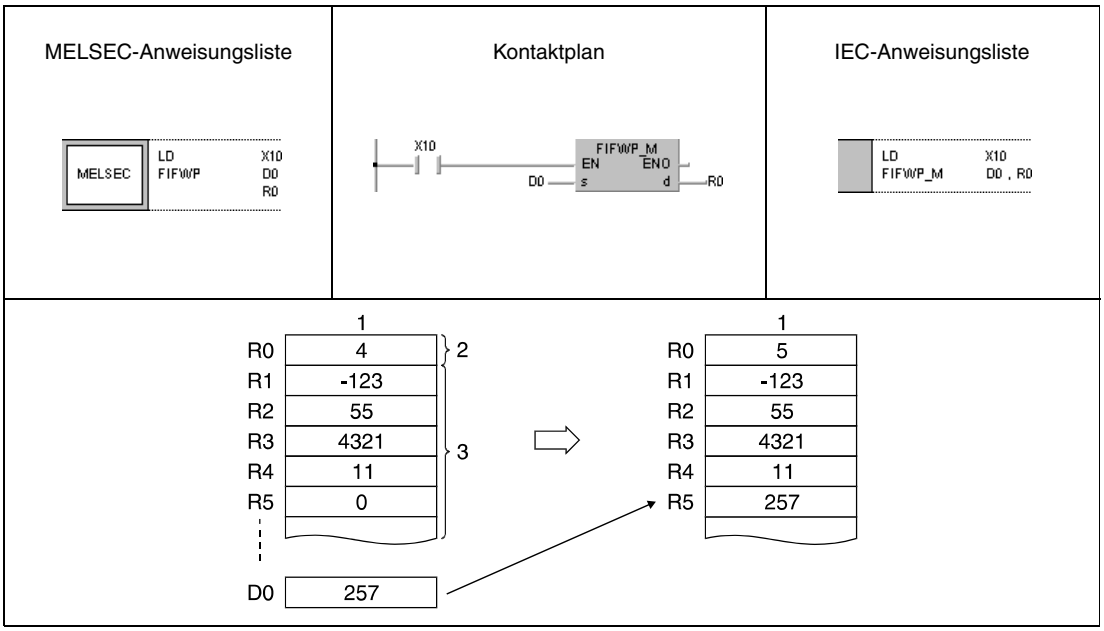
Fehler-quellen

Im folgenden Fall tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die Länge der FIFO-Liste überschreitet bei der Ausführung der FIFW-Anweisung den für die Speicherung vorgesehenen Bereich des Operanden
  (Q-Serie/System Q = Fehlercode 4101).

Beispiel 1 FIFWP

In folgendem Programm wird der Speicherbereich der Datenliste durch die Datenregister R0 bis R5 vergeben. Die Startadresse (R0) des Bereiches enthält den Positionszeiger, der die Anzahl der gespeicherten Datensätz angibt. Mit der positiven Flanke von X10 werden die Daten aus D0 in den nächsten freien Speicherpositionen der Datenliste (im Beispiel R5) abgelegt.



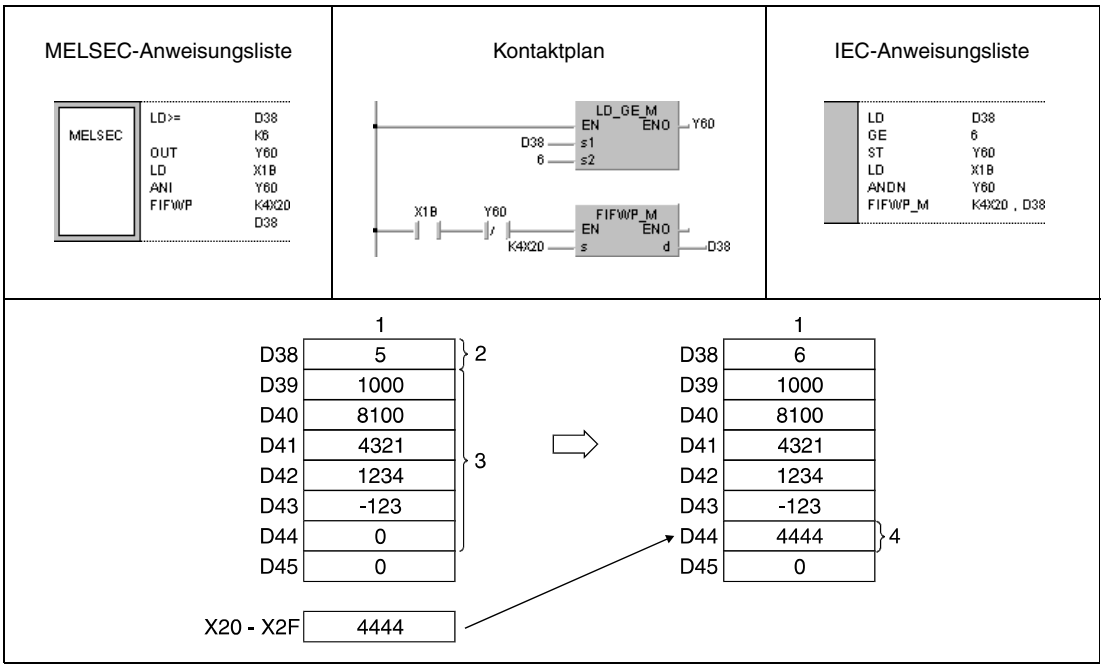
[1] Datenliste

[2] Positionszeiger

[3] Adressbereich der Datenliste

Beispiel 2 FIFWP

In folgendem Programm wird der Speicherbereich der Datenliste durch die Datenregister D38 bis D44 vorgegeben. Die Startadresse (D38) des Bereiches enthält den Positionszeiger, der die Anzahl der gespeicherten Datensätze angibt. Mit der positiven Flanke von X1B werden die Daten der Eingänge X20 bis X2F an der nächsten freien Speicherposition der Datenliste (im Beispiel D44) abgelegt. In der hier vorgegebenen Datenliste können maximal 6 Datensätze gespeichert werden. Y60 wird daher als Verriegelung der FIFW-Anweisung programmiert. Der Ausgang schaltet ein, sobald der Inhalt von D38 größer oder gleich 6 ist.



[1] Datenliste

[2] Positionszeiger

[3] Adressbereich der Datenliste

[4] Letzte Speicheradresse

7.7.2 FIFR, FIFRP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

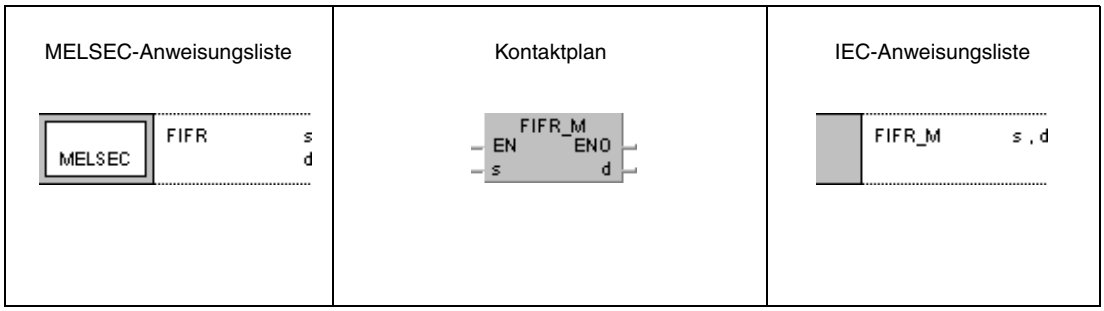
	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)								Konstante		Pointer		Ebene				
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
s		●	●	●	●	●	●	●	●	●	●	●	●	●	●							K1 ↓ K4	7 ● [1]	●		●
d								●	●	●	●	●														

[1] Die Anzahl der Schritte bei der Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

Operanden MELSEC Q

	Operanden								Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sonder-module U□\G□	Index-Register Zn	Konstanten	Andere	
	Bit	Wort		Bit	Wort					
s	●	●	●	●	●	●	●	—		SM0	3
d	—	●	●	—	—	—	—	—	—		

GX IEC Developer



GX Developer

Variablen

Operand	Befehlswert	Datentyp
s	Startadresse des Operanden, in welchem die ausgelesenen Daten gespeichert werden.	BIN-16-Bit
d	Startadresse der Datenliste.	

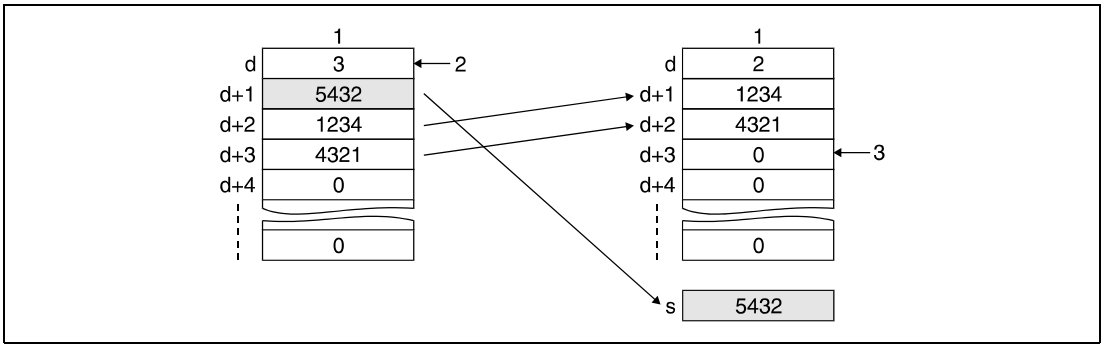
Funktionsweise

Lesen zuerst eingegebener Daten aus der Datenliste

FIFR Anweisung zum Auslesen der zuerst eingegebenen Daten

Die FIFR-Anweisung liest die Daten aus der Datenliste und speichert sie in einen vorgegebenen Adressenbereich. Das Auslesen der Daten beginnt mit der ersten Adresse d+1 nach dem Positionszeiger. Die Übertragung erfolgt an den in s angegebenen Adressenbereich.

Die Daten der Datenliste werden nacheinander in der Reihenfolge ihres Eingangs an den Anfang der Liste geschoben. Alle vorangegangenen Daten werden gelöscht. Nach dem Auslesen wird der Wert des Positionszeigers (erste Adresse in d) um 1 vermindert.



[1] Datenliste

[2] Positionszeiger

[3] Dieses Register wird mit dem Wert 0 beschrieben.

HINWEIS *Achten Sie darauf, dass die Anweisung nicht aufgerufen wird, wenn der Wert in d (Positionszeiger) gleich 0 ist.*

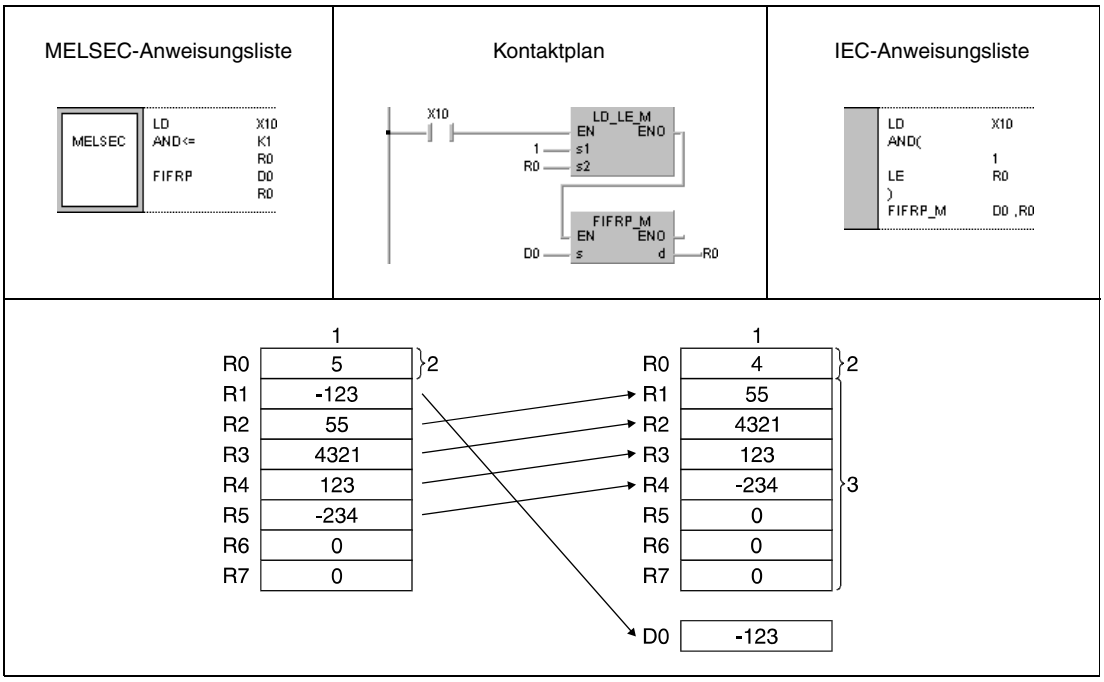
Fehlerquellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Eine FIFR-Anweisung wird ausgeführt, wenn der Wert des Positionszeigers gleich 0 ist (Q-Serie/System Q = Fehlercode 4100).
- Die Länge der Datenliste überschreitet bei der Ausführung der FIFR-Anweisung den Operandenbereich (Q-Serie/System Q = Fehlercode 4101).

Beispiel 1 FIFRP

Im folgenden Programm wird mit positiver Flanke von X10 der Datenwert der Adresse R1 (zuerst eingegebener Wert) der Datenliste von R0 bis R7 ausgelesen und im Register D0 gespeichert. Im Beispiel lautet der Wert des Positionszeigers zu Beginn der Ausführung 5, nach der Ausführung 4. Die vorgeschaltete Vergleichsanweisung verhindert die Ausführung der FIFR-Anweisung, wenn der Positionszeiger (R0) den Wert 0 besitzt.



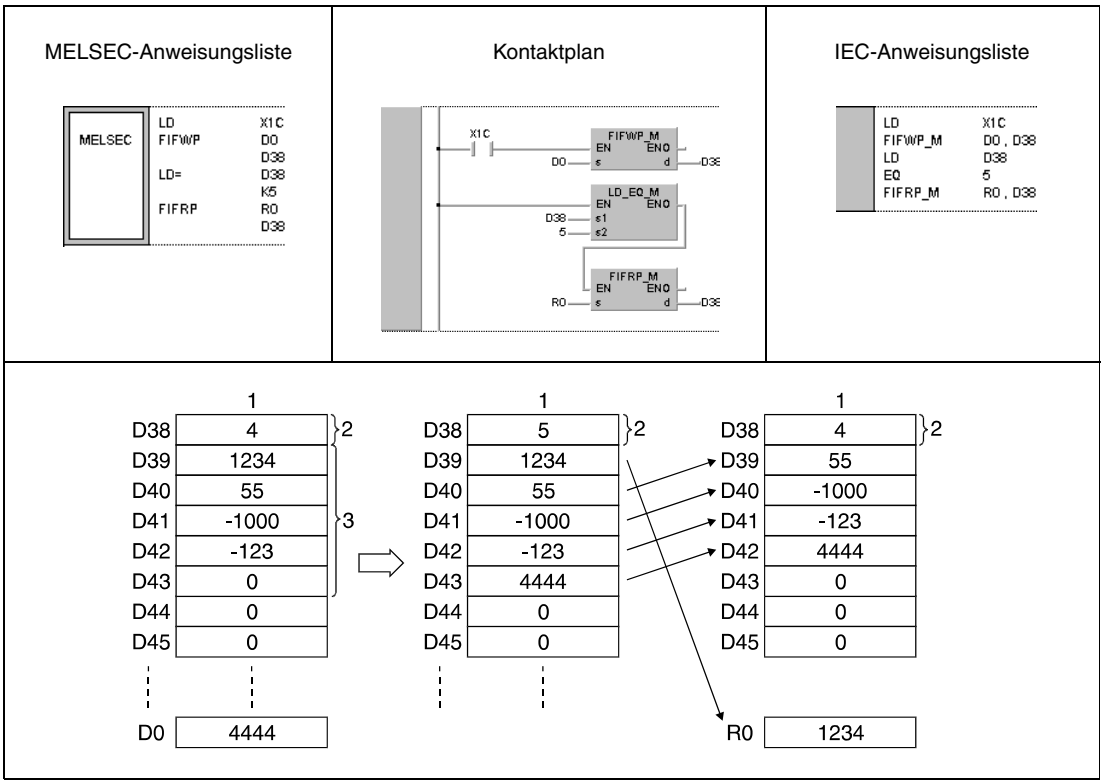
[1] Datenliste

[2] Positionszeiger

[3] Adressbereich der Datenliste

Beispiel 2 FIFRP

Das folgende Programm schreibt mit jeder positiven Flanke von X1C den Wert aus D0 in die Datenliste von D38 bis D43. Wenn der Wert des Positionszeigers 5 lautet, wird der erste Wert der FIFO-Liste ausgelesen und an R0 weitergegeben. Dieser Vorgang widerholt sich dann mit jeder positiven Flanke von X1C.



[1] Datenliste

[2] Positionszeiger

[3] Adressbereich der Datenliste

7.7.3 FPOP, FPOPP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Operanden MELSEC Q

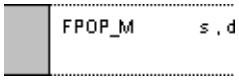
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten	Andere		
	Bit	Wort		Bit	Wort						
s	●	●	●	●	●	●	●	●	—	SM0	3
d	—	●	●	—	—	—	—	—	—		

GX IEC Developer

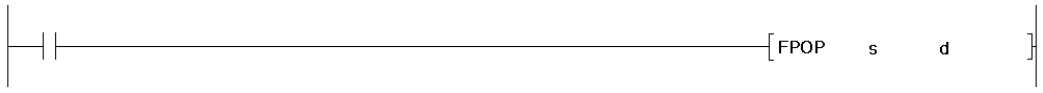
MELSEC-Anweisungsliste

Kontaktplan

IEC-Anweisungsliste



GX Developer



Variablen

Operand	Befehlswert	Datentyp
s	Startadresse des Operanden, in welchem die ausgelesenen Daten gespeichert werden.	BIN-16-Bit
d	Startadresse der Datenliste.	

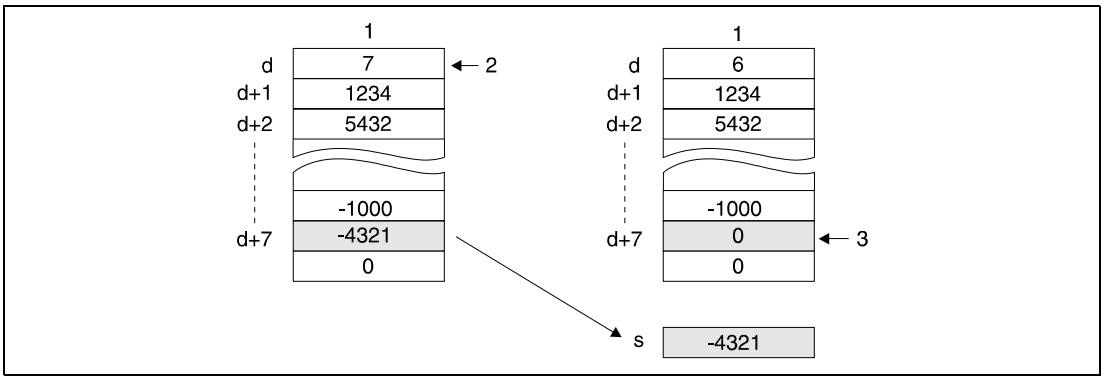
Funktions-weise

Lesen zuletzt eingegebener Daten aus der Datenliste

FPOP Anweisung zum Auslesen der zuletzt eingegebenen Daten

Die FPOP-Anweisung liest die Daten aus der Datenliste und speichert sie in einen vorgegebenen Adressenbereich. Das Auslesen der Daten beginnt mit der zuletzt beschriebenen Adresse d+n der Datenliste. Die Übertragung erfolgt an den in s angegebenen Adressenbereich.

Die ausgelesene Adresse der Datenliste wird mit dem Wert 0 beschieben. Nach dem Auslesen wird der Wert des Positionszeigers (erste Adresse in d) um 1 vermindert.



[1] Datenliste

[2] Positionszeiger

[3] Dieses Register wird mit dem Wert 0 beschrieben.

HINWEIS *Achten Sie darauf, dass die Anweisung nicht aufgerufen wird, wenn der Wert in d (Positionszeiger) gleich 0 ist.*

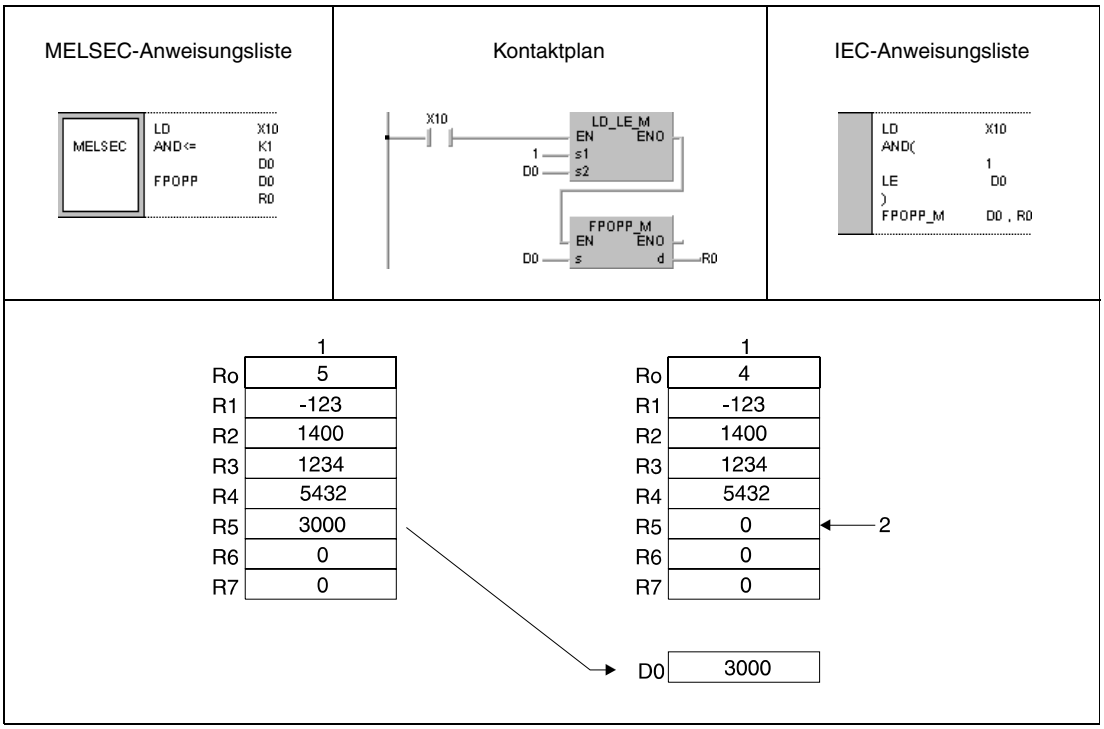
Fehler-quellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Eine FPOP-Anweisung wird ausgeführt, wenn der Wert des Positionszeigers gleich 0 ist (Fehlercode 4100).
- Die Länge der Datenliste überschreitet bei der Ausführung der FPOP-Anweisung den Operandenbereich (Fehlercode 4101).

Beispiel 1 FPOPP

Im folgenden Programm wird mit positiver Flanke von X10 der Datenwert der Adresse R5 (zuletzt eingegebener Wert) der Datenliste von R0 bis R7 ausgelesen und im Register D0 gespeichert. Im Beispiel lautet der Wert des Positionszeigers zu Beginn der Ausführung 5, nach der Ausführung 4. Die vorgeschaltete Vergleichsanweisung verhindert die Ausführung der FPOPP-Anweisung, wenn der Positionszeiger (R0) den Wert 0 besitzt.

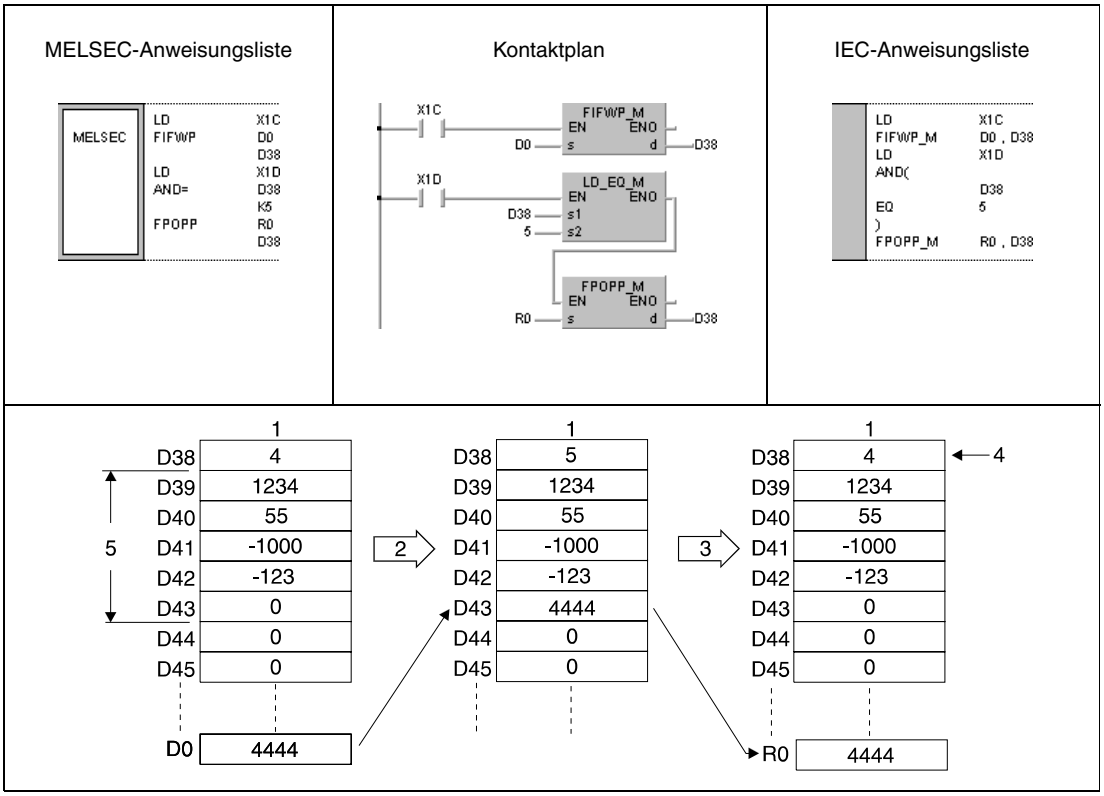


[1] Datenliste

[2] Dieses Register wird mit dem Wert 0 beschrieben.

Beispiel 2 FPOPP

Das folgende Programm schreibt mit jeder positiven Flanke von X1C den Wert aus D0 in die Datenliste von D38 bis D43. Wenn der Wert des Positionszeigers 5 beträgt, wird mit positiver Flanke von X1D der Wert des Registers D43 ausgelesen und an R0 weitergegeben.



[1] Datenliste

[2] Positive Flanke von X1C

[3] Positive Flanke von X1D

[4] Positionszeiger

[5] Aktueller Adressbereich der Datenliste

7.7.4 FDEL, FDELP, FINS, FINSP

CPU

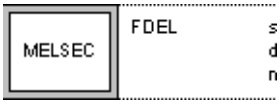
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Operanden MELSEC Q

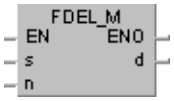
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	●	●	●	●	●	●	●	—	—	SM0	4
d	—	●	●	—	—	—	—	—	—		
n	●	●	●	●	●	●	●	●	—		

GX IEC Developer

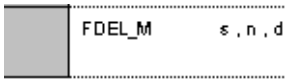
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

Operand	Befehlswert	Datentyp
s	Daten, die an einer bestimmten Adresse in die Datenliste eingefügt werden, oder Operand, in dem diese Daten gespeichert sind. Erste Adresse des Operanden, der die Daten, die an einer bestimmten Adresse in der Datenliste gelöscht werden, speichert.	BIN-16-Bit
d	Startadresse der Datenliste.	
n	Angabe der Adresse, an welcher Daten eingetragen oder gelöscht werden.	

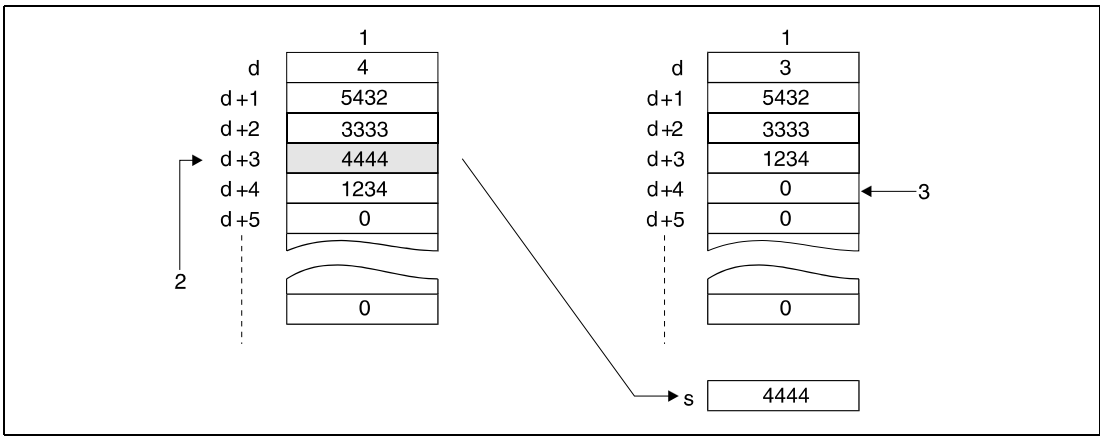
Funktionsweise

Löschen und Einfügen bestimmter Datenblöcke in der Datenliste

FDEL Löschen bestimmter Datenblöcke

Die FDEL-Anweisung löscht den n-ten Datenblock nach dem Positionszeiger in der in d angegebenen Datenliste und speichert diesen Wert in dem in s angegebenen Operanden.

Die Daten der Datenliste werden nach dem Löschen eines Datenblocks zusammengeschoben. Nach dem Auslesen wird der Wert des Positionszeigers (erste Adresse in d) um 1 vermindert.



[1] Datenliste

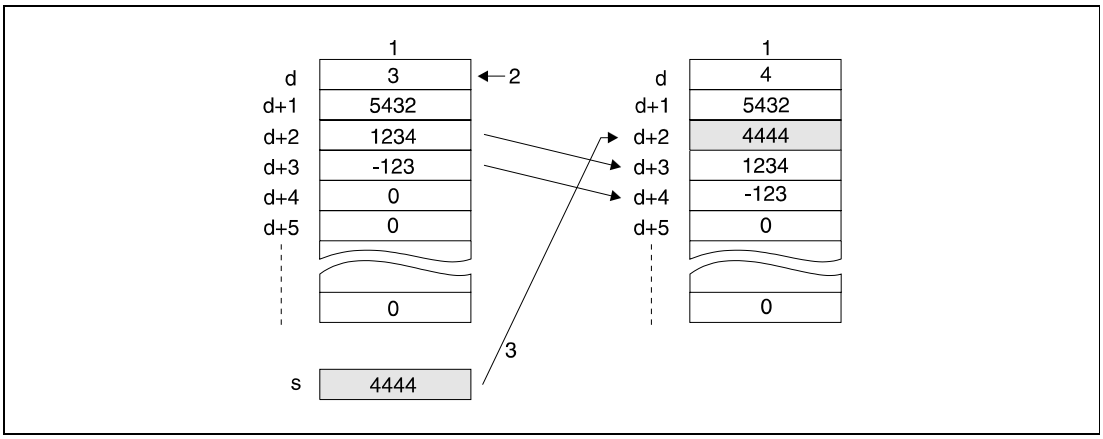
[2] Für n=3 wird der Datenblock d+3 gelöscht.

[3] Dieses Register wird mit dem Wert 0 beschrieben

FINS Einfügen bestimmter Datenblöcke

Die FINS-Anweisung fügt einen in s angegebenen 16-Bit-Datenblock an der n-ten Stelle nach dem Positionszeiger in der in d angegebenen Datenliste ein.

Die auf die Einfügestelle folgenden Datenblöcke werden um eine Adresse weitergeschoben. Nach dem Einfügen wird der Wert des Positionszeigers (erste Adresse in d) um 1 erhöht.



[1] Datenliste

[2] Positionszeiger

[3] Für n=2 wird der Datenblock als d+2 eingefügt

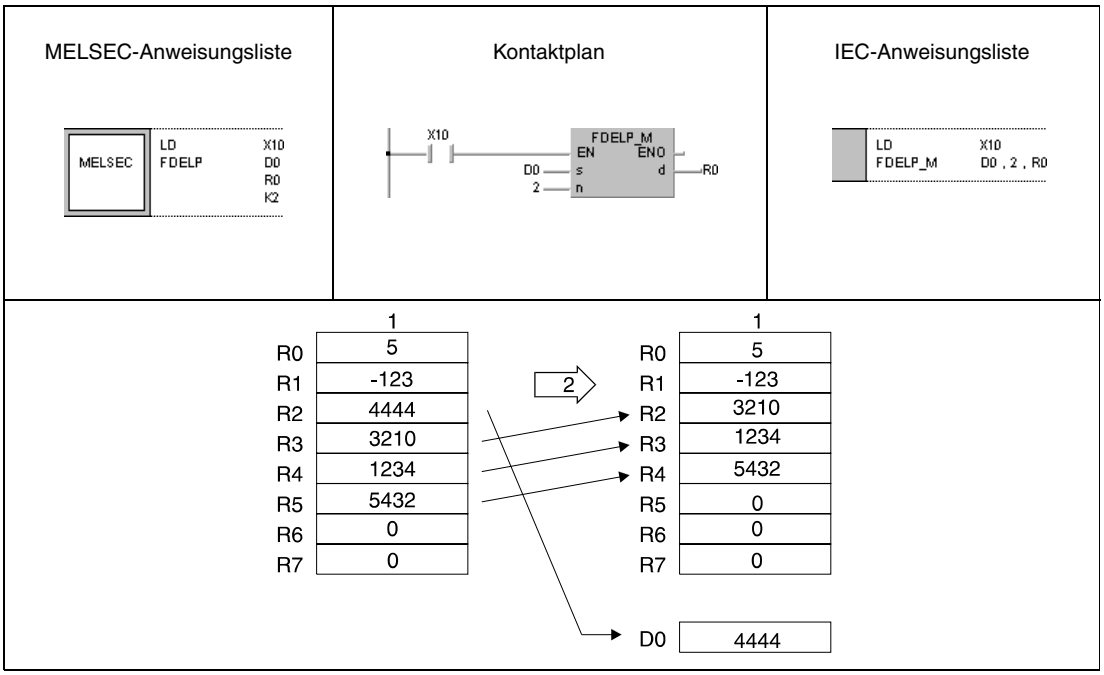
Fehler-quellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Der mit n angegebene Operand überschreitet bei der Ausführung der FDEL- oder FINS-Anweisung den Bereich des in d angegebenen Operanden (Fehlercode 4101).
- Der mit n angegebene Operand in d liegt bei der Ausführung der FDEL- oder FINS-Anweisung außerhalb des Adressbereichs der vorhandenen Datenblöcke + 1 (Fehlercode 4101).
- Die FDEL- oder FINS-Anweisung wurde ausgeführt, und in n war 0 eingetragen (Fehlercode 4100).
- Die FDEL-Anweisung wurde ausgeführt, und der Wert von d war 0 (Fehlercode 4100).
- Die Länge der Datenliste überschreitet bei der Ausführung der FPOP- oder FINS-Anweisung den Operandenbereich (Fehlercode 4101).

Beispiel 1 FDELP

Das folgende Programm löscht mit positiver Flanke von X10 den zweiten auf den Positionszeiger folgenden Datenblock (R2) in der Datenliste R0 bis R7 und speichert diesen Wert in D0.

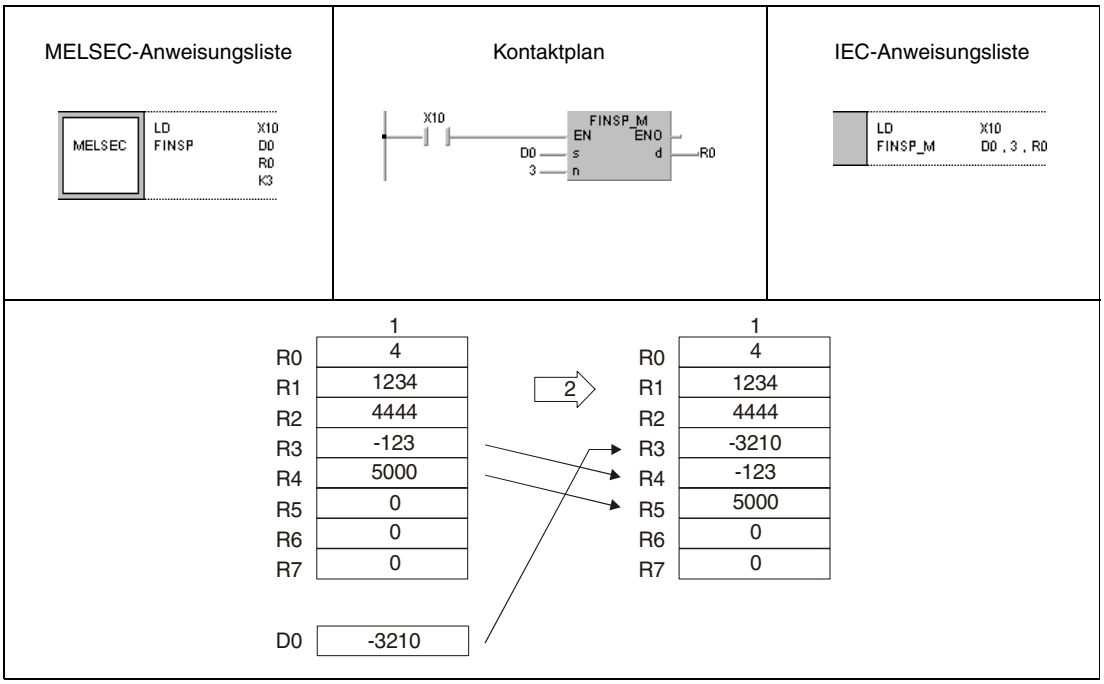


[1] Datenliste

[2] Positive Flanke von X10

Beispiel 2 FINSP

Das folgende Programm fügt mit positiver Flanke von X10 den in D0 angegebenen Datenblock als Datenblock R3 in die Datenliste R0 bis R7 ein.



[1] Datenliste

[2] Positive Flanke von X10

7.8 Anweisungen für den Pufferspeicherzugriff

Die im folgenden beschriebenen Anweisungen greifen auf den Pufferspeicher von Sondermodulen zu. Die Anweisungen erlauben der CPU Daten mit den entsprechenden Modulen auszutauschen. Die nachfolgende Tabelle enthält eine Übersicht der Anweisungen.

Funktion	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor
Daten aus einem Sondermodul lesen	FROM	FROM_M
	FROMP	FROMP_M
	DFRO	DFRO_M
	DFROP	DFROP_M
Daten in ein Sondermodul schreiben	TO	TO_M
	TOP	TOP_M
	DTO	DTO_M
	DTOP	DTOP_M

7.8.1 FROM, FROMP, DFRO, DFROP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
n1																	●	●				K1 ↓ K4 ●[2]	9 ●[3]	●		●
n2																										
d	●[1]	●[1]	●[1]	●[1]	●[1]	●[1]	●[1]	●[1]	●[1]	●[1]	●[1]	●[1]					●	●				K1 ↓ K8				
n3																	●	●								

[1] In Verbindung mit einer A3H CPU können keine Bit-Operanden verarbeitet werden.

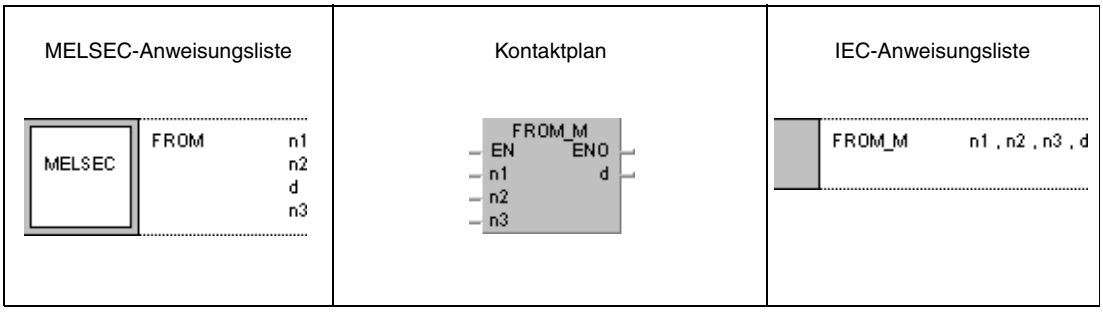
[2] Die Blocklänge kann bei Programmierung einer FROM/P-Anweisung zwischen K1 und K4 und bei einer DFRO/P-Anweisung zwischen K1 und K8 festgelegt werden.

[3] Die Anzahl der Schritte bei der Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

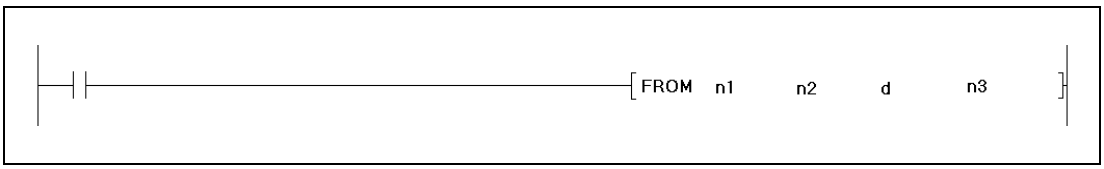
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort				U		
n1	●	●	●	●	●	●	●	●	●	SM0	5
n2	●	●	●	●	●	●	●	●	—		
d	●	●	●	—	—	—	—	—	—		
n3	●	●	●	●	●	●	●	●	—		

GX IEC Developer



GX Developer



Variablen

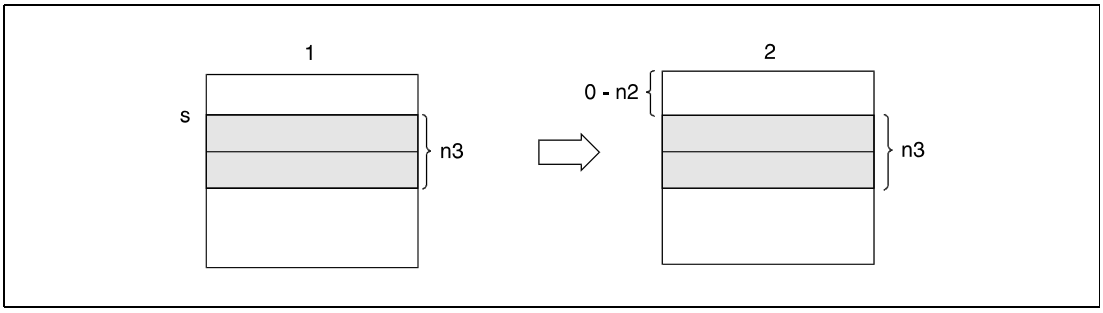
Operand	Befehlswert	Datentyp
n1	Kopfadresse des Sondermoduls auf dem Baugruppenträger.	BIN-16-Bit
n2	Startadresse des Adressenbereichs, aus dem gelesen wird.	BIN-16/-32-Bit
d	Startadresse des Adressenbereichs der CPU, in den geschrieben wird.	BIN-16-Bit
n3	Anzahl der zu lesenden Datenworte.	

Funktionsweise

Lesen von 1- und 2-Wort-Daten (32 Bit) aus einem Sondermodul

FROM Lesen von 1-Wort-Daten (16 Bit)

Die FROM-Anweisung liest 1-Wort-Daten aus dem Pufferspeicherbereich eines Sondermoduls und speichert sie in einem vorgegebenen Adressenbereich der CPU. Die Startadresse der zu lesenden Daten wird in n2, die Anzahl der Datenworte in n3 und die Adresse des Sondermoduls, die sich aus der Position des Moduls auf dem Baugruppenträger ergibt, in n1 vorgegeben. Der Adressenbereich der CPU, in dem die Daten gespeichert werden sollen, wird in d bestimmt.



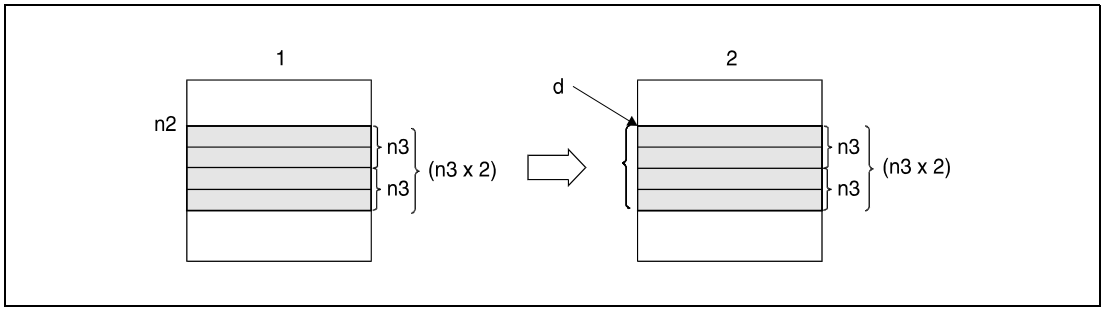
[1] Pufferspeicher des Sondermoduls

[2] Speicher der CPU

HINWEIS *Mit der FROM-Anweisung können im Multi-CPU-Betrieb auch Daten aus einer anderen CPU gelesen werden. Nähere Hinweise hierzu finden Sie in Kap. 9.6.2.*

DFRO Lesen von 2-Wort-Daten (32 Bit)

Die DFRO-Anweisung liest 2-Wort-Daten aus dem Pufferspeicher eines Sondermoduls. Die Startadresse der zu lesenden Daten wird in n2, die Anzahl der Datenworte (mal 2) in n3 und die Adresse des Sondermoduls in n1 vorgegeben. Der Adressenbereich der CPU, in dem die Daten gespeichert werden sollen, wird in d bestimmt.



[1] Pufferspeicher des Sondermoduls

[2] Speicher der CPU

HINWEIS *Eine CPU der Q-Serie oder des System Q kann außerdem direkt auf den Pufferspeicher von Sondermodulen zugreifen. In diesem Fall werden die Operanden in der Form U□\G□ (U(Kopfadresse des Sondermoduls)/G(Startadresse im Pufferspeicher)) angegeben.*

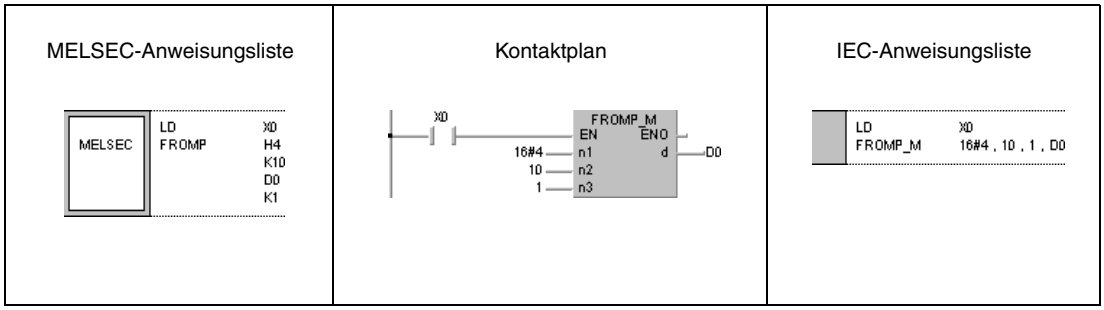
Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Vor Ausführung der Anweisung hat kein Datenaustausch mit dem Sondermodul stattgefunden (Q-Serie/System Q = Fehlercode 1412).
- Vor Ausführung der Anweisung ist in dem Sondermodul ein Fehler aufgetreten (Q-Serie/System Q = Fehlercode 1402).
- An der in n1 vorgegebenen E/A-Adresse befindet sich kein Sondermodul (Q-Serie/System Q = Fehlercode 2110).
- Die Anzahl der in n3 vorgegebenen Datenworte übersteigt den für die Speicherung vorgesehenen Bereich des Operanden in d (Q-Serie/System Q = Fehlercode 4101).
- Die in n2 vorgegebene Adresse liegt außerhalb des Bereichs des Pufferspeichers (Q-Serie/System Q = Fehlercode 4100).
- Die in n2 vorgegebene Adresse ist fehlerhaft (AJ71QC24) (Q-Serie/System Q = Fehlercode 4100).
- Der Zugriff auf ein Sondermodul ist nicht möglich.

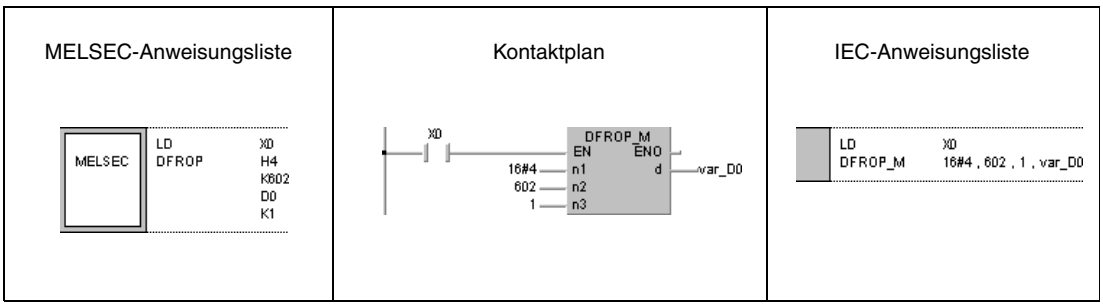
Beispiel 1 FROMP

Im folgenden Programm werden mit positiver Flanke von X0 die digitalen Werte des Kanals CH1 aus Adresse 10 des Pufferspeichers eines A68AD-Moduls gelesen. Der Adressenbereich des Moduls lautet 040 bis 05F. Die gelesenen Daten werden in D0 gespeichert.

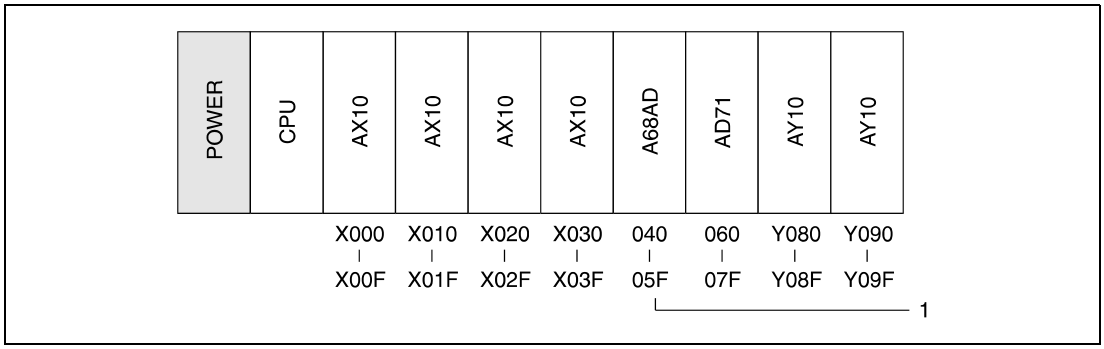


Beispiel 2 DFROP

Im folgenden Programm werden mit positiver Flanke von X0 die X-Achsen-Daten der Adressen 602 und 603 des Pufferspeichers eines AD71-Moduls gelesen. Der Adressenbereich des Moduls lautet 040 bis 05F. Die gelesenen Daten werden in D0 und D1 gespeichert.



HINWEISE *Die in n1 zu benennende Kopfadresse wird wie folgt festgelegt:*
n1 = 10 ⇒ Kopfadresse = 1
n1 = 20 ⇒ Kopfadresse = 2



[1] Startadresse des Sonderregisters: n1 = K4 oder H4

Das Programmbeispiel 2 ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.

7.8.2 TO, TOP, DTO, DTOP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)								Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
n1																	●	●				K1 ↓ K4				
n2																	●	●								
s	●[1]	●[1]	●[1]	●[1]	●[1]	●[1]	●[1]	●[1]	●[1]	●[1]	●[1]	●[1]					●[2]	●[2]				●[3] K1 ↓ K8	9/11 ●[4]	●		●
n3																	●	●								

[1] In Verbindung mit einer A3H CPU können keine Bit-Operanden verarbeitet werden.

[2] Der mögliche Bereich der Konstanten für s lautet: H0 bis FFFF, K-32768 bis 32767

[3] Die Blocklänge kann bei Programmierung einer TOP-Anweisung zwischen K1 und K4 und bei einer DTOP-Anweisung zwischen K1 und K8 festgelegt werden.

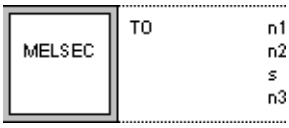
[4] Die Anzahl der Schritte bei der Verwendung einer AnA oder AnAS CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

Operanden MELSEC Q

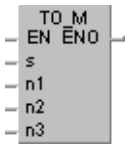
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sonder-module U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort				U		
n1	●	●	●	●	●	●	●	●	●	SM0	5
n2	●	●	●	●	●	●	●	●	—		
s	●	●	●	—	—	—	—	●	—		
n3	●	●	●	●	●	●	●	●	—		

GX IEC Developer

MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste

GX Developer

Variablen

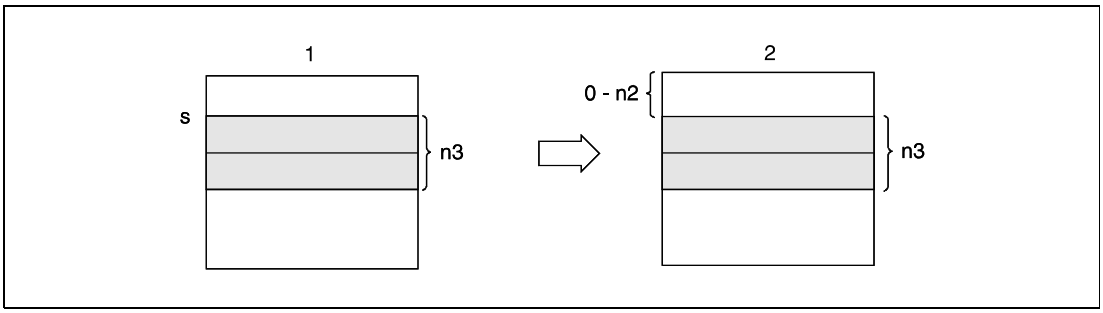
Operand	Befehlswert	Datentyp
n1	Kopfadresse des Sondermoduls auf dem Baugruppenträger.	BIN-16-Bit
n2	Startadresse des Adressenbereichs, in dem geschrieben wird.	
s	Zu schreibende Daten oder Startadresse des Adressenbereichs der CPU, in dem die zu schreibenden Daten gespeichert sind.	BIN-16-/32-Bit
n3	Anzahl der zu schreibenden Datenworte.	BIN-16-Bit

Funktionsweise

Schreiben in den Pufferspeicher

TO Schreiben von 1-Wort-Daten

Die TO-Anweisung schreibt 1-Wort-Daten aus dem Speicher der CPU in den Pufferspeicher eines Sondermoduls. Die Startadresse des Speicherbereiches, in den die Daten übertragen werden sollen, wird in n2, die Anzahl der Datenworte in n3 und die Adresse des Sondermoduls, die sich aus der Position des Moduls auf dem Baugruppenträger ergibt, in n1 vorgegeben. Die Startadresse des Adressenbereiches, aus dem die Daten gelesen werden sollen, wird in s bestimmt.

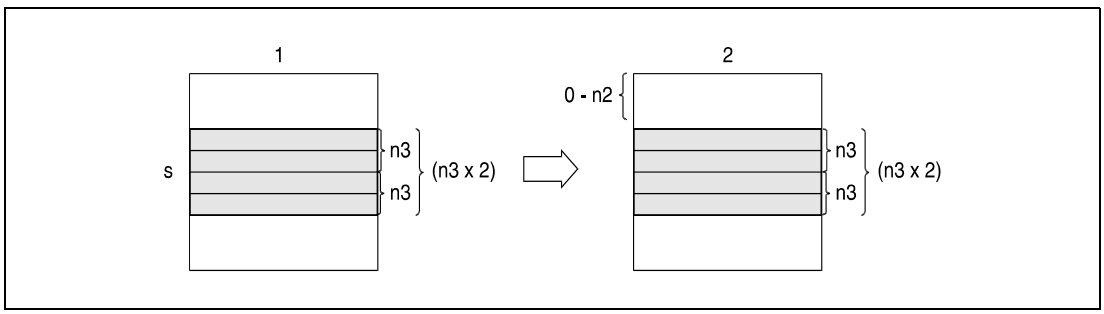


[1] Speicher der CPU

[2] Sondermodul-Pufferspeicher

DTO Schreiben von 2-Wort-Daten (32 Bit)

Die DTO-Anweisung schreibt 2-Wort-Daten (32 Bit) in den Pufferspeicher eines Sondermoduls. Die Startadresse des Speicherbereiches, in den die Daten übertragen werden sollen, wird in n2, die Anzahl der Datenworte (mal 2) in n3 und die Adresse des Sondermoduls in n1 vorgegeben. Die Startadresse des Adressenbereiches, aus dem die Daten gelesen werden sollen, wird in s bestimmt.



[1] Speicher der CPU

[2] Sondermodul-Pufferspeicher

Fehler-quellen

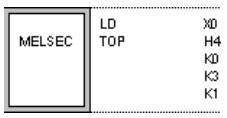
In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Vor der Ausführung der Anweisung hat kein Datenaustausch mit dem Sondermodul stattgefunden (Q-Serie/System Q = Fehlercode 1412).
- Vor Ausführung der Anweisung ist in dem Sondermodul ein Fehler aufgetreten (Q-Serie/System Q = Fehlercode 1402).
- An der in n1 vorgegebenen E/A-Adresse befindet sich kein Sondermodul (Q-Serie/System Q = Fehlercode 2110).
- Die Anzahl der in n3 vorgegebenen Datenworte übersteigt den für die Speicherung vorgesehenen Bereich des Operanden in s (Q-Serie/System Q = Fehlercode 4101).
- Die in n2 vorgegebene Adresse liegt außerhalb des Bereichs des Pufferspeichers (Fehlercode 4100).
- Die in n2 vorgegebene Adresse ist fehlerhaft (AJ71QC24) (Q-Serie/System Q =Fehlercode 4100).
- Der Zugiff auf ein Sondermodul ist nicht möglich.

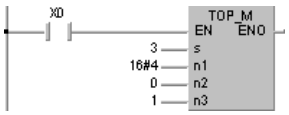
Beispiel 1 TOP

Im folgenden Programm wird mit der positiven Flanke von X0 die A/D-Wandlung der Kanäle CH1 und CH2 eines A68AD-Moduls eingeschaltet. Das Sondermodul belegt die Adressen 040 bis 05F. In den Pufferspeicher wird an Adresse 0 der Wert 3 geschrieben.

MELSEC-Anweisungsliste



Kontaktplan



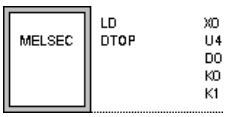
IEC-Anweisungsliste

```
LD    X0
TOP_M 3 , 16#4 , 0 , 1
```

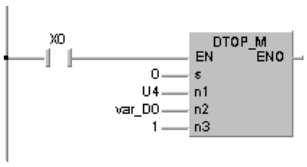
Beispiel 2 DTOP

Im folgenden Programm werden mit der positiven Flanke von X0 die X-Datenwerte an den Pufferspeicheradressen var_D0 eines AD71-Moduls auf 0 gesetzt. Das Sondermodul befindet sich an Adresse 040 bis 05F.

MELSEC-Anweisungsliste



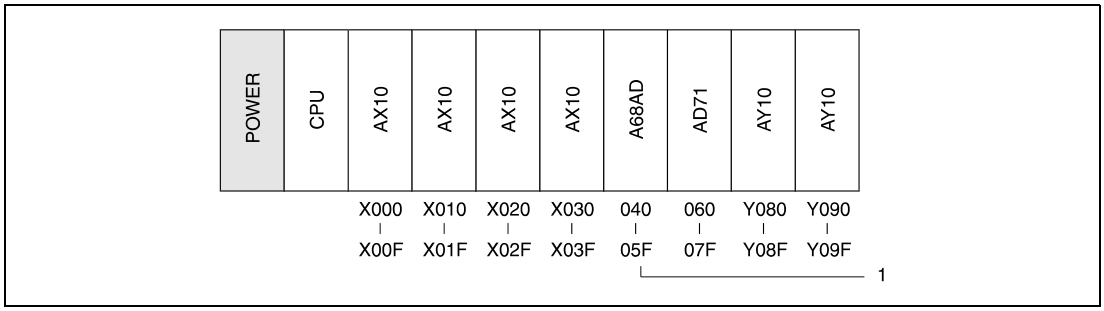
Kontaktplan



IEC-Anweisungsliste

```
LD    X0
DTOP_M 0 , U4 , var_D0 , 1
```

HINWEIS *Die in n1 zu benennende Kopfadresse wird wie folgt festgelegt:*
n1 = 10 ⇒ Kopfadresse = 1
n1 = 20 ⇒ Kopfadresse = 2



[1] Startadresse des Sonderregisters: n1 = K4 oder H4

Das Programmbeispiel 2 ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.

7.9 Display-Anweisungen

Die MELSEC Q- und die MELSEC A-Serie verfügen über eine Reihe von Anweisungen, mit deren Hilfe ASCII-Zeichen an die Ausgänge eines Ausgangsmoduls ausgegeben oder über das LED-Display an der Frontseite der dafür geeigneten CPU-Module angezeigt werden können. Insgesamt stehen 7 verschiedene Display-Anweisungen zur Verfügung.

Funktion	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor
ASCII-Ausgabe	PR	PR_M
	PRC	PRC_M
Anzeige von ASCII-Zeichen und Kommentaren	LED	LED_M
	LEDC	LEDC_M
	LEDA	LEDA_M
	LEDB	LEDB_M
Anzeige löschen	LEDR	LEDR_M

HINWEISE *Die LEDA- und LEDB-Anweisung kann in Verbindung mit einer A3A CPU nicht ohne Weiteres als Display-Anweisung verarbeitet werden. Die Anweisungen dienen hier als Startbefehl für die Erweiterten Applikationsanweisungen.*

Um die in diesem Abschnitt beschriebene Funktion der LEDA- und LEDB-Anweisung mit einer A3A CPU ausführen zu können, muss die Zeichenfolge der Daten mit Hilfe der Erweiterten Applikationsanweisungen der AnA oder AnAS CPUs geändert werden. Genaue Angaben hierzu enthält die gesonderte Programmieranleitung der AnA- und AnAS-Serie (Dedicated Instructions).

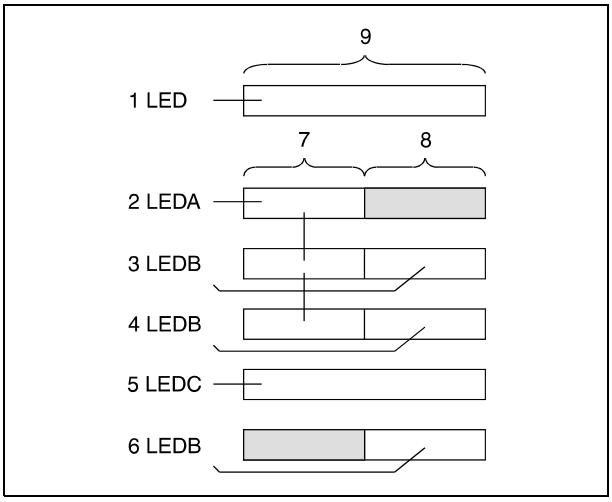
Die LED-Anzeige der CPU unterliegt folgender Anzeigenpriorität:

1. Fehleranzeige nach Selbstdiagnose
2. Anzeige über CHK-Anweisung
3. Anzeige der Nummer eines Fehlermerkers F
4. Anzeige von ASCII-Zeichen über LED(A,B,C)-Anweisung
5. „BATTERY ERROR“

Bei Verwendung einer A3A CPU kann diese Priorität beliebig geändert werden. Nähere Angaben sind den Handbüchern der AnA-Serie zu entnehmen.

Steht auf der Anzeige eine der ersten drei Meldungen, ändert die Ausführung einer Display-Anweisung den augenblicklichen Inhalt der Anzeige nicht. Steht dagegen auf der Anzeige die Meldung „BATTERY ERROR“, ändert sich der Inhalt der Anzeige mit der Ausführung der LED(A,B,C)-Anweisung.

Die Abbildung zeigt schematisch die Darstellungsweise auf dem LED-Display nach Ausführung einer LED(A,B,C)-Anweisung.



Bei Ausführung einer LED-Anweisung werden bis zu 16 Zeichen (9) auf der Anzeige dargestellt (1). Nach Ausführung einer LEDA-Anweisung werden die ersten 8 Zeichen (7) genutzt. Die letzten 8 Stellen (8) bleiben frei (2). Wird anschließend eine LEDB-Anweisung ausgeführt, werden die Daten in der zweiten Hälfte des Displays angezeigt (3). Eine wiederholte Ausführung der LEDB-Anweisung überschreibt die ursprünglichen Daten in der zweiten Hälfte. Die Daten der ersten 8 Stellen bleiben unverändert (4). Nach Ausführung einer LEDC-Anweisung wird ein vorgegebener Kommentar (15 Zeichen) zur Anzeige gebracht (5). Die Ausführung einer LEDB-Anweisung überschreibt die ursprünglichen Daten wieder. Die ersten 8 Stellen bleiben frei (6).

Auf dem LED-Display der dafür geeigneten CPUs können Zahlen zwischen 0 und 9, alle Buchstaben zwischen A und Z (nur Großbuchstaben ohne Umlaute) sowie die folgenden Sonderzeichen dargestellt werden: < > = x / ' + -

7.9.1 PR

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●[1]

[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC A

	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
s								●	●	●	●	●	●										7 ●[1]			●
d		●																								

[1] Die Anzahl der Schritte bei der Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

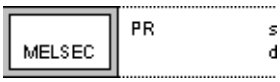
Operanden MELSEC Q

	Operanden								Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sonder-module U□\G□	Index-Register Zn	Konstanten $	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	—	—	●	●	—	—	3
d	●[1]	—	—	—	—	—	●	—	—		

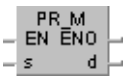
[1] Nur Y

GX IEC Developer

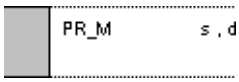
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

Operand	Befehlswert	Datentyp
s	Erste Adresse des Operanden, in dem der ASCII-Code gespeichert ist.	Zeichenfolge
d	Startadresse des Ausgabemoduls, an welchem der ASCII-Code ausgegeben wird.	BIT

Funktions-weise

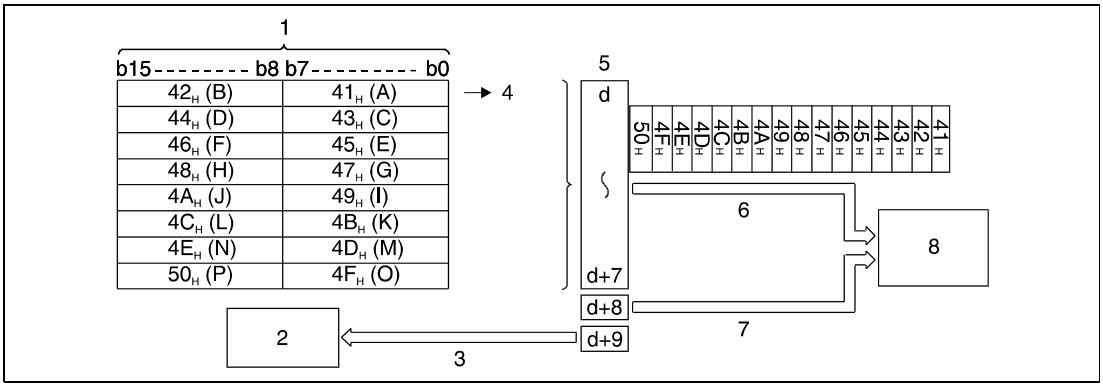
Ausgabe an ein peripheres Gerät

PR Ausgabe einer ASCII-Zeichenfolge

Die PR-Anweisung verfügt über zwei Funktionen. Die Funktion ist von dem Zustand des Sondermerkers M9049 (A-Serie) bzw. des Merkers SM701 (Q-Serie/System Q) abhängig.

M9049/SM701 gesetzt (1)(Funktion 1):

Ausgabe einer ASCII-Zeichenfolge mit 16 Zeichen an ein Ausgangsmodul. Die Zeichenfolge wird, aufgeteilt in 2 mal 8 Zeichen, aus dem Adressenbereich s gelesen und an die in d vorgegebenen Ausgänge ausgegeben.



[1] Operand, in dem der ASCII-Code gespeichert ist

[2] Unterprogramm

[3] Flag zur Anzeige der Verarbeitung der PR-Anweisung (wird als Interlock verwendet)

[4] Anfang der Ausgabe

[5] Ausgänge Y

[6] Ausgabe des ASCII-Codes

[7] Ausgabe des Strobe-Signals

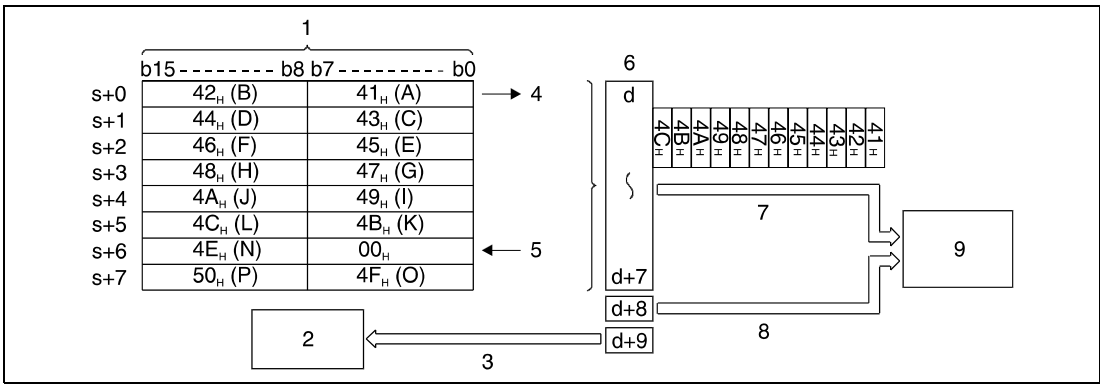
[8] Drucker des Anzeigegeräts

Die PR-Anweisung kann nur auf bereits gespeicherte ASCII-Daten zugreifen. Ändern sich die gespeicherten Daten, werden die aktuellen Daten ausgegeben. Zur Konvertierung von alphanumerischen Zeichen in einen ASCII-Code ist die ASC-Anweisung zu verwenden.

Während der Übertragung des aus 16 Zeichen bestehenden ASCII-Codes wird eine Ausführungskennung der PR-Anweisung in Adresse d+9 gesetzt. Das bedeutet, dass der Ausgang Y mit der Adresse d+9 so lange eingeschaltet ist, wie die Ausführung der PR-Anweisung andauert.

M9049/SM701 nicht gesetzt (0)(Funktion 2):

Ausgabe einer ASCII-Zeichenfolge bis zu dem Zeichencode "00H" aus dem Adressenbereich s als Hexadezimalcode an die in d vorgegebenen Ausgänge.



[1] Operand, in dem der ASCII-Code gespeichert ist

[2] Unterprogramm

[3] Flag zur Anzeige der Verarbeitung der PR-Anweisung (wird als Interlock verwendet)

[4] Anfang der Ausgabe

[5] Ende der Zeichenfolge (Übertragung)

[6] Ausgänge Y

[7] Ausgabe des ASCII-Codes

[8] Ausgabe des Strobe-Signals

[9] Drucker oder Anzeigegerät

HINWEIS *Mit einer A-Serie CPU kann ausschließlich die 1. Funktion ausgeführt werden.*

Wenn der Inhalt der Operandenadressen, die den ASCII-Code speichern, noch während der Ausgabe überschrieben wird, werden die neuen Daten ausgegeben.

Das Ende der ASCII-Zeichenfolge ist durch den Zeichencode "00H" gekennzeichnet.

Liegt der Hexadezimalcode "00H" in dem vorgegebenen Operand nicht vor, wird die Verarbeitung abgebrochen und eine Fehlerkennung gesetzt.

Während der Übertragung des ASCII-Codes wird eine Ausführungskennung der PR-Anweisung in Adresse Y= d+9 gesetzt.

Für die Ausführung der PRC-Anweisung wird ein Ausgangsmodul mit 10 aufeinanderfolgenden binären Ausgängen benötigt. Der Adressenbereich beginnt mit der in d vorgegebenen Ausgangsadresse.

Die Verarbeitungszeit eines Ausgangssignals in einem Ausgangsmodul beträgt 30 ms pro Zeichen. Bis zur Verarbeitung von n Zeichen werden somit n x 30 ms benötigt. Da die Ausgabesteuerung während der Übertragung mittels Interrupt-Verarbeitung in Intervallen zu 10 ms geregelt wird, läuft das Ablaufprogramm kontinuierlich ab.

Die Verarbeitung der 10 Ausgangsadressen im Ausgangsmodul erfolgt unabhängig von einer Aktualisierung der E-/A-Zustände (E-/A-Refresh) nach einer END-Anweisung in der Ablauffolge.

Zusätzlich zum ASCII-Code wird ein Strobe-Signal (EIN = 10 ms, AUS = 20 ms) über Adresse Y= d+8 ausgegeben.

PR- und PRC-Anweisungen können mehrfach hintereinander ausgeführt werden. Die Programmierung muss jedoch so erfolgen, dass die gleichzeitige Ausführung von zwei oder mehr PR-/PRC-Anweisungen ausgeschlossen werden kann. Aus diesem Grund sollte die Ausführungskennung (Ausgangsoperand in Y= d+9) als Verriegelung der Anweisung programmiert werden.

Beispiel PR

Das folgende Programm konvertiert mit positiver Flanke von X0 die Zeichenfolge "ABCDEF-GHIJKLMNOP" in einen ASCII-Code und speichert diesen in den Datenregistern D0 bis D7 ab. Nach dem Einschalten von X3 wird der ASCII-Code aus D0 bis D7 an die Ausgänge Y14 bis Y1D ausgegeben.

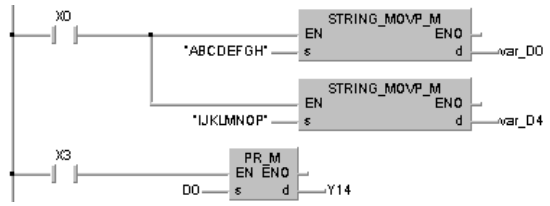
MELSEC-Anweisungsliste

```
MELSEC  LD      X0
        $MOVP   "ABCDEFGH"
                D0
        $MOVP   "IJKLMNOP"
                D4
        LD      X3
        PR      D0
                Y14
```

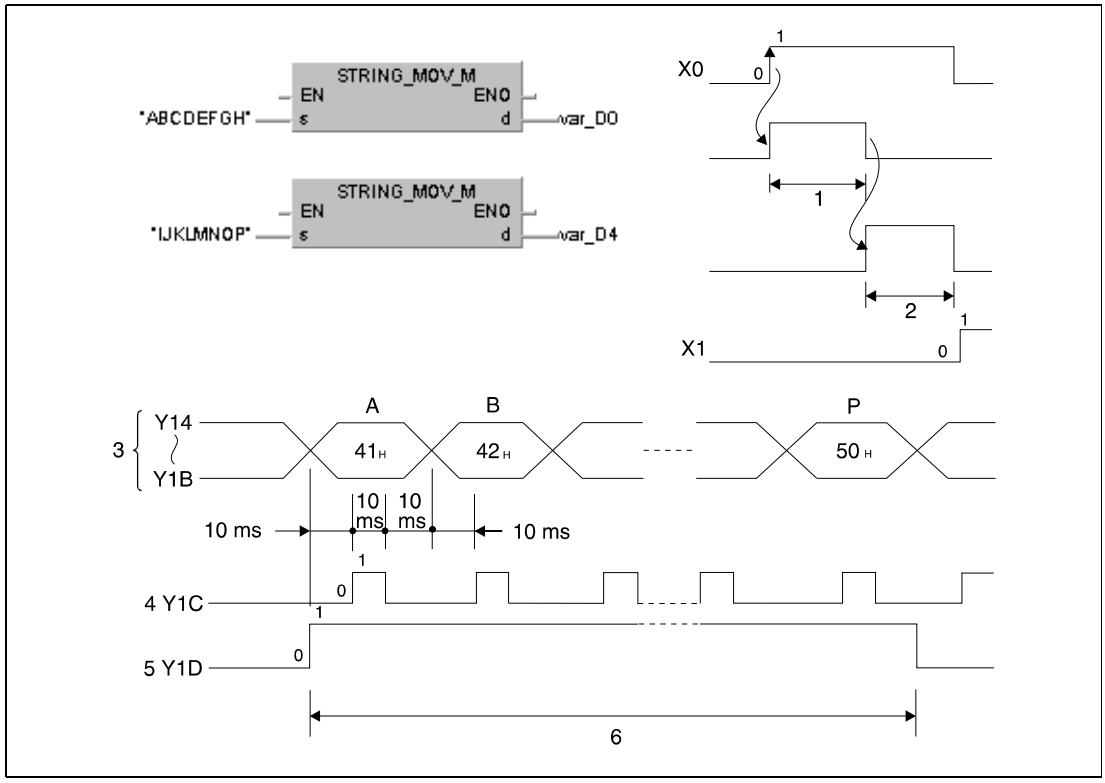
IEC-Anweisungsliste

```
        LD                X0
        STRING_MOVP_M     "ABCDEFGH" , var_D0
        STRING_MOVP_M     "IJKLMNOP" , var_D4
        LD                X3
        PR_M              D0 , Y14
```

Kontaktplan



Die folgenden Signalflussdiagramme veranschaulichen die Verarbeitungsweise des Programms.



[1] Speichern der Zeichenfolge "ABCDEFGH" in D0 bis D3

[2] Speichern der Zeichenfolge "IJKLMNOP" in D4 bis D7

[3] ASCII-Code

[4] Strobe-Signal

[5] Flag zur Anzeige der Verarbeitung der PR-Anweisung

[6] Verarbeitung der PR-Anweisung (Dauer = 480 ms)

HINWEISE *Wenn keine A-Serie CPU verwendet wird und SM701 nicht gesetzt ist, muss in Register D8 der Wert "00H" geschrieben werden. Ohne diesen Zeichencode würde im obenstehenden Beispiel ein Fehler auftreten.*

Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen. Compiler- oder Checker-Fehlermeldungen können die Folge sein.

7.9.2 PRC

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●[1]

[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC A

	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
s	●	●	●	●	●	●	●	●	●	●	●	●							●	●			7 ●[1]	●		
d		●																								

[1] Die Anzahl der Schritte bei der Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

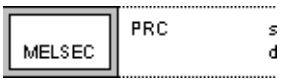
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sonder-module U□\G□	Index-Register Zn	Konstanten $	Andere		
	Bit	Wort		Bit	Wort				P, I, J, U		
s	●	●	●	●	●	●	—	—	●	—	3
d	●[1]	—	—	—	—	—	—	—	—	—	

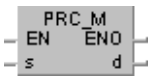
[1] Nur Y

GX IEC Developer

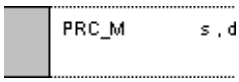
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

Operand	Befehlswert	Datentyp
s	Erste Adresse des Operanden, in dem der auszugebende Kommentar gespeichert ist.	BIN-16-Bit
d	Adresse des Ausgabemoduls, an welchem der Kommentar ausgegeben wird.	Bit

Funktions-weise

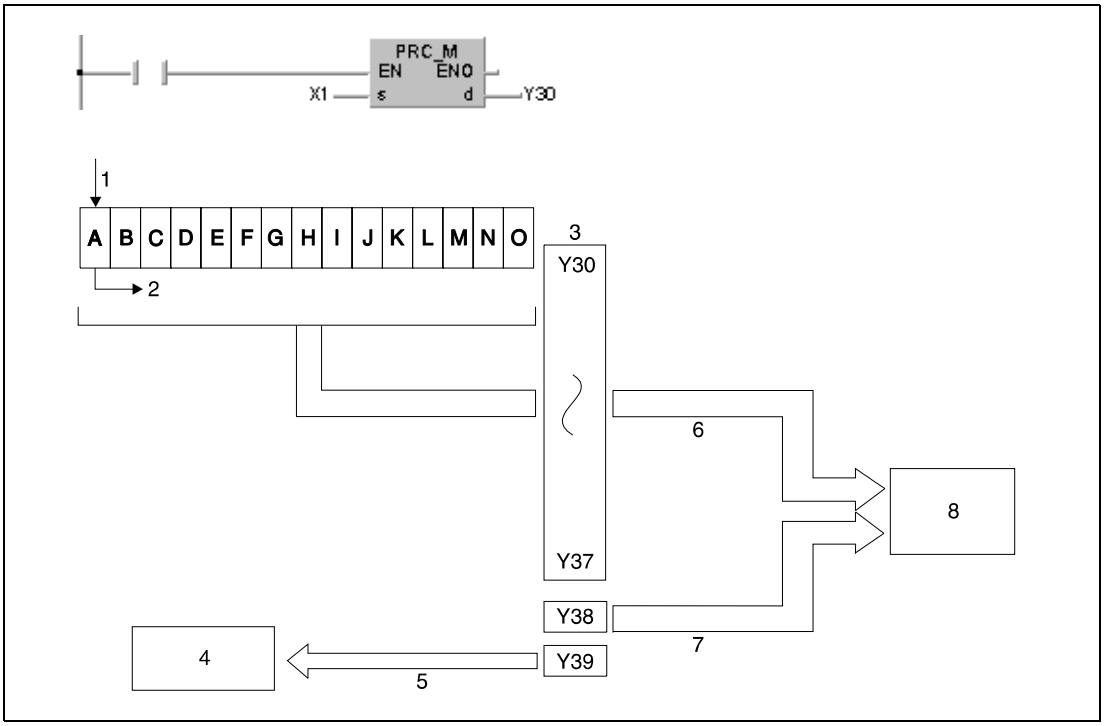
Ausgabe an ein peripheres Gerät

PRC Ausgabe eines Kommentares

Die PRC-Anweisung gibt einen Kommentar eines Operanden (in ASCII-Code) an ein Ausgangsmodul aus.

Bei der MELSEC A-Serie wird die Zeichenfolge, aufgeteilt in 2 mal 8 Zeichen, aus dem Adressenbereich s gelesen und an die in d vorgegebenen Ausgänge ausgegeben.

Bei der Q-Serie bzw. dem System Q kann eine Auswahl zwischen der Ausgabe von 16 oder 32 Zeichen getroffen werden. Die Auswahl erfolgt über den Sondermerker SM701. Ist SM701 gesetzt (1), werden 16 Zeichen ausgegeben, und ist SM701 nicht gesetzt (0), werden 32 Zeichen ausgegeben.



[1] Kommentar (ASCII-Code) ab X1

[2] Anfang der Ausgabe

[3] Ausgänge Y

[4] Unterprogramm

[5] Flag zur Anzeige der Verarbeitung der PR-Anweisung (wird als Interlock verwendet)

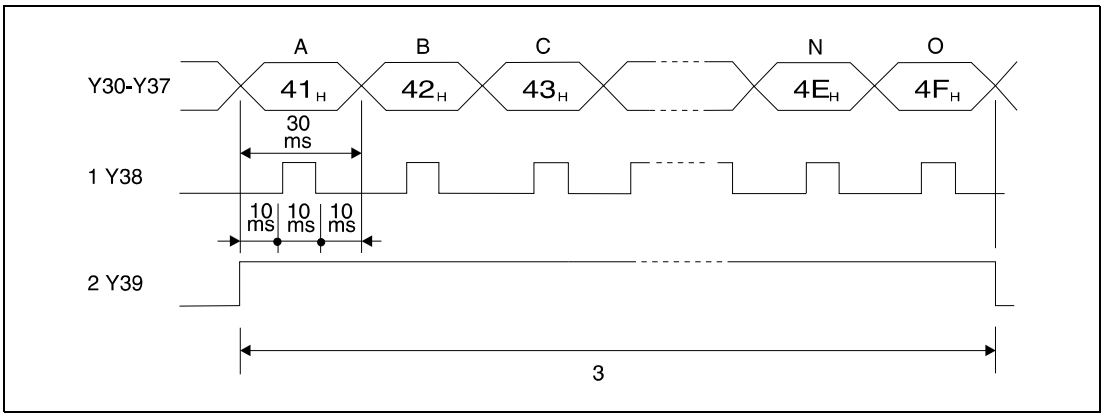
[6] Ausgabe des ASCII-Codes

[7] Ausgabe des Strobe-Signals

[8] Drucker oder Anzeigegerät

Die folgenden Signalflussdiagramme veranschaulichen die Verarbeitungsweise der PRC-Anweisung.

Signalverlauf bei einer QnA-CPU:

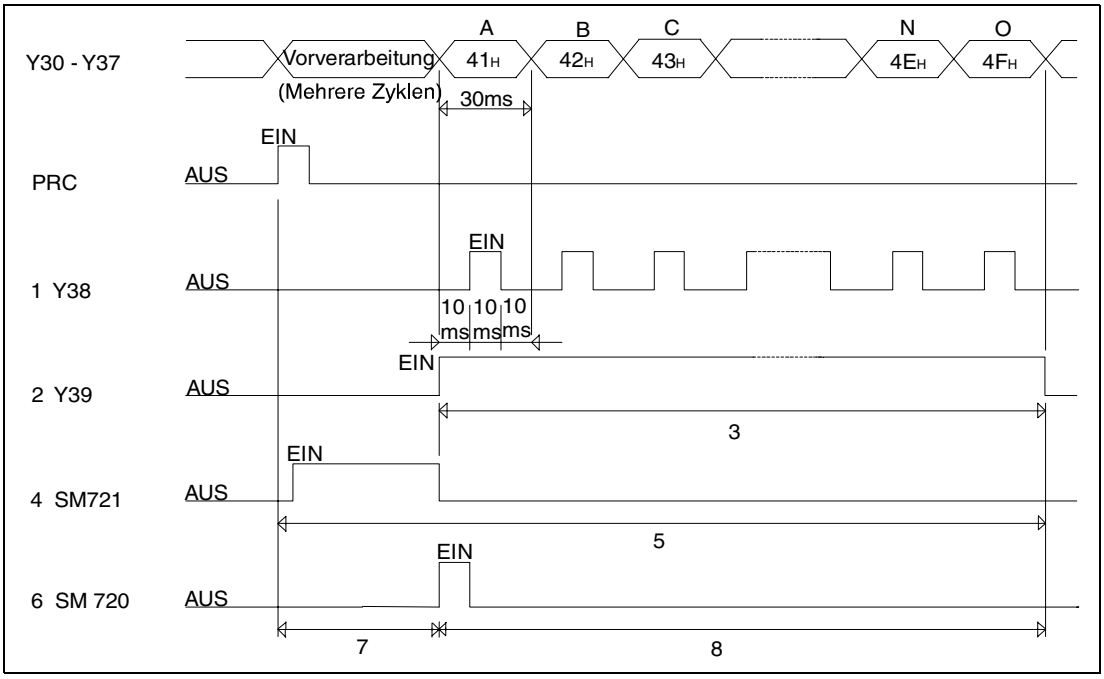


[1] Strobe-Signal

[2] Signal zur Anzeige der Verarbeitung der PRC-Anweisung

[3] Verarbeitung der PRC-Anweisung (Dauer = 16 x 30 ms = 480 ms)

Signalverlauf bei einer Multi-Prozessor-Q-CPU:



[1] Strobe-Signal

[2] Signal zur Anzeige der Signalausgabe

[3] Verarbeitung der PRC-Anweisung (Dauer = 16 x 30 ms = 480 ms)

[4] Anzeige, dass auf ein File zugegriffen wird

[5] Während dieser Zeit kann keine andere PRC-Anweisung ausgeführt werden

[6] Anzeige, das der Zugriff auf das File beendet ist

[7] Während dieser Zeit können keine anderen Anweisungen ausgeführt werden

[8] Alle Anweisungen ausser PRC, S.FREAD, S.FWRITE, PLOAD, PUNLOAD und PSWAPP können ausgeführt werden

Es werden 10 binäre Ausgänge eines digitalen Ausgangsmoduls belegt. Der Adressenbereich beginnt mit der in d vorgegebenen Ausgangsadresse Y.

Die Verarbeitungszeit eines Ausgangssignals in einem Ausgangsmodul beträgt 30 ms pro Zeichen. Bis zur Verarbeitung von n Zeichen werden somit n x 30 ms benötigt. Da die Ausgabesteuerung während der Übertragung mittels Interrupt-Verarbeitung in Intervallen zu 10 ms geregelt wird, läuft das Ablaufprogramm kontinuierlich ab.

Zusätzlich zum ASCII-Code wird ein Strobe-Signal (EIN = 10 ms, AUS = 20 ms) über Adresse Y = d+8 ausgegeben.

Während der Übertragung des aus 16 Zeichen bestehenden ASCII-Codes wird eine Ausführungskennung der PRC-Anweisung in Adresse d+9 gesetzt, d.h. der Ausgang Y mit der Adresse d+9 ist so lange eingeschaltet, wie die Ausführung der PRC-Anweisung andauert. PR- und PRC-Anweisungen können mehrfach hintereinander ausgeführt werden. Die Programmierung muss jedoch so erfolgen, dass die gleichzeitige Ausführung von zwei oder mehr PR-/PRC-Anweisungen ausgeschlossen werden kann. Aus diesem Grund sollte die Ausführungskennung (Ausgangsoperand in d+9) als Verriegelung der Anweisung programmiert werden.

Wenn im Adressenbereich s keine Daten vorhanden sind, wird die Anweisung nicht ausgeführt.

Die PRC-Anweisung kann nur auf bereits in der SPS gespeicherte Kommentare zugreifen. Zur Konvertierung von alphanumerischen Zeichen in einen ASCII-Code ist die ASC-Anweisung zu verwenden.

Der Sondermerker SM720 der Q-Serie/System Q wird nach Ausführung der Anweisung für einen Zyklus gesetzt. SM721 wird während der Ausführung der PRC-Anweisung gesetzt. Die PRC-Anweisung kann nicht angewählt werden, wenn SM721 gesetzt ist.

HINWEISE

Die Kommentare, auf die mit der PRC-Anweisung zugegriffen wird, müssen in der Speicherkarte abgelegt sein. Es kann nicht auf Kommentare im internen Speicher zugegriffen werden.

Die Kommentardatei auf die die PRC-Anweisung zugreift, wird im Parametriermodus unter „PC File Setting“ eingestellt. Eine Ausgabe von Kommentaren mit der PRC-Anweisung ist nicht möglich, wenn keine Datei eingestellt ist

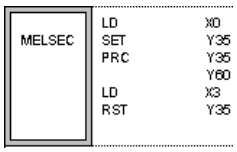
Die PRC-Anweisung darf nicht in einem Interruptprogramm ausgeführt werden. Fehlfunktionen können die Folge sein.

Die Kommentar-Operanden für die PRC-Anweisung werden auf der IC-Speicherkarte gespeichert. Im internen Speicher der CPU können keine Kommentare gespeichert werden (nur Q-Serie).

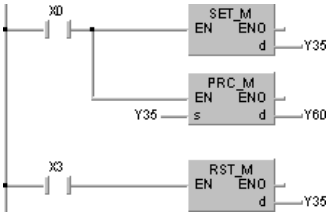
Beispiel PRC

Im folgenden Programm wird nach dem Einschalten von X0 Ausgang Y35 gesetzt und gleichzeitig der Kommentar von Y35 als ASCII-Code an die Ausgänge Y60 bis Y69 ausgegeben. Nach dem Einschalten von X3 wird Y35 zurückgesetzt.

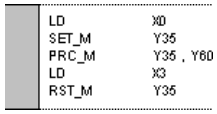
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



7.9.3 LED

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
	●[1]	●[2]	●	●[3]	

[1] Nur für A3N CPU
[2] Nur für A3A CPU
[3] Nicht für Q2A(S1) CPU

Operanden MELSEC A

	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)								Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
s								●	●	●	●	●											3 ●[1]	●		●

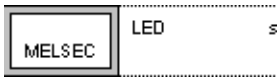
[1] Die Anzahl der Schritte bei der Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

Operanden MELSEC Q

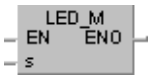
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten $	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	—	—	—	●	—	—	2

GX IEC Developer

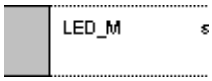
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



Variablen

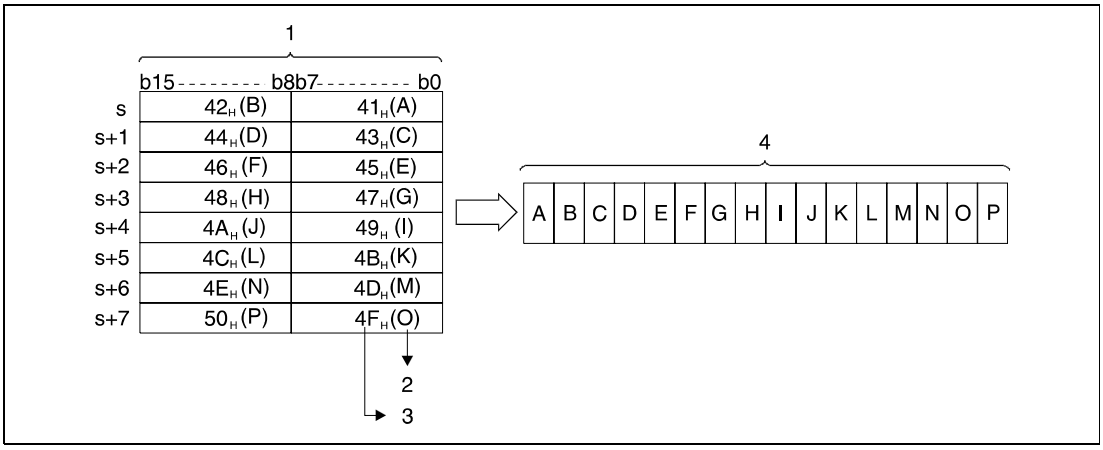
Operand	Befehlswert	Datentyp
s	Erste Adresse des Operanden, in welchem die darzustellenden ASCII-Daten gespeichert sind.	Zeichenfolge

Funktionsweise

Ausgabe an LED-Anzeige

LED Gespeicherte ASCII-Daten am LED-Display der CPU anzeigen

Die LED-Anweisung ruft ASCII-Daten (16 Zeichen) aus einem vorgegebenen Adressenbereich auf und bringt sie auf dem LED-Display einer dafür geeigneten CPU zur Anzeige. Die Startadresse des in 8 Adressen gespeicherten ASCII-Codes wird in s festgelegt (siehe folgendes Schema).



1 Darzustellende Daten

2 ASCII-Zeichen

3 ASCII-Code (hexadezimal)

4 LED-Display an der CPU-Front

Sind in dem angegebenen Adressenbereich keine ASCII-Daten vorhanden, bleibt die Anzeige bei Timern, Countern sowie Daten- und Link-Registern leer. Bei File-Registern R erfolgt eine willkürliche Anzeige. Die Anzeige bleibt leer, wenn die entsprechenden File-Register bereits gelöscht sind.

Auf dem LED-Display der dafür geeigneten CPUs können Zahlen zwischen 0 und 9, alle Buchstaben zwischen A und Z (nur Großbuchstaben ohne Umlaute) sowie die folgenden Sonderzeichen dargestellt werden: < > = x / ' + -

Die LED-Anweisung kann nur auf bereits gespeicherte ASCII-Daten zugreifen. Zur Konvertierung von alphanumerischen Zeichen in einen ASCII-Code ist die $MOV- bzw. ASC-Anweisung zu verwenden.

HINWEIS *Die LED-Anweisung kann nur von einer A3N, A3A, Q3A, Q4A und Q4AR CPU ausgeführt werden. Bei CPUs ohne LED-Anzeige erfolgt die Verarbeitung der Anweisung ohne Reaktion.*

Beispiel LED

Das folgende Programm konvertiert eine Zeichenfolge in einen ASCII-Code, speichert diesen in vorgegebenen Registern ab und gibt den Registerinhalt an das LED-Display der CPU aus. Im ersten Schritt wird hierzu nach dem Einschalten von X0 die Zeichenfolge "ABCDEFGHIJKLMNOP" in einen ASCII-Code gewandelt und in den Datenregistern D88 bis D95 gespeichert. Nach dem Einschalten von X16 werden die in D88 bis D95 gespeicherten ASCII-Daten auf dem Display der CPU angezeigt.

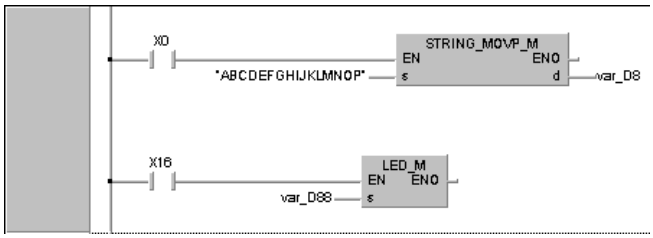
MELSEC-Anweisungsliste

```
MELSEC   LD     X0
         $MOVP  "ABCDEFGHIJKLMNOP"
                D88
         LD     X16
         LED    D88
```

IEC-Anweisungsliste

```
         LD              X0
         STRING_MOVP_M   "ABCDEFGHIJKLMNOP" , D88
         LD              X16
         LED_M           var_D88
```

Kontaktplan



7.9.4 LEDC

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
	●[1]	●[2]	●	●[3]	

[1] Nur fürA3N CPU.
[2] Nur fürA3A CPU.
[3] Nicht Q2A(S1) CPU

Operanden MELSEC A

	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)								Konstante		Pointer		Ebene				
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
	●	●	●	●	●	●	●	●	●	●	●	●							●	●			3 ●[1]	●		●

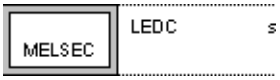
[1] Die Anzahl der Schritte bei der Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

Operanden MELSEC Q

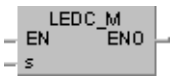
	Operanden								Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sonder-module U□\G□	Index-Register Zn	Konstanten $	Andere		
	Bit	Wort		Bit	Wort				BL, BL\S, BL\TR		
s	●	●	●	●	●	●	—	—	●	—	2

GX IEC Developer

MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste

Variablen

Operand	Befehlswert	Datentyp
s	Erste Adresse des Operanden, in dem der darzustellende Kommentar gespeichert ist.	Adresse

Funktions-weise

Ausgabe an LED-Anzeige

LEDC Gespeicherte Kommentare am LED-Display der CPU anzeigen

Die LEDC-Anweisung ruft Kommentare von Operanden (16 Zeichen) aus einem vorgegebenen Adressenbereich auf und bringt sie auf dem LED-Display einer dafür geeigneten CPU zur Anzeige. Liegen mehr als 16 Zeichen zur Anzeige an, werden nur die ersten 16 Zeichen angezeigt. Die Startadresse der Operanden, deren Kommentare angezeigt werden sollen, wird in s festgelegt.

Verfügt der in s vorgegebene Operand über keinen Kommentar, bleibt das Display an der CPU-Front leer, oder es wird gelöscht. Liegen die Daten außerhalb des Kommentarbereiches, erfolgt keine Verarbeitung der LEDC-Anweisung, und der zuvor bestehende Inhalt der Anzeige bleibt erhalten.

Enthält ein Kommentar Zeichen, die nicht auf dem Display der CPU dargestellt werden können, erfolgt eine fehlerhafte Anzeige. Auf dem LED-Display der dafür geeigneten CPUs können Zahlen zwischen 0 und 9, alle Buchstaben zwischen A und Z (nur Großbuchstaben ohne Umlaute) sowie die folgenden Sonderzeichen dargestellt werden: < > = x / ' + -

Die LED-Anweisung kann von der Q2ACPU(S1) nicht ausgeführt werden. Die Verarbeitung erfolgt ohne Reaktion.

HINWEIS *Die LEDC-Anweisung dient in den Erweiterten Applikationsanweisungen der AnA CPUs zum Setzen von Operanden. Zur Programmierung der LEDC-Anweisung in Verbindung mit einer A3A CPU sind die entsprechenden Hinweise in der gesonderten Programmieranleitung der AnA CPUs (Dedicated Instructions) zu beachten (nur A-Serie).*

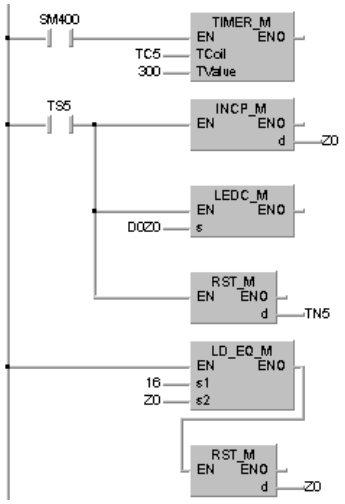
Beispiel LEDC

Im nachstehenden Programm werden die Kommentare aus D0 bis D15 im Abstand von 30 Sekunden zur Anzeige gebracht. Timer T5 schaltet hierzu im 30-Sekunden-Rhythmus als Eingangsbedingung zur LEDC-Anweisung. Sobald der Timer einschaltet, wird der Kommentar des Datenregisters D(0+Z) angezeigt. Mit jedem Einschalten wird der Wert von Z um 1 erhöht. Der Wert in Z wird wieder auf 0 gesetzt, sobald Z gleich 16 ist.

MELSEC-Anweisungsliste

```
MELSEC   LD    SM400
         OUT   T5
               K300
         LD    T5
         RST   T5
         LEDC  D0Z0
         INCP  Z0
         LD=   K16
               Z0
         RST   Z0
```

Kontaktplan



IEC-Anweisungsliste

```
LD      SM400
TIMER.M TC5 , 300
LD      TS5
R       TN5
LEDC    D0Z0
INCR_M  Z0
LD      TRUE
EQ_M    16 , Z0
RST_M   Z0
```

7.9.5 LEDA, LEDB

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
	●[1]				

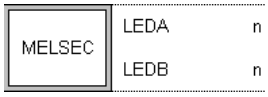
[1] Nur A3N CPU.

Operanden MELSEC A

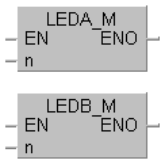
	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
n																							13			

GX IEC Developer

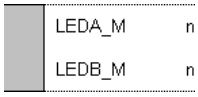
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



Variablen

Operand	Befehlswert	Datentyp
n	ASCII-Daten	Zeichenfolge

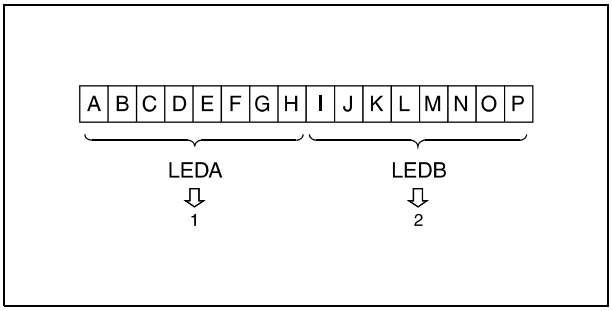
Funktions-weise

Ausgabe an LED-Anzeige

LEDA, LEDB ASCII-Zeichenfolge am LED-Display der CPU anzeigen

Die Anweisungen bringen eine ASCII-Zeichenfolge auf dem LED-Display einer dafür geeigneten CPU zur Anzeige. Die ASCII-Zeichenfolge besteht aus jeweils 8 Zeichen pro Anweisung und wird in der LEDA-/ LEDB-Anweisung vorgegeben.

Insgesamt kann mit beiden Anweisungen eine Zeichenfolge von bis zu 16 Zeichen dargestellt werden. Die LEDA-Anweisung bestimmt dabei die ersten 8 Zeichen (linke Hälfte) und die LEDB-Anweisung die letzten 8 Zeichen (rechte Hälfte) der LED-Anzeige.



[1] Definition der ersten acht Zeichen

[2] Definition der zweiten acht Zeichen

Auf dem LED-Display der dafür geeigneten CPUs können Zahlen zwischen 0 und 9, alle Buchstaben zwischen A und Z (nur Großbuchstaben ohne Umlaute) sowie die folgenden Sonderzeichen dargestellt werden: < > = x / ' + -

HINWEIS *Die LEDA-/LEDB-Anweisung dient in Verbindung mit einer AnA oder AnU CPU zur Kennzeichnung des Beginns der Erweiterten Applikationsanweisungen. Zur Programmierung der LEDA-/LEDB-Anweisung in Verbindung mit einer AnA oder AnU CPU sind die Hinweise in der gesonderten Programmieranleitung der AnA/ AnU CPUs (Dedicated Instructions) zu beachten.*

Beispiel LEDA, LEDB

Mit Hilfe des folgenden Programms wird nach dem Einschalten von XC die Zeichenfolge "ABCDEFGH IJKLMNOP" auf dem LED-Display der CPU angezeigt.

MELSEC-Anweisungsliste	Kontaktplan	IEC-Anweisungsliste
MELSEC LD XC LEDA "ABCDEFGH" LEDB "IJKLMNOP"		

HINWEIS *Anmerkung zur Anzeige der CPU*

Die zweite Hälfte einer mittels LED-Anweisung zur Anzeige gebrachten Zeichenfolge aus 16 Zeichen verlischt, wenn die ersten 8 Zeichen durch eine LEDA-Anweisung überschrieben werden. Umgekehrt verhält es sich mit einer LEDB-Anweisung, d.h. die erste Hälfte der Anzeige verlischt, wenn die letzten 8 Zeichen einer mittels LED-Anweisung angezeigten Zeichenfolge durch eine LEDB-Anweisung überschrieben werden.

7.9.6 LEDR

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●[1]

[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC A

	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)								Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
																							1			

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten	Andere		
	Bit	Wort		Bit	Wort						
—	—	—	—	—	—	—	—	—	—	—	1

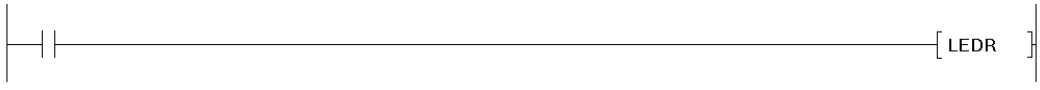
GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan

IEC-Anweisungsliste

GX Developer



Variablen

Operand	Befehlswert	Datentyp
—	—	—

Funktions-weise

Rücksetzen von Fehlermerkern und LED-Anzeigen

LEDR Rücksetzanweisung

Die LEDR-Anweisung setzt Fehlermerker zurück, die nach Auftreten eines Fehlers automatisch gesetzt werden. Bei CPU-Modulen mit LED-Display entspricht das Betätigen der Taste INDICATOR RESET der Funktion der LEDR-Anweisung (nur A-Serie).

Verarbeitung der LEDR-Anweisung bei durch Selbstdiagnose gesetztem Fehlermerker (nur Q-Serie):

Ist in der Selbstdiagnose ein Fehler aufgetreten, der den Betrieb der CPU nicht beeinflußt, werden bei Ausführung der LEDR-Anweisung die "ERROR"-LED und die Fehleranzeige im LED-Display der CPU gelöscht.

Zusätzlich ist es erforderlich, SM1 und SD0 des Anwenderprogramms zurückzusetzen, da diese über die LEDR-Anweisung nicht automatisch zurückgesetzt werden. Weitere mit dem Zurücksetzen des Fehlermerkers erforderliche Schritte werden ebenfalls nicht ausgeführt.

Verarbeitung der LEDR-Anweisung bei einem Batteriefehler (Q-Serie):

Wird die LEDR-Anweisung nach dem Ersetzen der Batterie ausgeführt, werden die "BAT. ARM"-LED an der Forderseite der CPU und die Fehleranzeige im LED-Display der CPU gelöscht. Dabei wird automatisch SM51 zurückgesetzt.

Verarbeitung der LEDR-Anweisung bei gesetztem Fehlermerker F bei einer CPU ohne LED-Display:

Nach Ausführung der LEDR-Anweisung laufen folgende Vorgänge ab.

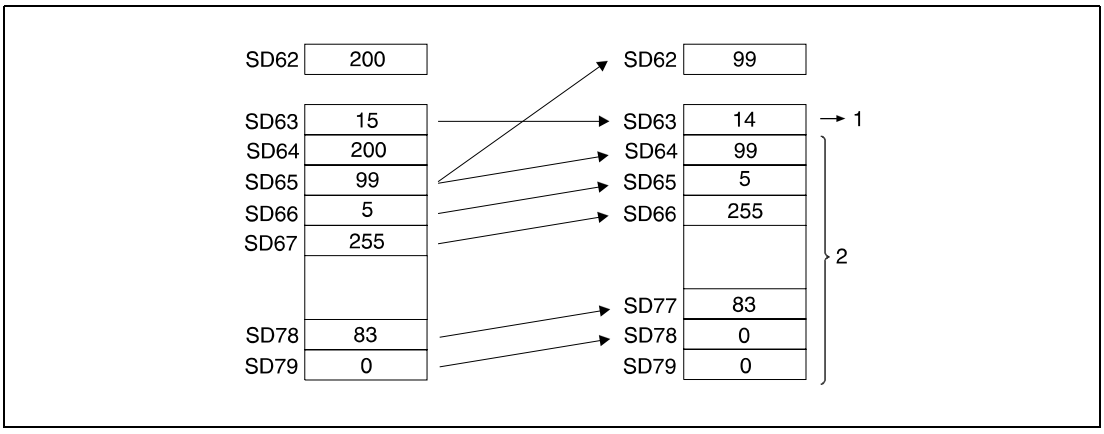
Die ERROR-LED an der CPU blinkt und verlischt kurz darauf.

Die in D9009 (A-Serie) bzw. SD62 (Q-Serie/System Q) gespeicherten Fehlermerker F werden zurückgesetzt.

Die Register D9009 und D9125 (A-Serie) bzw. SD62 und SD64 (Q-Serie/System Q) werden gelöscht und die in D9126 bis D9131 (A-Serie) bzw. SD65 bis SD79 (Q-Serie/System Q) gespeicherten Fehlermerker zur weiteren Verarbeitung nach vorne geschoben.

Die in D9125 (A-Serie) bzw. SD64 (Q-Serie/System Q) nachgeschobene Adresse des nächsten Fehlermerkers wird nach D9009 (A-Serie) bzw. SD62 (Q-Serie/System Q) übertragen.

Der Akkumulator der Fehlermerker in D9124 (A-Serie) bzw. SD63 (Q-Serie/System Q) wird um 1 vermindert. Ist D9124 (A-Serie) bzw. SD63 (Q-Serie/System Q) bereits 0, bleibt der Wert bestehen.



[1] Anzahl der gespeicherten Fehlermerker

[2] Speicherbereich der Fehlermerker

Verarbeitung der LEDR-Anweisung bei gesetztem Fehlermerker F bei einer CPU mit LED-Display:

Nach Ausführung der LEDR-Anweisung laufen folgende Vorgänge ab.

Der auf dem LED-Display der CPU angezeigte Fehlermerker wird gelöscht.

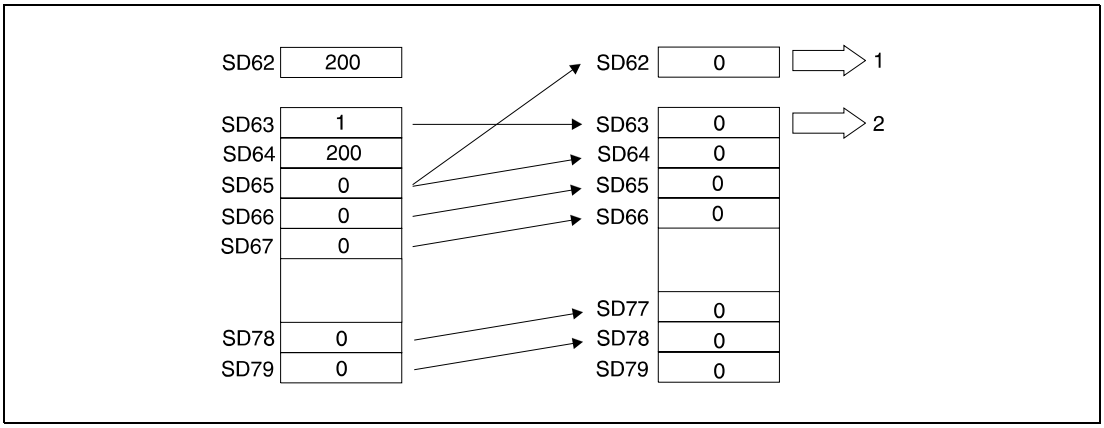
Der in D9009 (A-Serie) bzw. SD62 (Q-Serie) gespeicherte Fehlermerker F wird gelöscht.

Die Datenregister D9009 und D9125 (A-Serie) bzw. SD62 und SD64 (Q-Serie) werden zurückgesetzt und die in D9126 bis D9131 (A-Serie) bzw. SD65 bis SD79 (Q-Serie) gespeicherten Fehlermerker zur weiteren Verarbeitung nach vorne geschoben.

Die in D9125 (A-Serie) bzw. SD64 (Q-Serie) nachgeschobene Adresse des nächsten Fehlermerkers wird nach D9009 (A-Serie) bzw. SD62 (Q-Serie) übertragen.

Der Akkumulator der Fehlermerker in D9124 (A-Serie) bzw. SD63 (Q-Serie) wird um 1 vermindert. Ist D9124 (A-Serie) bzw. SD63 (Q-Serie) bereits 0, bleibt der Wert bestehen.

Der jetzt in D9009 (A-Serie) bzw. SD62 (Q-Serie) gespeicherte Fehlermerker wird auf dem Display angezeigt. Ist D9124 (A-Serie) bzw. SD63 (Q-Serie) gleich 0, erfolgt keine Anzeige.



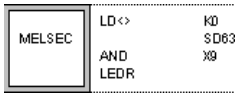
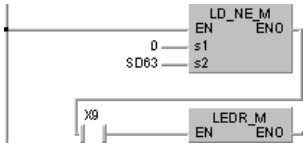
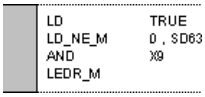
[1] Da SD63 den Wert 0 hat, wird kein Fehlermerker auf dem LED-Display angezeigt

[2] Anzahl der gespeicherten Fehlermerker

HINWEIS *Die LEDR-Anweisung dient im Zusammenhang mit einer AnA oder AnU CPU zum Abschluss der Erweiterten Applikationsanweisungen. Zur Programmierung der LEDR-Anweisung in Verbindung mit einer A3A CPU sind die entsprechenden Hinweise in der gesonderten Programmieranleitung der AnA CPUs (Dedicated Instructions) zu beachten (nur A-Serie).*

Beispiel LEDR

Im folgenden Programm wird die LEDR-Anweisung ausgeführt, wenn der Wert im Register SD63 von 0 verschieden ist, und wenn X9 gesetzt ist.

MELSEC-Anweisungsliste	Kontaktplan	IEC-Anweisungsliste
		

HINWEIS *In der folgenden Tabelle sind die Voreinstellungen für Fehlernummern und Prioritäten in den Diagnoseregistern SD207 bis SD209 dargestellt.*

Priorität	Fehlernummer (Hexadezimal)	Beschreibung	Bemerkung
1	1	AC DOWN	Unterbrechung der Spannungsversorgung
2	2	UNIT VERFY ERR. FUSE BREAK OFF P. UNIT ERROR	Fehler bei Überprüfung der E/A-Module Sicherung hat ausgelöst Fehler bei Überprüfung eines Sondermoduls
3	3	OPERATIN ERROR LINK PARA ERROR SFCP OPE. ERROR SFCP EXE. ERROR	Fehler bei der Ausführung einer Anweisung Fehler bei Netzwerkparametern Programm der Ablaufsprache fehlerhaft Fehler bei Ausführung des Schrittprogrammes
4	4	ICM.OPE ERROR FILE OPE ERROR EXTEND INST. ERROR	Speicherkartenfehler Fehler bei Zugriff auf ein File Fehler bei einer erweiterten Anweisung
5	5	PRG.TIME OVER	Konstante Zykluszeit oder WDT überschritten
6	6	CHK-Anweisung	Fehler wurde mit der CHK-Anweisung festgestellt
7	7	Fehlermerker	
8	8	LED-Anweisung	
9	9	BATTERY ERROR	Batterie fehlerhaft
10	A	Uhrendaten	

7.10 Fehlererkennung und Fehlerbeseitigung

Die Anweisungen zur Fehlererkennung und zur Fehlerbeseitigung (Debugging) dienen der Fehlerkontrolle, dem Setzen und Rücksetzen des Status Latchs, der Abtastüberwachung (Sampling Trace) und der Programmüberwachung (Program Trace). Die folgende Tabelle liefert eine Übersicht der Anweisungen.

Funktion	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor
Fehler-kontrolle	CHKST	CHKST_M
	CHK	CHK_M
	CHKCIR	CHKCIR_M
	CHKEND	CHKEND_MD
Status Latch setzen/rücksetzen	SLT	SLT_M
	SLTR	SLTR_M
Abtastüberwachung (Sampling Trace) setzen/rücksetzen	STRA	STRA_M
	STRAR	STRAR_M
Programmüberwachung (Program Trace) ausführen/setzen/rücksetzen	PTRA	PTRA_M
	PTRAR	PTRAR_M
	PTRAEXE	PTRAEXE_M
	PTRAEXEP	PTRAEXEP_M

HINWEIS *„Bitte prüfen Sie in Ihrer vorliegenden Version des GX IEC Developers, ob diese Anweisungen unterstützt werden“.*

7.10.1 CHKST, CHK (Q-Serie/System Q)

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●[1]

[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

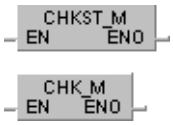
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort				DY		
—	—	—	—	—	—	—	—	—	—	SM0	1

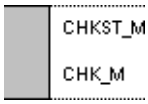
GX IEC Developer

MELSEC-Anweisungsliste

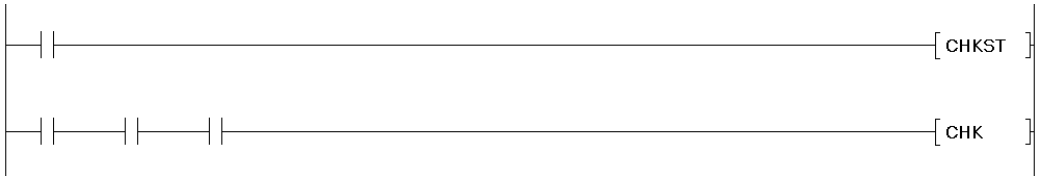
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

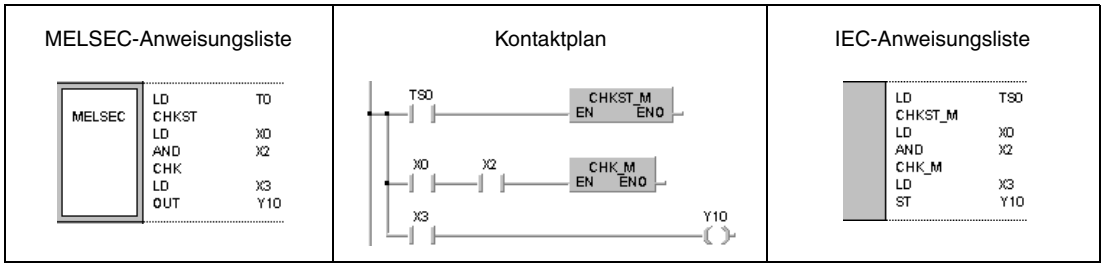
Operand	Befehlswert	Datentyp
—	—	—

Funktions-weise

Fehlerkontrolle bei wechselseitigen Schaltvorgängen (Q-Serie und System Q)

CHKST Startanweisung zur CHK-Anweisung

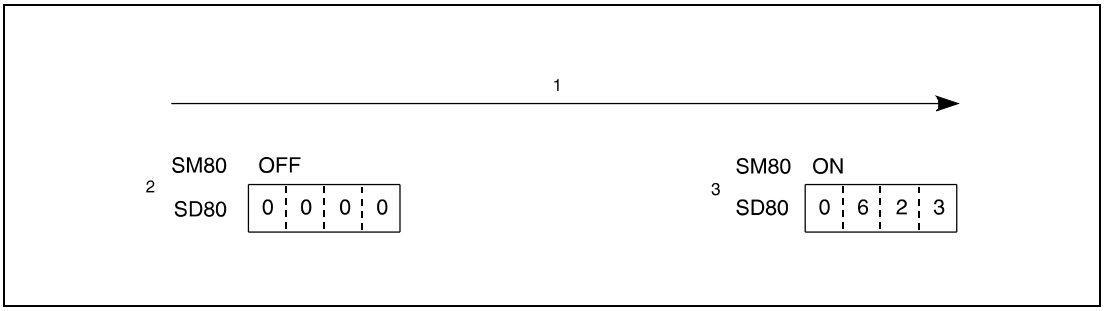
Die CHKST-Anweisung startet die Ausführung der CHK-Anweisung. Ist die Ausführungsbedingung der CHKST-Anweisung nicht gesetzt (0), wird der nächste auf die CHK-Anweisung folgende Programmschritt ausgeführt. Bei gesetzter Ausführungsbedingung der CHKST-Anweisung (1) wird die CHK-Anweisung ausgeführt. In der folgenden Abbildung sind diese Anweisungen programmiert.



CHK Anweisung zur Fehlerkontrolle

Die CHK-Anweisung ermöglicht bei einigen CPU-Typen (in Abhängigkeit der Verarbeitungsart) eine Fehlerkontrolle in einer Kontaktanordnung mit Grenzschaltern, die zur Störungskontrolle bei wechselseitigen Bewegungsvorgängen dienen. Sobald ein Fehler in dieser Anordnung vorliegt, wird der Merker SM80 gesetzt und ein entsprechender Fehlercode im Diagnoseregister SD80 gespeichert.

Bei Verwendung der Q-Serie wird der Fehlercode in Form eines 4-stelligen BCD-Datenwertes in dem Diagnoseregister SD80 gespeichert. In den oberen 3 Digits wird die Kontaktnummer des entsprechenden Kontaktes gespeichert (im Beispiel Kontakt 62) und im unteren Digit wird die Nummer des Fehlerprüfnetzwerks (Fehlerzustand 1-6) gespeichert (im Beispiel Fehlerzustand 3).



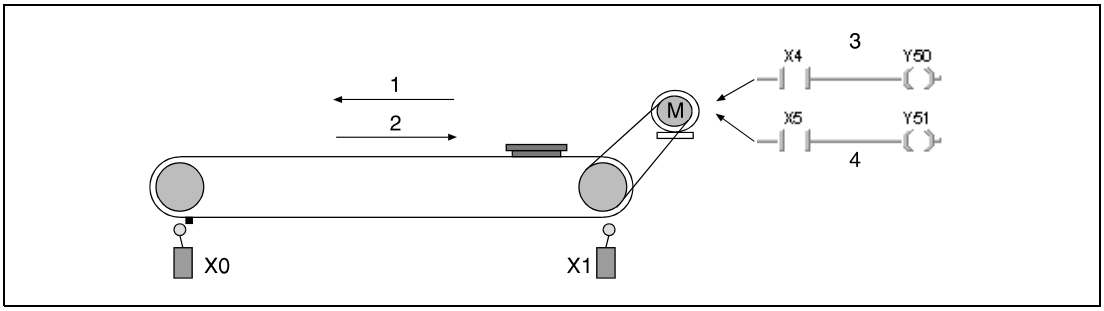
[1] Kontakt 62; Spule 3 (bei Fehlererkennung)

[2] vor der Fehlererkennung

[3] nach der Fehlererkennung

Die vor der CHK-Anweisung programmierten Eingangskontakte dienen nicht als Ausführungsbedingung der CHK-Anweisung, sondern der Vorgabe der Kontrollbedingungen.

Im folgenden soll anhand eines konkreten Beispiels der Programmaufbau zur Fehlerkontrolle mittels CHK-Anweisung näher erläutert werden. Die folgende Abbildung zeigt ein Transportband, das sich nach Erreichen der rechten oder linken Grenze in die Gegenrichtung bewegt. Die jeweiligen Endpunkte werden über Grenzschalter (X0 und X1) festgelegt. Startkontakt für die Vorwärtsbewegung des Bandes ist X4 und für die Rückwärtsbewegung X5.



[1] Vorwärtsrichtung

[2] Rückwärtsrichtung

[3] Start vorwärts

[4] Start rückwärts

Die folgende Abbildung zeigt die Programmgestaltung für den Betrieb und die Fehlerkontrolle des oben gezeigten Transportbandes bei Verwendung einer Q-Serie CPU. Bei störungsfreiem Betrieb springt die Verarbeitung zum auf die CHK-Anweisung folgenden Programmschritt. Bei positiver Flanke von X4 wird das Band in Vorwärtsrichtung bewegt und Y0 zur Fehlererkennung gesetzt. Bei positiver Flanke von X5 setzt sich das Band in Rückwärtsrichtung in Bewegung, und Y0 wird zurückgesetzt. Mit Timer T0 wird die Dauer des Arbeitszyklus überwacht. Bei Zeitüberschreitung wird die CHKST-Anweisung über den Kontakt TS0 gesetzt. Im nächsten Programmschritt wird die CHK-Anweisung ausgeführt, und der Fehlercode im Diagnoseregister SD80 gespeichert.

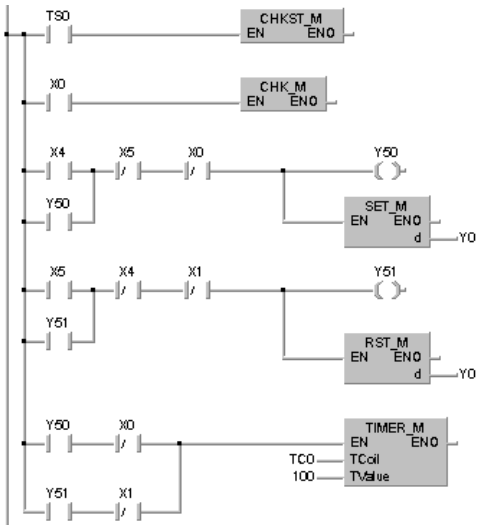
MELSEC-Anweisungsliste

```
MELSEC
        LD      T0
        CHKST
        LD      X0
        CHK
        LD      X4
        OR      Y50
        ANI     X5
        ANI     X0
        OUT     Y50
        SET     Y10
        LD      X5
        OR      Y51
        ANI     X4
        ANI     X1
        OUT     Y51
        RST     Y10
        LD      Y50
        ANI     X0
        LD      Y51
        ANI     X1
        ORB
        OUT     T0
                K100
```

IEC-Anweisungsliste

```
        LD      TS0
        CHKST_M
        LD      X0
        CHK_M
        LD      X4
        OR      Y50
        ANDN    X5
        ANDN    X0
        ST      Y50
        S       Y10
        LD      X5
        OR      Y51
        ANDN    X4
        ANDN    X1
        ST      Y51
        R       Y10
        LD      Y50
        ANDN    X0
        OR(
                Y51
        ANDN    X1
        )
        TIMER_M TC0 , 100
```

Kontaktplan

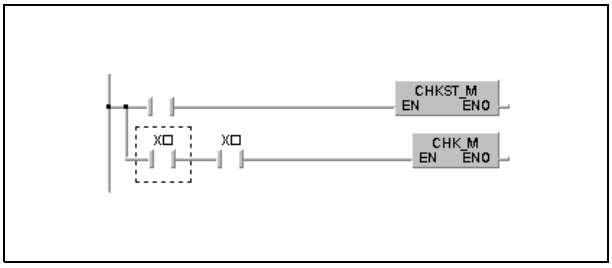


Die Verarbeitungsweise der CHK-Anweisung lässt sich anhand der folgenden Kontaktpläne erläutern, die in der Funktion mit der Abarbeitung der CHK-Anweisung übereinstimmen. Die Kontaktadressen des Grenzschalters für Vorwärtsbewegung X☐ und des Grenzschalters für Rückwärtsbewegung X☐+1 müssen aufeinanderfolgend vergeben sein. Die Adresse des Grenzschalters für Vorwärtsbewegung X☐ muss immer niedriger sein als die Adresse des Grenzschalters für Rückwärtsbewegung X☐+1. Der Kontaktadresse für den Vorwärtsgrenzschalter X☐ ist ein Ausgang Y☐ mit identischer Adresse zugeordnet. Dieser Ausgang ist gemäß Programmierung (siehe Beispiel) beim Betrieb in Vorwärtsrichtung gesetzt und beim Betrieb in Rückwärtsrichtung zurückgesetzt.

Zur besseren Veranschaulichung des vorangegangenen Beispiels sind hier die Kontakte X0 (X☐), X1 (X☐+1) und Y0 (Y☐) zur Vorgabe der Kontrollbedingungen direkt eingesetzt. An ihrer Stelle können in Abhängigkeit des Programms beliebige andere Adressen eingesetzt werden.

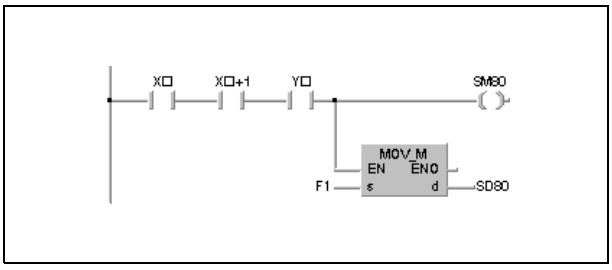
HINWEIS *Die Ausgänge Y☐ werden als interne Merker behandelt und können nicht extern abgefragt werden.*

Im folgenden sind die CHK-Anweisungen angegeben. Der markierte Kontakt X☐ steht als Variable für maximal 150 Kontakte (150 Transportbänder oder vergleichbare Anwendungen).



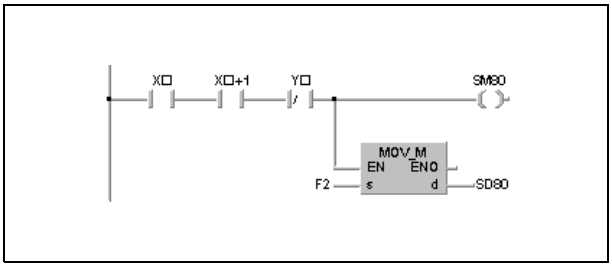
Fehlerprüfnetzwerk 1 (Fehlerzustand 1):

Beide Grenzschalter sprechen im Vorwärtsbetrieb des Transportbandes an.



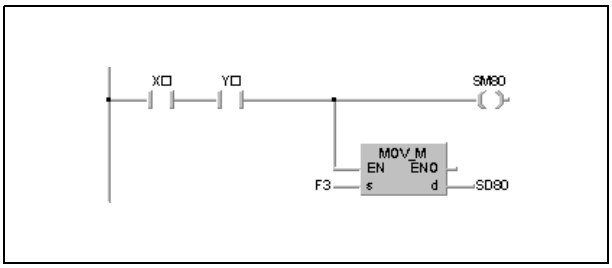
Fehlerprüfnetzwerk 2 (Fehlerzustand 2):

Beide Grenzschalter sprechen im Rückwärtsbetrieb des Transportbandes an.



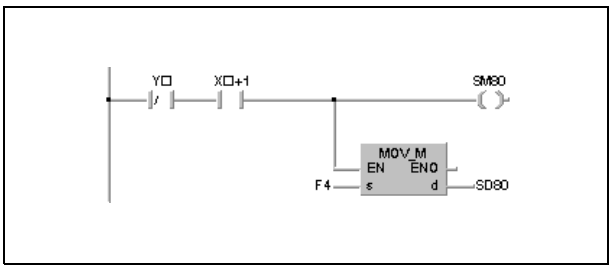
Fehlerprüfnetzwerk 3 (Fehlerzustand 3):

Aufruf des Vorwärtsbetriebs bei eingeschaltetem Grenzschalter "Vorwärts".



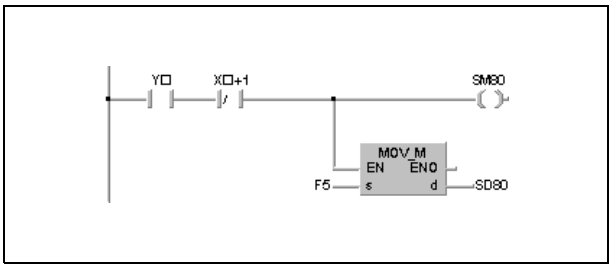
Fehlerprüfnetzwerk 4 (Fehlerzustand 4):

Aufruf des Rückwärtsbetriebs bei eingeschaltetem Grenzschalter "Rückwärts".



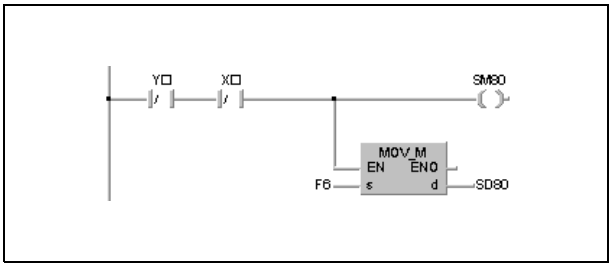
Fehlerprüfnetzwerk 5 (Fehlerzustand 5):

Aufruf des Vorwärtsbetriebs bei nicht eingeschaltetem Grenzschalter "Rückwärts".

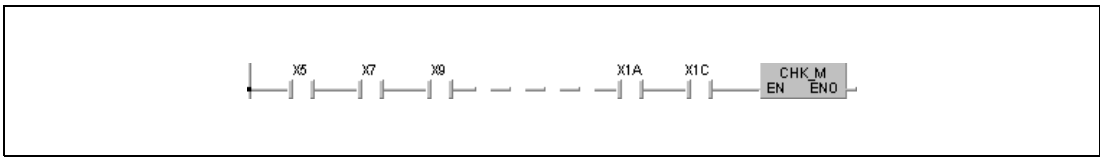


Fehlerprüfnetzwerk 6 (Fehlerzustand 6):

Aufruf des Rückwärtsbetriebs bei nicht eingeschaltetem Grenzschalter "Vorwärts".



Bei Verwendung der CHK-Anweisung können maximal 150 Kontaktadressen für Grenzschalter für die Vorwärtsrichtung angegeben werden. Bei der Angabe der Kontaktadressen wird jeweils die Adresse für den Grenzschalter für die Rückwärtsrichtung übersprungen.

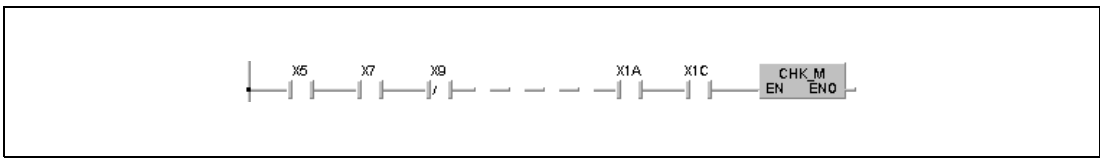


Der Merker SM80 und das Diagnoseregister SD80 und müssen nach der Ausführung der CHK-Anweisung zurückgesetzt werden, da sie nach dem Setzen durch die CHK-Anweisung ihren Zustand beibehalten. Sind die Operanden bei erneuter Ausführung der CHK-Anweisung nicht zurückgesetzt, kann die CHK-Anweisung nicht ausgeführt werden.

Die CHKST-Anweisung muss vor der CHK-Anweisung programmiert werden.

Die CHK-Anweisung kann in jeden beliebigen Programmschritt des Ablaufprogramms geschrieben werden. Die CHK-Anweisung kann bis zu zwei mal in einer Programm-Organisationseinheit (POE) ausgeführt werden.

Die Kontrollbedingungen müssen mittels einer LD- oder AND-Anweisung vor der CHK-Anweisung programmiert werden. Andere Eingangsanweisungen sind nicht möglich. Wenn in den Kontrollbedingungen eine LDI- oder ANI-Anweisung programmiert wird, kann die Fehlerkontrolle der CHK-Anweisung nicht ausgeführt werden. Die an der Fehlerkontrolle beteiligten Kontaktadressen können hingegen von LDI- und ANI-Anweisungen angesprochen werden. In der folgenden Abbildung wird der Schalter mit der Adresse X9 ignoriert, da es sich um einen NC-Kontakt handelt.



Der Prüfalgorithmus hängt vom Zustand des Merkers SM710 ab:

SM710 ist zurückgesetzt (0):

In diesem Fall erfolgt die Fehlerkontrolle beginnend mit der ersten angegebenen Grenzschalteradresse mit dem ersten Fehlerprüfnetzwerk. Wenn die Kontrolle mit dem ersten Prüfnetzwerk für alle mit den Vorwärtsgrenzschalteradressen angegebenen Schaltungsanordnungen durchgeführt wurde, erfolgt die weitere Fehlerkontrolle erneut bei der ersten Grenzschalteradresse mit dem zweiten Fehlerprüfnetzwerk. Die Fehlerkontrolle ist abgeschlossen, wenn die letzte Schaltungsanordnung mit dem letzten (6.) Prüfnetzwerk überprüft wurde.

SM710 ist gesetzt (1):

Die Fehlerkontrolle erfolgt in der Reihenfolge der Eingangskontakte, die als Kontrollvariablen vor der CHK-Anweisung programmiert sind. Das bedeutet, dass jede mit der Adresse des Vorwärtsgrenzschalters angegebene Schaltungsanordnung mit den 6 Fehlerprüfnetzwerken kontrolliert wird. Die nächste Schaltungsanordnung wird erst nach nach dem vollständigen Abarbeiten aller 6 Fehlerprüfnetzwerke kontrolliert. Die Fehlerkontrolle ist abgeschlossen, wenn die letzte Schaltungsanordnung mit den 6 Fehlerprüfnetzwerken kontrolliert wurde.

Werden zwei oder mehr Fehler erkannt, wird nur der Fehlercode mit der höheren Priorität gespeichert. Da der Fehler mit der höchsten Priorität mit dem zuerst gefundenen Fehler identisch ist, wird der Fehlercode des zuerst aufgetretenen Fehlers gespeichert.

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- In den Kontrollbedingungen der CHK-Anweisung werden zwei Eingangskontakte parallelgeschaltet (Fehlercode 4235).
- Die Kontrollbedingungen der CHK-Anweisung enthalten mehr als 150 Eingangsoperanden (Fehlercode 4235).
- Hinter der CHKST-Anweisung ist keine CHK-Anweisung programmiert (Fehlercode 4235).
- Die CHK-Anweisung wird ohne vorheriges Ausführen der CHKST-Anweisung ausgeführt (Fehlercode 4235).

7.10.2 CHK (A-Serie)

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●[1]	●[1]	●	●		

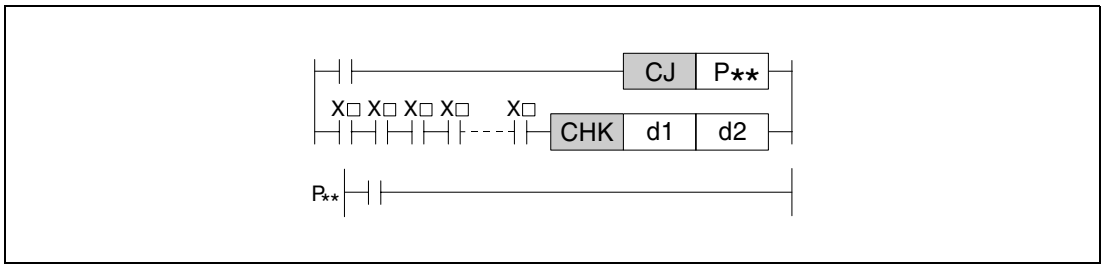
[1] Nur bei Direktverarbeitung

Operanden MELSEC A

	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
d1		●	●	●	●	●	●																5 ●[1]			
d2		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●										

[1] Die Anzahl der Schritte bei der Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

GX IEC Developer



Variablen

Operand	Befehlswert	Datentyp
d1	Operand, der bei einer Fehlererkennung gesetzt wird.	Bit
d2	Operand, in dem der Fehlercode gespeichert wird.	BIN-16-Bit

Funktions-weise

Fehlerkontrolle bei wechselseitigen Schaltvorgängen (A-Serie)

CHK Anweisung zur Fehlerkontrolle

Die Funktion der CHK-Anweisung ist von der gewählten Verarbeitungsart abhängig. Bei A1S- und AnN-CPUs ermöglicht die CHK-Anweisung in der Verarbeitung nach dem Prozessabbild, die Erzeugung eines Flip-Flops.

Bei Direktverarbeitung der E-/A-Zustände (außer AnA-, AnAS-, AnU- und A2C-CPUs) dient die Anweisung zur Fehlerkontrolle bei wechselseitigen Schaltvorgängen.

Die Programmierung der CHK-Anweisung ist aufgrund des Pointers 254 nur in der MELSEC MEDOC-Anweisungsliste möglich.

Die CHK-Anweisung ermöglicht bei einigen CPU-Typen (in Abhängigkeit der Verarbeitungsart) eine Fehlerkontrolle in einer Kontaktanordnung mit Grenzschaltern, die zur Störungskontrolle bei wechselseitigen Bewegungsvorgängen dienen. Sobald ein Fehler in dieser Anordnung vorliegt, wird der Operand in d1 gesetzt und ein entsprechender Fehlercode in d2 gespeichert.

Die vor der CHK-Anweisung programmierten Eingangskontakte dienen nicht als Ausführungsbedingung der CHK-Anweisung, sondern zur Vorgabe der Kontrollbedingungen.

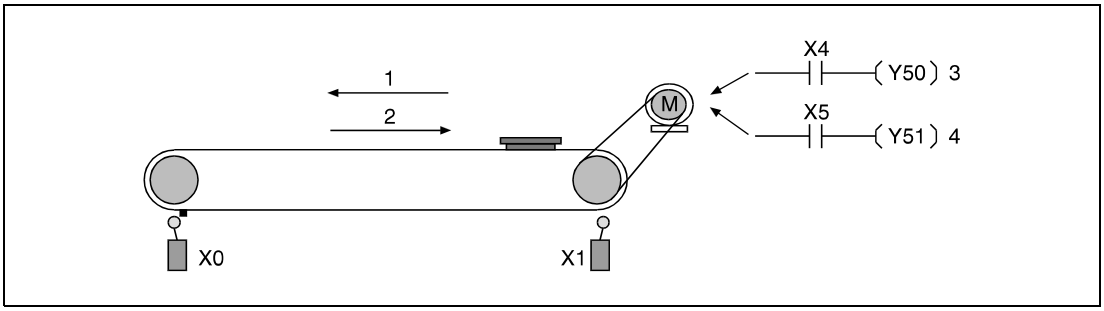
Eine CHK-Anweisung hat die Funktion, das Auftreten und die Ursache eines Fehlers, wie z.B. die Zeitüberschreitung eines Arbeitszyklus, anzuzeigen. Die Ausführung des Programmteils, der die CHK-Anweisung enthält, sollte bei fehlerfreier Verarbeitung übersprungen werden. Zum Überspringen des CHK-Anweisungsteils kann eine CJ-, SCJ- oder JMP-Anweisung verwendet werden.

Die Ausführung der CHK-Anweisung erfolgt mit jedem Programmzyklus und ist unabhängig von dem Zustand der Eingangsoperanden, die der Anweisung als Kontrollbedingung vorangestellt sind.

In dem folgenden Programm wird Y60 nach einer Zeitüberschreitung eines Arbeitszyklus eingeschaltet und die CHK-Anweisung ausgeführt. Nach Erfassung des Fehlers durch die CHK-Anweisung wird M0 gesetzt und ein Programmsprung zu Sprungzieladresse P31 (im Beispiel nicht dargestellt) ausgeführt. An Sprungmarke P31 könnte sich beispielsweise ein Programmteil zur Fehlerverarbeitung befinden. Liegt keine Zeitüberschreitung vor, wird der Programmteil zur Fehlerkontrolle übersprungen und Schritt 18 an Sprungzieladresse P30 ausgeführt. Die Programmierung dieses Programms ist aufgrund des Pointers 254 nur in der MELSEC MEDOC-Anweisungsliste möglich.

```
100   LDI   Y060
101   CJ    P30
104   LD    M0
105   CJ    P30
108   P254
109   LD    X010
110   AND   X015
111   AND   X008
112   AND   X01A
113   CHK   M0    D0
118   P30
118   LD    M10
119   OUT   Y040
121   END
```

Im folgenden soll anhand eines konkreten Beispiels der Programmaufbau zur Fehlerkontrolle mittels CHK-Anweisung näher erläutert werden. Die folgende Abbildung zeigt ein Transportband, das sich nach Erreichen der rechten oder linken Grenze in die Gegenrichtung bewegt. Die jeweiligen Endpunkte werden über Grenzschalter (X0 und X1) festgelegt. Startkontakt für die Vorwärtsbewegung des Bandes ist X4 und für die Rückwärtsbewegung X5.



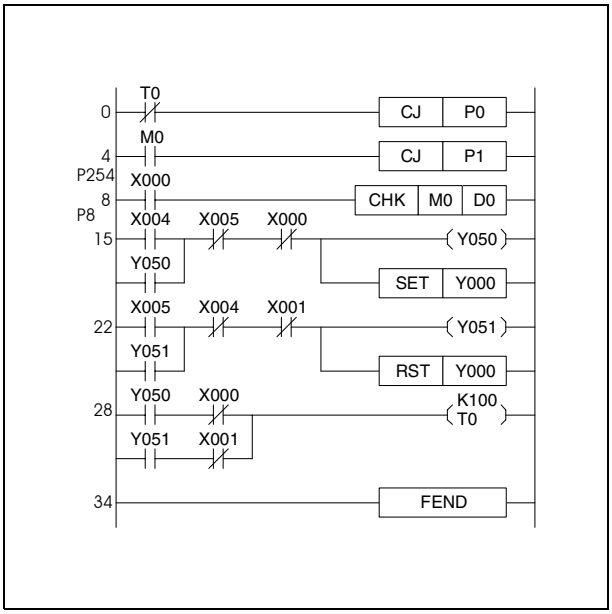
[1] Vorwärtsrichtung

[2] Rückwärtsrichtung

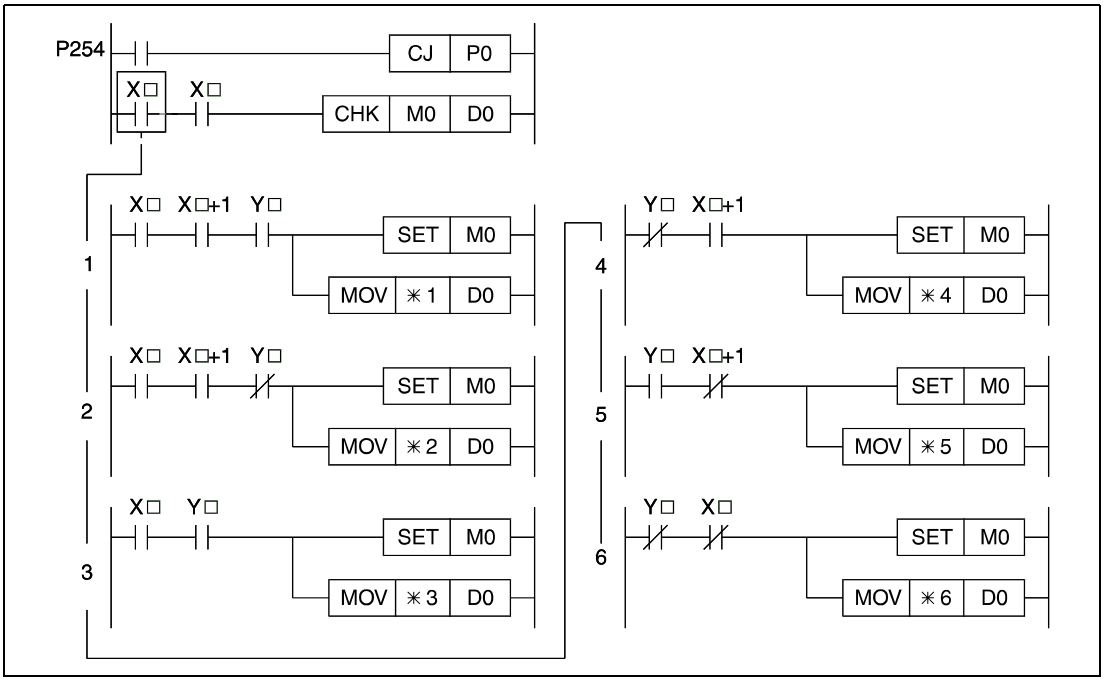
[3] Start vorwärts

[4] Start rückwärts

Die folgende Abbildung zeigt die Programmgestaltung für den Betrieb und die Fehlerkontrolle des oben gezeigten Transportbandes. Die Programmierung dieses Programms ist aufgrund des Pointers 254 nur in der MELSEC MEDOC-Anweisungsliste möglich. Bei störungsfreiem Betrieb springt die Verarbeitung zu Sprungzieladresse P0. Nach dem Einschalten von X4 wird das Band in Vorwärtsrichtung bewegt und Y0 zur Fehlererkennung gesetzt. Nach dem Einschalten von X5 setzt sich das Band in Rückwärtsrichtung in Bewegung, und Y0 wird zurückgesetzt. Mit Timer T0 wird die Dauer des Arbeitszyklus überwacht. Bei Zeitüberschreitung wird M0 über die CHK-Anweisung gesetzt und der Fehlercode in D0 gespeichert. Die Programmverarbeitung wird zur weiteren Fehlerverarbeitung an Sprungzieladresse P1 (Schritt 35) fortgesetzt.



Die Verarbeitungsweise der CHK-Anweisung lässt sich anhand des folgenden Kontaktplans erläutern, der in seiner Funktion mit der Abarbeitung CHK-Anweisung übereinstimmt. Zur besseren Veranschaulichung des vorangegangenen Beispiels sind hier die Kontakte X0, X1 und Y0 zur Vorgabe der Kontrollbedingungen direkt eingesetzt. An ihrer Stelle können in Abhängigkeit des Programms beliebige andere Adressen eingesetzt werden.



Folgende Fehlerzustände können vorliegen:

Zustand **1**: Beide Grenzschalter sprechen im Vorwärtsbetrieb des Transportbandes an.

Zustand **2**: Beide Grenzschalter sprechen im Rückwärtssbetrieb des Transportbandes an.

Zustand **3**: Aufruf des Vorwärtsbetriebs bei eingeschaltetem Grenzschalter „Vorwärts“.

Zustand **4**: Aufruf des Rückwärtsbetriebs bei eingeschaltetem Grenzschalter „Rückwärts“.

Zustand **5**: Aufruf des Vorwärtsbetriebs bei nicht eingeschaltetem Grenzschalter „Rückwärts“.

Zustand **6**: Aufruf des Rückwärtsbetriebs bei nicht eingeschaltetem Grenzschalter „Vorwärts“.

Die Nummer des in D0 gespeicherten Fehlercodes entspricht der Nummer des oben aufgeführten Fehlerzustandes.

Die Fehlerkontrolle der CHK-Anweisung erfolgt nach dem gezeigten Muster. Die Struktur des Kontaktplanmusters kann nicht geändert werden.

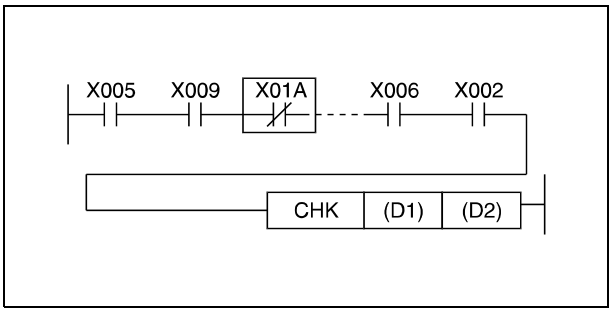
Die Operanden in d1 und d2 müssen nach der Ausführung der CHK-Anweisung zurückgesetzt werden, da sie nach dem Setzen durch die CHK-Anweisung ihren Zustand beibehalten. Sind die Operanden bei erneuter Ausführung der CHK-Anweisung nicht zurückgesetzt, kann die CHK-Anweisung nicht ausgeführt werden.

Vor einer Programmzeile mit CHK-Anweisung muss immer der Pointer P254 als Sprungzieladresse stehen. Der Pointer kennzeichnet den Beginn der Fehlerkontrolle.

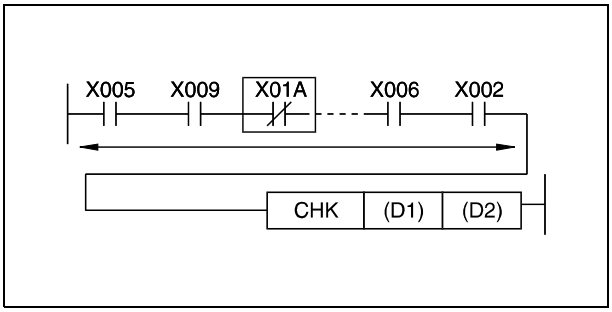
Die CHK-Anweisung kann an jedem beliebigen Programmschritt des Ablaufprogramms programmiert werden. Insgesamt darf die CHK-Anweisung nur einmal im Programm vorkommen.

Das Schreiben CHK-Anweisung im RUN-Betrieb der CPU ist nicht möglich.

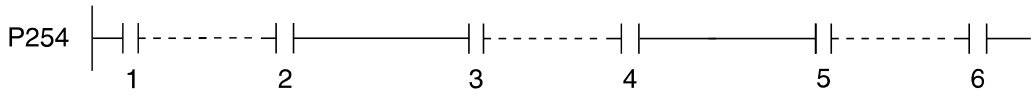
Die Kontrollbedingungen müssen mittels einer LD- oder AND-Anweisung vor der CHK-Anweisung programmiert werden. Andere Eingangsanweisungen sind nicht möglich. Wenn in den Kontrollbedingungen eine ANI-Anweisung programmiert wird, kann die Fehlerkontrolle der CHK-Anweisung nicht ausgeführt werden.



Die Fehlerkontrolle erfolgt in der Reihenfolge der Eingangskontakte, die als Kontrollvariablen vor der CHK-Anweisung programmiert sind. Werden zwei oder mehr Fehler erkannt, wird nur der Fehlercode mit der höheren Priorität gespeichert.



Der Datenwert des in d2 gespeicherten Fehlercodes ist von verschiedenen Faktoren abhängig.



Fehlerzustände	Eingangskontakte 1 bis 50	Eingangskontakte 51 bis 100	Eingangskontakte 101 bis 150
Fehlerzustand 1 (Fehlercode Nr. 1)	100 + (2 x (Kontaktadr.) - 1	400 + (2 x (Kontaktadr.) - 1	700 + (2 x (Kontaktadr.) - 1
Fehlerzustand 2 (Fehlercode Nr. 2)	101 + (2 x (Kontaktadr.) - 1	401 + (2 x (Kontaktadr.) - 1	701 + (2 x (Kontaktadr.) - 1
Fehlerzustand 3 (Fehlercode Nr. 3)	200 + (2 x (Kontaktadr.) - 1	500 + (2 x (Kontaktadr.) - 1	800 + (2 x (Kontaktadr.) - 1
Fehlerzustand 4 (Fehlercode Nr. 4)	201 + (2 x (Kontaktadr.) - 1	501 + (2 x (Kontaktadr.) - 1	801 + (2 x (Kontaktadr.) - 1
Fehlerzustand 5 (Fehlercode Nr. 5)	300 + (2 x (Kontaktadr.) - 1	600 + (2 x (Kontaktadr.) - 1	900 + (2 x (Kontaktadr.) - 1
Fehlerzustand 6 (Fehlercode Nr. 6)	301 + (2 x (Kontaktadr.) - 1	601 + (2 x (Kontaktadr.) - 1	901 + (2 x (Kontaktadr.) - 1

[1] Kontaktadresse 1

[2] Kontaktadresse 50

[3] Kontaktadresse 51

[4] Kontaktadresse 100

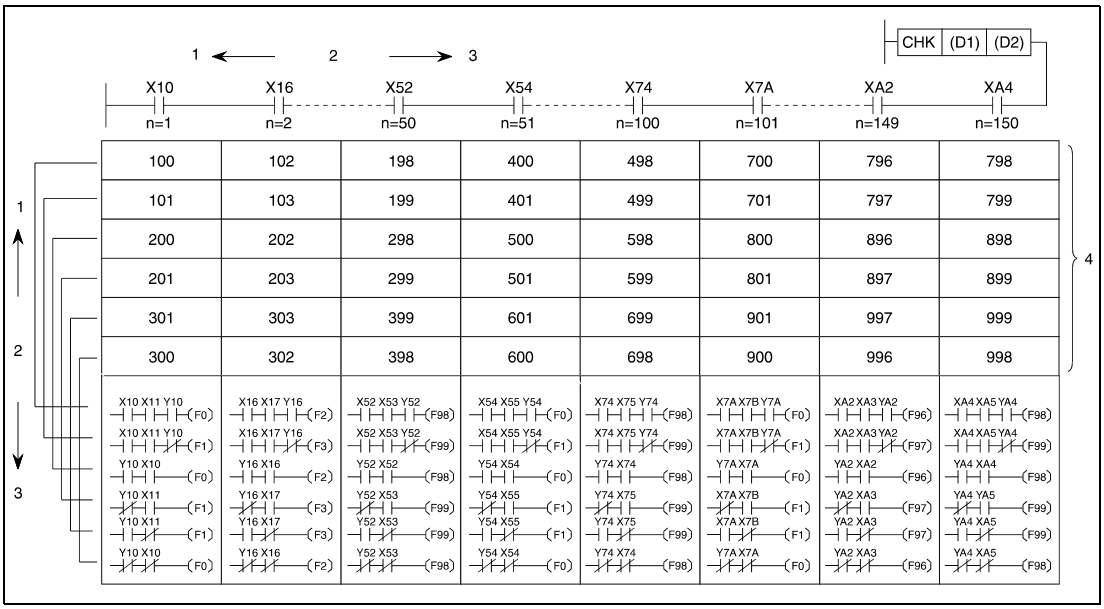
[5] Kontaktadresse 101

[6] Kontaktadresse 150

Die nach Ausführung der CHK-Anweisung ausgegebenen Fehlercodes geben Auskunft über die Art des aufgetretenen Fehlers. Zur schnellen Fehlerdiagnose empfiehlt es sich, eine Übersichtstabelle der Fehlercodes anzufertigen.

Fehlercode	Fehlerursache	Fehlerbehebung
301	Fehler am Bandförderer 1: Aufruf des Rückwärtsbetriebs, obwohl der Grenzschalter für Vorwärtsbetrieb nicht eingeschaltet ist.	Grenzschalter X1 überprüfen, Bandförderer überprüfen
302	Fehler am Bandförderer 1: …	…
…	… …	…

Übersicht der Fehlercodeadressen



[1] hoch (Priorität)

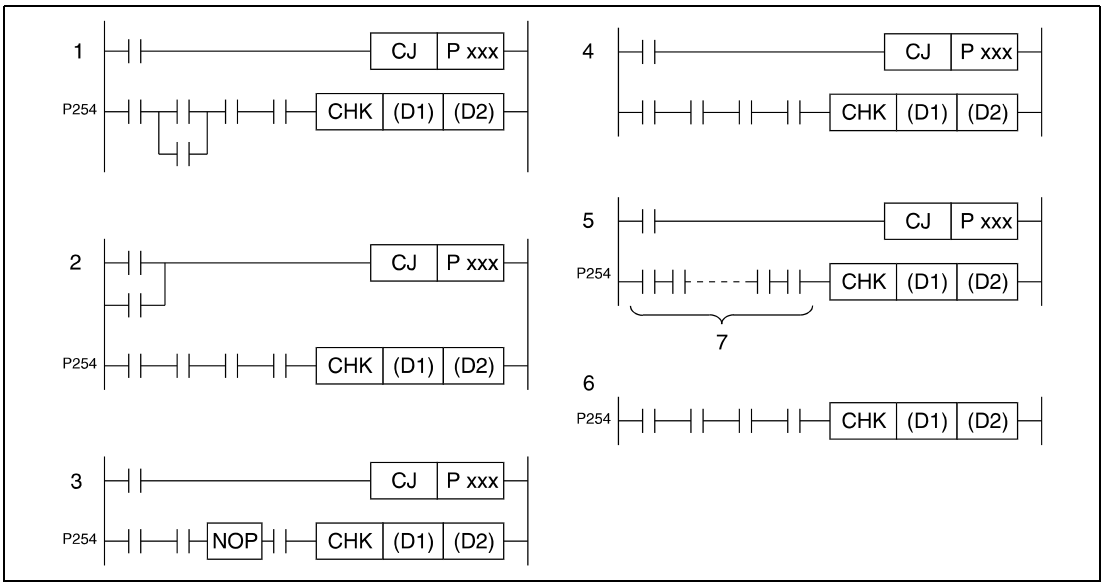
[2] Priorität

[3] niedrig (Priorität)

[4] Fehlercodeadresse

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error-Flag wird gesetzt (Die Angaben in den Klammern beziehen sich auf die folgende Abbildung):

- In den Kontrollbedingungen der CHK-Anweisung (1) oder im Anweisungsblock der CJ-Anweisung (2) werden zwei Eingangskontakte parallelgeschaltet.
- Innerhalb der Kontrollbedingungen der CHK-Anweisung befindet sich eine NOP-Anweisung (3).
- Die Sprungzieladresse P254 fehlt im Programm (4).
- Die Kontrollbedingungen der CHK-Anweisung enthalten mehr als 150 Eingangsoperanden (5).
- Vor dem CHK-Anweisungsblock fehlt die Sprunganweisung (CJ)(6).



[7]mehr als 150 Kontakte

7.10.3 CHKCIR, CHKEND

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●[2]

[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden (nur CHKEND).

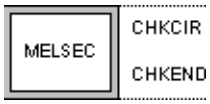
[1] Nicht für Q00JCPU, Q00CPU und Q01CPU.

Operanden MELSEC Q

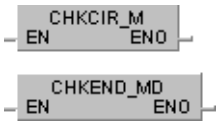
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sonder-module U□\G□	Index-Register Zn	Konstanten	Andere		
	Bit	Wort		Bit	Wort						
—	—	—	—	—	—	—	—	—	—	SM0	1

GX IEC Developer

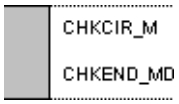
MELSEC-Anweisungsliste



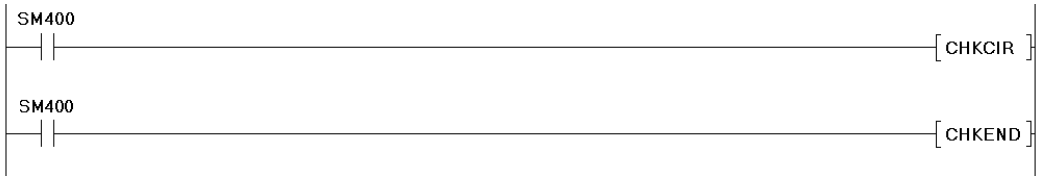
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

Operand	Befehlswert	Datentyp
—	—	—

Funktionsweise

Erzeugen von Prüfnetzwerken für die CHK-Anweisung

CHKCIR, CHKEND **Start- und End-Anweisung für den Programmbereich mit den erzeugten Prüfnetzwerken**

Die CHKCIR- und CHKEND-Anweisungen dienen der Veränderung der Prüfnetzwerke, mit denen die CHK-Anweisung arbeitet. Es kann jedes erforderliche Prüfnetzwerkformat erzeugt werden. Die eigentliche Fehlerkontrolle wird mit den CHKST- und CHK-Anweisungen ausgeführt.

Die Fehlerkontrolle wird mit den Fehlerprüfnetzwerken durchgeführt, die zwischen der CHKCIR- und CHKEND-Anweisung programmiert sind.

HINWEIS

Wenn die Prüfnetzwerkformate für die CHK-Anweisung durch die CHKCIR- und CHKEND-Anweisung verändert werden ist es erforderlich, angeschlossene Peripheriegeräte im Standard-Modus zu starten, und anschließend eine Programmerweiterung durchzuführen.

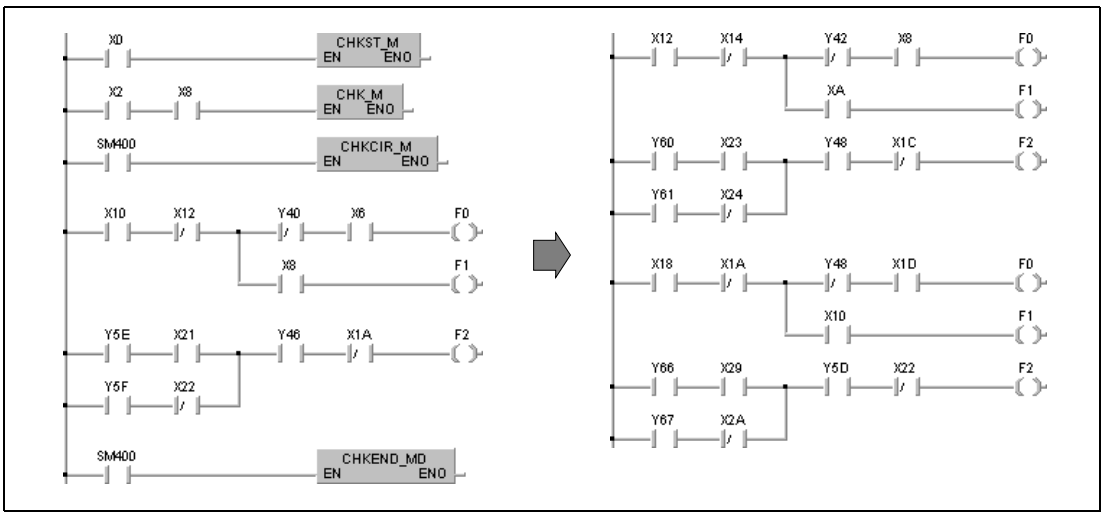
Für den Fall, das ein Peripheriegerät durch eine Q2A, Q2AS, Q3A oder Q4A CPU gestartet wird, und der Versuch unternommen wird, geänderte Fehlerprüfnetzwerke mittels CHKCIR- und CHKEND-Anweisung für die CHK-Anweisung zu erstellen, ist eine einwandfreie Verarbeitung nicht mehr gewährleistet

Aus den zwischen der CHKCIR- und CHKEND-Anweisung programmierten Fehlerprüfnetzwerken werden durch indizierte Adressierung neue Fehlerprüfnetzwerke erzeugt. Den zwischen den Anweisungen programmierten Fehlernetzwerken können 9 Fehlermerker (F1 - F9) zugewiesen werden. Die indizierte Adressierung wird durch Addition der vor der CHK-Anweisung angegebenen Kontaktadressen mit den Kontaktadressen der Fehlerprüfnetzwerke vorgenommen. Der Kontakt X10 in den Fehlerprüfnetzwerken der folgenden Abbildung wird beispielsweise durch die vor der CHK-Anweisung programmierten Kontakte X2 und X8 in den indiziert adressierten Prüfnetwerken X12 und X18 lauten.

Der Fehlerkontrollalgorithmus ist von dem Zustand des Merkers SM710 wie folgt abhängig.

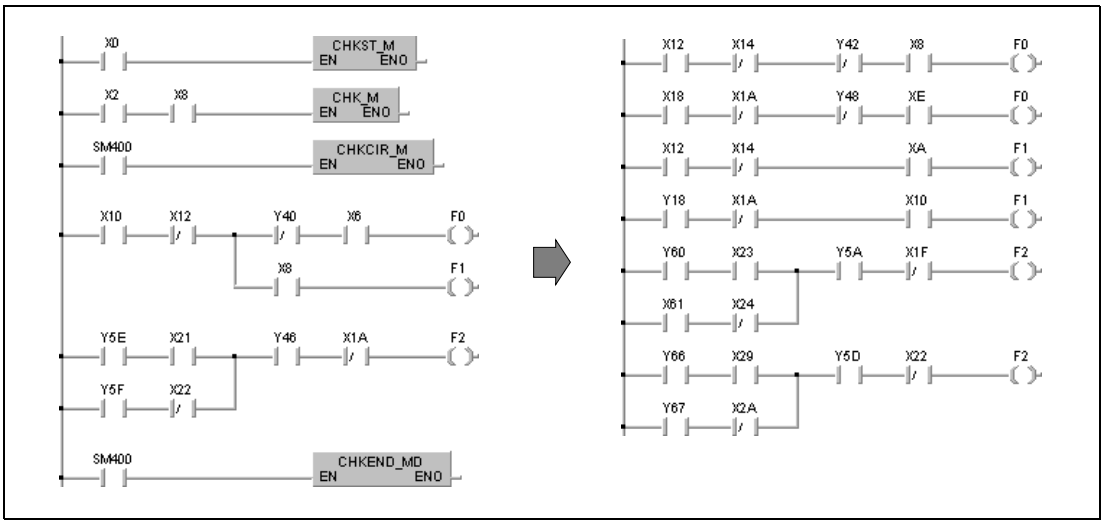
SM710 ist zurückgesetzt (0):

In diesem Fall wird jede Kontaktadresse der zwischen der CHKCIR- und CHKEND-Anweisung programmierten Fehlerprüfnetzwerke zunächst mit der ersten vor der CHK-Anweisung angegebenen Kontaktadresse indiziert adressiert. Anschließend wird wiederum jedes programmierte Prüfnetzwerk mit der zweiten vor der CHK-Anweisung angegebenen Kontaktadresse indiziert adressiert. Dieser Vorgang ist abgeschlossen, wenn für jedes programmierte Prüfnetzwerk mit zugewiesenem Fehlermerker (F) eine zu den Eingangskontakten der CHK-Anweisung äquivalente Anzahl von neuen Prüfnetzwerken existiert.



SM710 ist gesetzt (1):

In diesem Fall wird zunächst das erste programmierte Fehlerprüfnetzwerk mit einem zugewiesenen Fehlermerker mit allen vor der CHK-Anweisung programmierten Kontaktadressen indiziert adressiert. Darauffolgend wird das nächste Prüfnetzwerk mit allen vor der CHK-Anweisung programmierten Kontaktadressen indiziert adressiert. Dieser Vorgang ist abgeschlossen, wenn für jedes programmierte Prüfnetzwerk mit zugewiesenem Fehlermerker (F) eine zu den Eingangskontakten der CHK-Anweisung äquivalente Anzahl von neuen Prüfnetzwerken existiert.



Während der Fehlerkontrolle der indiziert adressierten Fehlerprüfnetzwerke werden die mit der OUT F-Anweisung gesetzten Ausgänge (F) auf ihren Zustand abgefragt. Ist ein Ausgang (F) gesetzt, wird der Merker SM80 gesetzt. Der Fehlercode bestehend aus Kontaktnummer und Fehlerprüfnetzwerk (F1 - F9) wird im BCD-Datenformat in dem Diagnoseregister SD80 gespeichert.

Die Fehlerprüfnetzwerke zwischen der CHKCIR- und CHKEND-Anweisung können mit folgenden Anweisungen programmiert werden.

Kontakte:

LD, LDI, AND, ANI, OR, ORI, ANB, ORB, MPS, MPP, MRD und Vergleichsanweisungen.

Spulen:

OUT F

Als Operanden sind für Kontakte die Eingänge X und die Ausgänge Y zu programmieren.

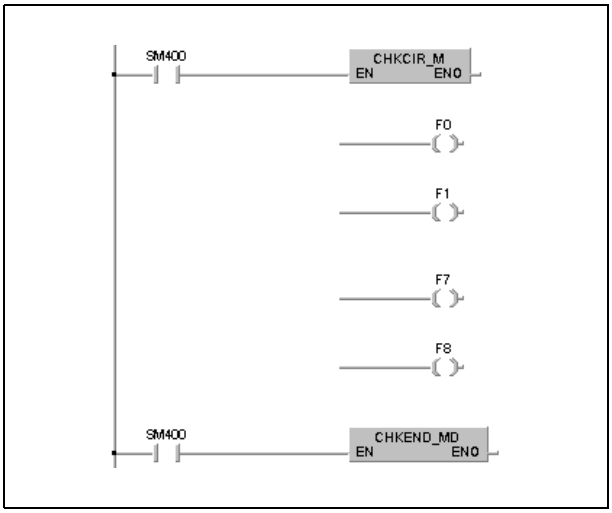
Als Ausgänge der Fehlerprüfnetzwerke sind nur Fehlermerker (F) zu verwenden. Die Bezeichnung der Fehlerprüfnetzwerke kann ab dem Fehlermerker F0 beliebig erfolgen, da diese Ausgänge als Dummys verarbeitet werden. Aus diesem Grund treten bei der Verarbeitung überlappender Fehlermerker (F) keine Probleme auf.

Die Kontrolle der Zustände der Fehlermerker (F) erfolgt auch dann einwandfrei, wenn ein identischer Fehlermerker (F) gleichzeitig in einem anderen Zusammenhang außerhalb CHK-Anweisung programmiert ist, da diese beiden Fehlermerker getrennt verarbeitet werden.

Da die von der CHK-Anweisung verwendeten Fehlermerker (F) ihren Zustand (0/1) nicht aktuell anpassen, werden die Fehlermerker (F) auch dann nicht gesetzt, wenn sie von einem Peripheriegerät überwacht werden.

Die zwischen der CHKCIR- und CHKEND-Anweisung programmierten Fehlerprüfnetzwerke können aus 256 Programmschritten (Kontaktzweigen) und über 9 von OUT F-Anweisungen angesprochenen Ausgängen (Fehlermerkern F1 - F9) bestehen.

Die Bezeichnung der Fehlerprüfnetzwerke zwischen der CHKCIR und CHKEND-Anweisung erfolgt von oben beginnend mit Prüfnetzwerk 1 (F0) bis Prüfnetzwerk 9 (F8).



Die CHKCIR- und CHKEND-Anweisungen können an jedem beliebigen Programmschritt des Ablaufprogramms programmiert werden. Insgesamt dürfen diese Anweisungen nur zwei mal in allen Programmdateien, die ausgeführt werden, und einmal in einem Projekt vorhanden sein.

Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die CHKCIR- und CHKEND-Anweisungen kommen in einem Projekt drei mal oder öfter vor (Fehlercode 4235).
- Die CHKEND-Anweisung wird nicht nach der CHKCIR-Anweisung ausgeführt (Fehlercode 4230).
- Die CHKEND-Anweisung wird ausgeführt ohne das zuvor eine CHKCIR-Anweisung ausgeführt wird (Fehlercode 4230).
- Es werden zehn oder mehr Fehlermerker (F) (Fehlerprüfnetzwerke) programmiert (Fehlercode 4235).
- In den Fehlerprüfnetzwerken werden unzulässige Operanden verwendet (Fehlercode 4235).
- Einer der in den Fehlerprüfnetzwerken programmierten Operanden wird bereits indiziert adressiert (Fehlercode 4235).

HINWEIS

Folgende an einem Peripheriegerät während der Programmerweiterung auftretenden Fehler können die Ausführung der Programmerweiterung verhindern:

- *In den Fehlerprüfnetzwerken werden unzulässige Operanden verwendet.*
- *Einer der in den Fehlerprüfnetzwerken programmierten Operanden wird bereits indiziert adressiert.*

Für den Fall, dass einer der hier beschriebenen Fehler auftritt, sind die Fehlerprüfnetzwerke entsprechend zu korrigieren.

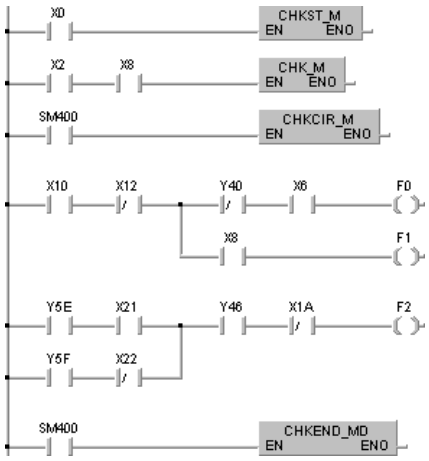
Beispiel CHKCIR, CHKEND

Das folgende Programm dient der Erzeugung indiziert adressierter Fehlerprüfnetzwerke. Die Arbeitsweise dieses Programmbeispiels ist unter dem Punkt Funktionsweise erläutert. Ergänzend zum Kontaktplan sind hier die MELSEC- und IEC-Anweisungslisten angegeben.

MELSEC-Anweisungsliste

```
MELSEC
LD      X0
CHKST
LD      X2
AND     X8
LD      SM400
CHKCIR
LD      X10
ANI     X12
ANI     X6
OUT     F0
LD      X10
ANI     X12
AND     X8
OUT     F1
LD      Y5E
AND     X21
LD      Y5F
ANI     X22
ORB
AND     Y46
ANI     X1A
OUT     F2
LD      SM400
CHKEND
```

Kontaktplan



IEC-Anweisungsliste

```
LD      X0
CHKST_M
LD      X2
AND     X8
LD      SM400
CHKCIR_M
LD      X10
ANDN    X12
ANDN    X6
ST      F0
LD      X10
ANDN    X12
AND     X8
ST      F1
LD      Y5E
AND     X21
OR(
        Y5F
ANDN    X22
)
AND     Y46
ANDN    X1A
ST      F2
LD      SM400
CHKEND_MD
```

7.10.4 SLT, SLTR

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●[1]	●	●	●	

[1] Nicht in Verbindung mit einer A1N CPU.

Operanden MELSEC A

	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
																							1			

Operanden MELSEC Q

	Operanden								Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sonder-module U□\G□	Index-Register Zn	Konstanten	Andere	
	Bit	Wort		Bit	Wort					
—	—	—	—	—	—	—	—	—	—	—	1

GX IEC Developer

MELSEC-Anweisungsliste	Kontaktplan	IEC-Anweisungsliste
		

Variablen

Operand	Befehlswert	Datentyp
—	—	—

Funktions-weise

Status Latch setzen und zurücksetzen

SLT Status Latch setzen

Während der Programmüberwachung mit Hilfe des GX IEC Developers kann nicht jeder Operandenzustand zu jedem Zeitpunkt übertragen und angezeigt werden. Für diesen Zweck existiert im Speicherbereich der CPU ein dafür vorgesehener Zustandsspeicher (Status Latch). Der Status-Latch-Speicher muss über Parameter definiert werden. Es werden die Daten eines Zyklus gespeichert. (Siehe GX IEC Developer)

Die SLT-Anweisung führt die Zwischenspeicherung der definierten Operandendaten aus. Die Daten werden im Status-Latch-Speicher gesichert und können überprüft und angezeigt werden.

Die SLT-Anweisung kann nur einmal in einem Programmzyklus ausgeführt werden. Für eine weitere Ausführung muss die SLT-Anweisung mit der SLTR-Anweisung zurückgesetzt werden.

SLTR Status Latch zurücksetzen

Mit der SLTR-Anweisung werden die im Status-Latch-Bereich zwischengespeicherten Daten gelöscht und die SLT-Anweisung zurückgesetzt.

Die erneute Ausführung einer SLT-Anweisung ist erst nach Ausführung der SLTR-Anweisung möglich.

HINWEISE

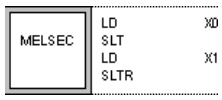
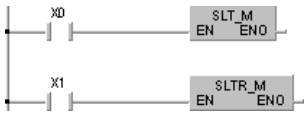
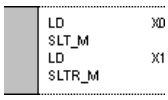
Bitte prüfen Sie in Ihrer GX IEC Developer-Version, ob Status-Latch-Funktionalität vorhanden und einsatzbereit ist.

Weitere Informationen zum Thema Status Latch können Sie in den Bedienungsanleitungen der CPUs und im GX IEC Developer-Benutzerhandbuch nachlesen.

Mit der Ausführung einer SLT-Anweisung nimmt die Programmzykluszeit des in der CPU verarbeiteten Programms unterschiedlich stark zu. Die entsprechenden Verarbeitungszeiten der SLT-Anweisung sind den Bedienungsanleitungen der entsprechenden CPU zu entnehmen. Der Sollwert des Watch-Dog-Timers muss unter Berücksichtigung der aufgeführten Werte festgelegt werden (nur A-Serie).

Beispiel SLT/SLTR

Im folgenden Programm wird die SLT-Anweisung für die Einschaltdauer von X0 ausgeführt. Die SLTR-Anweisung setzt für die Einschaltdauer von X1 die SLT-Anweisung zurück.

MELSEC-Anweisungsliste	Kontaktplan	IEC-Anweisungsliste
		

7.10.5 STRA, STRAR

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	

Operanden MELSEC A

	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)								Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
																							1			

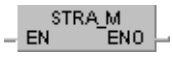
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten	Andere		
	Bit	Wort		Bit	Wort						
—	—	—	—	—	—	—	—	—	—		1

GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan



IEC-Anweisungsliste

Variablen

Operand	Befehlswert	Datentyp
—	—	—

Funktionsweise

Sampling Trace setzen und rücksetzen

STRA Sampling Trace (Abtastüberwachung) setzen

Die Funktion des Sampling Trace überwacht die Daten und Zustände ausgewählter Operanden über einen bestimmten Zeitraum und speichert die kumulierten Daten der abgetasteten Operanden in einem gesonderten Speicherbereich. Die Auswahl der Operanden und der Zeitraum der Abtastung wird über Parameter festgelegt.

STRAR Sampling Trace (Abtastüberwachung) zurücksetzen

Mit der STRAR-Anweisung werden die Daten der Sampling Trace-Programmdatei gelöscht und die STRA-Anweisung sowie Sondermerker M9043 (A-Serie) bzw. die Merker SM801 - SM805 (Q-Serie/System Q) zurückgesetzt.

Die erneute Ausführung einer STRA-Anweisung ist erst nach Ausführung der STRAR-Anweisung möglich.

HINWEISE

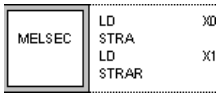
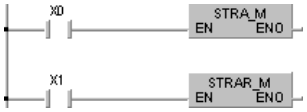
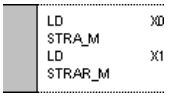
***Bitte prüfen Sie in Ihrer GX IEC Developer-Version*, ob die *Sampling Trace-Funktionalität vorhanden und einsatzbereit*.**

Weitere Informationen zum Thema Sampling Trace können Sie in den Bedienungsanleitungen der CPUs und im GX IEC Developer-Benutzerhandbuch nachlesen.

Während des Zugriffes auf ein ROM ist die Ausführung der STRA- bzw. STRAR-Anweisung nicht möglich (nur A-Serie).

Beispiel STRA/STRAR

Im folgendem Programm wird die STRA-Anweisung für die Einschaltdauer von X0 ausgeführt. Die STRA-Anweisung wird durch die Ausführung der STRAR-Anweisung zurückgesetzt. Die STRAR-Anweisung wird für die Einschaltdauer von X1 ausgeführt.

MELSEC-Anweisungsliste	Kontaktplan	IEC-Anweisungsliste
		

7.10.6 PTRA, PTRAR, PTRAEXE, PTRAEXEP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	

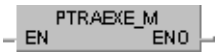
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten	Andere		
	Bit	Wort		Bit	Wort						
—	—	—	—	—	—	—	—	—	—	—	1

GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan



IEC-Anweisungsliste

Variablen

Operand	Befehlswert	Datentyp
—	—	—

Funktionsweise

Program Trace ausführen, setzen und zurücksetzen

PTRA Program Trace (Programmüberwachung) setzen

Die Funktion des Program Trace überwacht die Daten und Zustände von Programmen ausgewählter Operanden über einen bestimmten Zeitraum und speichert die kumulierten Daten der abgetasteten Programme in einem gesonderten Speicherbereich.

Mit der Ausführung der PTRA-Anweisung wird die Abtastung der Programme für die vorgegebene Anzahl von Abtastzyklen und die Zwischenspeicherung der Daten in einem gesonderten Speicherbereich der CPU für die Program Trace-Funktion ermöglicht. Die eigentliche Program Trace-Ausführung wird mit der PTRAEXE-Anweisung gestartet. Für die Speicherung ist der gesetzte Zustand (1) der Merker SM810 - SM812 Voraussetzung.

Bei der Ausführung der PTRA-Anweisung wird der Merker SM813 gesetzt. Nach der Ausführung der vorgegebenen Anzahl von Abtastzyklen werden die Daten für die weitere Verarbeitung gespeichert und die Program Trace-Ausführung abgebrochen.

Wird der Merker SM811 während der Ausführung des Program Trace zurückgesetzt, wird der Abtastvorgang abgebrochen.

Nach vollendeter Ausführung der PTRA-Anweisung wird der Merker SM815 gesetzt.

Vor einer erneuten Ausführung der PTRA-Anweisung ist die PTRAR-Anweisung auszuführen.

Die Ergebnisse der Program Trace-Funktion können von einem Peripheriegerät überwacht werden.

PTRAR Program Trace (Programmüberwachung) zurücksetzen

Mit der PTRAR-Anweisung werden die Daten der Program Trace-Programmdatei gelöscht und die PTRA-Anweisung und die Merker SM811 - SM815 zurückgesetzt.

Die erneute Ausführung einer PTRA-Anweisung ist erst nach Ausführung der PTRAR-Anweisung möglich.

PTRAEXE Program Trace (Programmüberwachung) ausführen

Mit der Ausführung der PTRAEXE-Anweisung wird die Program Trace-Ausführung gestartet.

Wird der Merker SM811 während der Ausführung des Program Trace zurückgesetzt, wird der Abtastvorgang abgebrochen.

Bei nicht gesetzter Ausführungsbedingung der PTRAEXE-Anweisung wird kein Program Trace durchgeführt.

HINWEISE

Bitte prüfen Sie in Ihrer GX IEC Developers-Version, ob die Program Trace-Funktionalität vorhanden und einsatzbereit ist.
Weitere Informationen zum Thema Program Trace können Sie in den Bedienungsanleitungen der CPUs und im GX IEC Developer-Benutzerhandbuch nachlesen.

7.11 Verarbeitungsanweisungen für Zeichenfolgen

Funktion	MELSEC-Anweisung im MELSEC-EDITOR	MELSEC-Anweisung im IEC-EDITOR
Konvertierung von 16-/32-Bit-Binärdaten in Dezimalzahlen im ASCII-Code	BINDA	BINDA_MD
		BINDA _K_MD
		BINDA_S_MD
	BINDAP	BINDA_P_MD
		BINDA_K_P_MD
		BINDA_P_S_MD
	DBINDA	DBINDA_MD
		DBINDA_K_MD
		DBINDA_S_MD
	DBINDAP	DBINDA_P_MD
		DBINDA_K_P_MD
		DBINDA_P_S_MD
Konvertierung von 16-/32-Bit-Binärdaten in Hexadezimalzahlen im ASCII-Code	BINHA	BINHA_MD
		BINHA_K_MD
		BINHA_S_MD
	BINHAP	BINHA_P_MD
		BINHA_K_P_MD
		BINHA_P_S_MD
	DBINHA	DBINHA_MD
		DBINHA_K_MD
		DBINHA_S_MD
	DBINHAP	DBINHA_P_MD
		DBINHA_K_P_MD
		DBINHA_P_S_MD
Konvertierung von 4-/8-stelligen BCD-Daten in den ASCII-Code	BCDDA	BCDDA_MD
		BCDDA_K_MD
		BCDDA_S_MD
	BCDDAP	BCDDA_P_MD
		BCDDA_K_P_MD
		BCDDA_P_S_MD
	DBCDDA	DBCDDA_MD
		DBCDDA_K_MD
		DBCDDA_S_MD
	DBCDDAP	DBCDDA_P_MD
		DBCDDA_K_P_MD
		DBCDDA_P_S_MD

Funktion	MELSEC-Anweisung im MELSEC-EDITOR	MELSEC-Anweisung im IEC-EDITOR
Konvertierung dezimaler ASCII-Daten in 16-/32-Bit-Binärdaten	DABIN	DABIN_MD
		DABIN_S_MD
	DABINP	DABIN_P_MD
		DABIN_P_S_MD
	DDABIN	DDABIN_MD
		DDABIN_S_MD
	DDABINP	DDABIN_P_MD
		DDABIN_P_S_MD
Konvertierung hexadezimaler ASCII-Daten in 16-/32-Bit-Binärdaten	HABIN	HABIN_MD
		HABIN_S_MD
	HABINP	HABIN_P_MD
		HABIN_P_S_MD
	DHABIN	DHABIN_MD
		DHABIN_S_MD
	DHABINP	DHABIN_P_MD
		DHABIN_P_S_MD
Konvertierung dezimaler ASCII-Daten in 4-/8-stellige BCD-Daten	DABCD	DABCD_MD
		DABCD_S_MD
	DABCDP	DABCD_P_MD
		DABCD_P_S_MD
	DDABCD	DDABCD_MD
		DDABCD_S_MD
	DDABCDP	DDABCD_P_MD
		DDABCD_P_S_MD
Auslesen von Kommentardaten	COMRD	COMRD_MD
		COMRD_S_MD
	COMRDP	COMRD_P_MD
		COMRD_P_S_MD
Erfassung der Länge von Zeichenfolgen	LEN	LEN_E
		LEN_MD
		LEN_S_MD
	LENP	LEN_P_S_MD
Konvertierung von 16-/32-Bit-Binärdaten in Zeichenfolgen	STR	STR_MD
		STR_K_MD
		STR_S_MD
	STRP	STR_P_MD
		STR_K_P_MD
		STR_P_S_MD
	DSTR	DSTR_MD
		DSTR_K_MD
		DSTR_S_MD
	DSTRP	DSTR_P_MD
		DSTR_K_P_MD
		DSTR_P_S_MD

Funktion	MELSEC-Anweisung im MELSEC-EDITOR	MELSEC-Anweisung im IEC-EDITOR
Konvertierung von Zeichenfolgen in 16-/32-Bit-Binärdaten	VAL	VAL_MD
		VAL_S_MD
	VALP	VAL_P_MD
		VAL_P_S_MD
	DVAL	DVAL_MD
		DVAL_S_MD
	DVALP	DVAL_P_MD
		DVAL_P_S_MD
Konvertierung von Gleitkommazahlen in Zeichenfolgen	ESTR	ESTR_M
	ESTRP	ESTRP_M
Konvertierung von Zeichenfolgen in dezimale Gleitkommazahlen	EVAL	EVAL_M
	EVALP	EVALP_M
Konvertierung von alphanumerischen Zeichenfolgen in den ASCII-Code	ASC	ASC_MD
		ASC_K_MD
		ASC_S_MD
	ASCP	ASC_P_MD
		ASC_P_S_MD
		ASC_K_P_MD
Konvertierung von hexadezimalen ASCII-Werten in Binärwerte	HEX	HEX_MD
		HEX_K_MD
		HEX_S_MD
	HEXP	HEX_P_MD
		HEX_K_P_MD
		HEX_P_S_MD
Auszug der Zeichenfolgendaten (rechter Teil der Zeichenfolge)	RIGHT	RIGHT_M
	RIGHTP	RIGHTP_M
Auszug der Zeichenfolgendaten (linker Teil der Zeichenfolge)	LEFT	LEFT_M
	LEFTP	LEFTP_M
Speichern und Verschieben von Zeichenfolgenteilen	MIDR	MIDR_M
	MIDRP	MIDRP_M
	MIDW	MIDW_M
	MIDWP	MIDWP_M
Suche von Zeichenfolgen	INSTR	INSTR_M
	INSTRP	INSTRP_M
Gleitkommazahlumrechnung in das BCD-Format	EMOD	EMOD_M
	EMODP	EMODP_M
Gleitkommazahlumrechnunug in das Dezimal-Format	EREXP	EREXP_M
	EREXPP	EREXPP_M

7.11.1 BINDA, BINDAP, DBINDA, DBINDAP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●[2]

[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

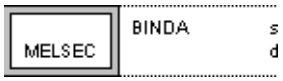
[2] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

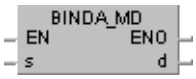
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	●	●	●	●	●	●	●	●	—	—	3
d	—	●	●	—	—	—	—	—	—	—	

GX IEC Developer

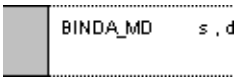
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

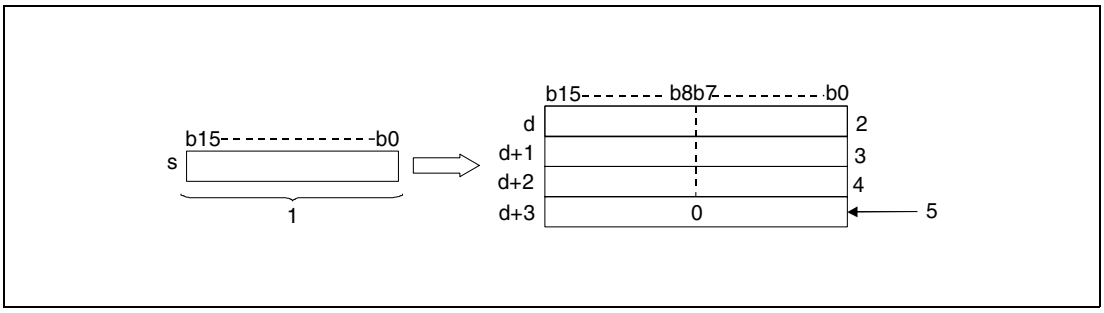
Operand	Befehlswert	Datentyp	
		MELSEC	IEC
s	Binärdaten, die ins ASCII-Format konvertiert werden.	BIN-16-/32-Bit	ANY16/32
d	Erste Adresse des Operanden, in dem das Konvertierungsergebnis gespeichert wird.	Zeichenfolge	Array [1..4]/ [1..6] of ANY16

Funktions-weise

Konvertierung von 16-/32-Bit-Binärdaten in Dezimalzahlen im ASCII-Code

BINDA Konvertierung von 16-Bit-Binärdaten

Die BINDA-Anweisung konvertiert die in s angegebene 16-Bit-Binärzahl in eine Dezimalzahl im ASCII-Code und speichert sie in d (Array_d[1]) bis d+3 (Array_d[4]).



[1] 16-Bit-Binärdaten

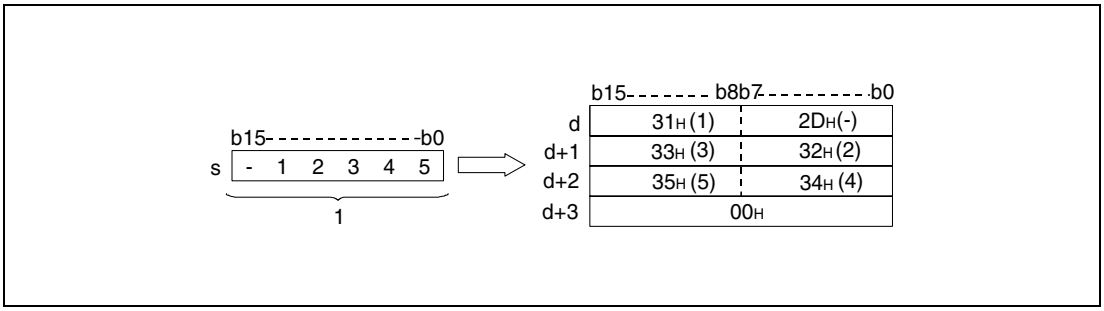
[2] Zehntausender-Stelle im ASCII-Code/ Vorzeichen

[3] Hunderter-Stelle im ASCII-Code/ Tausender-Stelle im ASCII-Code

[4] Einer-Stelle im ASCII-Code/ Zehner-Stelle im ASCII-Code

[5] Bei nicht gesetztem Merker SM701

Der in s angegebene Wert wird beginnend bei d (Array_d[1]) fortschreitend bis d+3 (Array_d[4]) als Dezimalzahl im ASCII-Code gespeichert.



[1] Binärzahl

Die in s angegebene 16-Bit-Binärzahl kann in dem Bereich zwischen -32768 und 32767 liegen.

Die Ergebnisse der Konvertierungsoperationen werden wie folgt ab d gespeichert:

Ist die 16-Bit-Binärzahl positiv, wird das Vorzeichen als "20H" gespeichert.

Bei einer negativen Binärzahl, wird das Vorzeichen als "2DH" gespeichert.

Das gespeicherte Vorzeichen "20H" ersetzt die vorangestellten Nullen.

In der Zahl 00325 werden die Nullen der Zehntausender- und Tausender-Stellen durch "20H" ersetzt, so dass nur die tatsächlich benötigten Stellen gespeichert werden.

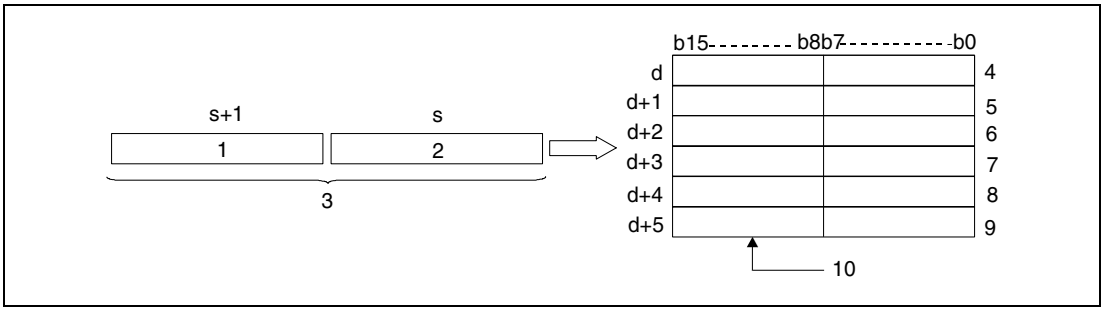
Die Speicherung der Daten in dem in d+3 (Array_d[4]) angegebenen Operanden hängt vom Status des Merkers SM701 ab.

Ist der Merker nicht gesetzt, wird eine Null "00H" in den Bereich d+3 (Array_d[4]) geschrieben.

Ist der Merker gesetzt, bleibt der Wert in d+3 (Array_d[4]) unverändert.

DBINDA Konvertierung von 32-Bit-Binärdaten

Die DBINDA-Anweisung konvertiert die in s und s+1 angegebenen 32-Bit-Binärdaten in eine Dezimalzahl im ASCII-Code und speichert sie in d (Array_d[1]) bis d+5 (Array_d[6]).



[1] Obere 16 Bits

[2] Untere 16 Bits

[3] 32-Bit-Binärdaten

[4] Vorzeichen/ Milliarden-Stelle im ASCII-Code

[5] Zehnmillionen-/ Hundertmillionen-Stelle im ASCII-Code

[6] Hunderttausender-/ Millionen-Stelle im ASCII-Code

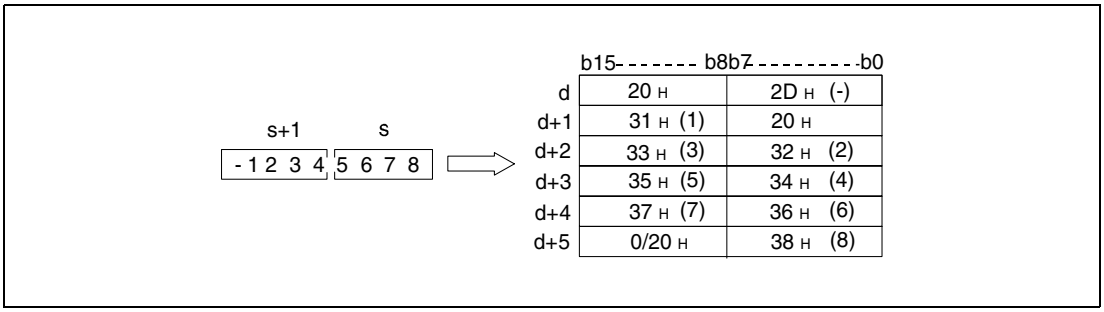
[7] Tausender-/ Zehntausender-Stelle im ASCII-Code

[8] Zehner-/ Hunderter-Stelle im ASCII-Code

[9] 0 oder 20H/ Einer-Stelle im ASCII-Code

[10] Bei nicht gesetztem Merker SM701 (0)/ Bei gesetztem Merker SM701 (20H)

Der in s und s+1 angegebene Wert wird beginnend bei d (Array_d[1]) fortschreitend bis d+5 (Array_d[6]) als Dezimalzahl im ASCII-Code gespeichert.



Die in s angegebene 32-Bit-Binärzahl kann in dem Bereich zwischen -2147483648 und 2147483647 liegen.

Die Ergebnisse der Konvertierungsoperationen werden wie folgt in d (Array_d[1]) bis d+5 (Array_d[6]) gespeichert:

Ist die Binärzahl positiv, wird das Vorzeichen als "20H" gespeichert.

Bei einer negativen Binärzahl, wird das Vorzeichen als "2DH" gespeichert.

Das gespeicherte Vorzeichen "20H" ersetzt die vorangestellten Nullen.

In der Zahl 0012034560 werden die Nullen der Milliarden- und Hundertmillionen-Stellen durch "20H" ersetzt, so dass nur die tatsächlich benötigten Stellen gespeichert werden.

Die Speicherung der Daten in den höherwertigen 8 Bits des Operanden d+5 (Array_d[6]) hängt vom Status des Merkers SM701 wie folgt ab.

Ist der Merker nicht gesetzt, wird eine Null "00H" in den Bereich d+5 (Array_d[6]) geschrieben.

Ist der Merker gesetzt, wird ein Leerzeichen (20H) in den Bereich d+5 (Array_d[6]) geschrieben.

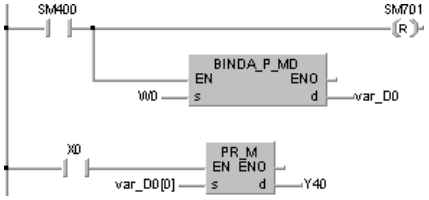
Beispiel 1 BINDAP

Das folgende Programm gibt mit positiver Flanke von SM400 den Wert der 16-Bit-Binärdaten aus W0 mittels der BINDAP-Anweisung als Dezimalwert im ASCII-Code aus. Die Zeichen werden mit der PR-Anweisung an Y40 bis Y48 ausgegeben.

MELSEC-Anweisungsliste

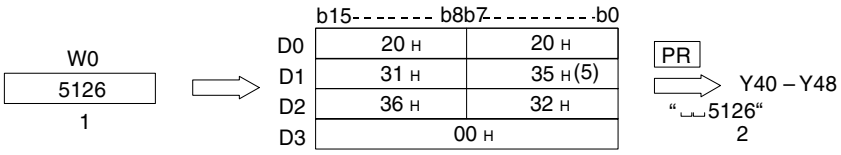
```
MELSEC   LD      SM400
         RST     SM701
         BINDAP  W0
                 D0
         LD      X0
         PR      D0
                 Y40
```

Kontaktplan



IEC-Anweisungsliste

```
LD          SM400
R           SM701
BINDA_P_MD  W0 , var_D0
LD          X0
PR_M        var_D0[0] , Y40
```



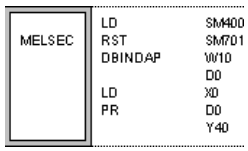
[1] Binärzahl

[2] Ausgabe

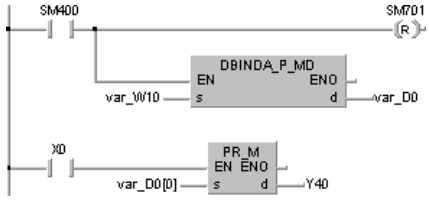
Beispiel 2 DBINDAP

Das folgende Programm gibt mit positiver Flanke von SM400 den Wert der 32-Bit-Binärdaten aus W10 und W11 mittels der DBINDAP-Anweisung als Dezimalwert im ASCII-Code aus. Die Zeichen werden mit der PR-Anweisung an Y40 bis Y48 ausgegeben.

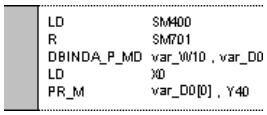
MELSEC-Anweisungsliste

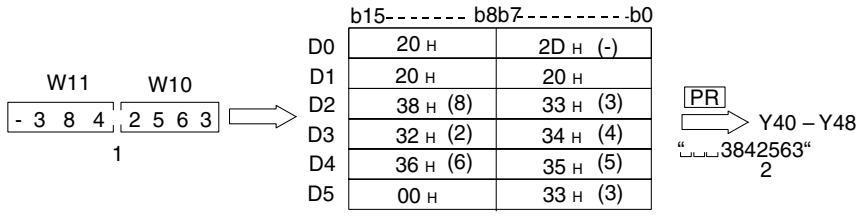


Kontaktplan



IEC-Anweisungsliste





[1] Ausgabe

[2] Binärzahl

HINWEIS *Diese Programmbeispiele sind ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.11.2 BINHA, BINHAP, DBINHA, DBINHAP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●[2]

[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

[2] Nicht für Q00JCPU, Q00CPU und Q01CPU

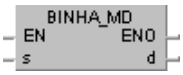
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	●	●	●	●	●	●	●	●	—	—	3
d	—	●	●	—	—	—	—	—	—	—	

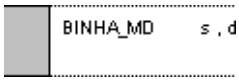
GX IEC Developer

MELSEC-Anweisungsliste

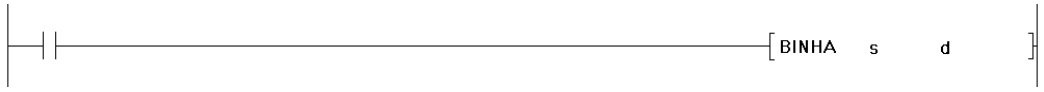
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

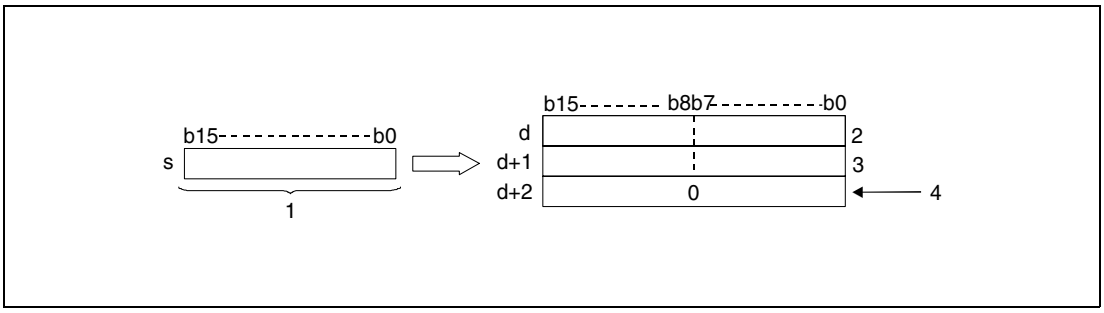
Operand	Befehlswert	Datentyp	
		MELSEC	IEC
s	Binärdaten, die ins ASCII-Format konvertiert werden.	BIN-16-/32-Bit	ANY16/32
d	Erste Adresse des Operanden, in dem das Konvertierungsergebnis gespeichert wird.	Zeichenfolge	Array [1..3]/ [1..5] of ANY16

Funktions-weise

Konvertierung von 16-/32-Bit-Binärdaten in Hexadezimalzahlen im ASCII-Code

BINHA Konvertierung von 16-Bit-Binärdaten

Die BINHA-Anweisung konvertiert die in s angegebenen 16-Bit-Binärdaten in eine Hexadezimalzahl im ASCII-Code und speichert sie in d (Array_d[1]) bis d+2 (Array_d[3]).



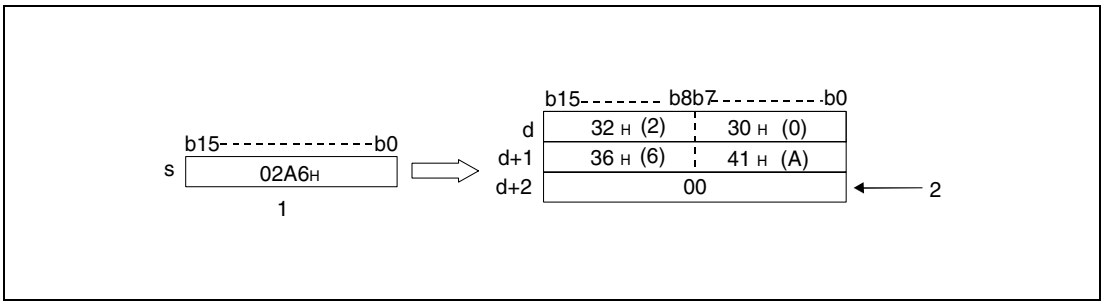
[1] 16-Bit-Binärdaten

[2] ASCII-Code der 3. Stelle/ ASCII-Code der 4. Stelle

[3] ASCII-Code der 1. Stelle/ ASCII-Code der 2. Stelle

[4] Bei nicht gesetztem Merker SM701

Der in s angegebene Wert wird in d (Array_d[1]) bis d+2 (Array_d[3]) im ASCII-Code gespeichert.



[1] 16-Bit-Binärdaten

[2] Bei nicht gesetztem Merker SM701

Die in s angegebenen 16-Bit-Binärdaten dürfen in dem Bereich zwischen 0H und FFFFH liegen.

Das Ergebnis der Konvertierungsoperation wird als 4-stelliger Hexadezimalwert in d (Array_d[1]) bis d+2 (Array_d[3]) gespeichert.

Ist eine der Stellen eine Null, wird dieser Wert auch als Null verarbeitet (es wird keine Nullstellenunterdrückung durchgeführt).

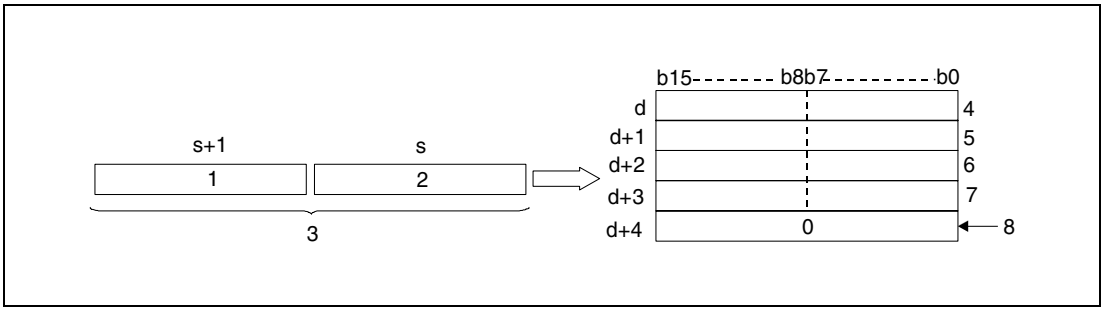
Die Speicherung der Daten in dem in d+2 (Array_d[3]) angegebenen Operanden hängt vom Status des Merkers SM701wie folgt ab:

Ist der Merker nicht gesetzt, wird eine Null "00H" in den Bereich d+2 (Array_d[3]) geschrieben.

Ist der Merker gesetzt, bleibt der Wert in d+2 (Array_d[3]) unverändert.

DBINHA Konvertierung von 32-Bit-Binärdaten

Die in s und s+1 angegebenen 32-Bit-Binärdaten werden in d (Array_d[1]) bis d+4 (Array_d[5]) in hexadezimaler Schreibweise im ASCII-Code gespeichert.



[1] Obere 8 Bits

[2] Untere 8 Bits

[3] 32-Bit-Binärdaten

[4] ASCII-Code der 7. Stelle/ ASCII-Code der 8. Stelle

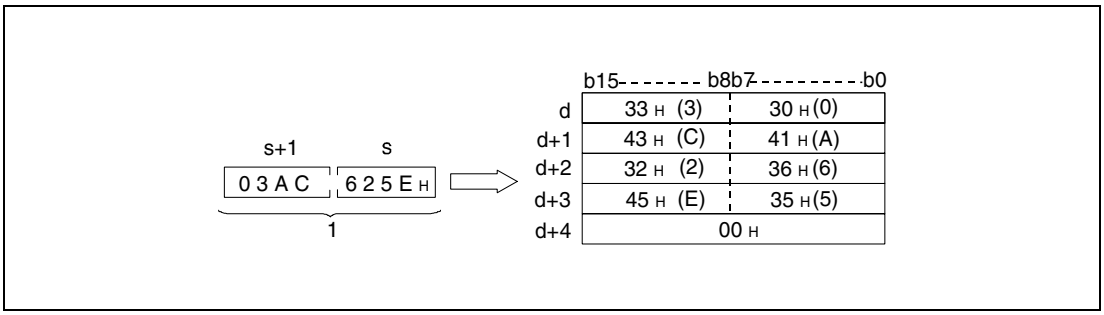
[5] ASCII-Code der 5. Stelle/ ASCII-Code der 6. Stelle

[6] ASCII-Code der 3. Stelle/ ASCII-Code der 4. Stelle

[7] ASCII-Code der 1. Stelle/ ASCII-Code der 2. Stelle

[8] Bei nicht gesetztem Merker SM701

Der in s und s+1 angegebene Wert "03AC625E$_H$" wird in folgender Weise ab d gespeichert:



[1] BIN-32-Bit-Daten

Der in s und s+1 angegebene 32-Bit-Binärwert kann in dem Bereich zwischen 0$_H$ und FFFFFFFF$_H$ liegen.

Das Ergebnis der Konvertierungsoperation wird als 8-stelliger Hexadezimalwert in d (Array_d[1]) bis d+4 (Array_d[5]) gespeichert.

Ist eine der Stellen eine Null, wird dieser Wert auch als Null verarbeitet (es wird keine Null-stellenunterdrückung durchgeführt).

Die Speicherung der Daten in dem in d+4 (Array_d[5]) angegebenen Operanden hängt vom Status des Merkers SM701 wie folgt ab:

Ist der Merker nicht gesetzt, wird eine "00$_H$" (Null) in den Bereich d+4 (Array_d[5]) geschrieben.
Ist der Merker gesetzt, bleibt der Wert in d+4 (Array_d[5]) unverändert.

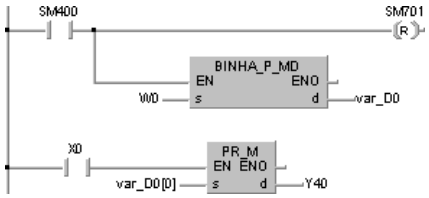
Beispiel 1 BINHAP

Das folgende Programm gibt mit positiver Flanke von SM400 den Wert der 16-Bit-Binärdaten aus W0 mittels der BINHAP-Anweisung als Dezimalwert im ASCII-Code aus. Die Zeichen werden mit der PR-Anweisung an Y40 bis Y48 ausgegeben.

MELSEC-Anweisungsliste

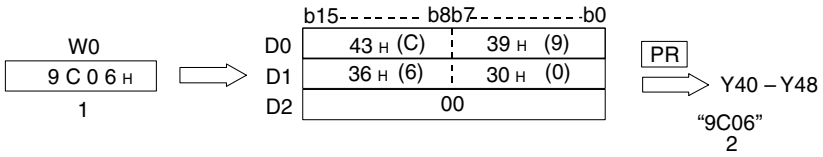
```
MELSEC   LD      SM400
         RST     SM701
         BINHAP  W0
                 D0
         LD      X0
         PR      D0
                 Y40
```

Kontaktplan



IEC-Anweisungsliste

```
LD          SM400
R           SM701
BINHA_P_MD  W0 , var_D0
LD          X0
PR_M        var_D0[0] , Y40
```



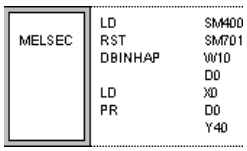
[1] Ausgabe

[2] Binärzahl

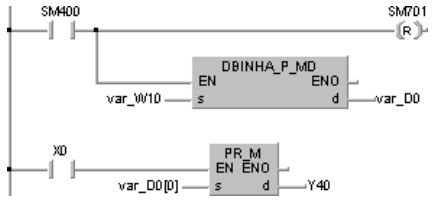
Beispiel 2 DBINHAP

Das folgende Programm gibt mit positiver Flanke von SM400 den Wert der 32-Bit-Binärdaten aus W10 und W11 mittels der DBINHAP-Anweisung als Dezimalwert im ASCII-Code aus. Die Zeichen werden mit der PR-Anweisung an Y40 bis Y48 ausgegeben.

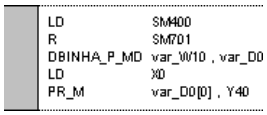
MELSEC-Anweisungsliste

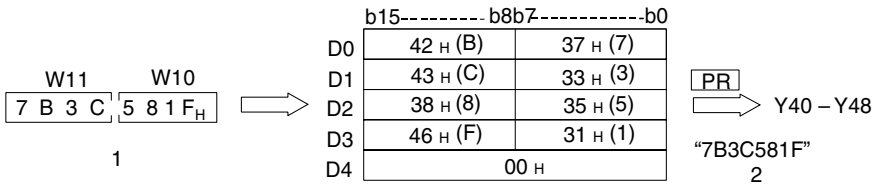


Kontaktplan



IEC-Anweisungsliste





[1] Ausgabe

[2] Binärzahl

HINWEIS *Diese Programmbeispiele sind ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.11.3 BCDDA, BCDDAP, DBCDDA, DBCDDAP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●[2]

[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

[2] Nicht für Q00JCPU, Q00CPU und Q01CPU

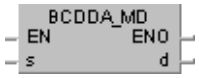
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten $	Andere		
	Bit	Wort		Bit	Wort						
s	●	●	●	●	●	●	●	●	—	SM0	3
d	—	●	●	—	—	—	—	—	—		

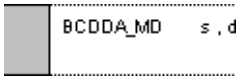
GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

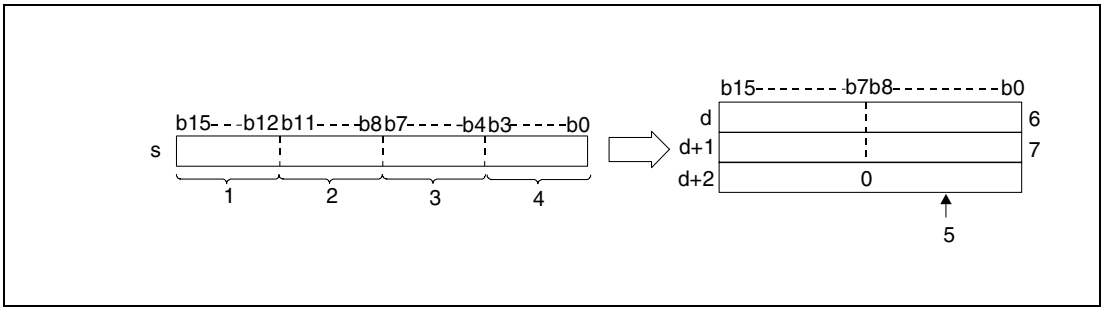
Operand	Befehlswert	Datentyp	
		MELSEC	IEC
s	BCD-Daten, die ins ASCII-Format konvertiert werden.	Wort	ANY16/32
d	Erste Adresse, ab der das Konvertierungsergebnis gespeichert wird.	Zeichenfolge	Array [1..3]/ [1..5] of ANY16

Funktions-weise

Konvertierung von 4-/8-stelligen BCD-Daten in den ASCII-Code

BCDDA Konvertierung von 4-stelligen BCD-Daten

Die BCDDA-Anweisung konvertiert die in s angegebenen 4-stelligen BCD-Daten in das ASCII-Format und speichert sie in d (Array_d[1]) bis d+2 (Array_d[3]).



[1] Tausender-Stelle

[2] Hunderter-Stelle

[3] Zehner-Stelle

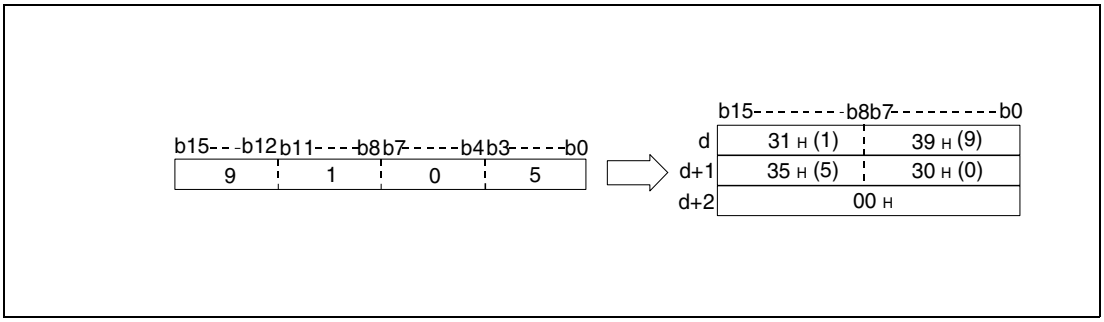
[4] Einer-Stelle

[5] Bei nicht gesetztem Merker SM701

[6] ASCII-Code der 3. Stelle/ ASCII-Code der 4. Stelle

[7] ASCII-Code der 1. Stelle/ ASCII-Code der 2. Stelle

Der in s angegebene Wert 9105 wird in folgender Weise in d gespeichert:



Der in s angegebene BCD-Wert kann in dem Bereich zwischen 0 und 9999 liegen.

Das Ergebnis der Konvertierungsoperation wird in d (Array_d[1]) bis d+2 (Array_d[3]) gespeichert.

Ist eine der Stellen eine Null, wird dieser Wert als "30H" gespeichert (es wird keine Nullstellenunterdrückung durchgeführt).

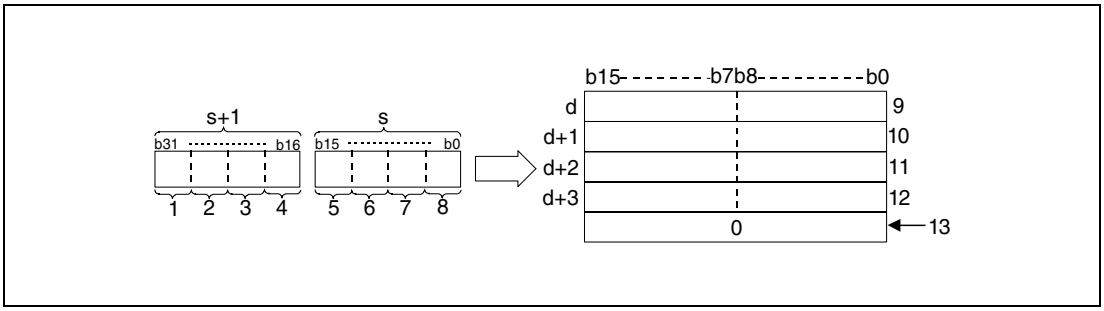
Die Speicherung der Daten in dem in d+2 (Array_d[3]) angegebenen Operanden hängt vom Status des Merkers SM701 wie folgt ab:

Ist der Merker nicht gesetzt, wird eine "00H" (Null) in den Bereich d+2 (Array_d[3]) geschrieben.

Ist der Merker gesetzt, bleibt der Wert in d+2 (Array_d[3]) unverändert.

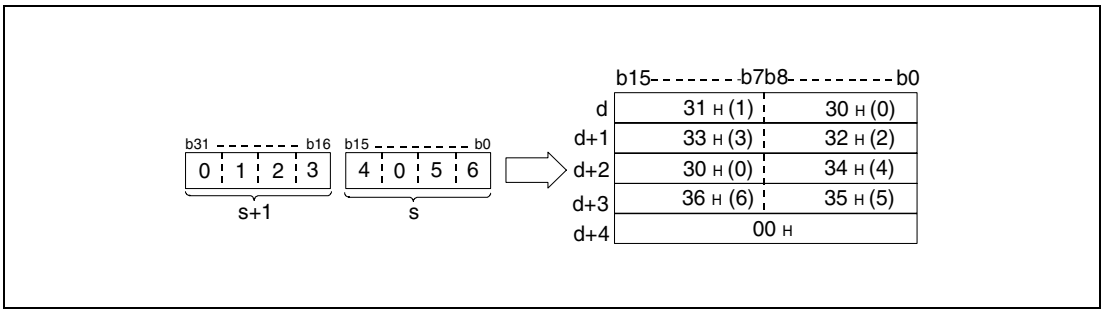
DBCDDA Konvertierung von 8-stelligen BCD-Daten

Die DBCDDA-Anweisung konvertiert die in s und s+1 angegebenen 8-stelligen BCD-Daten in das ASCII-Format und speichert sie in d (Array_d[1]) bis d+4 (Array_d[5]).



[1] Zehnmillionen-Stelle
[2] Millionen-Stelle
[3] Hunderttausender-Stelle
[4] Zehntausender-Stelle
[5] Tausender-Stelle
[6] Hunderter-Stelle
[7] Zehner-Stelle
[8] Einer-Stelle
[9] ASCII-Code der 7. Stelle/ ASCII-Code der 8. Stelle
[10] ASCII-Code der 5. Stelle/ ASCII-Code der 6. Stelle
[11] ASCII-Code der 3. Stelle/ ASCII-Code der 4. Stelle
[12] ASCII-Code der 1. Stelle/ ASCII-Code der 2. Stelle
[13] Bei nicht gesetztem Merker SM701

Der in s und s+1 angegebene Wert 01234056 wird in folgender Weise ab d gespeichert:



Der in s und s+1 angegebene BCD-Wert kann in dem Bereich zwischen 0 und 99999999 liegen.

Das Ergebnis der Konvertierungsoperation wird in d (Array_d[1]) bis d+4 (Array_d[5]) gespeichert.

Ist eine der Stellen eine Null, wird dieser Wert als "30H" gespeichert (es wird keine Nullstellenunterdrückung durchgeführt).

Die Speicherung der Daten in dem in d+4 (Array_d[5]) angegebenen Operanden hängt vom Status des Merkers SM701 ab.

Ist der Merker nicht gesetzt, wird eine Null "00H" in den Bereich d+4 (Array_d[5]) geschrieben.

Ist der Merker gesetzt, bleibt der Wert in d+4 (Array_d[5]) unverändert.

Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die BCD-Daten in s liegen während der Verarbeitung der BCDDA-Anweisung außerhalb des Bereichs von 0 bis 9999 (Fehlercode 4100).
- Die BCD-Daten in s liegen während der Verarbeitung der DBCDDA-Anweisung außerhalb des Bereichs von 0 bis 99999999 (Fehlercode 4100)

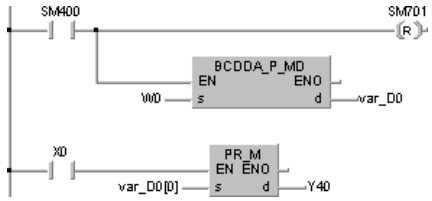
Beispiel 1 BCDDAP

Das folgende Programm gibt mit positiver Flanke von SM400 den Wert der 4-stelligen BCD-Daten aus W0 mittels der BCDDAP-Anweisung als Dezimalwert im ASCII-Code aus. Die Zeichen werden mit der PR-Anweisung an Y40 bis Y48 ausgegeben.

MELSEC-Anweisungsliste

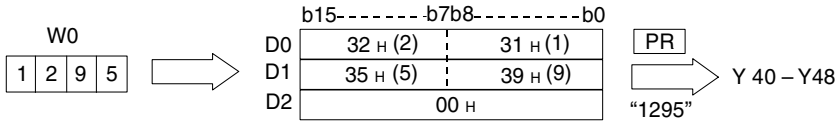
```
MELSEC   LD     SM400
         RST    SM701
         BCDDAP W0
                D0
         LD     X0
         PR     D0
                Y40
```

Kontaktplan



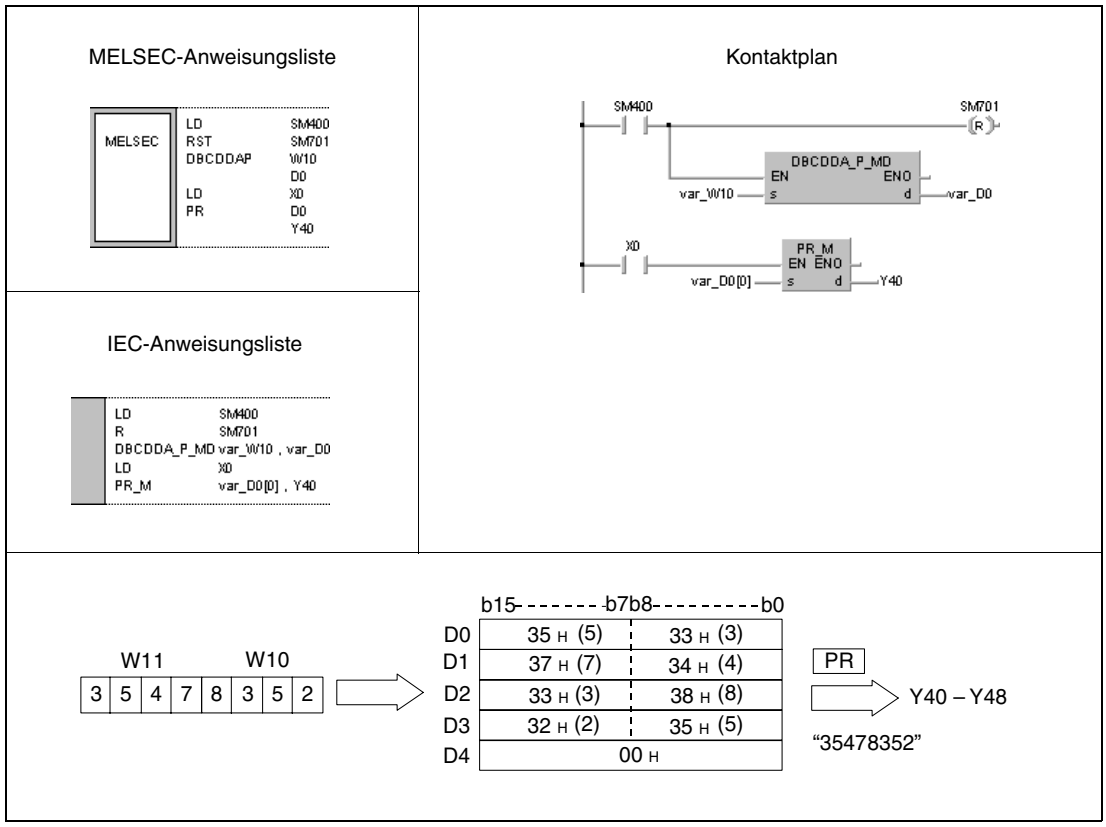
IEC-Anweisungsliste

```
LD          SM400
R           SM701
BCDDA_P_MD  W0 , var_D0
LD          X0
PR_M        var_D0[0] , Y40
```



Beispiel 2 DBCDDAP

Das folgende Programm gibt mit positiver Flanke von SM400 den Wert der 8-stelligen BCD-Daten aus W10 und W11 mittels der DBCDDAP-Anweisung als Dezimalwert im ASCII-Code aus. Die Zeichen werden mit der PR-Anweisung an Y40 bis Y48 ausgegeben.



HINWEIS *Diese Programmbeispiele sind ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer" dieser Programmieranleitung zu entnehmen.*

7.11.4 DABIN, DABINP, DDABIN, DDABINP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●[2]

[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

[2] Nicht für Q00JCPU, Q00CPU und Q01CPU

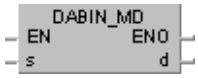
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten $	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	—	—	—	●	—	SM0	3
d	●	●	●	●	●	●	●	—	—		

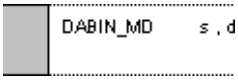
GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

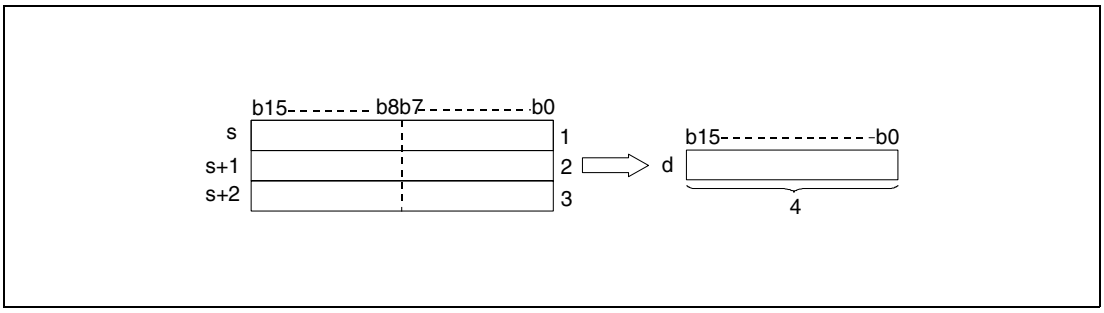
Operand	Befehlswert	Datentyp	
		MELSEC	IEC
s	Speicherbereich, in dem die zu konvertierenden ASCII-Daten gespeichert sind.	Zeichenfolge	Array [1..3]/ [1..6] of ANY16
d	Speicherbereich, in dem das Konvertierungsergebnis gespeichert wird.	BIN-16-/32-Bit	ANY16/32

Funktionsweise

Konvertierung dezimaler ASCII-Daten in 16-/32-Bit-Binär-Daten

DABIN Konvertierung in 16-Bit-Binärdaten

Die DABIN-Anweisung konvertiert die in s (Array_s[1]) bis s+2 (Array_s[3]) angegebenen dezimalen ASCII-Daten in das Format BIN-16-Bit und speichert sie in d.



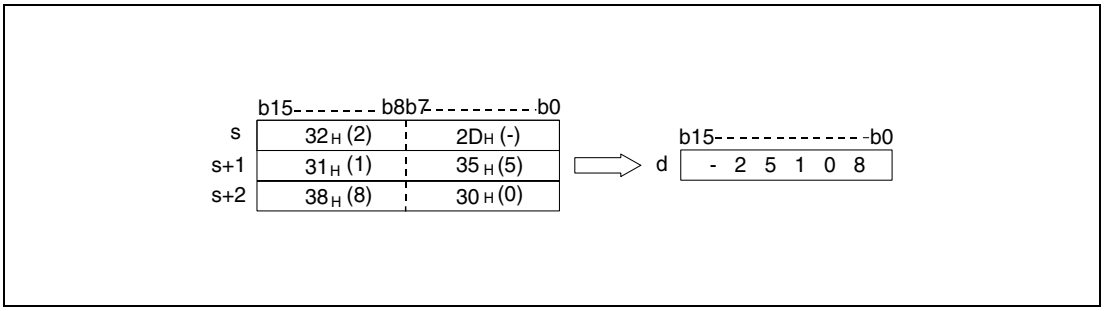
[1] ASCII-Code der Zehntausender-Stelle/ Vorzeichen

[2] ASCII-Code der Hunderter-Stelle/ ASCII-Code der Tausender-Stelle

[3] ASCII-Code der Einer-Stelle/ ASCII-Code der Zehner-Stelle

[4] 16-Bit-Binärdaten

Der in s (Array_s[1]) bis s+2 (Array_s[3]) angegebene Wert wird in folgender Weise in d als -25018H gespeichert:



Der in s (Array_s[1]) bis s+2 (Array_s[3]) angegebene ASCII-Wert kann in dem Bereich zwischen -32768 und 32767 liegen.

Das Vorzeichen wird als "20H" gespeichert, wenn der Binärwert positiv ist.

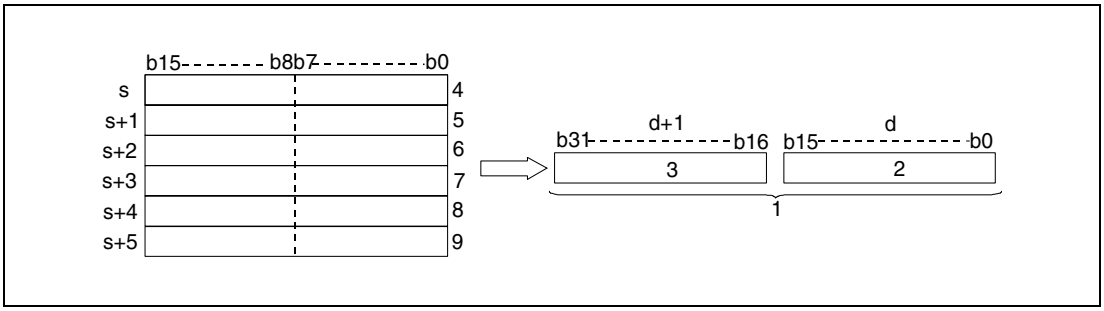
Bei negativem Wert wird der Wert als "2DH" gespeichert.

Jede gespeicherte Stelle des ASCII-Code kann einen Wert zwischen "30H" und "39H" annehmen.

Enthält eine Stelle den Wert "20H" oder "00H", wird dieser Wert automatisch mit dem Wert "30H" überschrieben.

DDABIN Konvertierung in 32-Bit-Binärdaten

Die DDABIN-Anweisung konvertiert die in s (Array_s[1]) bis s+5 (Array_s[6]) angegebenen dezimalen ASCII-Daten in das Format BIN-32-Bit und speichert sie in d und d+1.



[1] 32-Bit-Binärdaten

[2] Untere 16 Bits

[3] Obere 16 Bits

[4] ASCII-Code der Milliarden-Stelle/ Vorzeichen

[5] ASCII-Code der Zehnmillionen-Stelle/ ASCII-Code der Hundertmillionen-Stelle

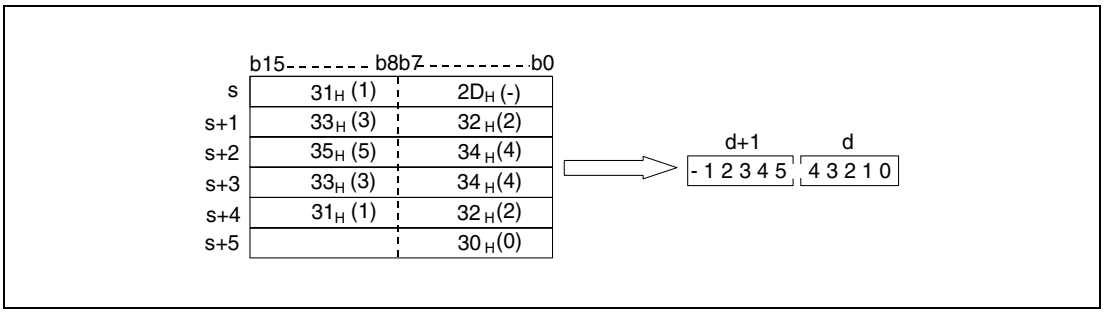
[6] ASCII-Code der Hunderttausender-Stelle/ ASCII-Code der Millionen-Stelle

[7] ASCII-Code der Tausender-Stelle/ ASCII-Code der Zehntausender-Stelle

[8] ASCII-Code der Zehner-Stelle/ ASCII-Code der Hunderter-Stelle

[9] Wird ignoriert/ ASCII-Code der Zehner-Stelle

Der in s (Array_s[1]) bis s+5 (Array_s[6]) angegebene Wert wird in folgender Weise ab d als -1234543210H gespeichert:



Der in s (Array_s[1]) bis s+5 (Array_s[6]) angegebene ASCII-Wert kann in dem Bereich zwischen -2147483648 und 2147483647 liegen.

Das Vorzeichen wird als "20H" gespeichert, wenn der Binärwert positiv ist.

Bei negativem Wert wird der Wert als "2DH" gespeichert.

Jede gespeicherte Stelle des ASCII-Code kann einen Wert zwischen "30H" und "39H" annehmen.

Enthält eine Stelle den Wert "20H" oder "00H", wird dieser Wert automatisch mit dem Wert "30H" überschrieben.

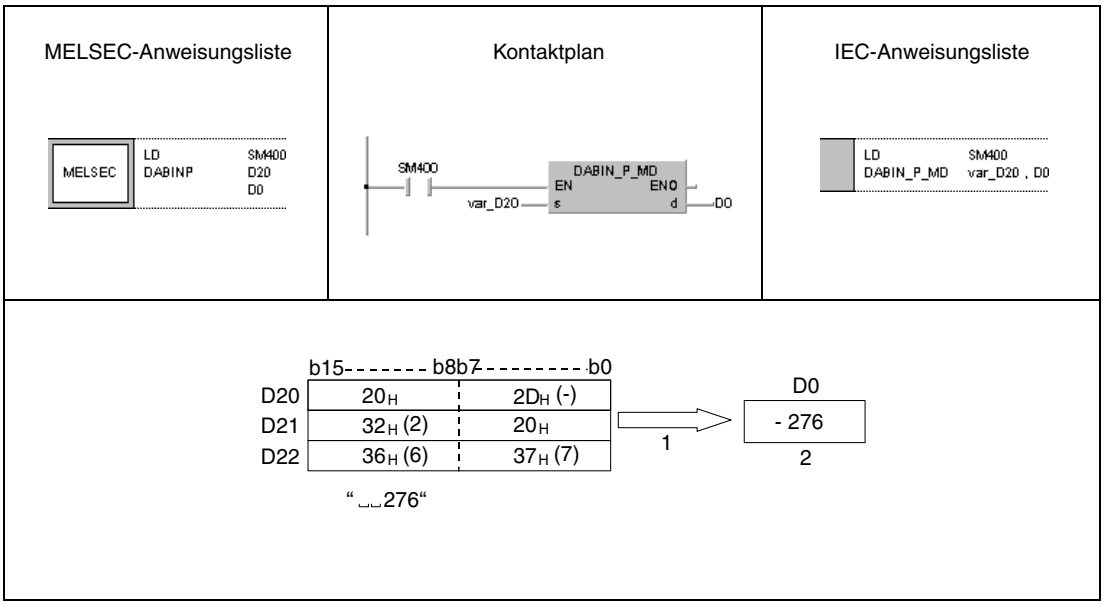
Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Das Vorzeichen in den unteren 8 Bits des Operanden s (Array_s[1]) enthält einen anderen Wert als "20H" oder "2DH" (Fehlercode 4100).
- Der ASCII-Code in s (Array_s[1]) bis s+5 (Array_s[6]) enthält andere Werte als "30H" bis "39H, "20H" oder "00H" (Fehlercode 4100).
- Der ASCII-Code in s (Array_s[1]) bis s+5 (Array_s[6]) liegt außerhalb folgender Bereiche:
  Bei der DABIN-Anweisung -32768 bis 32767
  Bei der DDABIN-Anweisung -2147483648 bis 2147483647 (Fehlercode 4100)

Beispiel 1 DABINP

Das folgende Programm konvertiert mit positiver Flanke von SM400 den fünfstelligen dezimalen ASCII-Wert aus D20 (var_D20 Array [0]) bis D22 (var_D20 Array [2]) in einen Binärwert und speichert ihn in D0.

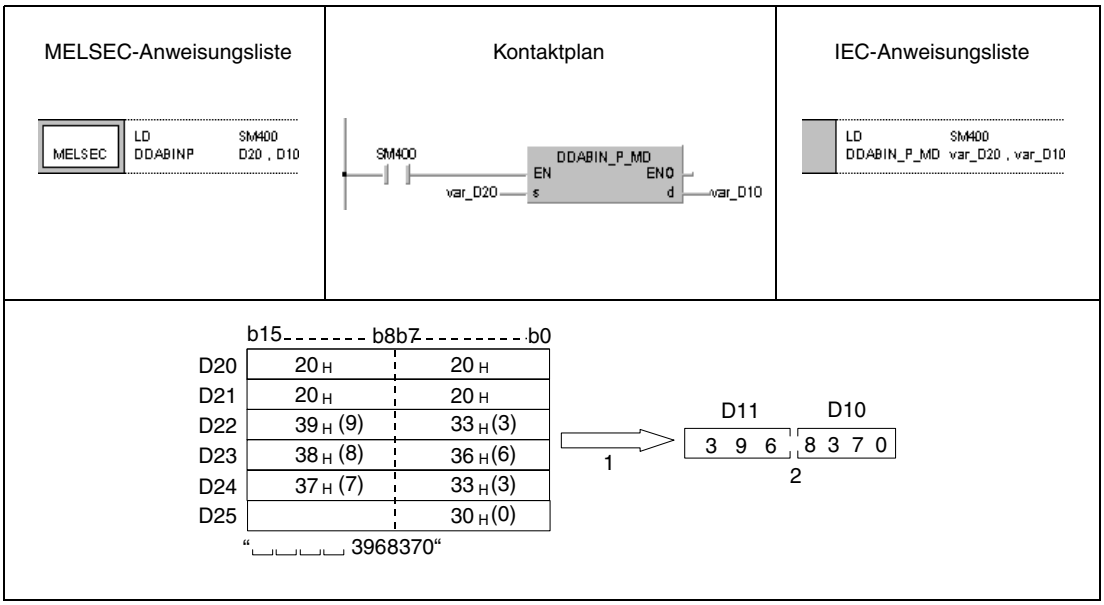


[1] Wird als -00276 gelesen

[2] Binärwert

Beispiel 2 DDABINP

Das folgende Programm konvertiert mit positiver Flanke von SM400 den zehnstelligen dezimalen ASCII-Wert aus D20 (var_D20 Array [0]) bis D25 (var_D20 Array [5]) in einen Binärwert und speichert ihn in D10 und D11.



[1] Wird als +0003968370 gelesen

[2] Binärwert

HINWEIS *Diese Programmbeispiele sind ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.11.5 HABIN, HABINP, DHABIN, DHABINP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●[2]

[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

[2] Nicht für Q00JCPU, Q00CPU und Q01CPU

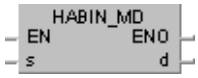
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten $	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	—	—	—	●	—	SM0	3
d	●	●	●	●	●	●	●	—	—		

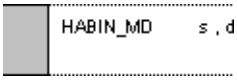
GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

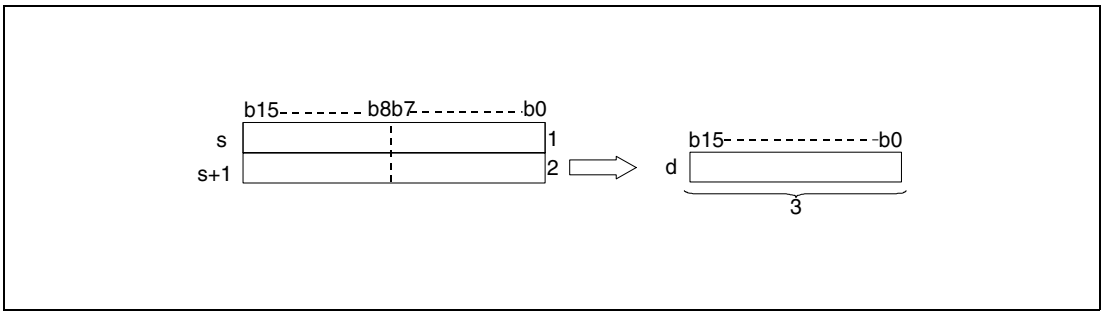
Operand	Befehlswert	Datentyp	
		MELSEC	IEC
s	Speicherbereich, in dem die zu konvertierenden ASCII-Daten gespeichert sind.	Zeichenfolge	ANY32/Array [1..4] of ANY16
d	Speicherbereich, in dem das Konvertierungsergebnis gespeichert wird.	BIN-16-/32-Bit	ANY16/32

Funktions-weise

Konvertierung hexadezimaler ASCII-Daten in 16-/32-Bit-Binärdaten

HABIN Konvertierung in 16-Bit-Binärdaten

Die HABIN-Anweisung konvertiert die hexadezimalen ASCII-Daten in s und s+1 in das BIN-16-Bit-Datenformat und speichert sie in d.

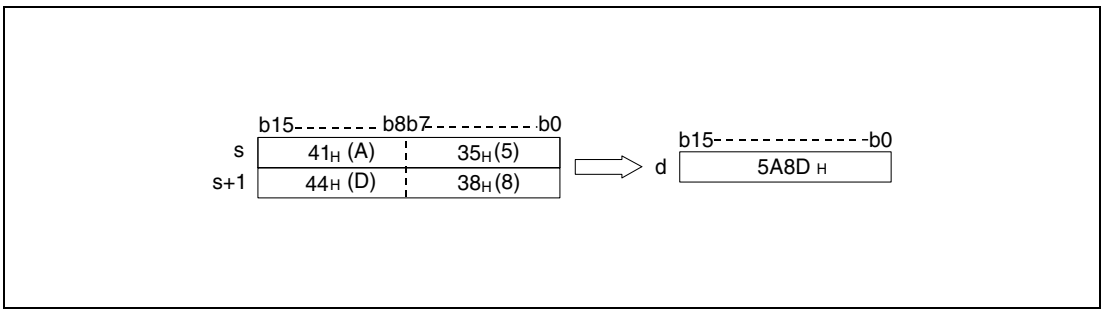


[1] ASCII-Code für die 3. Stelle/ ASCII-Code für die 4. Stelle

[2] ASCII-Code für die 1. Stelle/ ASCII-Code für die 2. Stelle

[3] BIN-16-Bit-Binärdaten

Der in s bis s+1 angegebene Wert "5A8D$_H$" wird nach der Verarbeitung in folgender Weise in d gespeichert:

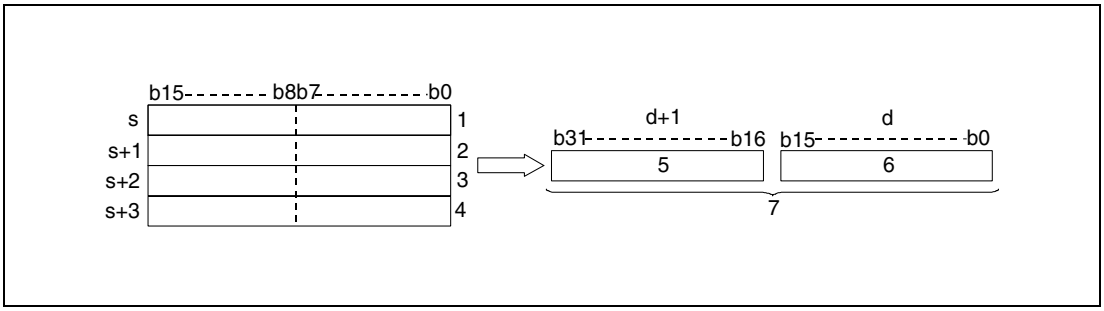


Der in s bis s+1 angegebene ASCII-Wert kann in dem Bereich zwischen 0000$_H$ und FFFF$_H$ liegen.

Jede gespeicherte Stelle des ASCII-Code kann einen Wert zwischen "30$_H$" und "39$_H$" und "41$_H$" und "46$_H$" annehmen.

DHABIN Konvertierung in 32-Bit-Binärdaten

Die DHABIN-Anweisung konvertiert die im Bereich s (Array_s[1]) bis s+3 (Array_s[4]) angegebenen hexadezimalen ASCII-Daten in das Format BIN-32-Bit und speichert sie in d und d+1.



[1] ASCII-Code der 7. Stelle/ ASCII-Code der 8. Stelle

[2] ASCII-Code der 5. Stelle/ ASCII-Code der 6. Stelle

[3] ASCII-Code der 3. Stelle/ ASCII-Code der 4. Stelle

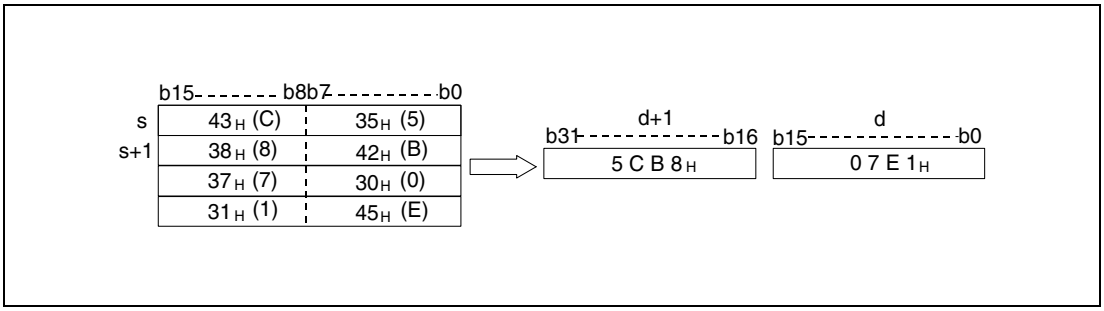
[4] ASCII-Code der 1. Stelle/ ASCII-Code der 2. Stelle

[5] obere 16 Bits

[6] untere 16 Bits

[7] 32-Bit-Binärdaten

Der in s (Array_s[1]) bis s+3 (Array_s[4]) angegebene Wert "5CB807E1" wird nach der Verarbeitung in folgender Weise in d und d+1 gespeichert:



Der in s (Array_s[1]) bis s+3 (Array_s[4]) angegebene ASCII-Wert kann in dem Bereich zwischen 00000000H und FFFFFFFFH liegen.

Jede gespeicherte Stelle des ASCII-Code kann einen Wert zwischen "30H" und "39H" und "41H" und "46H" annehmen.

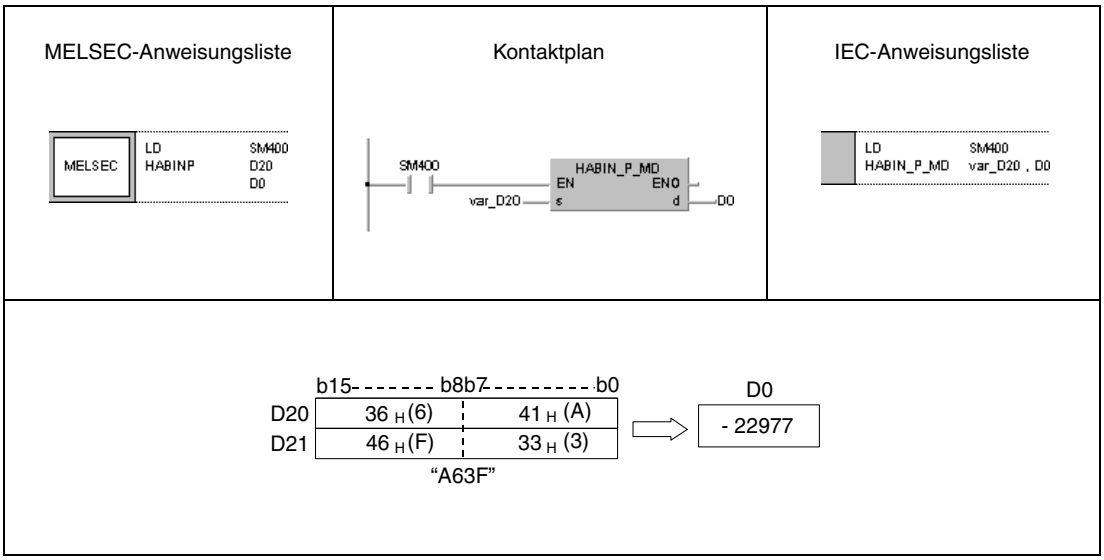
Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Der ASCII-Code in s (Array_s[1]) bis s+3 (Array_s[4]) liegt außerhalb des Bereichs von "30H" bis "39H" und "41H" bis "46H" (Fehlercode 4100).

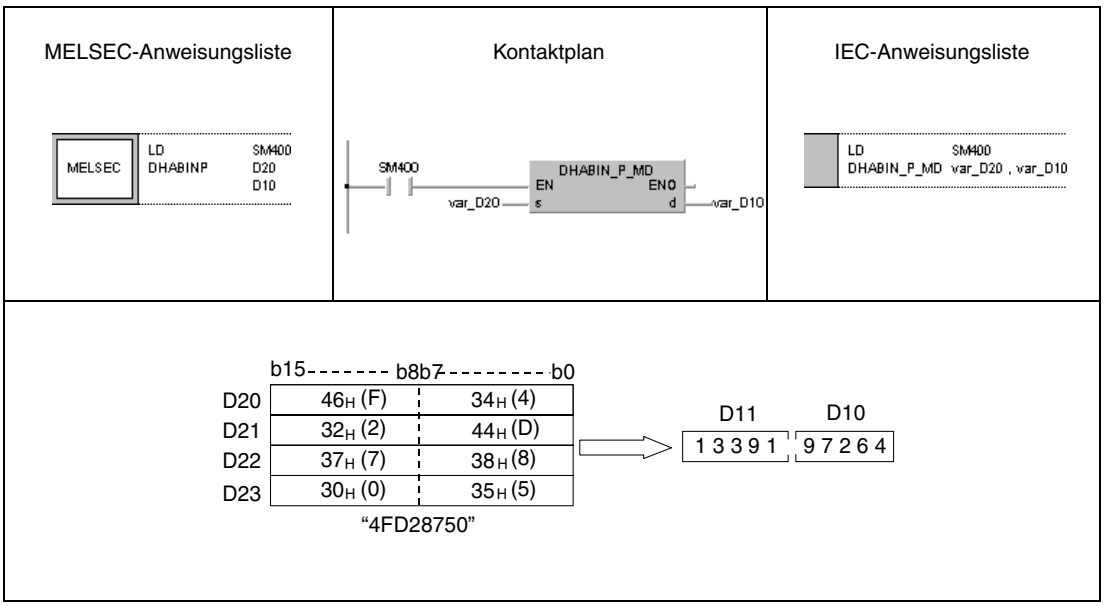
Beispiel 1 HABINP

Das folgende Programm konvertiert mit positiver Flanke von SM400 den 4-stelligen ASCII-Wert aus D20 (var_D20 Array [0]) bis D21 (var_D20 Array [1]) in einen Binärwert und speichert ihn in D0.



Beispiel 2 DHABINP

Das folgende Programm konvertiert mit positiver Flanke von SM400 den 8-stelligen ASCII-Wert aus D20 (var_D20 Array [0]) bis D23 (var_D20 Array [3]) in einen Binärwert und speichert ihn in D10 und D11 ab.



HINWEIS *Diese Programmbeispiele sind ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.11.6 DABCD, DABCDP, DDABCD, DDABCDP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●[2]

[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

[2] Nicht für Q00JCPU, Q00CPU und Q01CPU

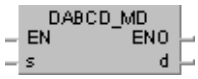
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten $	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	—	—	—	●	—	SM0	3
d	●	●	●	●	●	●	●	—	—		

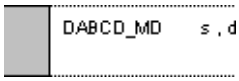
GX IEC Developer

MELSEC-Anweisungsliste

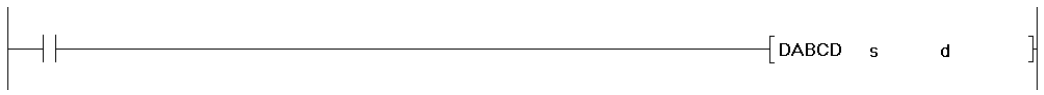
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

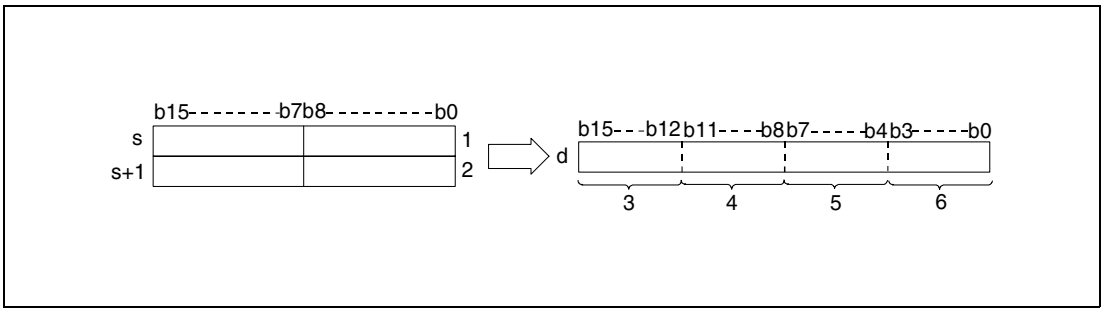
Operand	Befehlswert	Datentyp	
		MELSEC	IEC
s	Speicherbereich, in dem die zu konvertierenden ASCII-Daten gespeichert sind.	Zeichenfolge	ANY32/ Array [1..4] of ANY16
d	Speicherbereich, in dem das Konvertierungsergebnis gespeichert wird.	4-/8-stellige BCD-Daten	ANY16/32

Funktions-weise

Konvertierung dezimaler ASCII-Daten in 4-/8-stellige BCD-Daten

DABCD Konvertierung in 4-stellige BCD-Daten

Die DABCD-Anweisung konvertiert die dezimalen ASCII-Daten in s und s+1 in das 4-stellige BCD-Daten-Format und speichert die Daten in d.



[1] ASCII-Code der Hunderter-Stelle/ ASCII-Code der Tausender-Stelle

[2] ASCII-Code der Einer-Stelle/ ASCII-Code der Zehner-Stelle

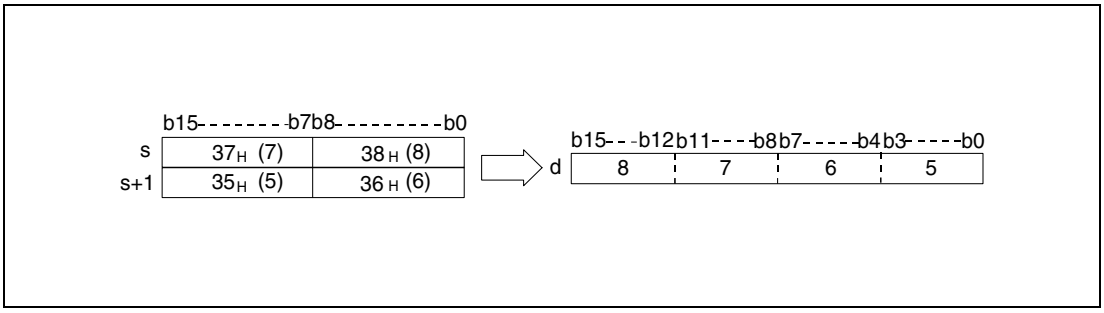
[3] Tausender-Stelle

[4] Hunderter-Stelle

[5] Zehner-Stelle

[6] Einer-Stelle

Der in s und s+1 angegebene Wert 8765 wird in folgender Weise in d gespeichert:



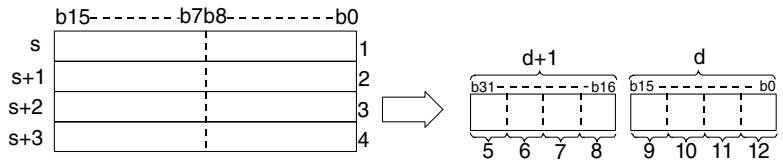
Der in s bis s+1 angegebene ASCII-Wert kann in dem Bereich zwischen 0 und 9999 liegen.

Jede gespeicherte Stelle des ASCII-Code kann einen Wert zwischen "30_H" und "39_H" annehmen.

Enthält eine Stelle den Wert "20_H" oder "00_H", wird der Wert automatisch mit dem Wert "30_H" überschrieben.

DDABCD Konvertierung in 8-stellige BCD-Daten

Die DDABCD-Anweisung konvertiert die in s (Array_s[1]) bis s+3 (Array_s[4]) angegebenen dezimalen ASCII-Daten in das 8-stellige BCD-Format und speichert sie in d und d+1.



[1] ASCII-Code der Millionen-Stelle/ ASCII-Code der Zehnmillionen-Stelle

[2] ASCII-Code der Zehntausender-Stelle/ ASCII-Code der Hunderttausender-Stelle

[3] ASCII-Code der Hunderter-Stelle/ ASCII-Code der Tausender-Stelle

[4] ASCII-Code der Einer-Stelle/ ASCII-Code der Zehner-Stelle

[5] Zehnmillionen-Stelle

[6] Millionen-Stelle

[7] Hunderttausender-Stelle

[8] Zehntausender-Stelle

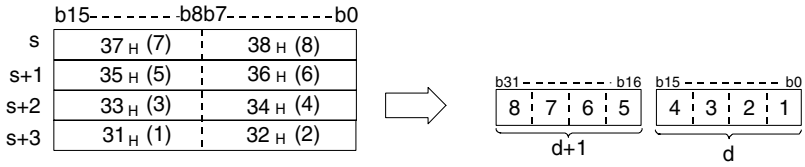
[9] Tausender-Stelle

[10] Hunderter-Stelle

[11] Zehner-Stelle

[12] Einer-Stelle

Der in s (Array_s[1]) bis s+3 (Array_s[4]) angegebene Wert 87654321 wird in folgender Weise in d und d+1 gespeichert:



Der in s (Array_s[1]) bis s+3 (Array_s[4]) angegebene ASCII-Wert kann in dem Bereich zwischen 0 und 99999999 liegen.

Jede gespeicherte Stelle des ASCII-Code kann einen Wert zwischen "30H" und "39H" annehmen.

Enthält eine Stelle den Wert "20H" oder "00H", wird der Wert automatisch mit dem Wert "30H" überschrieben.

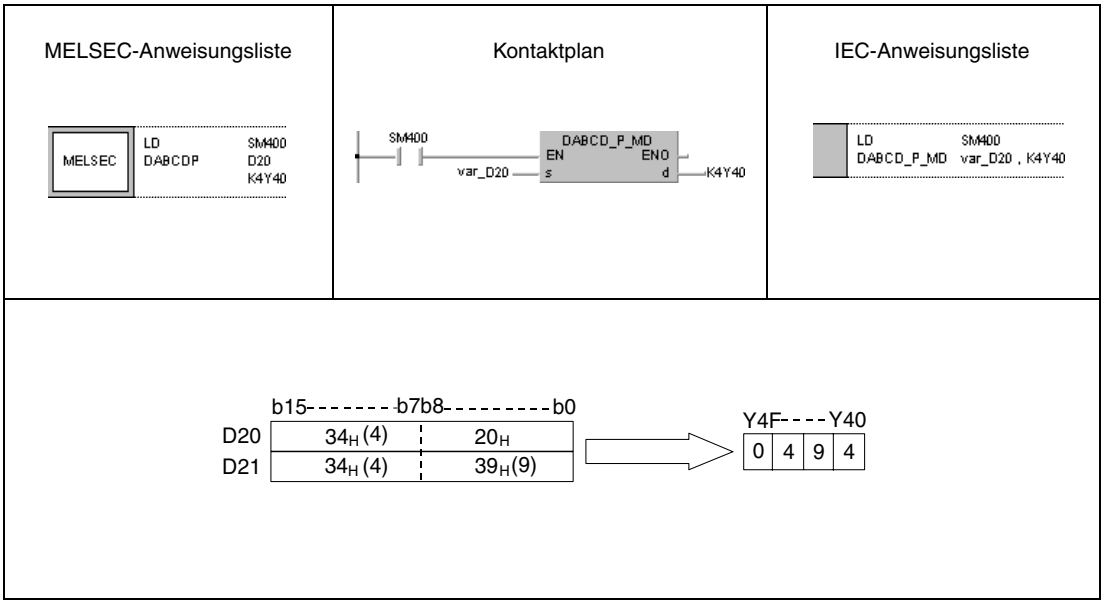
Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Der ASCII-Code in s (Array_s[1]) bis s+3 (Array_s[4]) liegt außerhalb des erlaubten Bereichs von "30_H" bis "39_H" (Fehlercode 4100).

Beispiel 1 DABCDP

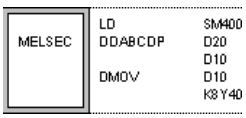
Das folgende Programm konvertiert mit positiver Flanke von SM400 den ASCII-Wert aus D20 (var_D20 Array [0]) bis D21 (var_D20 Array [1]) in einen 4-stelligen BCD-Wert und gibt ihn an Y40 bis Y4F aus.



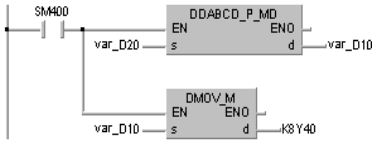
Beispiel 2 DDABCDP

Das folgende Programm konvertiert mit positiver Flanke von SM400 den ASCII-Wert aus D20 (var_D20 [0]) bis D23 (var_D20 [3]) in einen 8-stelligen BCD-Wert, speichert das Ergebnis in D10 und D11 und gibt es an Y40 bis Y5F aus.

MELSEC-Anweisungsliste

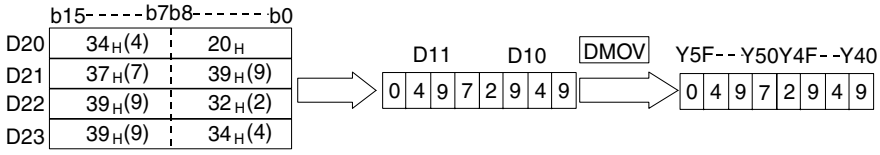


Kontaktplan



IEC-Anweisungsliste

```
LD      SM400
DDABCD_P_MD var_D20 , var_D10
DMOV_M      var_D10 , K8Y40
```



7.11.7 COMRD, COMRDP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●[2]

[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

[2] Nicht für Q00JCPU, Q00CPU und Q01CPU

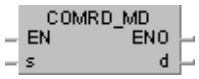
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort				BL\S, BL\TR, BL, P, I, J, U		
s	●	●	●	●	●	●	—	—	●	SM0	3
d	—	●	●	—	—	—	—	—	—		

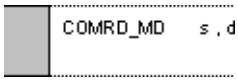
GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

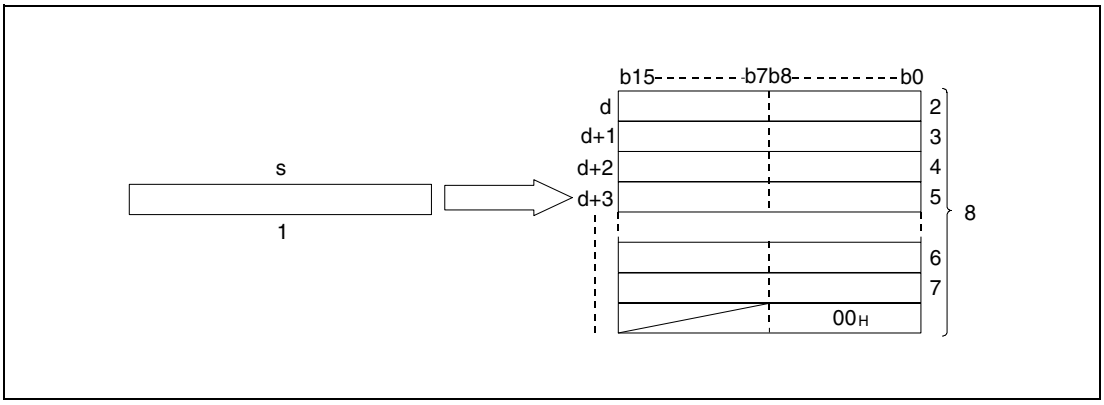
Operand	Befehlswert	Datentyp	
		MELSEC	IEC
s	Erste Adresse des Operanden, aus dem ein gespeicherter Kommentar ausgelesen werden soll.	Adresse	ANY16
d	Erste Adresse des Operanden, in dem die Kommentardaten gespeichert werden sollen.	Zeichenfolge	Array [1..8] of ANY16

Funktionsweise

Auslesen von Kommentardaten

COMRD Leseanweisung

Die COMRD-Anweisung liest Kommentardaten aus s und speichert sie als ASCII-Code in d (Array_d[1]) bis d+7 (Array_d[8]) ab.



[1] Kommentar

[2] ASCII-Code des 2. Zeichens/ ASCII-Code des 1. Zeichens

[3] ASCII-Code des 4. Zeichens/ ASCII-Code des 3. Zeichens

[4] ASCII-Code des 6. Zeichens/ ASCII-Code des 5. Zeichens

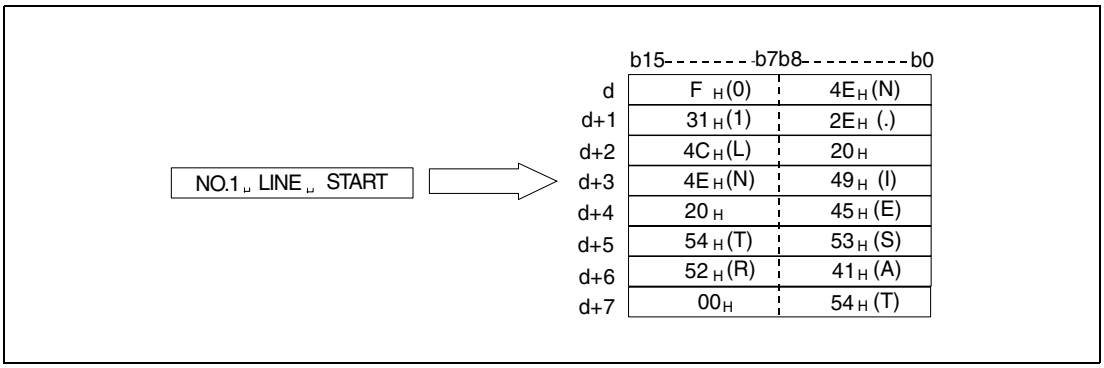
[5] ASCII-Code des 8. Zeichens/ ASCII-Code des 7. Zeichens

[6] ASCII-Code des 30. Zeichens/ ASCII-Code des 29. Zeichens

[7] ASCII-Code des 32. Zeichens/ ASCII-Code des 31. Zeichens

[8] Speichert maximal 32 Zeichen

Der in s abgelegte Kommentar mit der Zeichenfolge "NO.1 LINE START" wird in folgender Weise ab d (Array_d[1]) gespeichert:



Der Adressbereich in s muss innerhalb des Adressbereichs für Kommentare liegen.

Wird kein Kommentar in s angegeben, werden die Zeichen in Leerzeichen umgewandelt.

Die Größe eines Kommentares darf die maximale Anzahl von 32 Zeichen nicht überschreiten.

Der Inhalt des Bytes nach dem letzten Zeichen ist vom Zustand des Diagnosemerkers SM701 wie folgt abhängig:
Ist SM701 nicht gesetzt, wird eine Null gespeichert.
Ist SM701 gesetzt, gibt es keine Änderungen.

Der SM720 einer Q-CPU der Q-Serie bzw. des System Q wird für einen Zxklus gesetzt, wenn die Anweisung ausgeführt wurde. SM721 wird während der Ausführung gesetzt. Die COMRD(P)-Anweisung kann nicht angewählt werden, wenn SM721 gesetzt ist.

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Der Adressbereich in s liegt außerhalb des Kommentarbereiches (Fehlercode 4100).
- Die COMRD(P)-Anweisung wird ausgeführt während ein Kommentar im Zustand RUN geschrieben wird (Fehlercode 4100).
- Die angewählte Datei existiert nicht (Fehlercode 2410).

HINWEISE

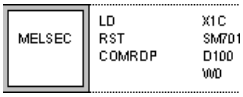
Eine CPU des System Q braucht für die Abarbeitung mehrere Programmdurchläufe, die QnA-CPU führt die Anweisung unmittelbar aus.
Bevor eine COMRD(P)/PRC-Anweisung angewählt wird, muss eine laufende COMRD(P)/PRC-Anweisung abgeschlossen sein (SM720 muss "1" gewesen sein).
Auf zwei oder mehr Kommentare kann nicht gleichzeitig zugegriffen werden.
Folgende Anweisungen können nicht gleichzeitig ausgeführt werden, weil sie den Sondermerker SM721 benutzen:

Anweisung	Während Ausführung gesetzt	Für einen Zyklus nach Ausführung gesetzt	Gesetzt nach fehlerhafter Abarbeitung
S.FREAD S.FWRITE	SM721	In Anweisung angegebenes Bit	In Anweisung angegebenes Bit + nächstes Bit
PRC COMRD		SM720	—

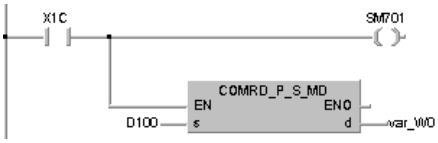
Beispiel COMRDP

Das folgende Programm speichert mit positiver Flanke von X1C den Kommentar in D100 als ASCII-Code in W0 (var_W0 Array [0]) bis W7 (var_W0 Array [7]).

MELSEC-Anweisungsliste

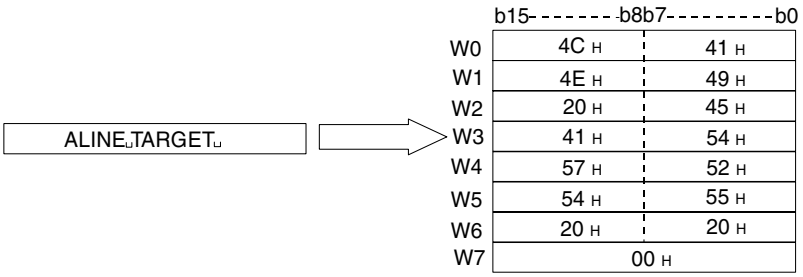


Kontaktplan



IEC-Anweisungsliste

```
LD      X1C
R       SM701
COMRD_P_S_MD  D100 , var_W0
```



HINWEIS *Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.11.8 LEN, LENP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●[2]

[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

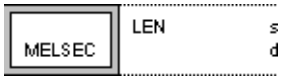
[2] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

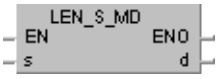
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten $	Andere		
	Bit	Wort		Bit	Wort				U		
s	—	●	●	—	—	—	—	●	—	SM0	3
d	●	●	●	●	●	●	●	—	—		

GX IEC Developer

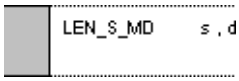
MELSEC-Anweisungsliste



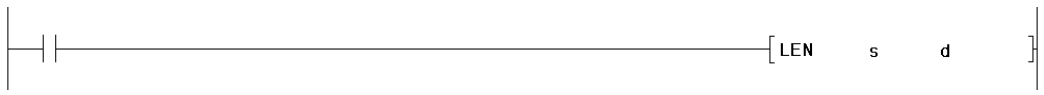
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

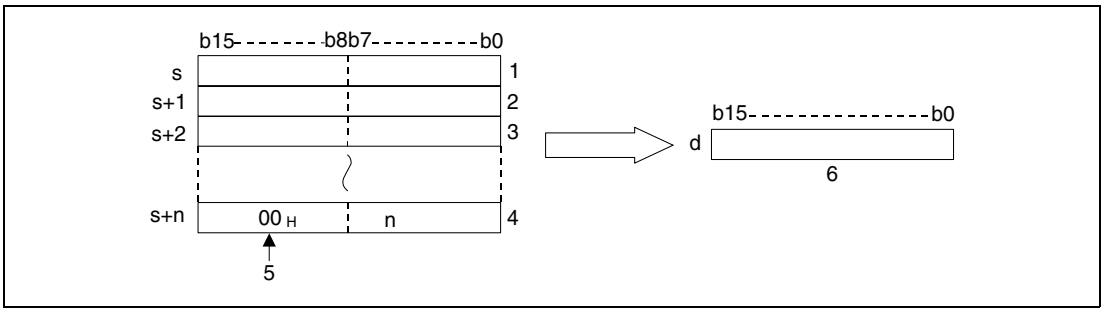
Operand	Befehlswert	Datentyp
s	Erste Adresse des Operanden, in dem die Zeichenfolge gespeichert ist, deren Länge bestimmt werden soll.	Zeichenfolge
d	Adressbereich, in dem die erkannte Zeichenfolgenlänge gespeichert wird.	BIN-16-Bit

Funktions-weise

Erfassung der Länge von Zeichenfolgen

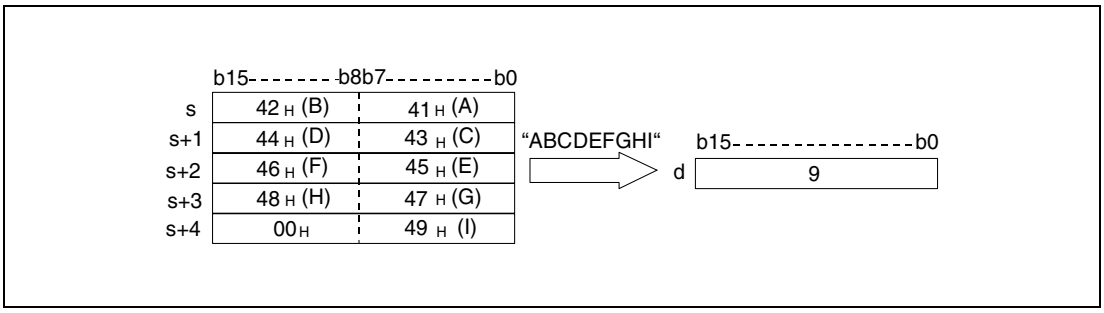
LEN Längenerfassung

Die LEN-Anweisung erfaßt die Länge einer Zeichenfolge, die in s angegeben ist, und speichert das Ergebnis in d.



[1] 2. Zeichen/ 1. Zeichen
[2] 4. Zeichen/ 3. Zeichen
[3] 6. Zeichen/ 5. Zeichen
[4] n-tes Zeichen
[5] Zeichenfolgenende
[6] Zeichenfolgenlänge

Die in s abgelegte Zeichenfolge "ABCDEFGHI" wird in folgender Weise als 9 in d gespeichert:



Die in s abgelegte Zeichenfolge wird bis zum Erreichen des Zeichen-Codes "00H" verarbeitet.

Das Ergebnis wird in d gespeichert.

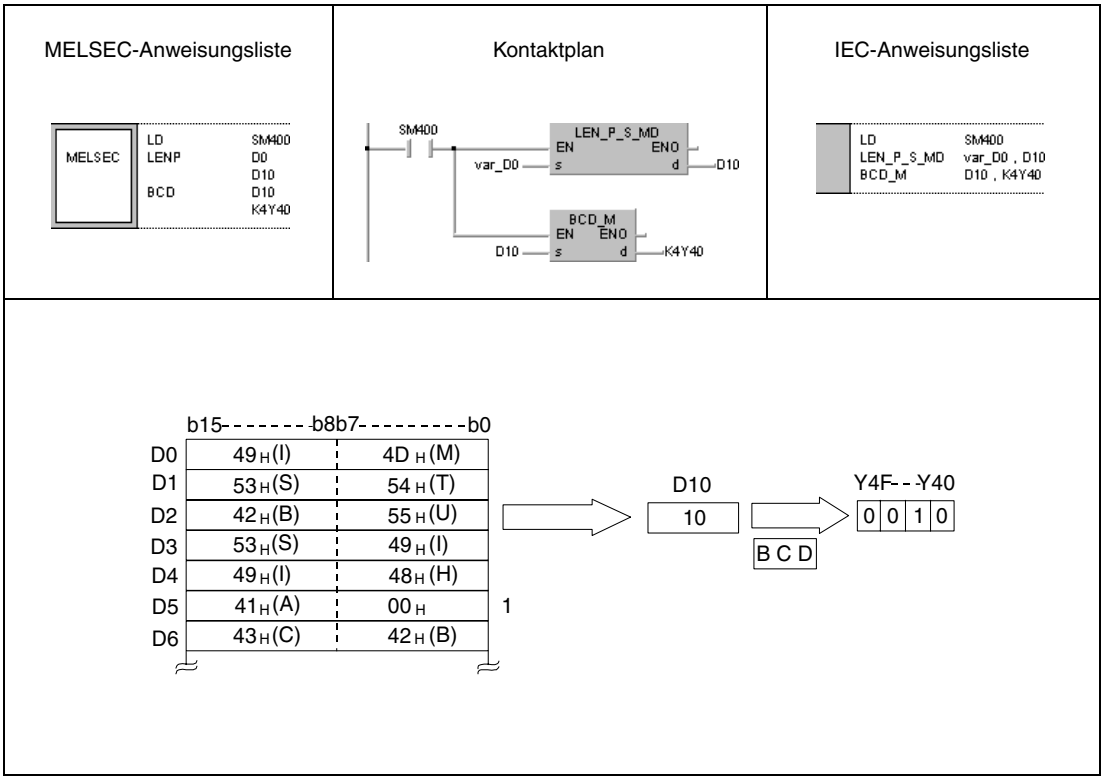
Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Im letzten Byte ab s fehlt der Zeichen-Code "00H" (Fehlercode 4101).

Beispiel LENP

Das folgende Programm bestimmt mit positiver Flanke von SM400 die Länge der Zeichenfolge ab D0 bis zum Zeichencode "00H". Das Ergebnis wird an Y40 – Y4F ausgegeben.



[1] Zeichen nach dem Zeichen-Code "00H" werden unterdrückt
(es wird nur die Länge der Zeichenfolge "MITSUBISHI" erfasst)

HINWEIS *Diese Programmbeispiele sind ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.11.9 STR, STRP, DSTR, DSTRP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●[2]

[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

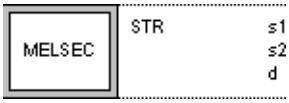
[2] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

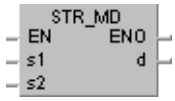
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	●	●	●	●	●	●	●	—	—	SM0	4
s2	●	●	●	●	●	●	●	●	—		
d	—	●	●	—	—	—	—	—	—		

GX IEC Developer

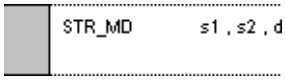
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

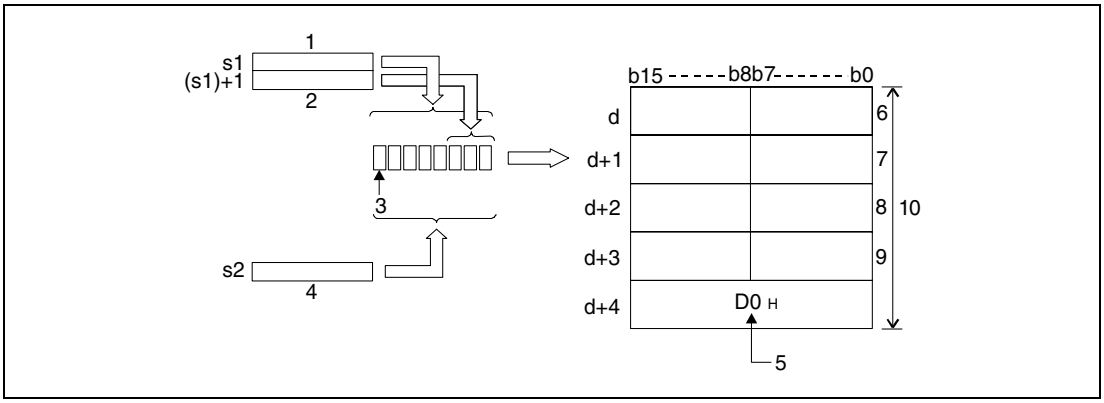
Operand	Befehlswert	Datentyp	
		MELSEC	IEC
s1	Erste Adresse des Operanden, in dem die Anzahl der Stellen des zu konvertierenden numerischen Wertes gespeichert ist.	BIN-16-Bit	ANY32
s2	Zu konvertierende Binärdaten.	BIN-16-/32-Bit	ANY16/32
d	Erste Adresse des Operanden, in dem die konvertierte Zeichenfolge gespeichert wird.	Zeichenfolge	Array [1..5]/ [1..6] of ANY16

Funktionsweise

Konvertierung von 16-/32-Bit-Binärdaten in Zeichenfolgen

STR Konvertierung von 16-Bit-Binärdaten

Die STR-Anweisung fügt dem 16-Bit-Binärdatenwert in s2 ein Dezimalkomma an der Stelle hinzu, die in s1 und (s1)+1 angegeben ist. Das Ergebnis wird in eine Zeichenfolge konvertiert und in d (Array_d[1]) bis d+4 (Array_d[5]) gespeichert.



[1] Anzahl aller Stellen

[2] Nachkommastellen

[3] Vorzeichen

[4] Binärwert

[5] Automatisch gesetztes Zeichenfolgenende

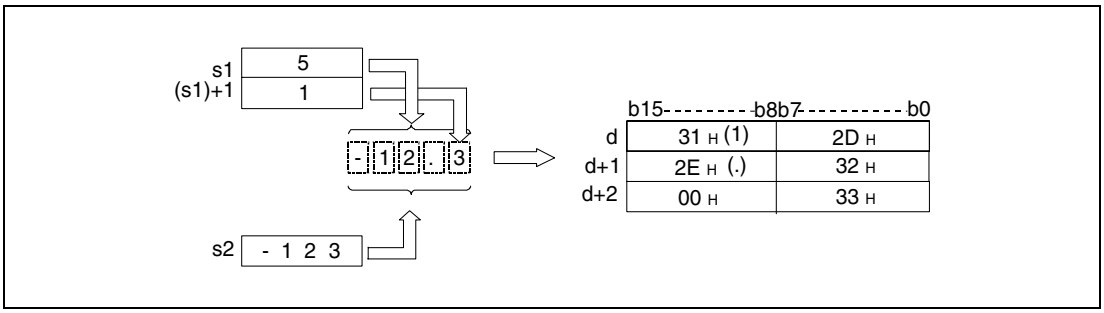
[6] ASCII-Code des Zeichens gesamte Stellenanzahl -1/ ASCII-Code des Vorzeichens

[7] ASCII-Code des Zeichens gesamte Stellenanzahl -3/ ASCII-Code des Zeichens gesamte Stellenanzahl -2

[8] ASCII-Code des Zeichens gesamte Stellenanzahl -5/ ASCII-Code des Zeichens gesamte Stellenanzahl -4

[9] ASCII-Code des Zeichens gesamte Stellenanzahl -7/ ASCII-Code des Zeichens gesamte Stellenanzahl -6

[10] Anzahl aller Stellen



Die Anzahl der Stellen, die in s1 gespeichert werden kann, liegt zwischen 2 und 8.

Die Anzahl der Nachkommastellen, die in (s1)+1 gespeichert werden kann, liegt zwischen 0 und 5 und darf nicht größer als die Anzahl der Stellen minus 3 sein.

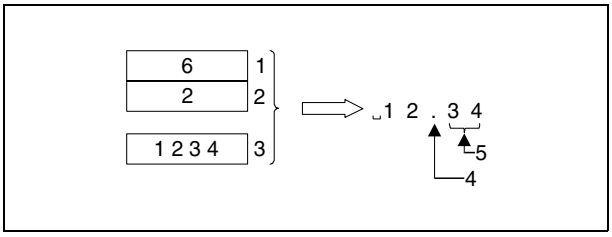
Die 16-Bit-Binärdaten in s2 dürfen in dem Bereich zwischen -32768 und 32767 liegen.

Nach der Konvertierung zu einer Zeichenfolge wird diese in d (Array_d[1]) bis d+4 (Array_d[5]) wie folgt gespeichert:

Als positives Vorzeichen der Binärdaten wird das ASCII-Zeichen "20H" (Leerzeichen) genutzt.

Als negatives Vorzeichen der Binärdaten wird das ASCII-Zeichen "2DH" (Minuszeichen) genutzt.

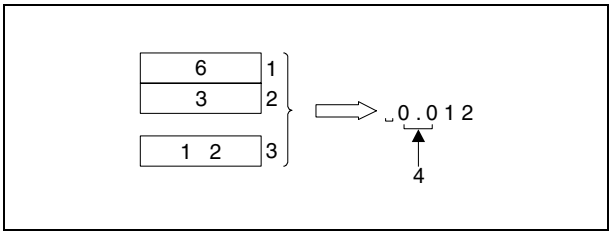
Ist die Anzahl der Nachkommastellen größer als Null, wird automatisch das Dezimalkomma "2E$_H$" (.) vor der ersten angegebenen Stelle gesetzt.



[1] Anzahl aller Stellen
[2] Anzahl der Nachkommastellen
[3] Binärwert
[4] Automatisch gesetztes Dezimalkomma
[5] Nachkommastellen

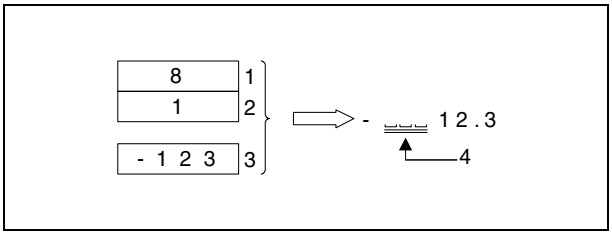
Ist die Anzahl der Nachkommastellen gleich Null, wird das Zeichen des Dezimalkommas "2D$_H$" (.) nicht gesetzt.

Ist die Anzahl der Nachkommastellen größer als die Anzahl der Stellen des Binärwertes, werden automatisch die fehlenden Stellen mit Nullen ausgefüllt, die Zahl des Binärwertes nach rechts verschoben und das Dezimalkomma an entsprechender Stelle gesetzt (0.□□□□□).



[1] Anzahl aller Stellen
[2] Anzahl der Nachkommastellen
[3] Binärwert
[4] Automatisch gesetzte Nullen und Dezimalkomma

Ist die Anzahl der Stellen inclusive Vorzeichen und Dezimalkomma größer als die Anzahl der Stellen des Binärwertes, werden die fehlenden Stellen zwischen Vorzeichen und numerischem Wert automatisch mit "20$_H$" Leerzeichen ausgefüllt.

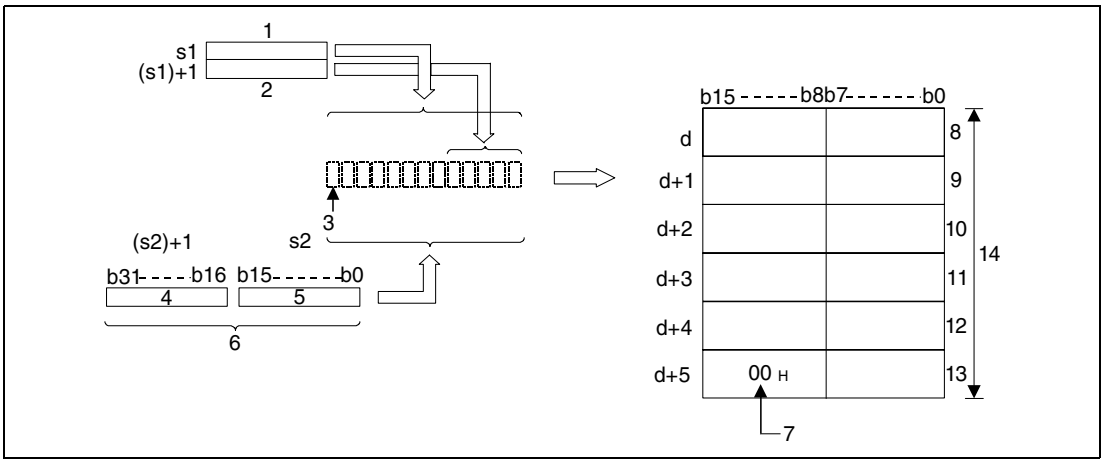


[1] Anzahl aller Stellen
[2] Anzahl der Nachkommastellen
[3] Binärwert
[4] Automatisch gesetzte Leerzeichen

Am Ende der konvertierten Zeichenfolge wird automatisch der Zeichen-Code "00$_H$" gespeichert.

DSTR Konvertierung von BIN-32-Bit-Daten

Die DSTR-Anweisung fügt dem BIN-32-Bit-Binärwert in s2 und (s2)+1 ein Dezimalkomma an der Stelle hinzu, die in s1 und (s1)+1 angegeben ist. Das Ergebnis wird in eine Zeichenfolge konvertiert und in d (Array_d[1]) bis d+5 (Array_d[6]) gespeichert.



[1] Anzahl aller Stellen

[2] Nachkommastellen

[3] Vorzeichen

[4] obere 16 Bits

[5] untere 16 Bits

[6] Binärwert

[7] Automatisch gesetztes Zeichenfolgenende

[8] ASCII-Code des Zeichens gesamte Stellenanzahl -1/ ASCII-Code des Vorzeichens

[9] ASCII-Code des Zeichens gesamte Stellenanzahl -3/ ASCII-Code des Zeichens gesamte Stellenanzahl -2

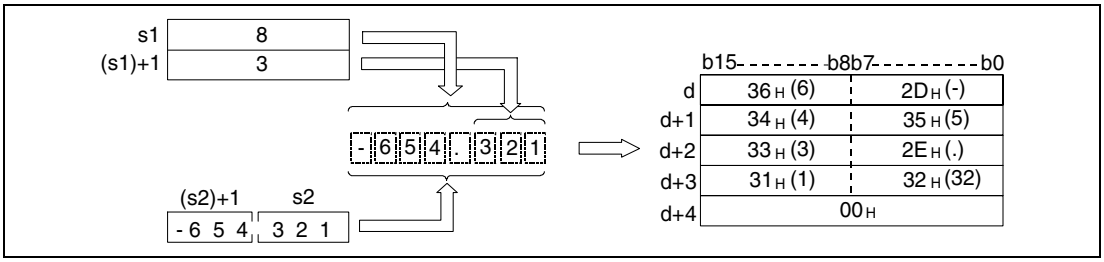
[10] ASCII-Code des Zeichens gesamte Stellenanzahl -5/ ASCII-Code des Zeichens gesamte Stellenanzahl -4

[11] ASCII-Code des Zeichens gesamte Stellenanzahl -7/ ASCII-Code des Zeichens gesamte Stellenanzahl -6

[12] ASCII-Code des Zeichens gesamte Stellenanzahl -9/ ASCII-Code des Zeichens gesamte Stellenanzahl -8

[13] Markierung des Zeichenfolgenendes/ ASCII-Code des Zeichens gesamte Stellenanzahl -10

[14] Anzahl aller Stellen



Die Anzahl der Stellen, die in s1 gespeichert werden können, liegt zwischen 2 und 13.

Die Anzahl der Nachkommastellen, die in (s1)+1 gespeichert werden können, liegt zwischen 0 und 10 und darf nicht größer als die Anzahl der Stellen minus 3 sein.

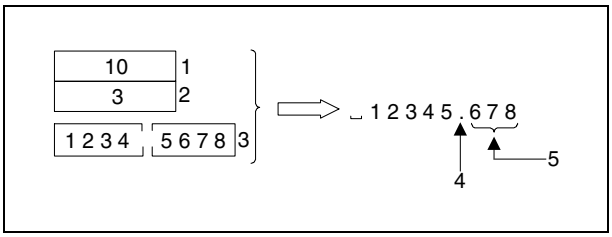
Die 32-Bit-Binärdaten, die in s2 und (s2)+1 gespeichert werden, dürfen in dem Bereich zwischen -2147483648 und 32147483647 liegen.

Nach der Konvertierung zu einer Zeichenfolge wird diese in d (Array_d[1]) bis d+5 (Array_d[6]) wie folgt gespeichert:

Als positives Vorzeichen der Binärdaten wird das ASCII-Zeichen "20H" (Leerzeichen) genutzt.

Als negatives Vorzeichen der Binärdaten wird das ASCII-Zeichen "2DH" (Minuszeichen) genutzt.

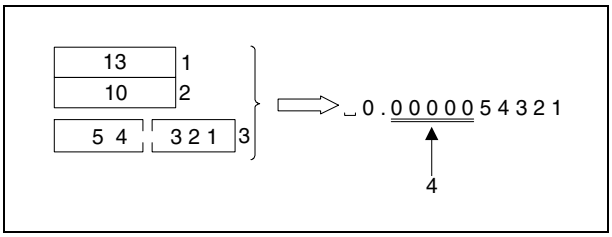
Ist die Anzahl der Nachkommastellen größer als Null, wird automatisch das Dezimalkomma "2EH" (.) vor der ersten angegebenen Stelle gesetzt.



[1] Anzahl aller Stellen
[2] Nachkommastellen
[3] Binärwert
[4] Automatisch gesetztes Dezimalkomma
[5] Anzahl der Nachkommastellen

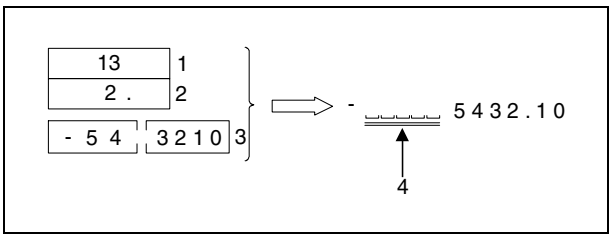
Ist die Anzahl der Nachkommastellen gleich Null, wird das Zeichen des Dezimalkommas "2DH" (.) nicht gesetzt.

Ist die Anzahl der Nachkommastellen größer als die Anzahl der Stellen des Binärwertes, werden automatisch die fehlenden Stellen mit Nullen ausgefüllt, die Zahl des Binärwertes nach rechts verschoben und das Dezimalkomma an entsprechender Stelle gesetzt (0.□□□□).



[1] Anzahl aller Stellen
[2] Nachkommastellen
[3] Binärwert
[4] Automatisch gesetzte Nullen und Dezimalkomma

Ist die Anzahl der Stellen inclusive Vorzeichen und Dezimalkomma größer als die Anzahl der Stellen des Binärwertes, werden die fehlenden Stellen zwischen Vorzeichen und numerischem Wert automatisch mit Leerzeichen "20H" ausgefüllt.



[1] Anzahl aller Stellen
[2] Anzahl der Nachkommastellen
[3] Binärwert
[4] Automatisch gesetzte Leerzeichen

Am Ende der konvertierten Zeichenfolge wird automatisch der Zeichen-Code "00H" gespeichert.

Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die in s1 gespeicherte Anzahl aller Stellen liegt außerhalb der unten angegebenen Wertebereiche (Fehlercode 4100).
  Wertebereich für die STR-Anweisung.....2 bis 8
  Wertebereich für die DSTR-Anweisung.....2 bis 13
- Die in (s1)+1 gespeicherte Anzahl der Nachkommastellen liegt außerhalb der unten angegebenen Wertebereiche (Fehlercode 4100).
  Wertebereich für die STR-Anweisung.....0 bis 5
  Wertebereich für die DSTR-Anweisung.....0 bis 10
- Die in s1 und (s1)+1 gespeicherten Werte entsprechen nicht der folgenden Beziehung: Die Anzahl aller Stellen minus 3 ist größer oder gleich der Anzahl der Nachkommastellen (Fehlercode 4100).
- Die Anzahl der Stellen, die in s1 und (s1)+1 gespeichert sind, ist kleiner als die Anzahl der Stellen des Binärwertes in s2 und (s2)+1 (Fehlercode 4100).
- Der Speicherbereich, in dem die ab d (Array_d[1]) angegebene Zeichenfolge gespeichert wird, liegt außerhalb des für die Speicherung vorgesehenen Bereiches (Fehlercode 4100).

Beispiel 1 STRP

Das folgende Programm konvertiert mit positiver Flanke von X0 den in D10 angegebenen Binärwert entsprechend der in D0 und D1 angegebenen Anzahl von Stellen und speichert das Ergebnis in dem Bereich von D20 (var_D20 Array [1]) bis D23 (var_D20 Array [4]).

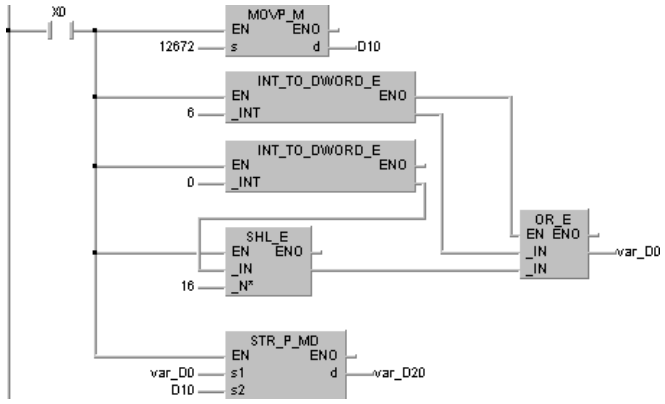
MELSEC-Anweisungsliste

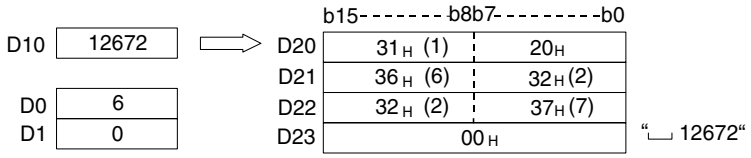
```
MELSEC  LD      X0
        MOVP    K12672
                D10
        MOVP    K6
                D0
        MOVP    K0
                D1
        STRP    D0
                D10
                D20
```

IEC-Anweisungsliste

```
LD              X0
MOVP_M          12672 , D10
INT_TO_DWORD_E  6 , var_Merker
INT_TO_DWORD_E  0 , var_D0
SHL_E           var_D0 , K16 , var_D0
LD              var_Merker
OR              var_D0
ST              var_D0
LD              X0
STR_P_MD        var_D0 , D10 , var_D20
```

Kontaktplan





Beispiel 2 DSTRP

Das folgende Programm konvertiert mit positiver Flanke von X0 den in D11 und D12 angegebenen Binärwert entsprechend der in D0 und D1 angegebenen Anzahl von Stellen und speichert das Ergebnis in dem Bereich von D30 (var_D20 Array [1]) bis D36 (var_D20 Array [7]) ab.

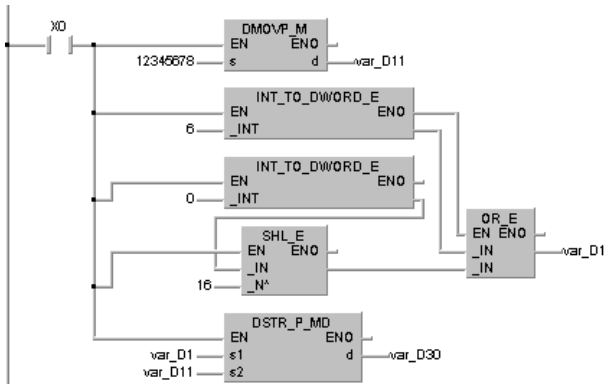
MELSEC-Anweisungsliste

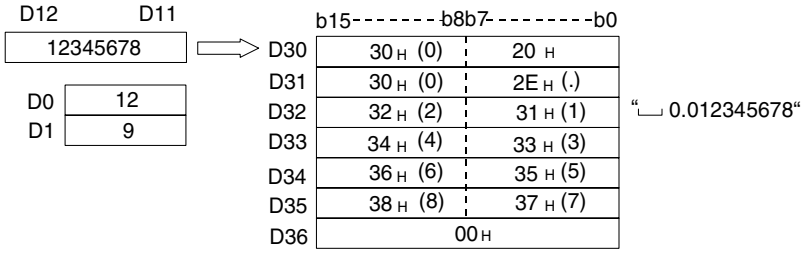
```
MELSEC   LD      X0
         DMOVP   K12345678
                 D11
         MOVP    K6
                 D2
         MOVP    K0
                 D3
         DSTRP   D2
                 D11
                 D30
```

IEC-Anweisungsliste

```
LD              X0
DMOVP_M         12345678 , var_D11
INT_TO_DWORD_E  6 , var_Merker
INT_TO_DWORD_E  0 , var_D1
SHL_E           var_D1 , K16 , var_D1
LD              var_Merker
OR              var_D1
ST              var_D1
LD              X0
DSTR_P_MD       var_D1 , var_D11, var_D30
```

Kontaktplan





HINWEIS *Diese Programmbeispiele sind ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.11.10 VAL, VALP, DVAL, DVALP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●[2]

[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

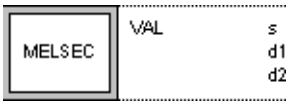
[2] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

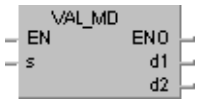
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten $	Andere		
	Bit	Wort		Bit	Wort						
	—	●	●	—	—	—	—	●	—	SM0	4
	●	●	●	—	—	—	—	—	—		
	●	●	●	●	●	●	●	—	—		

GX IEC Developer

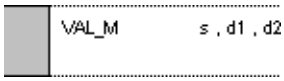
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

Operand	Befehlswert	Datentyp	
		MELSEC	IEC
s	Erste Adresse des Operanden, in dem die Zeichenfolge der zu konvertierenden Binärdaten gespeichert ist.	Zeichenfolge	Array [1..5]/ [1..7] of ANY16
d1	Erste Adresse des Operanden, in dem die Anzahl der Stellen der Binärdaten nach der Konvertierung gespeichert ist.	BIN-16-Bit	ANY32
d2	Startadresse des Operanden, in dem die konvertierten Binärdaten gespeichert werden.	BIN-16-/32-Bit	ANY16/32

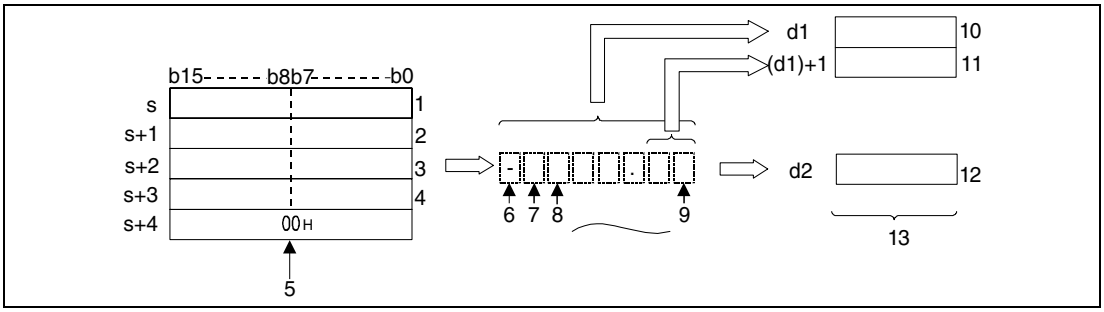
Funktions-weise

Konvertierung von Zeichenfolgen in 16-/32-Bit-Binärdaten

VAL Konvertierung in 16-Bit-Binärdaten

Die VAL-Anweisung konvertiert die in s (Array_s[1]) bis s+4 (Array_s[5]) gespeicherten Zeichenfolgen in 16-Bit-Binärdaten. Die Anzahl der Stellen und der Binärwert werden in d1, (d1)+1 und d2 gespeichert.

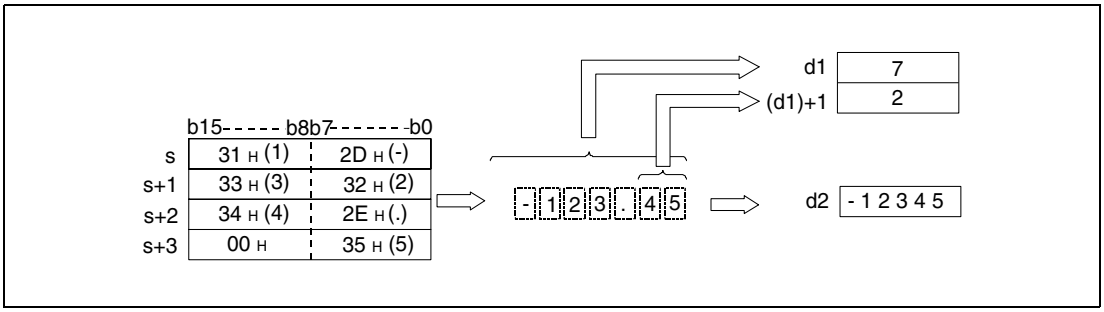
Zur Konvertierung in das BIN-16-Bit-Datenformat werden alle Daten in dem Bereich s (Array_s[1]) bis s+4 (Array_s[5]) bis zum Zeichencode "00H" als Zeichenfolge erkannt.



[1] ASCII-Code für das 1. Zeichen/ ASCII-Code für das Vorzeichen
[2] ASCII-Code für das 3. Zeichen/ ASCII-Code für das 2. Zeichen
[3] ASCII-Code für das 5. Zeichen/ ASCII-Code für das 4. Zeichen
[4] ASCII-Code für das 7. Zeichen/ ASCII-Code für das 6. Zeichen
[5] Kennzeichnet das Zeichenfolgenende
[6] Vorzeichen
[7] 1. Zeichen
[8] 2. Zeichen
[9] 7. Zeichen
[10] Anzahl aller Stellen
[11] Anzahl der Nachkommastellen
[12] Integerwert, das Dezimalkomma wird bei der Verarbeitung nicht berücksichtigt
[13] 16-Bit-Binärdaten

Die Zeichenfolge"-123.45" in s (Array_s[1]) bis s+4 (Array_s[5]) wird konvertiert.

Das Ergebnis wird in folgender Weise in d1, (d1)+1 und d2 gespeichert.



Die Anzahl aller Zeichen, die in s (Array_s[1]) bis s+4 (Array_s[5]) gespeichert sind, kann in dem Bereich zwischen 2 und 8 liegen.

Die Anzahl der Zeichen, die als Nachkommastellen in Frage kommen und in s (Array_s[1]) bis s+4 (Array_s[5]) abgelegt sind, kann zwischen 0 und 5 liegen. Generell darf die Anzahl der Nachkommastellen nicht größer sein als die Anzahl aller Stellen minus 3.

Der numerische Wert einer Zeichenfolge, die konvertiert werden soll, darf bei ignoriertem Dezimalkomma zwischen -32768 und 32767 liegen. Der numerische Wert der ASCII-Zeichenfolge darf ohne Berücksichtigung des Vorzeichens und des Dezimalkommas nur einen Wert zwischen "30H" und "39H" annehmen.

Als positives Vorzeichen wird das ASCII-Zeichen "20H" (Leerzeichen) genutzt.

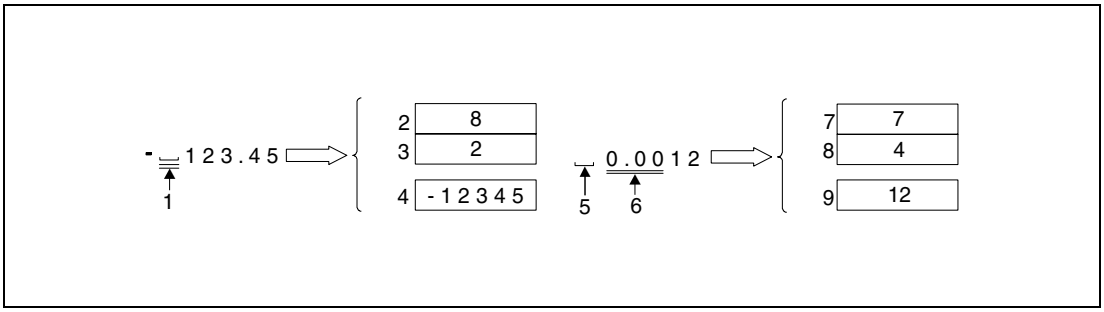
Als negatives Vorzeichen wird das ASCII-Zeichen "2DH" (Minuszeichen) genutzt.

Das ASCII-Zeichen "2EH" wird als Dezimalkomma gesetzt.

Die Anzahl aller Stellen, die in d1, (d1)+1 und d2 gespeichert sind, beinhaltet alle Zeichen, die den numerischen Wert darstellen als auch das Vorzeichen d1 und die Nachkommastellen (d1)+1.

In den Binärdaten, die in d2 nach der Konvertierung gespeichert sind, wird das Dezimalkomma ignoriert.

Wenn zwischen Vorzeichen und dem ersten numerischen Wert die Zeichen "20H" (Leerzeichen) oder "30H" (Null) stehen, werden diese während der Konvertierung ignoriert.



1 Diese Zeichen werden bei der Verarbeitung nicht berücksichtigt

2 Anzahl aller Stellen

3 Anzahl der Nachkommastellen

4 Binärwert

5 Vorzeichen

6 Diese Zeichen werden bei der Verarbeitung nicht berücksichtigt

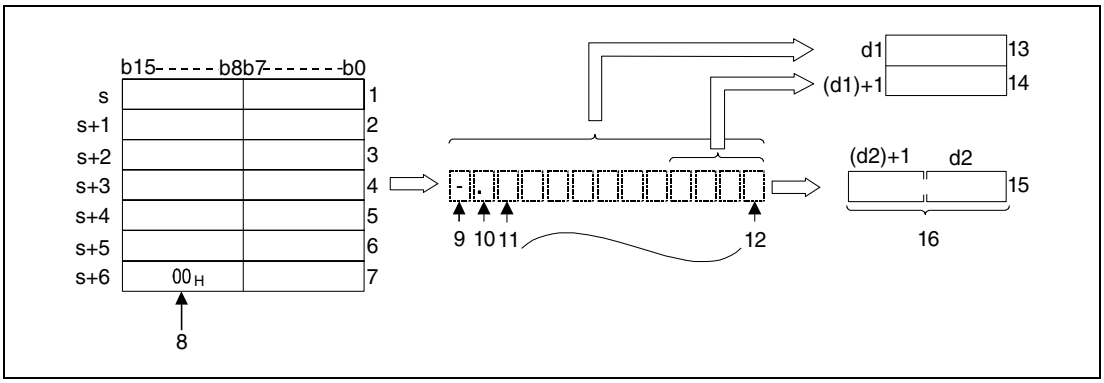
7 Anzahl aller Stellen

8 Anzahl der Nachkommastellen

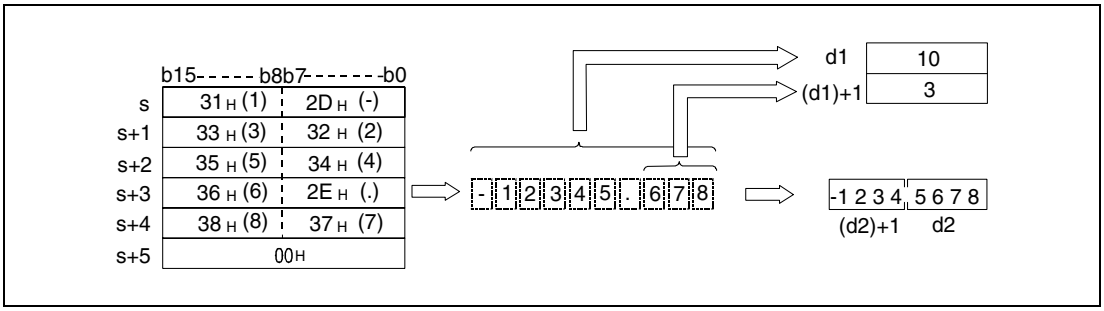
9 Binärwert

DVAL Konvertierung in 32-Bit-Binärdaten

Die DVAL-Anweisung konvertiert die in s (Array_s[1]) bis s+6 (Array_s[7]) gespeicherten Zeichenfolgen in 32-Bit-Binärdaten. Die Anzahl der Stellen und der Binärwert werden in d1, (d1)+1, d2 und (d2)+1 gespeichert.
Zur Konvertierung in das 32-Bit-Binärformat werden alle Daten in dem Bereich s (Array_s[1]) bis s+6 (Array_s[7]) bis zu dem Zeichencode "00H" als Zeichenfolge erkannt.



[1] ASCII-Code für das 1. Zeichen/ ASCII-Code für das Vorzeichen
[2] ASCII-Code für das 3. Zeichen/ ASCII-Code für das 2. Zeichen
[3] ASCII-Code für das 5. Zeichen/ ASCII-Code für das 4. Zeichen
[4] ASCII-Code für das 7. Zeichen/ ASCII-Code für das 6. Zeichen
[5] ASCII-Code für das 9. Zeichen/ ASCII-Code für das 8. Zeichen
[6] ASCII-Code für das 11. Zeichen/ ASCII-Code für das 10. Zeichen
[7] ASCII-Code für das Null Zeichen/ ASCII-Code für das 12. Zeichen
[8] Kennzeichnet das Zeichenfolgenende
[9] Vorzeichen
[10] 1. Zeichen
[11] 2. Zeichen
[12] 12. Zeichen
[13] Anzahl aller Stellen
[14] Anzahl der Nachkommastellen
[15] Integerwert, das Dezimalkomma wird bei der Verarbeitung nicht berücksichtigt
[16] 32-Bit-Binärdatenwert



Die Anzahl aller Zeichen, die in s (Array_s[1]) bis s+6 (Array_s[7]) gespeichert sind, kann in dem Bereich zwischen 2 und 13 liegen.

Die Anzahl der Zeichen, die als Nachkommastellen in Frage kommen und in s (Array_s[1]) bis s+6 (Array_s[7]) abgelegt sind, kann zwischen 0 und 10 liegen. Generell darf die Anzahl der Nachkommastellen nicht größer sein als die Anzahl aller Stellen minus 3.

Der numerische Wert einer Zeichenfolge, die konvertiert werden soll, kann bei ignoriertem Dezimalkomma zwischen -2147483648 und 2147483647 liegen. Der numerische Wert der ASCII-Zeichenfolge kann ohne Berücksichtigung des Vorzeichens und des Dezimalkommas nur einen Wert zwischen "30H" und "39H" annehmen.

Als positives Vorzeichen wird das ASCII-Zeichen "20H" (Leerzeichen) genutzt.

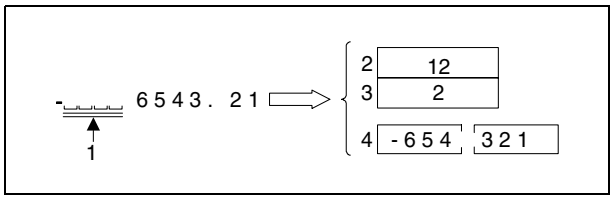
Als negatives Vorzeichen wird das ASCII-Zeichen "2DH" (Minuszeichen) genutzt.

Das ASCII-Zeichen "2EH" wird als Dezimalkomma gesetzt.

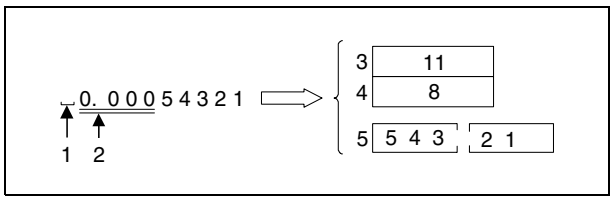
Die Anzahl aller Stellen, die in d1, (d1)+1, d2 und (d2)+1 gespeichert sind, beinhaltet alle Zeichen, die den numerischen Wert darstellen als auch das Vorzeichen d1 und die Nachkommastellen (d1)+1.

In den Binärdaten, die in d2 und (d2)+1 nach der Konvertierung gespeichert worden sind, wird das Dezimalkomma ignoriert.

Wenn zwischen Vorzeichen und dem ersten numerischen Wert die Zeichen "20H" (Leerzeichen) oder "30H" Null stehen, werden diese während der Konvertierung ignoriert.



[1] Diese Zeichen werden bei der Verarbeitung nicht berücksichtigt
[2] Anzahl aller Stellen
[3] Anzahl der Nachkommastellen
[4] 32-Bit-Binärwert



[1] Vorzeichen
[2] Diese Zeichen werden bei der Verarbeitung nicht berücksichtigt
[3] Anzahl aller Stellen
[4] Anzahl der Nachkommastellen
[5] 32-Bit-Binärwert

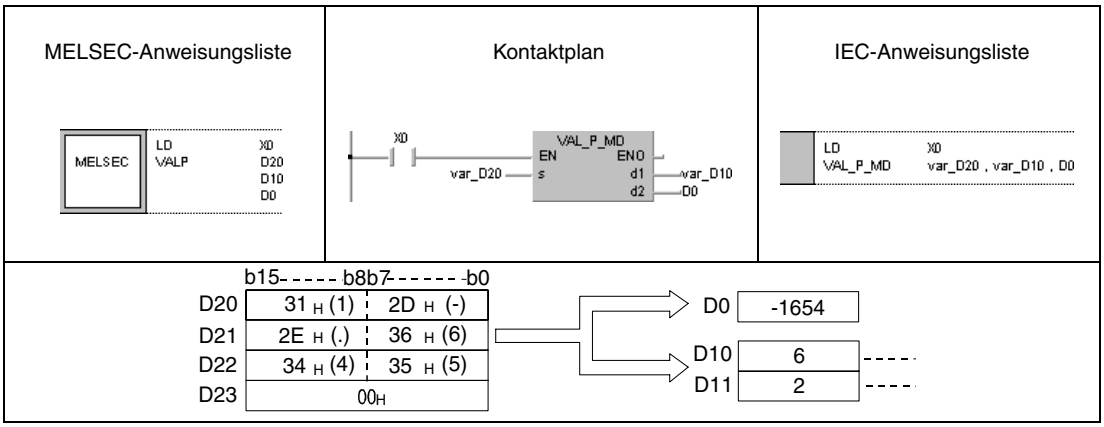
Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die ab s (Array_s[0]) gespeicherte Anzahl aller Stellen liegt außerhalb der Wertebereiche 2 bis 8 (VAL) bzw. 2 bis13 (DVAL) (Fehlercode 4101).
- Die in (d1)+1 gespeicherte Anzahl der Nachkommastellen liegt außerhalb des Wertebereiches 0 bis 5 (VAL) bzw. 0 bis 10 (DVAL) (Fehlercode 4100).
- Die in d1 und (d1)+1 gespeicherte Anzahl aller Stellen minus 3 ist größer oder gleich der Anzahl der Nachkommastellen (Fehlercode 4100).
- Es wurden andere ASCII-Zeichen als "20H" oder "2DH" für das Vorzeichen verwendet (Fehlercode 4100).
- Es wurden andere ASCII-Zeichen als "30H", "39H" oder "2EH" in einer Zahl verwendet (Fehlercode 4100).
- Es ist mehr als ein Dezimalkomma in der Zahl enthalten (Fehlercode 4100).
- Der Binärwert liegt nach der Konvertierung außerhalb der Wertebereiche -32768 bis 32767 (VAL) bzw. -2147483648 bis 2147483647 (DVAL) (Fehlercode 4100).
- Das ASCII-Zeichen "00H" ist an der falschen Stelle gesetzt worden (Fehlercode 4100).

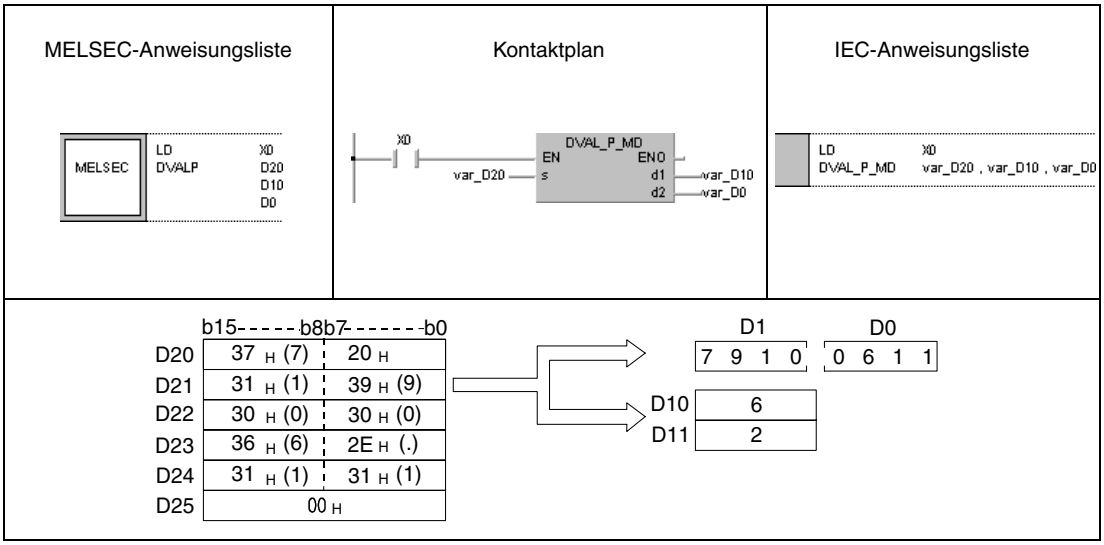
Beispiel 1 VALP

Das folgende Programm konvertiert mit positiver Flanke von X0 die im Bereich D20 (var_ D20 Array [1]) bis D23 (var_ D20 Array [4]) gespeicherte Zeichenfolge in einen Integerwert, konvertiert diesen Wert in einen 16-Bit-Binärwert und speichert ihn in D0.



Beispiel 2 DVALP

Das folgende Programm konvertiert mit positiver Flanke von X0 die im Bereich D20 (var_ D20 Array [1]) bis D24 (var_ D20 Array [5]) gespeicherte Zeichenfolge in einen Integerwert, konvertiert diesen Wert in einen 32-Bit-Binärwert und speichert ihn in D0 und D1.



HINWEIS *Diese Programmbeispiele sind ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.11.11 ESTR, ESTRP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●[1]

[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

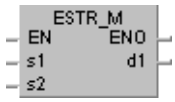
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□N□		Sondermodule U□NG□	Index-Register Zn	Konstanten E	Andere		
	Bit	Wort		Bit	Wort						
s1	—	●	●	—	●	●	—	●	—	SM0	4
s2	—	●	●	—	—	—	—	—	—		
d	—	●	●	—	—	—	—	—	—		

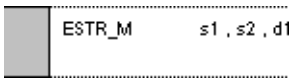
GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

Operand	Befehlswert	Datentyp	
		MELSEC	IEC
s1	Zu konvertierende Fließkommadaten oder die Startadresse des Operanden, in dem diese Daten gespeichert sind.	reelle Zahl	reelle Zahl
s2	Erste Adresse der Operanden, in dem das Wiedergabeformat der zu konvertierenden numerischen Daten gespeichert ist.	BIN-16-Bit	Array [1..3] of ANY16
d1	Erste Adresse des Operanden, in dem die konvertierten Daten gespeichert werden.	Zeichenfolge	Zeichenfolge

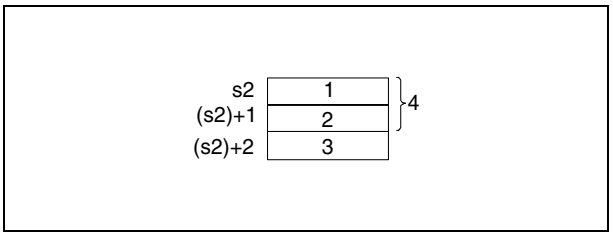
Funktions-weise

Konvertierung von Gleitkommazahlen in Zeichenfolgen

ESTR Konvertierung von Gleitkommazahlen

Die Anweisung konvertiert die Gleitkommazahlen (reellen Zahlen) in s1 und (s1)+1 in eine Zeichenfolge. Das Format dieser Zeichenfolge wird in s2 (Array_s2[1]) bis (s2)+2 (Array_s2[3]) angegeben. Das Ergebnis wird ab d1 gespeichert.

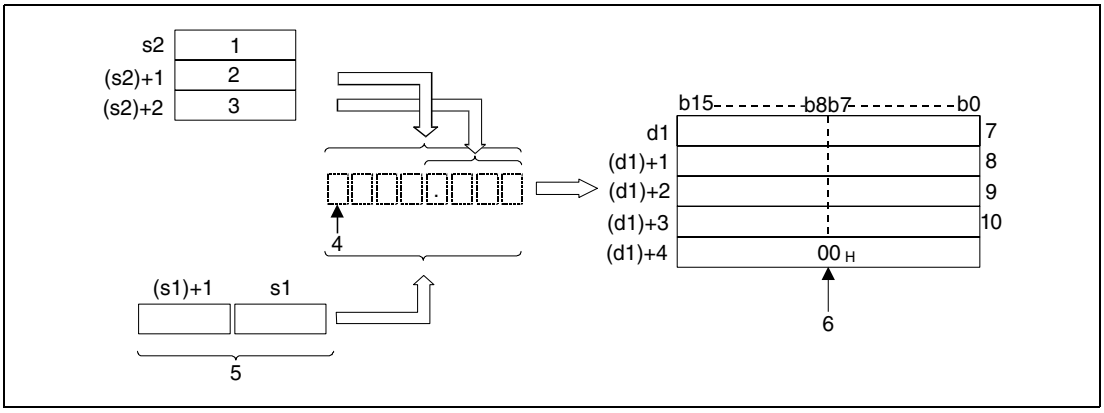
Das Darstellungsformat der Daten nach der Konvertierung hängt von dem Format in s2 (Array_s2[1]) bis (s2)+2 (Array_s2[3]) ab.



[1] Darstellungsformat (Dezimaldarstellung "0"/ Exponentialdarstellung "1")

[2] Anzahl aller Stellen

[3] Anzahl der Nachkommastellen



[1] Darstellungsformat (Dezimaldarstellung "0"/ Exponentialdarstellung "1")

[2] Anzahl aller Stellen

[3] Anzahl der Nachkommastellen

[4] Vorzeichen

[5] Gleitkommazahl (reelle Zahl)

[6] Automatisch gesetztes Zeichenfolgenende

[7] ASCII-Code des Zeichens gesamte Stellenanzahl -1/ ASCII-Code des Vorzeichens

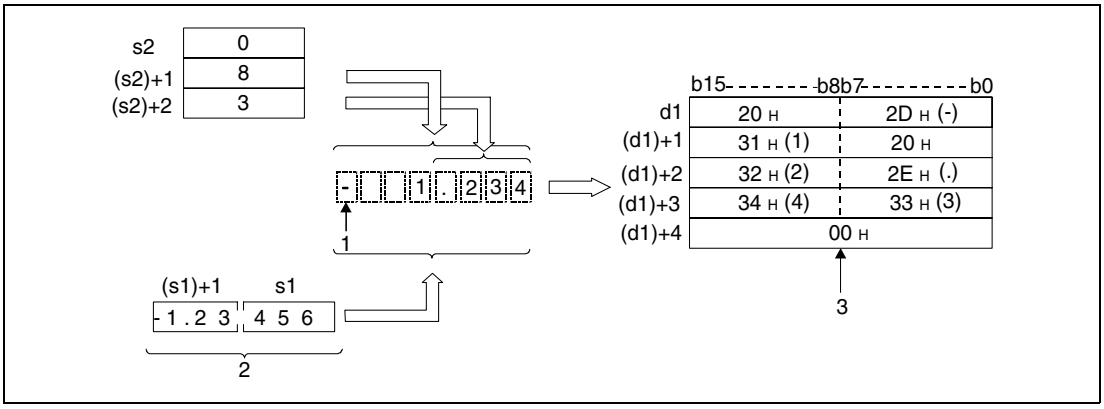
[8] ASCII-Code des Zeichens gesamte Stellenanzahl -3/ ASCII-Code des Zeichens gesamte Stellenanzahl -2

[9] ASCII-Code des Zeichens gesamte Stellenanzahl -5/ ASCII-Code des Zeichens gesamte Stellenanzahl -4

[10] ASCII-Code des Zeichens gesamte Stellenanzahl -7/ ASCII-Code des Zeichens gesamte Stellenanzahl -6

Dezimaldarstellung

Die reelle Zahl -1.23456 wird zu einer Zeichenfolge mit insgesamt 8 Stellen (davon 3 Nachkommastellen) konvertiert. Das Ergebnis wird ab d1 gespeichert.



[1] Vorzeichen

[2] Gleitkommazahl (reelle Zahl)

[3] Automatisch gesetztes Zeichenfolgenende

Die Anzahl aller Stellen der zu konvertierenden Zahl in (s2)+1 (Array_s2[2]) stellt sich wie folgt dar:

Wenn die Anzahl der Nachkommastellen Null beträgt, ist die Anzahl aller Stellen >= 2.

Wenn die Anzahl der Nachkommastellen einen anderen Wert annimmt, ist die Anzahl aller Stellen 3 plus der Anzahl der Nachkommastellen.

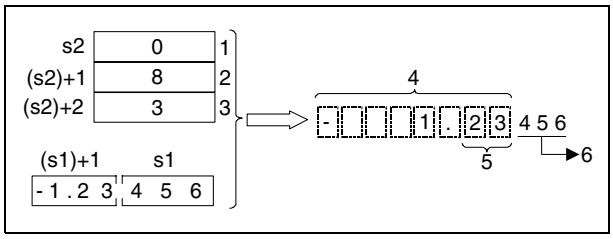
Die Anzahl der Nachkommastellen, die angegeben werden, muss zwischen 0 und 7 liegen. Generell gilt, dass die Anzahl der Nachkommastellen kleiner oder gleich der Anzahl aller Stellen minus 3 betragen muss.

Nach der Konvertierung wird die Zeichenfolge ab d1 in folgender Weise gespeichert:

Als positives Vorzeichen der Gleitkommazahl wird das ASCII-Zeichen "20H" (Leerzeichen) genutzt.

Als negatives Vorzeichen der Gleitkommazahl wird das ASCII-Zeichen "2DH" (Minuszeichen) genutzt.

In den Fällen, in denen die tatsächliche Anzahl der Nachkommastellen der Gleitkommazahl größer ist als die eingegebene Anzahl der Nachkommastellen der konvertierten Zahl, werden die überzähligen Stellen abgeschnitten.



[1] Darstellungsformat (Dezimaldarstellung) (0)

[2] Anzahl aller Stellen

[3] Anzahl der Nachkommastellen

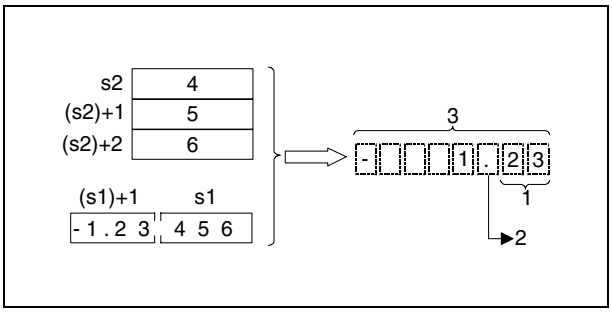
[4] Anzahl aller Stellen

[5] Anzahl der Nachkommastellen

[6] Diese Ziffern werden abgeschnitten

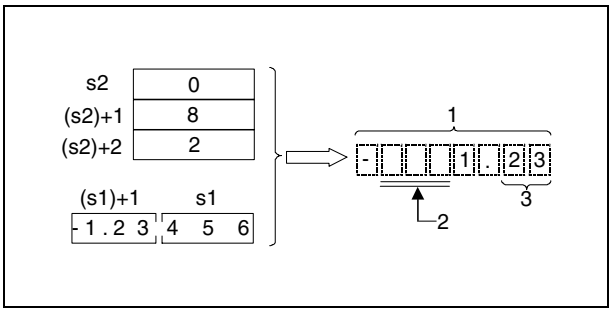
Wenn die Anzahl der Nachkommastellen auf einen andern Wert als Null gesetzt wird, wird das Dezimalkomma "2EH" (.) automatisch an der angegebenen Stelle gesetzt.

Wenn die Anzahl der Nachkommastellen auf Null gesetzt wird, wird das Dezimalkomma "2EH" (.) nicht gesetzt.



[1] Anzahl der Nachkommastellen
[2] Dezimalkomma wird automatisch gesetzt und gespeichert
[3] Anzahl aller Stellen

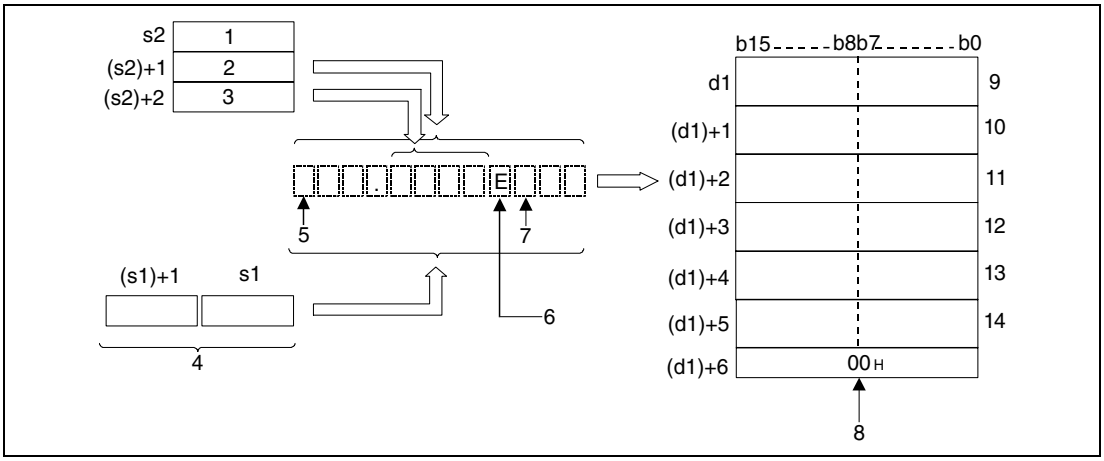
Wenn die Anzahl aller darzustellenden Stellen ohne Vorzeichen kleiner ist als die Anzahl der Komma und Nachkommastellen, werden die Stellen zwischen Vorzeichen und erster darzustellender Stelle mit dem Zeichencode "20H" (Leerzeichen) ausgefüllt.



[1] Anzahl aller Stellen
[2] Leerzeichen "20H" werden gespeichert
[3] Anzahl der Nachkommastellen

Der Zeichencode "00H" wird automatisch am Ende der Zeichenfolge gespeichert.

Exponentialdarstellung



[1] Darstellungsformat (Exponentialdarstellung) (1)

[2] Anzahl aller Stellen

[3] Anzahl der Nachkommastellen

[4] Gleitkommazahl (reelle Zahl)

[5] Vorzeichen des Integerwertes

[6] Das "E" wird automatisch gesetzt

[7] Vorzeichen des Exponenten

[8] Automatisch gesetztes Zeichenfolgenende

[9] ASCII-Code des Zeichens gesamte Stellenanzahl -1/ ASCII-Code des Vorzeichens

[10] ASCII-Code des Zeichens gesamte Stellenanzahl -3/ ASCII-Code des Zeichens gesamte Stellenanzahl -2

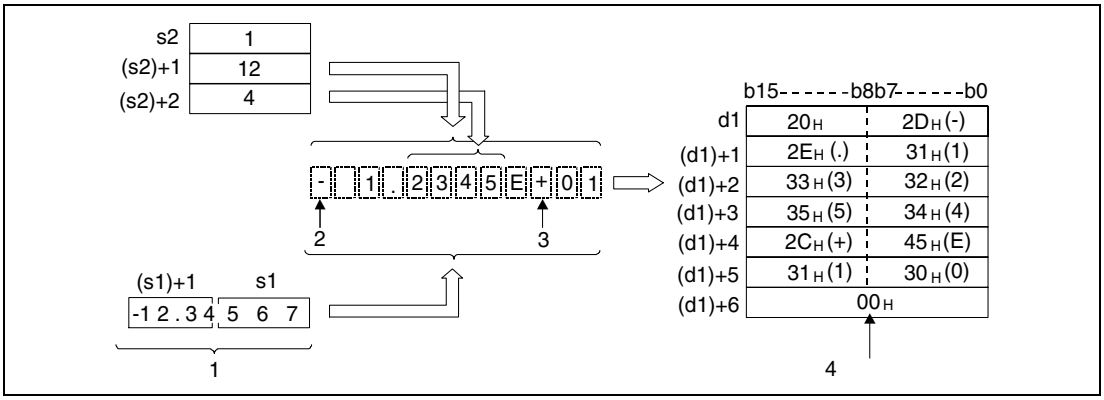
[11] ASCII-Code des Zeichens gesamte Stellenanzahl -5/ ASCII-Code des Zeichens gesamte Stellenanzahl -4

[12] ASCII-Code des Zeichens gesamte Stellenanzahl -7/ ASCII-Code des Zeichens gesamte Stellenanzahl -6

[13] Vorzeichen des Exponenten/ 45H (E)

[14] ASCII-Code des Zeichens gesamte Stellenanzahl -11 (Exponent)/ ASCII-Code des Zeichens gesamte Stellenanzahl -10 (Exponent)

Die reelle Zahl -12.34567 soll in exponentieller Schreibweise dargestellt werden. Die Anzahl aller Stellen beträgt 12. Davon sollen 4 Nachkommastellen im Dezimalteil dargestellt werden. Das Ergebnis wird ab d1 gespeichert.



[1] Gleitkommazahl (reelle Zahl)

[2] Vorzeichen des Integerwertes

[3] Vorzeichen des Exponenten

[4] Automatisch gesetztes Zeichenfolgenende

Die Anzahl aller Stellen der zu konvertierenden Zahl in (s2)+1 (Array_s2[2]) stellt sich wie folgt dar:

Wenn die Anzahl der Nachkommastellen Null beträgt, ist die Anzahl aller Stellen >= 2.

Wenn die Anzahl der Nachkommastellen einen anderen Wert annimmt, ist die Anzahl aller Stellen 7 plus der Anzahl der Nachkommastellen.

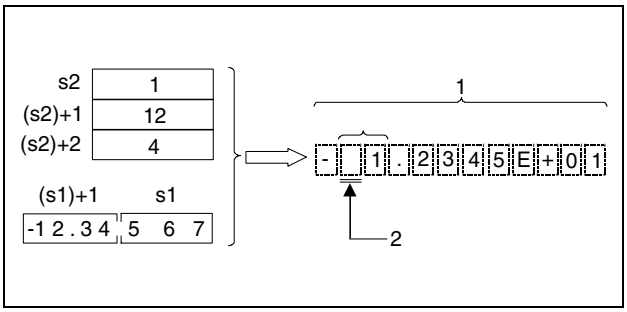
Die Anzahl der Nachkommastellen, die angegeben werden, kann zwischen 0 und 7 liegen. Generell gilt, dass die Anzahl der Nachkommastellen kleiner oder gleich der Anzahl aller Stellen minus 7 betragen muss.

Nach der Konvertierung wird die Zeichenfolge ab d1 in folgender Weise gespeichert:

Als positives Vorzeichen der Gleitkommazahl des Dezimalteils wird das ASCII-Zeichen "20H" (Leerzeichen) genutzt.

Als negatives Vorzeichen der Gleitkommazahl des Dezimalteils wird das ASCII-Zeichen "2DH" (Minuszeichen) genutzt.

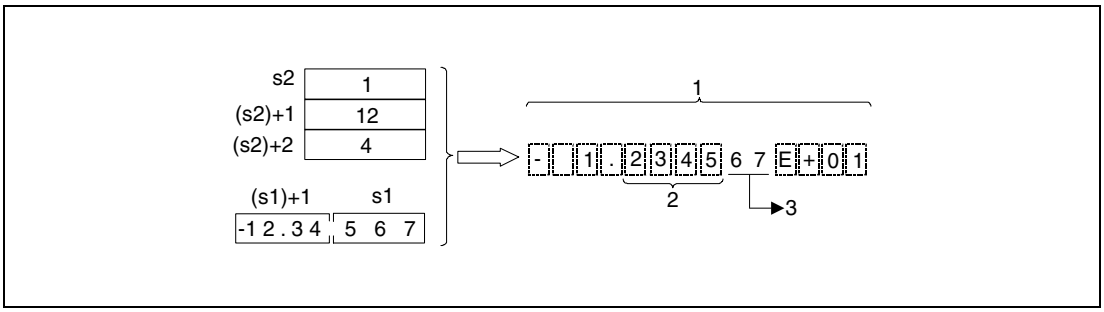
Der Integerbereich ist auf 2 Stellen festgelegt. Wenn der Integerbereich nur eine Stelle hat, wird zwischen dem Vorzeichen und der Integerstelle ein Leerzeichen im ASCII-Code gesetzt und gespeichert.



[1] Anzahl aller Stellen (12)

[2] Wird zum Leerzeichen

Ist die Gleitkommazahl des Dezimalbereichs länger als der vorgesehene Speicherbereich, werden die Stellen, die nicht gespeichert werden können, abgeschnitten.

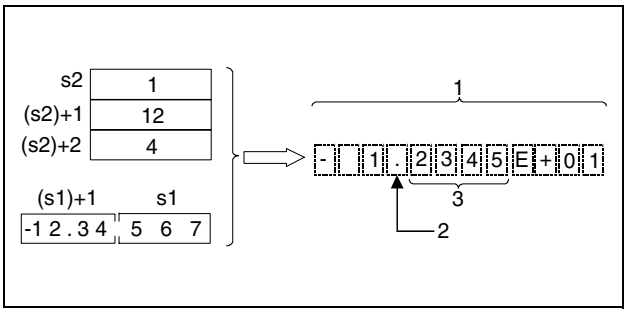


[1] Anzahl aller Stellen (12)

[2] Anzahl der Stellen im Dezimalbereich (4)

[3] Diese Ziffern werden abgeschnitten

Wird die Anzahl der Nachkommastellen auf einen anderen Wert als Null gesetzt, wird das Dezimalkomma "2E$_H$" (.) automatisch an der angegebenen Stelle gesetzt.



[1] Anzahl aller Stellen (12)

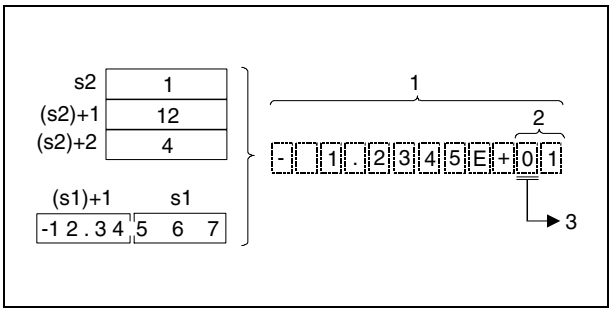
[2] Wird automatisch gesetzt

[3] Anzahl der Stellen im Dezimalbereich (4)

Wird die Anzahl der Stellen im Dezimalbereich auf Null gesetzt, wird das Dezimalkomma "2E$_H$" (.) nicht gesetzt.

Der ASCII-Code "2C$_H$" (+) wird für einen positiven und der ASCII-Code "2D$_H$" (-) für einen negativen Exponenten gesetzt und gespeichert.

Der Exponentialbereich ist auf 2 Stellen festgelegt. Steht die Stellenanzahl des Exponentialbereichs auf 1, wird der ASCII-Code "30H" (0) zwischen Exponentenvorzeichen und Exponent gesetzt und gespeichert.



[1] Anzahl aller Stellen (12)

[2] Ist auf 2 Stellen festgesetzt

[3] Wird automatisch auf Null gesetzt

Der ASCII-Code "00$_H$" wird automatisch am Ende der Zeichenfolge gesetzt und gespeichert.

Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die in s1 und (s1)+1 angegebenen Werte sind nicht Null oder liegen nicht innerhalb des Wertebereichs von $\pm 2^{-127} \leq s1 < \pm 2^{129}$ (Fehlercode 4100).
- Das Format in s2 (Array_s2[1]) ist weder 0 noch 1 (Fehlercode 4100).
- Die Anzahl aller Stellen in (s2)+1 (Array_s2[2]) liegt außerhalb der folgenden Wertebereiche (Fehlercode 4100):

  Im Dezimalformat
  Die Anzahl der Nachkommastellen beträgt Null (Anzahl aller Stellen ≥ 2).
  Die Anzahl der Nachkommastellen nimmt einen anderen Wert als Null an
  (Anzahl aller Stellen ≥ (Anzahl der Nachkommastellen + 3)).

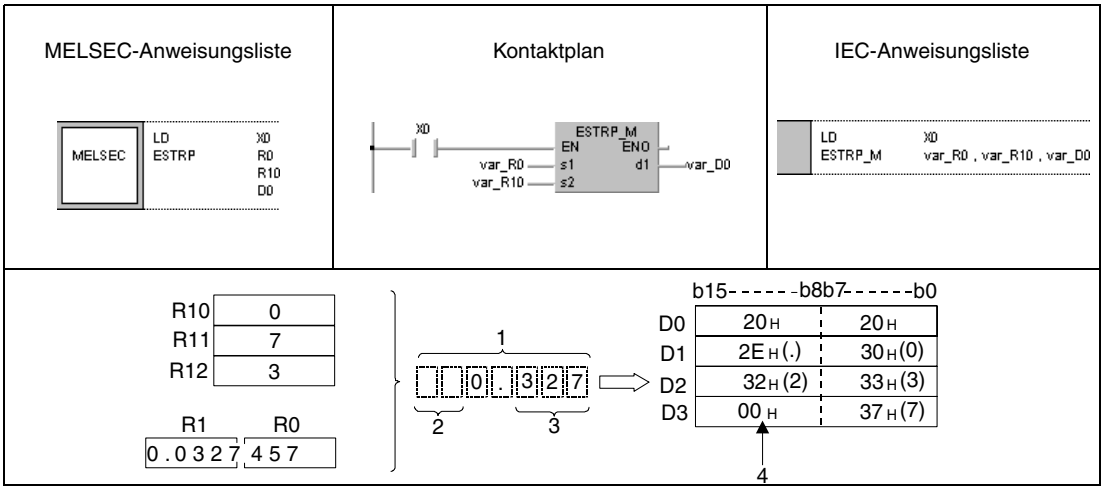
  Im Exponentialformat
  Die Anzahl der Nachkommastellen beträgt Null (Anzahl aller Stellen ≥ 2).
  Die Anzahl der Nachkommastellen nimmt einen anderen Wert als Null an
  (Anzahl aller Stellen ≥ (Anzahl der Nachkommastellen + 7)).
- Die Anzahl der Stellen in (s2)+2 (Array_s2[3]), die den Dezimalteil bilden, liegt außerhalb des Wertebereichs (Fehlercode 4100):

  Im Dezimalformat
  Die Anzahl der Stellen, die den Dezimalteil bilden, ist kleiner oder gleich der Anzahl aller Stellen minus 3.

  Im Exponentialformat
  Die Anzahl der Stellen, die den Dezimalteil bilden, ist kleiner oder gleich der Anzahl aller Stellen minus 7.
- Der Speicherbereich ab d1 liegt außerhalb des für die Speicherung vorgesehenen Bereichs des Operanden (Fehlercode 4101).

Beispiel 1 ESTRP

Das folgende Programm konvertiert mit positiver Flanke von X0 eine Gleitkommazahl (reelle Zahl), die in R0 und R1 angegeben ist, in das in R10 (var_R10 Array [1]) bis R12 (var_R10 Array [3]) angegebene Format und speichert das Ergebnis in D0 bis D3.



[1] Anzahl aller Stellen

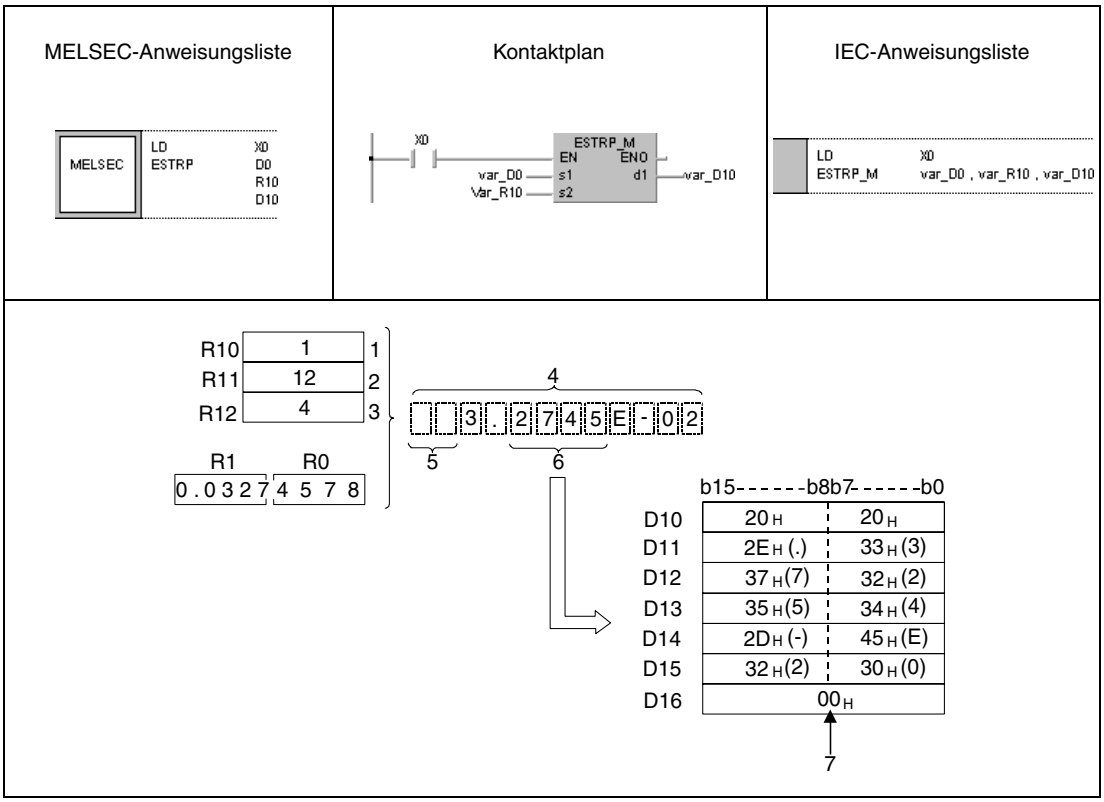
[2] Leerzeichen

[3] Anzahl der Nachkommastellen

[4] Wird automatisch gespeichert

Beispiel 2 ESTRP

Das folgende Programm konvertiert mit positiver Flanke von X0 eine Gleitkommazahl (reelle Zahl), die in D0 und D1 angegeben ist, in das in R10 (var_R10 Array [1]) bis R12 (var_R10 Array [3]) angegebene Format und speichert das Ergebnis in D10 bis D16.



[1] Darstellungsformat (Exponentialdarstellung) (1)

[2] Anzahl aller Stellen

[3] Anzahl der Nachkommastellen

[4] Anzahl aller Stellen

[5] Leerzeichen

[6] Anzahl der Nachkommastellen im Dezimalteil

[7] Wird automatisch gespeichert.

7.11.12 EVAL, EVALP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●[1]

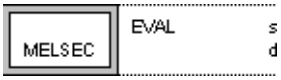
[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

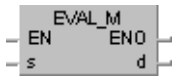
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten $	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	—	—	—	●	—	SM0	3
d	—	●	●	—	●	●	—	—	—		

GX IEC Developer

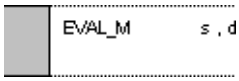
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

Operand	Befehlswert	Datentyp
s	In eine dezimale Gleitkommazahl (reelle Zahl) zu konvertierende Zeichenfolge oder Startadresse des Operanden, in dem diese Daten gespeichert sind.	Zeichenfolge
d	Erste Adresse des Operanden, in der die dezimale Gleitkommazahl (reelle Zahl) nach der Konvertierung gespeichert wird.	reelle Zahl

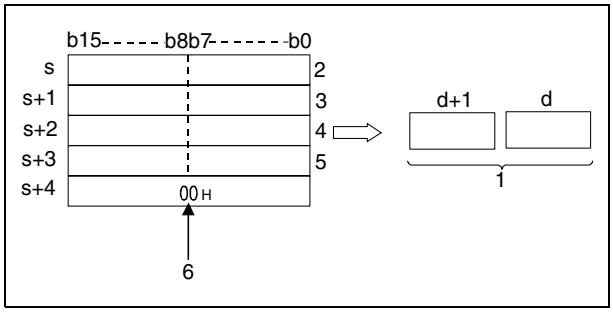
Funktions-weise

Konvertierung von Zeichenfolgen in dezimale Gleitkommazahlen

EVAL Konvertierung von Zeichenfolgen

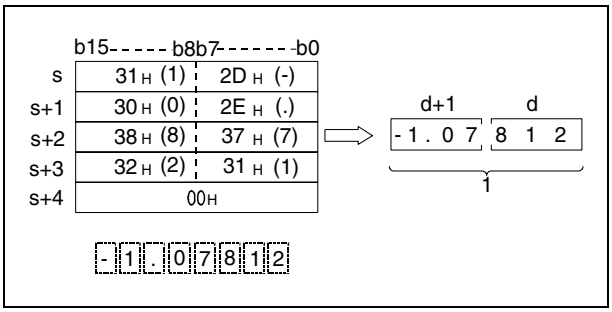
Die Anweisung konvertiert die Zeichenfolge in s bis s+4 in eine dezimale Gleitkommazahl (reelle Zahl). Das Ergebnis wird ab d gespeichert.

Die zur Konvertierung bestimmte Zeichenfolge kann in das dezimale Gleitkommaformat als auch in das Exponentialformat umgewandelt werden.



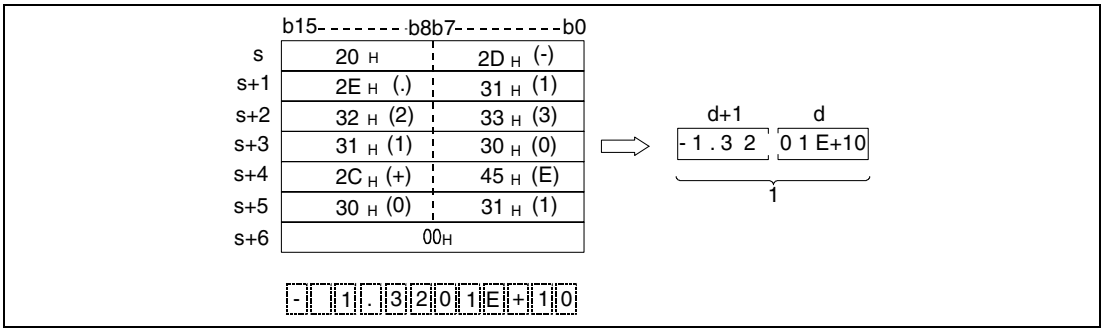
[1] Dezimale Gleitkommazahl (reelle Zahl)
[2] ASCII-Code des 1. Zeichens/ ASCII-Code des Vorzeichens
[3] ASCII-Code des 3. Zeichens/ ASCII-Code des 2. Zeichens
[4] ASCII-Code des 5. Zeichens/ ASCII-Code des 4. Zeichens
[5] ASCII-Code des 7. Zeichens/ ASCII-Code des 6. Zeichens
[6] Kennzeichnet das Ende der Zeichenfolge

Dezimaldarstellung



[1] Dezimale Gleitkommazahl (reelle Zahl)

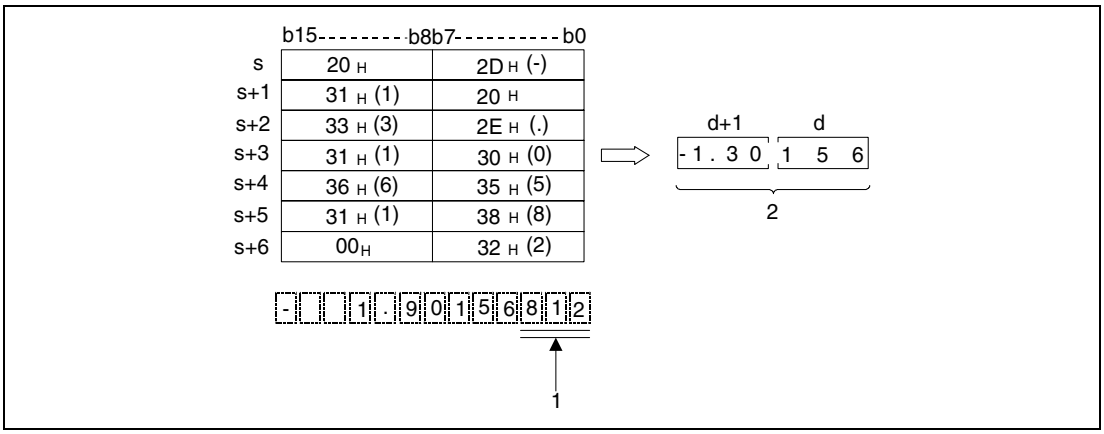
Exponentialdarstellung



[1] Dezimale Gleitkommazahl (reelle Zahl)

Es werden 6 Stellen (ohne Vorzeichen, Dezimalkomma und Exponentenstellen des Resultates) der Zeichenfolge ab s zu einer dezimalen Gleitkommazahl konvertiert. Ab der 7. Stelle werden die weiteren Stellen vom Resultat abgeschnitten.

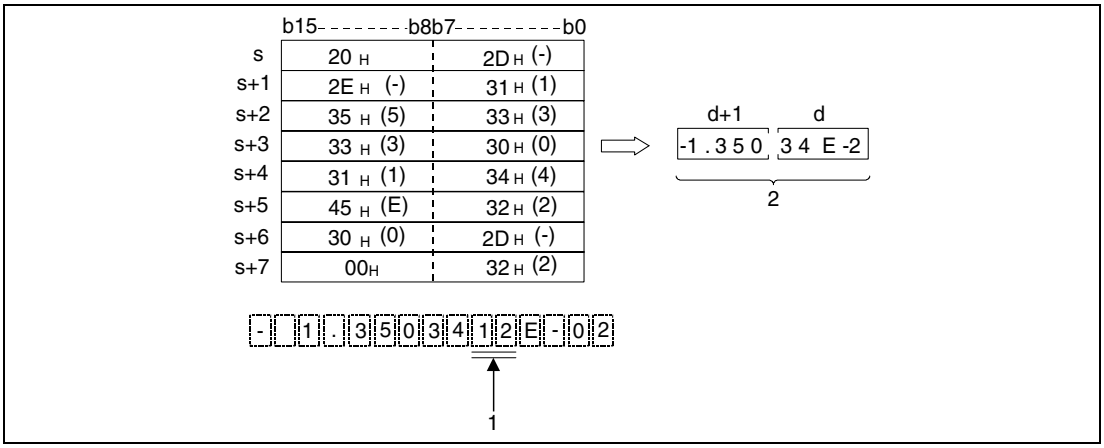
Dezimaldarstellung



[1] Diese Ziffern werden abgeschnitten.

[2] Dezimale Gleitkommazahl (reelle Zahl)

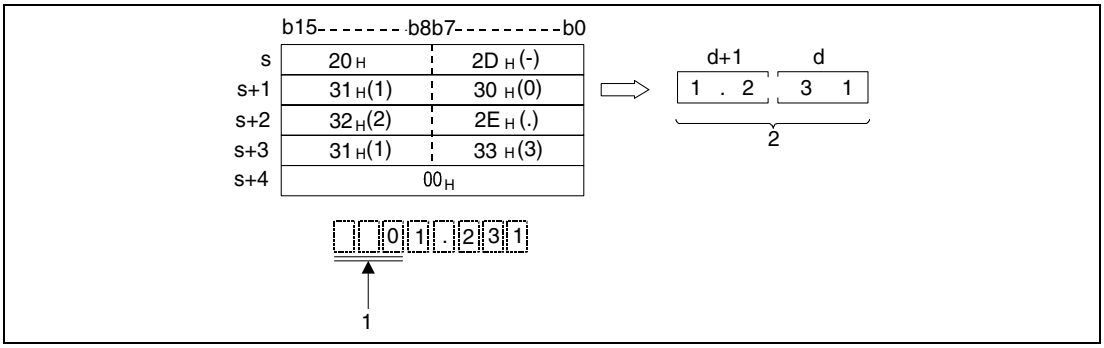
Exponentialdarstellung



[1] Diese Ziffern werden abgeschnitten.

[2] Dezimale Gleitkommazahl (reelle Zahl)

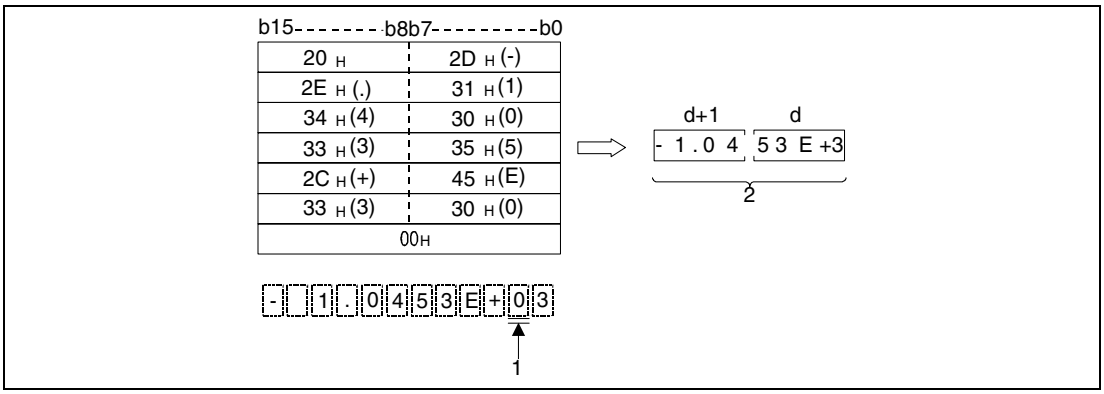
Wenn in der Zeichenfolge ab s der ASCII-Code für "20H" (Leerzeichen) oder für "30H" (Null) vor der ersten darzustellenden Ziffer gesetzt ist, werden diese Zeichen bei der Konvertierung ignoriert.



[1] Diese Ziffern werden bei der Verarbeitung nicht berücksichtigt

[2] Dezimale Gleitkommazahl (reelle Zahl)

Wenn der ASCII-Code für (Null) "30H" zwischen dem Zeichen "E" und der Zeichenfolge für das Exponentialformat gesetzt ist, wird dieses Zeichen bei der Konvertierung ignoriert.



[1] Diese Ziffer wird bei der Verarbeitung nicht berücksichtigt.

[2] Dezimale Gleitkommazahl (reelle Zahl)

Eine zu konvertierende Zeichenfolge kann aus maximal 24 Zeichen bestehen.

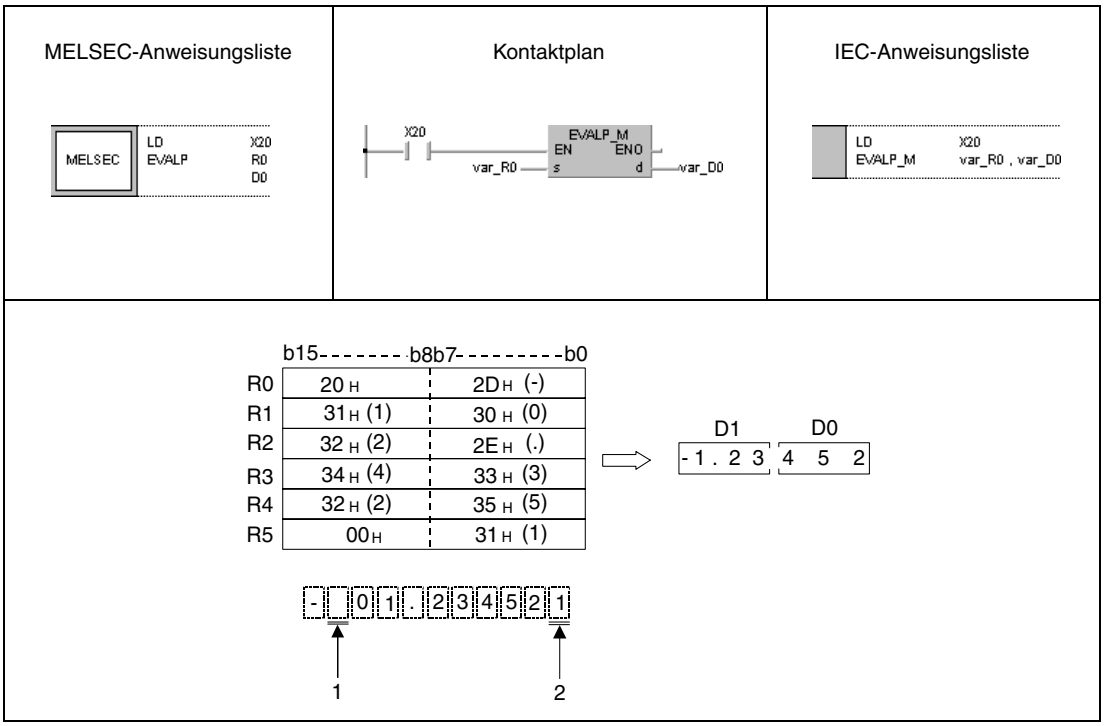
Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die Zeichenfolge beginnt mit einem anderen Zeichen als "20H" (Leerzeichen) oder "2DH" (Minus) (Fehlercode 4100).
- Die Stellen vor dem Komma oder die Nachkommastellen weisen andere Zeichen auf als aus dem Wertebereich zwischen "30H" (0) bis "39H" (9) (Fehlercode 4100).
- Das Zeichen "2EH" wurde mehr als einmal in der Zeichenfolge verwendet (Fehlercode 4100).
- Im Exponententeil sind andere Zeichen verwendet worden als "45H (E), 2CH (+)" oder "45H (E), 2DH (-)", oder es ist mehr als ein Exponent vorhanden (Fehlercode 4100).
- Der Wert ist nicht Null oder größer als 1.0×2^{129} oder kleiner als 1.0×2^{-127} (Fehlercode 4100).
- Die Endmarke "00H" liegt nicht innerhalb des für die Speicherung der Zeichenfolge vorgesehenen Bereichs (Fehlercode 4100).
- Die Anzahl der Zeichen in der Zeichenfolge beträgt Null oder mehr als 24.

Beispiel 1 EVALP

Das folgende Programm konvertiert mit positiver Flanke von X20 die in R0 bis R5 angegebene Zeichenfolge in eine dezimale Gleitkommazahl (reelle Zahl) und speichert das Ergebnis in D0 und D1.

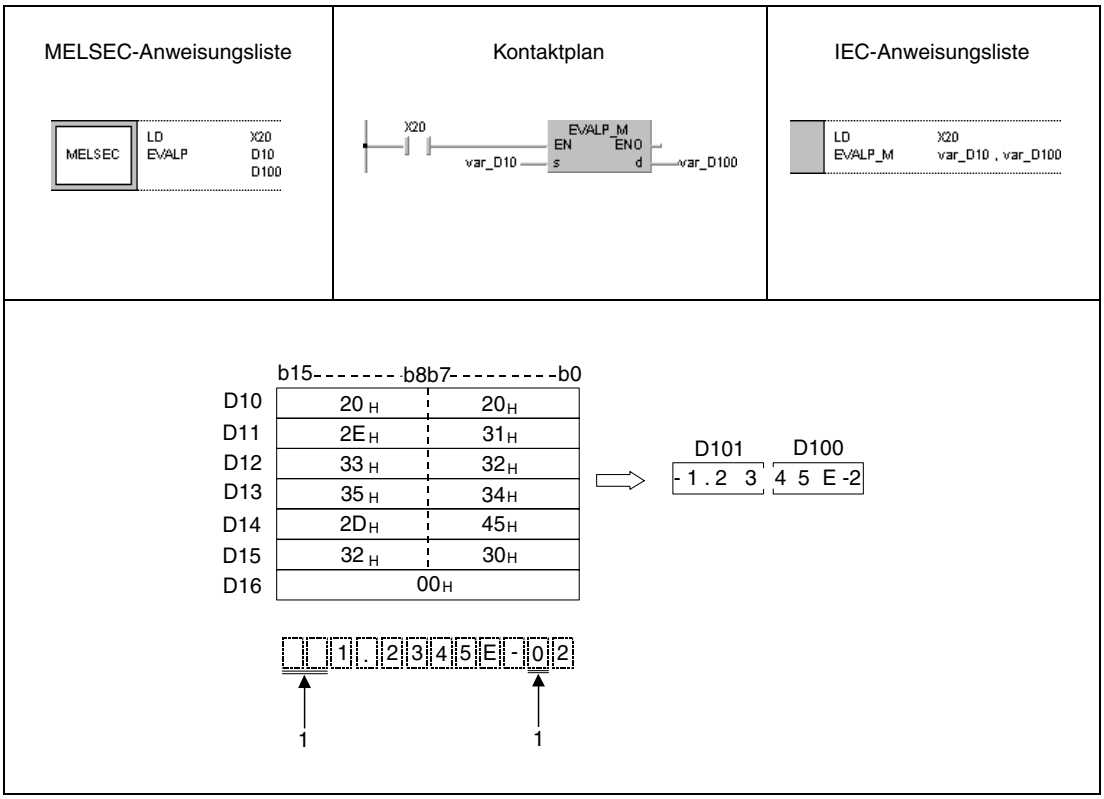


[1] Diese Stelle wird bei der Verarbeitung nicht berücksichtigt.

[2] Diese Ziffer wird abgeschnitten.

Beispiel 2 EVALP

Das folgende Programm konvertiert mit positiver Flanke von X20 die in D10 bis D16 angegebene Zeichenfolge in eine dezimale Gleitkommazahl (reelle Zahl) und speichert das Ergebnis in D100 und D101.



[1] Diese Stellen werden bei der Verarbeitung nicht berücksichtigt.

HINWEIS *Diese Programmbeispiele sind ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser der Programmieranleitung zu entnehmen.*

7.11.13 ASC, ASCP (Q-Serie und System Q)

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●[2]

[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

[2] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	—	—	—	—	—	SM0	4
d	—	●	●	—	—	—	—	—	—		
n	●	●	●	●	●	●	●	●	—		

GX IEC Developer

MELSEC-Anweisungsliste

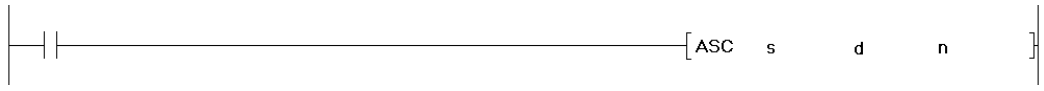
Kontaktplan

IEC-Anweisungsliste

Die Befehle ASC und ASCP funktionieren nicht in den IEC-Editoren. Sie können nur in der MELSEC-Anweisungsliste programmiert werden.

Abhilfe: Verschieben Sie die hexadezimalen ASC-Werte direkt in die Datenworte.

GX Developer



Variablen

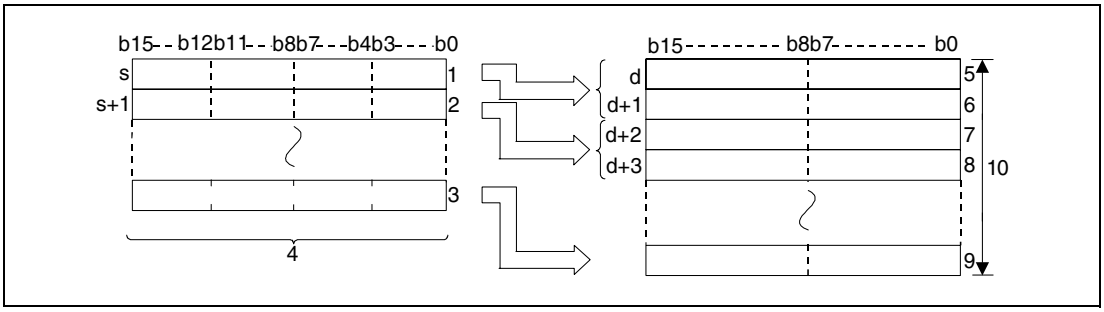
Operand	Befehlswert	Datentyp
s	Erste Adresse des Operanden, in dem die zu konvertierenden Binärdaten gespeichert sind.	BIN-16-Bit
n	Anzahl der zu konvertierenden Zeichen.	
d	Erste Adresse des Operanden, in dem die konvertierten Zeichenfolgen gespeichert werden.	Zeichenfolge

Funktions-weise

Konvertierung BIN-16-Bit-Daten in den ASCII-Code

ASC/ASCP Konvertierungsanweisung

Die ASC-Anweisung konvertiert die ab s gespeicherten 16-Bit-Binärdaten in das hexadezimale ASCII-Format und speichert das Resultat unter Berücksichtigung der in n angegebenen Anzahl von Zeichen beginnend ab d.



[1] Erste Stelle/ Zweite Stelle/ Dritte Stelle/ Vierte Stelle

[2] Erste Stelle/ Zweite Stelle/ Dritte Stelle/ Vierte Stelle

[3] Erste Stelle/ Zweite Stelle/ Dritte Stelle/ Vierte Stelle

[4] Binärdaten

[5] ASCII-Code der 1. Stelle/ ASCII-Code der 2. Stelle

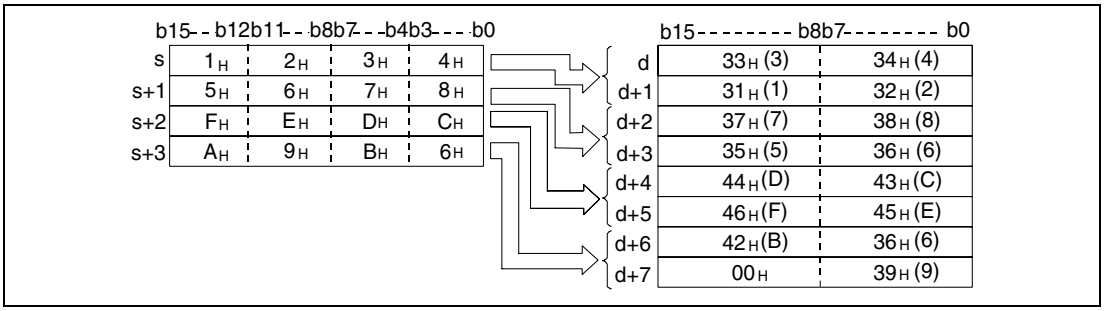
[6] ASCII-Code der 3. Stelle/ ASCII-Code der 4. Stelle

[7] ASCII-Code der 5. Stelle/ ASCII-Code der 6. Stelle

[8] ASCII-Code der 7. Stelle/ ASCII-Code der 8. Stelle

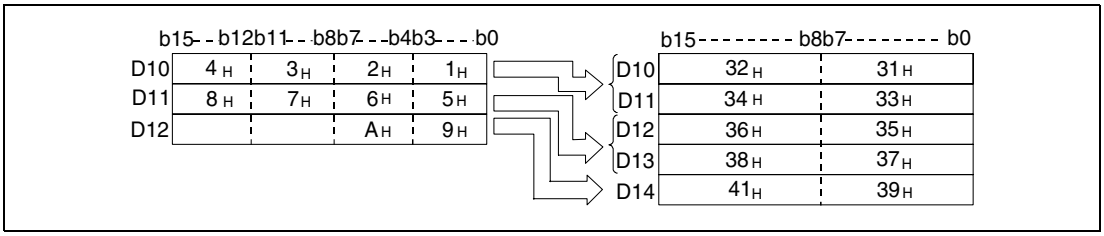
[9] ASCII-Code der 9. Stelle/ ASCII-Code der 10. Stelle

[10] Anzahl der in n angegebenen Zeichen

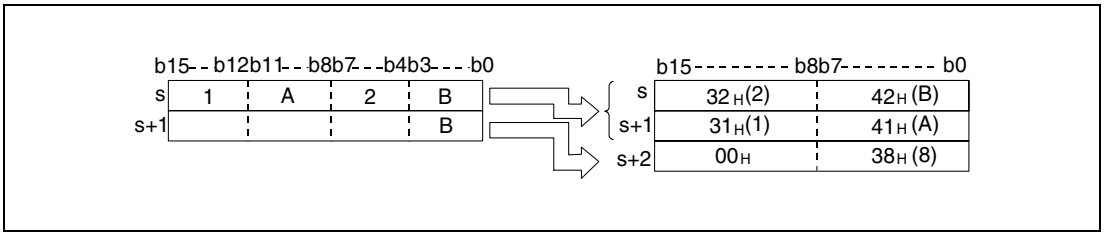


Die Anzahl von Zeichen, die in n angegeben ist, bedingt die Wertebereiche der ab s und d angegebenen Operanden. Die ab s angegebenen Operanden beinhalten die zu konvertierenden Binärdaten. In den ab d angegebenen Operanden wird die konvertierte Zeichenfolge gespeichert.

Die Programmverarbeitung erfolgt auch präzise und ohne Fehlermeldung, wenn der Speicherbereich der zu konvertierenden Binärdaten mit dem der konvertierten ASCII-Daten überlappt.



Wenn eine ungerade Anzahl von Zeichen in n angegeben ist, wird das ASCII-Zeichen "00H" automatisch in die höherwertigen 8 Bits der letzten Adresse des Bereichs, in dem die Zeichenfolge gespeichert werden soll, gesetzt.



Wenn die Anzahl der Zeichen, die in n angegeben ist, Null beträgt, wird das Programm nicht ausgeführt.

Fehlerquellen

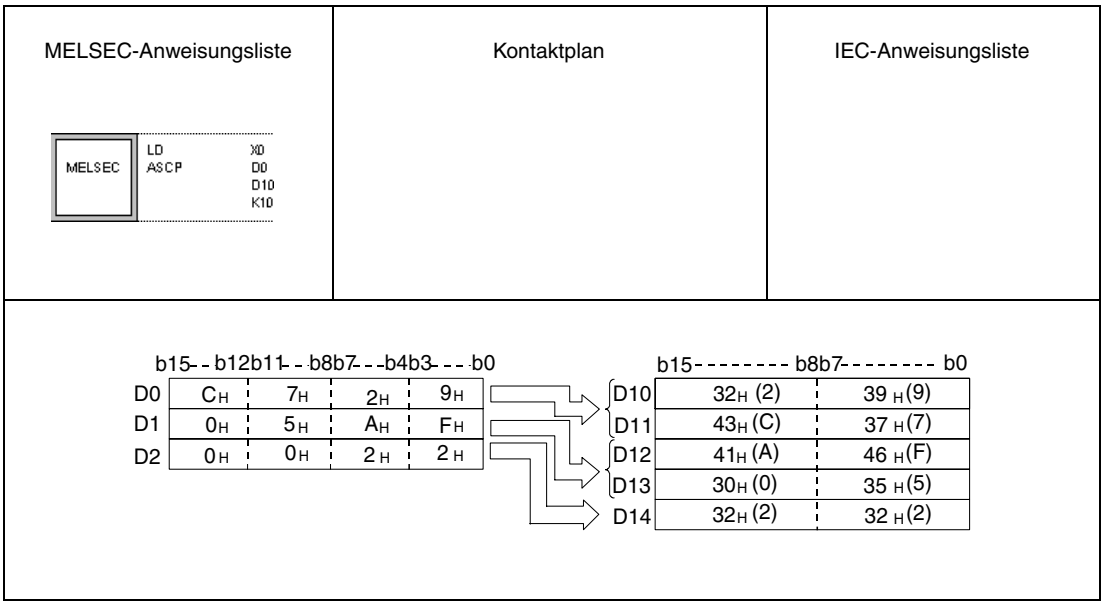
In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die in n angegebene Anzahl von Zeichen und somit die ab s benötigte Anzahl von Registern liegt außerhalb des für die Speicherung vorgesehenen Bereichs des Operanden (Fehlercode 4101).
- Die in n angegebene Anzahl von Zeichen und somit die ab d benötigte Anzahl von Registern liegt außerhalb des für die Speicherung vorgesehenen Bereichs des Operanden (Fehlercode 4101).

Beispiel

ASCP

Das folgende Programm liest mit positiver Flanke von X0 die in D0 gespeicherten Binärdaten als Hexadezimalwerte und konvertiert sie in eine Zeichenfolge. Das Ergebnis wird in D10 bis D14 gespeichert.



7.11.14 ASC (A-Serie)

CPU

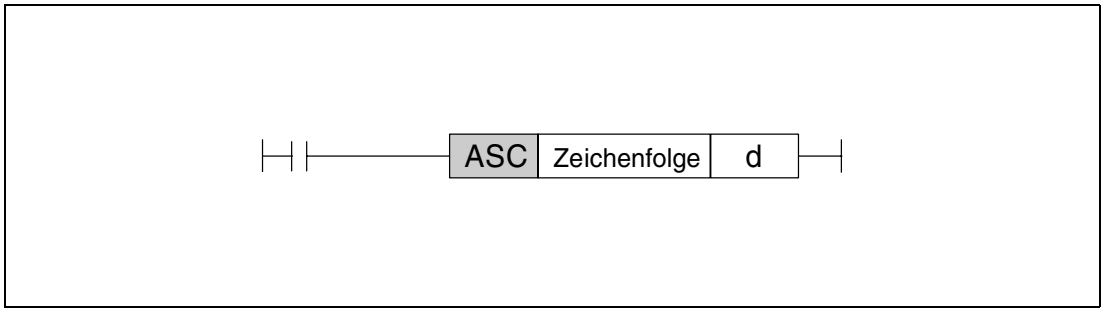
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●		

Operanden MELSEC A

	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)								Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
d								●	●	●	●	●											13 ●[1]	●		●

[1] Die Anzahl der Schritte bei Verwendung einer AnA, AnU oder AnAS CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

GX IEC Developer



Variablen

Operand	Befehlswert	Datentyp
d	Operand, in dem die konvertierten Zeichen gespeichert werden.	Zeichenfolge

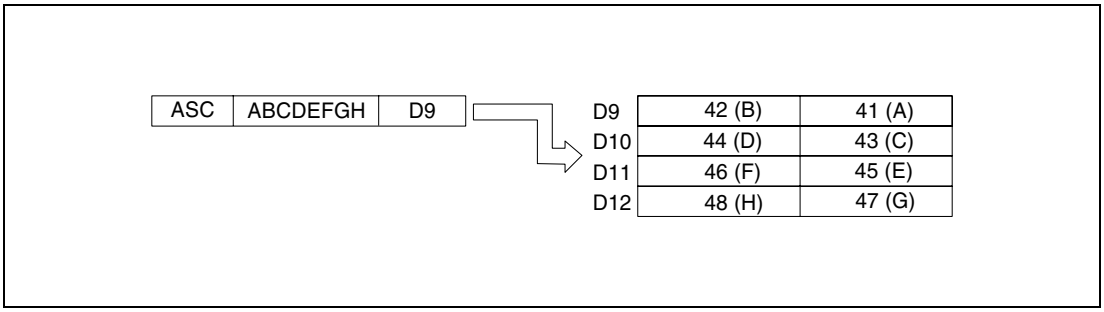
Funktions-weise

Konvertierung von Zeichenfolgen in den ASCII-Code

ASC Konvertierung von alphanumerischen Zeichenfolgen

Die ASC-Anweisung konvertiert alphanumerische Zeichenfolgen mit bis zu 8 Zeichen in den ASCII-Code. Das Ergebnis wird ab d gespeichert.

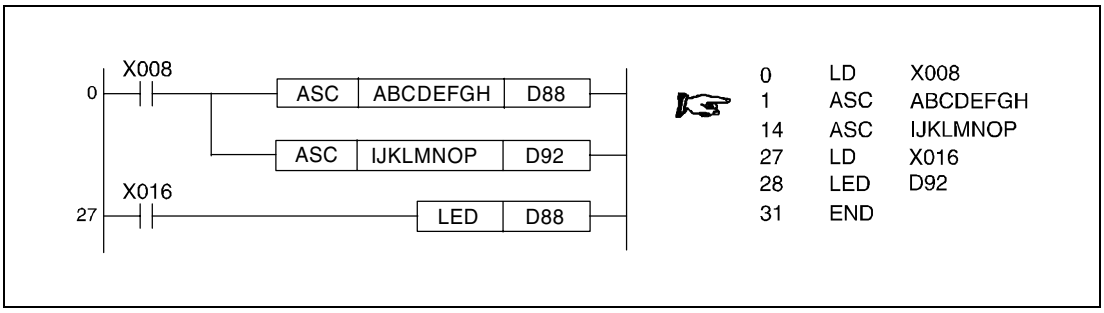
Ein Ausdruck des gespeicherten ASCII-Codes ist über die PR-/ PRC-Anweisung möglich. Eine Anzeige über das LED-Display der dafür geeigneten CPUs ist mittels der LED-Anweisung möglich.



Beispiel

ASCP

Das folgende Programm konvertiert nach dem Einschalten von X8 die Zeichenfolge "ABCDEF-GHIJKLMNOP" in einen ASCII-Code und speichert das Ergebnis in D88 bis D91 und D92 bis D95. Nach dem Einschalten von X16 werden die ASCII-Daten aus D88 bis D95 über die LEDs an der Frontseite der CPU angezeigt.



7.11.15 HEX, HEXP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●[2]

[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

[2] Nicht für Q00JCPU, Q00CPU und Q01CPU

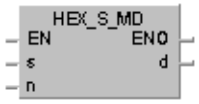
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	—	—	—	—	—	SM0	4
d	—	●	●	—	—	—	—	—	—		
n	●	●	●	●	●	●	●	●	—		

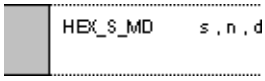
GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

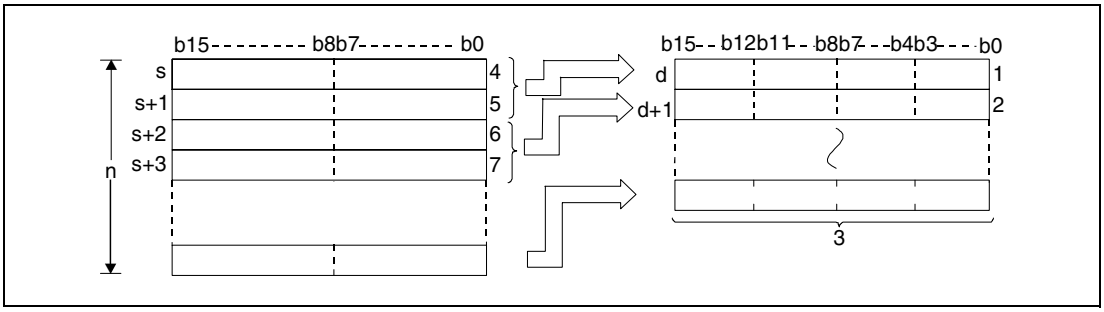
Operand	Befehlswert	Datentyp
s	Erste Adresse des Operanden, in dem die in Binärdaten zu konvertierenden Zeichenfolgen gespeichert sind.	Zeichenfolge
d	Erste Adresse des Operanden, in dem die konvertierten Binärdaten gespeichert werden.	BIN-16-Bit
n	Anzahl der zu konvertierenden Zeichen.	

Funktions-weise

Konvertierung von hexadezimalen ASCII-Werten in Binärwerte

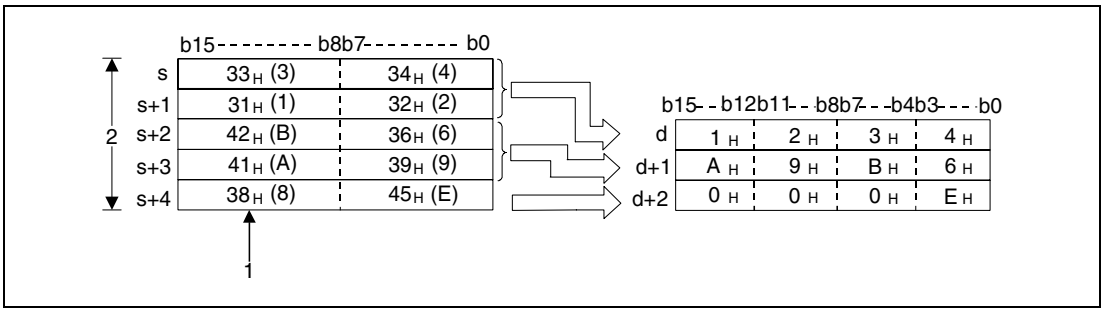
HEX Konvertierung von hexadezimalen ASCII-Werten

Die Anweisung konvertiert die hexadezimalen ASCII-Zeichen ab s in Binärwerte. Die Anzahl der zu konvertierenden Zeichen wird in n festgelegt. Das Ergebnis der Konvertierung wird ab d gespeichert.



[1] Vierte Stelle/ Dritte Stelle/ Zweite Stelle/ Erste Stelle
[2] Vierte Stelle/ Dritte Stelle/ Zweite Stelle/ Erste Stelle
[3] Binärdaten
[4] ASCII-Code der 2. Stelle/ ASCII-Code der 1. Stelle
[5] ASCII-Code der 4. Stelle/ ASCII-Code der 3. Stelle
[6] ASCII-Code der 2. Stelle/ ASCII-Code der 1. Stelle
[7] ASCII-Code der 4. Stelle/ ASCII-Code der 3. Stelle

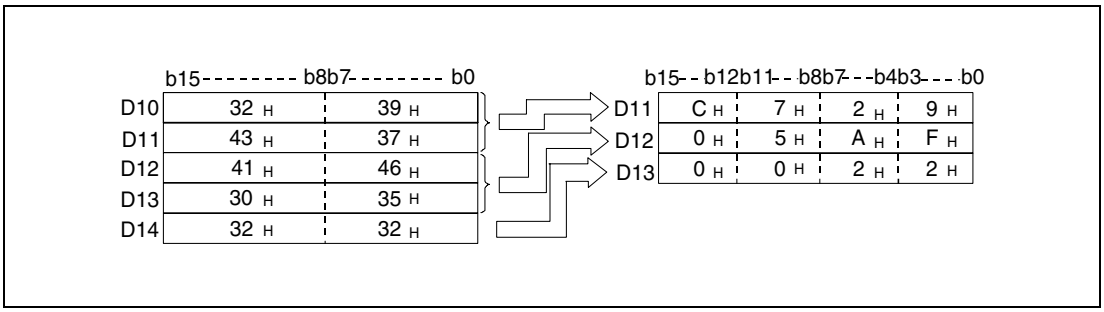
Die Anzahl der Zeichen in n beträgt 9.



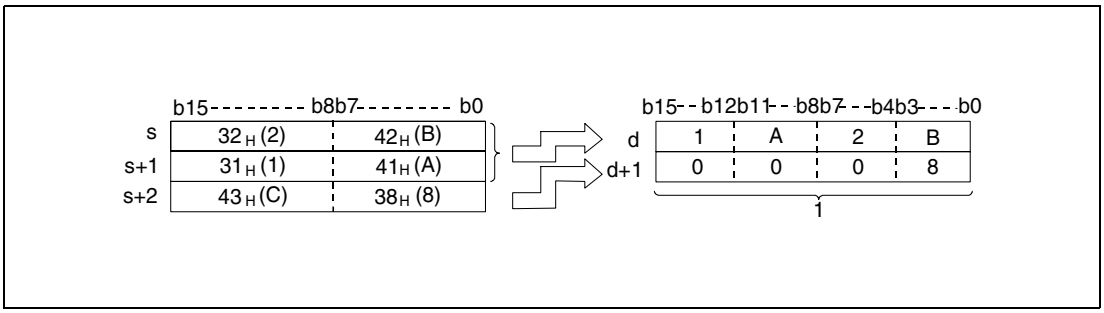
[1] Da eine Zeichenfolge mit 9 Zeichen angegeben ist, wird die "38H" nicht verändert und nicht übertragen.
[2] n = 9

Durch die Angabe der Zeichenanzahl in n wird der Wertebereich der Zeichenfolge ab s und der Binärdaten ab d automatisch gesetzt.

Obwohl der Wertebereich des ASCII-Codes, der konvertiert werden soll, mit dem der konvertierten Binärwerte überlappt, verarbeitet diese Anweisung die Daten korrekt.



Wenn die Anzahl der Zeichen in n nicht durch 4 teilbar ist, wird automatisch eine Null nach der angegebenen Anzahl von Zeichen in die letzten Register, in denen die konvertierten Binärwerte gespeichert werden, geschrieben.



[1] Der Wert Null wird automatisch gespeichert

Wenn die Anzahl der Zeichen in n gleich Null beträgt, wird der Konvertierungsvorgang nicht durchgeführt.

Der ASCII-Code ab s kann in den Bereichen von "30H" bis "39H" und "41H" bis "46H" liegen.

Fehlerquellen

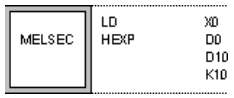
In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die ab s angegebenen Operanden enthalten andere Zeichen als die aus dem Bereich von "30H" bis "39H" und "41H" und "46H" (Fehlercode 4100).
- Die in n angegebene Anzahl von Zeichen und somit die ab s benötigte Anzahl von Registern liegt außerhalb des für die Speicherung vorgesehenen Bereichs des Operanden (Fehlercode 4101).
- Die in n angegebene Anzahl von Zeichen und somit die ab d benötigte Anzahl von Registern liegt außerhalb des für die Speicherung vorgesehenen Bereichs des Operanden (Fehlercode 4101).
- Der Wert in n ist negativ.

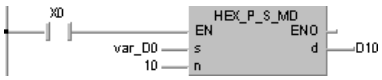
Beispiel HEXP

Das folgende Programm konvertiert mit positiver Flanke von X0 die Zeichenfolge "6B52A71379", die in D0 bis D4 gespeichert ist, in Binärdaten und speichert das Ergebnis in D10 bis D14.

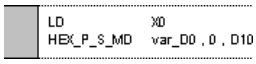
MELSEC-Anweisungsliste

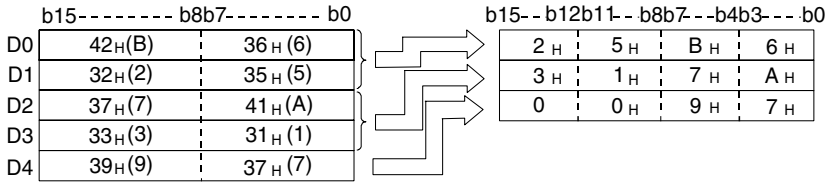


Kontaktplan



IEC-Anweisungsliste





HINWEIS *Diese Programmbeispiele sind ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.11.16 RIGHT, RIGHTP, LEFT, LEFTP

CPU

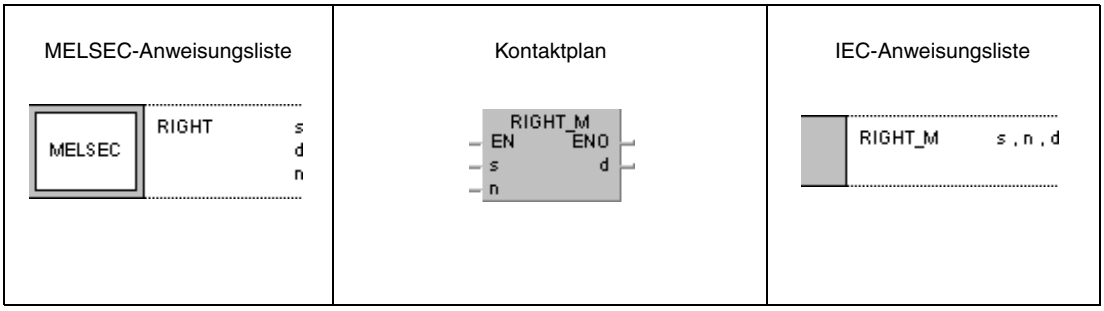
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●[1]

[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

	Operanden										Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten		Andere		
	Bit	Wort		Bit	Wort			K, H (16#)	$	U		
s	—	●	●	—	—	—	—	—	●	—	SM0	4
d	—	●	●	—	—	—	—	—	—	—		
n	●	●	●	●	●	●	●	●	—	—		

GX IEC Developer



GX Developer

Variablen

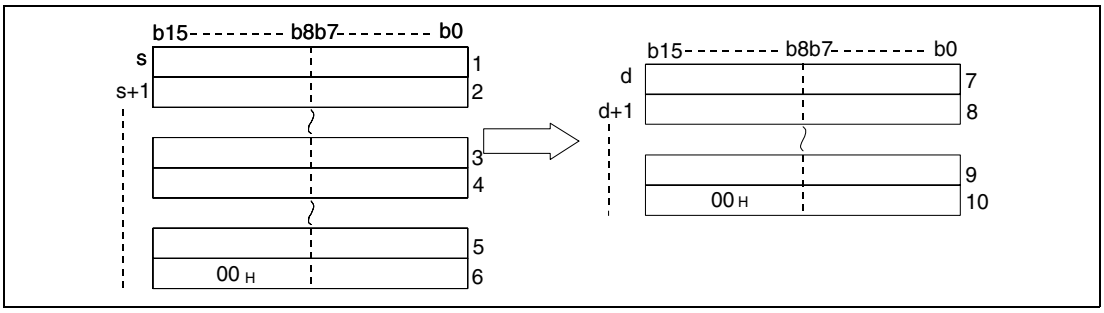
Operand	Befehlswert	Datentyp
s	Erste Adresse des Operanden, in dem die Zeichenfolge gespeichert ist.	Zeichenfolgen
d	Erste Adresse des Operandenbereichs, in dem sich die ermittelten Zeichen der Zeichenfolge befinden.	
n	Anzahl der Zeichen, die von links oder von rechts gespeichert werden.	BIN-16-Bit

Funktions-weise

Auszug der Zeichenfolgendaten von rechts oder von links

RIGHT Auszug der Zeichenfolgendaten von rechts

Die RIGHT-Anweisung speichert n Zeichen von der rechten Seite der Zeichenfolge (Ende der Zeichenfolge) ab s. Die Zeichen werden ab d gespeichert.



[1] ASCII-Code des 2. Zeichens/ ASCII-Code 1. Zeichens

[2] ASCII-Code des 4. Zeichens/ ASCII-Code 3. Zeichens

[3] ASCII-Code des letzten Zeichens minus n+2/ ASCII-Code des letzten Zeichens minus n+1

[4] ASCII-Code des letzten Zeichens minus n+4/ ASCII-Code des letzten Zeichens minus n+3

[5] ASCII-Code des letzten Zeichens minus 1/ ASCII-Code des letzten Zeichens minus 2

[6] "00H"/ ASCII-Code des letzten Zeichens

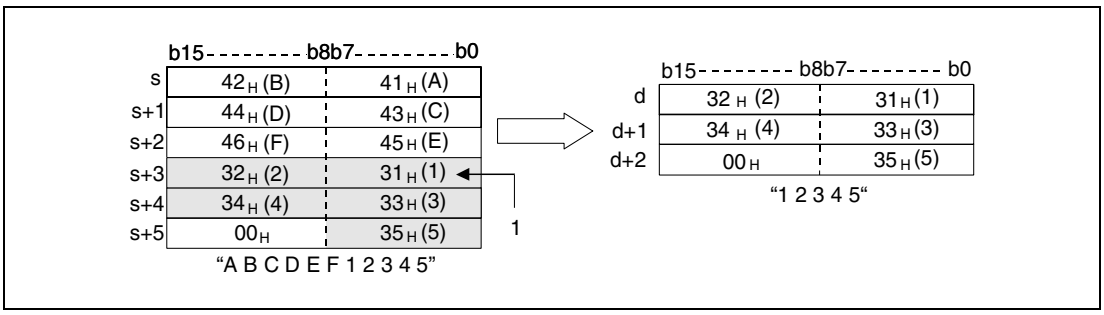
[7] ASCII-Code des letzten Zeichens minus n+2/ ASCII-Code des letzten Zeichens minus n+1

[8] ASCII-Code des letzten Zeichens minus n+4/ ASCII-Code des letzten Zeichens minus n+3

[9] ASCII-Code des letzten Zeichens minus 1/ ASCII-Code des letzten Zeichens minus 2

[10] "00H"/ ASCII-Code des letzten Zeichens

Bei n = 5

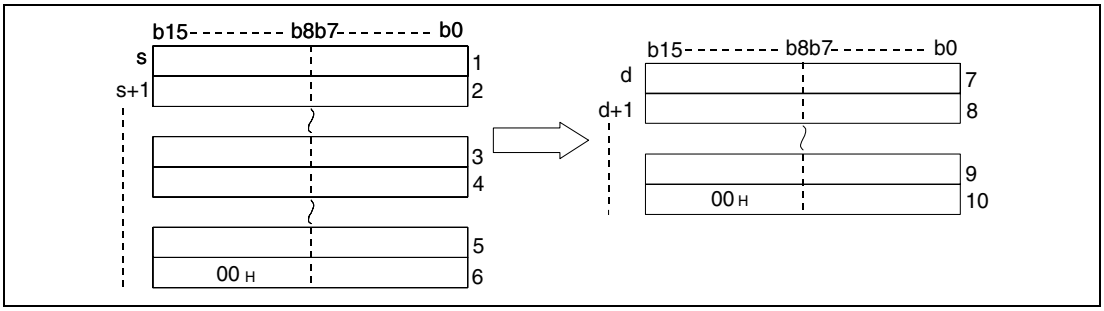


[1] ASCII-Code für das 5. Zeichen

Wenn die Anzahl der Zeichen in n gleich Null beträgt, wird der Zeichencode "00H" ab d gespeichert.

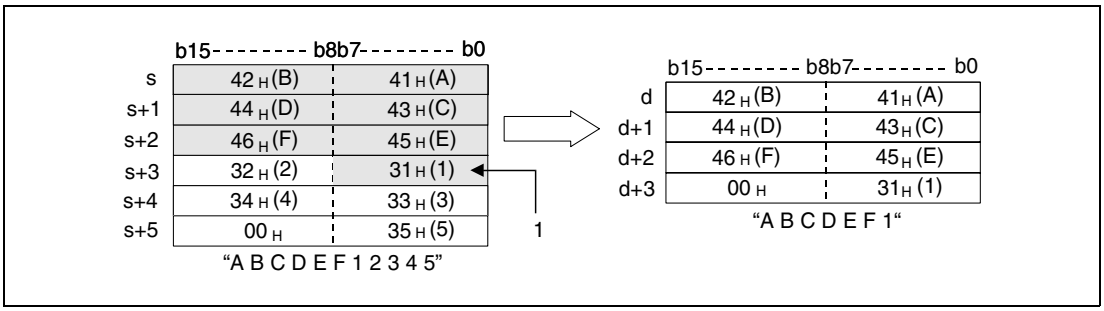
LEFT Auszug der Zeichenfolgendaten von links

Die LEFT-Anweisung speichert n Zeichen von der linken Seite der Zeichenfolge (Anfang der Zeichenfolge) ab s. Die Zeichen werden ab d gespeichert.



[1] ASCII-Code des 2. Zeichens/ ASCII-Code 1. Zeichens
[2] ASCII-Code des 4. Zeichens/ ASCII-Code 3. Zeichens
[3] ASCII-Code des Zeichens n-1/ ASCII-Code Zeichens n-2
[4] ASCII-Code des Zeichens n+1/ ASCII-Code des n-ten Zeichens
[5] "00H"/ ASCII-Code des letzten Zeichens
[6] ASCII-Code des 2. Zeichens/ ASCII-Code 1. Zeichens
[7] ASCII-Code des 4. Zeichens/ ASCII-Code 3. Zeichens
[8] ASCII-Code des Zeichens n-1/ ASCII-Code des Zeichens n- 2
[9] "00H"/ ASCII-Code des n-ten Zeichens

Bei n=7



[1] ASCII-Code des 7. Zeichens

Wenn die Anzahl der Zeichen in n gleich Null beträgt, wird der Zeichencode "00H" ab d gespeichert.

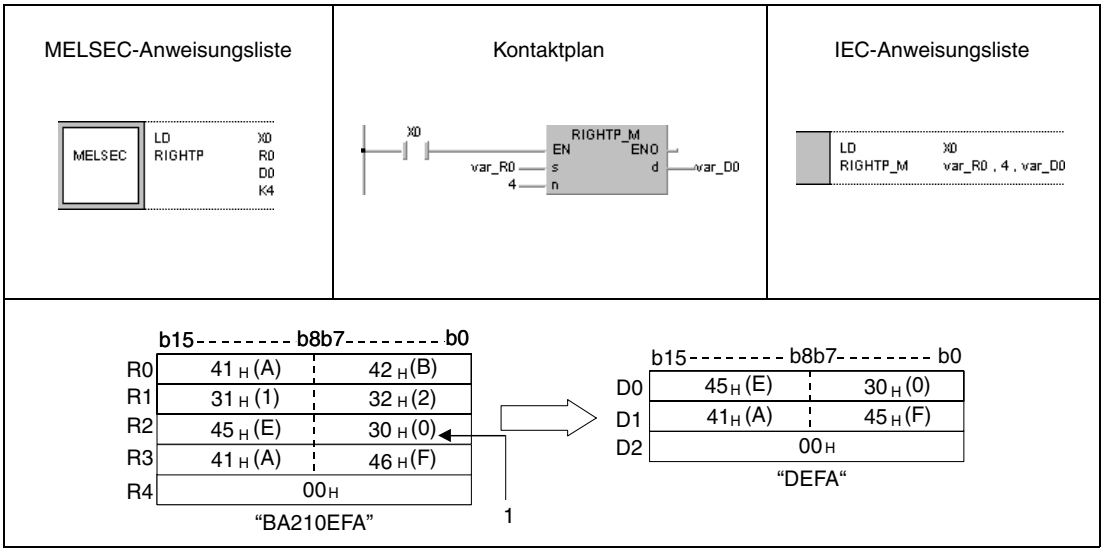
Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Der Wert in n ist größer als die vorhandenen Zeichen ab s (Fehlercode 4101).
- Der mit n angegebene Bereich liegt außerhalb des für die Speicherung vorgesehenen Bereichs des Operanden ab d (Fehlercode 4101).

Beispiel 1 RIGHTP

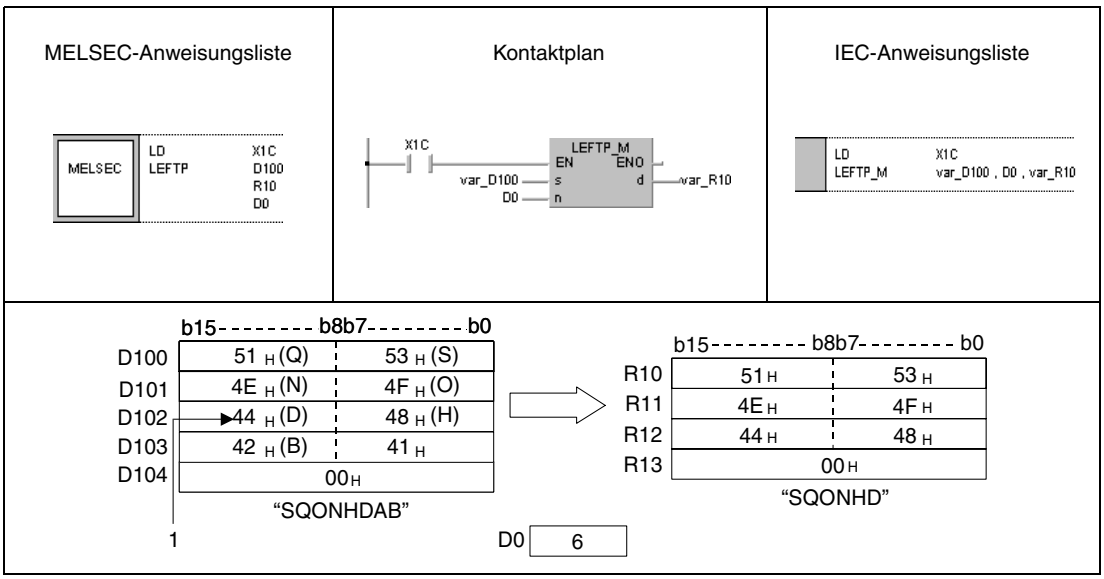
Das folgende Programm ermittelt mit positiver Flanke von X0 4 Zeichen der Daten von der rechten Seite der in R0 bis R4 gespeicherten Zeichenfolge und speichert sie in D0 bis D2.



[1] ASCII-Code des 4. Zeichens

Beispiel 2 LEFTP

Das folgende Programm ermittelt mit positiver Flanke von X1C die in D0 angegebene Anzahl von Zeichen von der linken Seite der in D100 bis D104 angegebenen Zeichenfolge und speichert sie in R10 bis R13.



[1] ASCII-Code des 6. Zeichens

HINWEIS *Diese Programmbeispiele sind ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.11.17 MIDR, MIDRP, MIDW, MIDWP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●[1]

[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

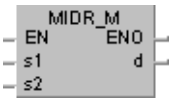
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten $	Andere		
	Bit	Wort		Bit	Wort						
s1	—	●	●	—	—	—	—	●	—	SM0	4
d	—	●	●	—	—	—	—	—	—		
s2	●	●	●	●	●	●	●	—	—		

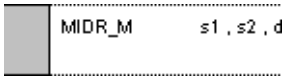
GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

Operand	Befehlswert	Datentyp	
		MELSEC	IEC
s1	Erste Adresse des Operanden, in dem die Zeichenfolge gespeichert ist.	Zeichenfolge	Zeichenfolge
d	Erste Adresse des Operanden, in dem die Zeichenfolge des Verarbeitungsresultates gespeichert wird.		
s2	Erste Adresse des Operanden, in dem das 1. Zeichen und die Anzahl der Zeichen gespeichert ist. (s2)+0 (Array_s2[1]): Register des 1. Zeichens (s2)+1 (Array_s2[2]): Anzahl der Zeichen	BIN-16-Bit	Array [1..2] of ANY16

Funktionsweise

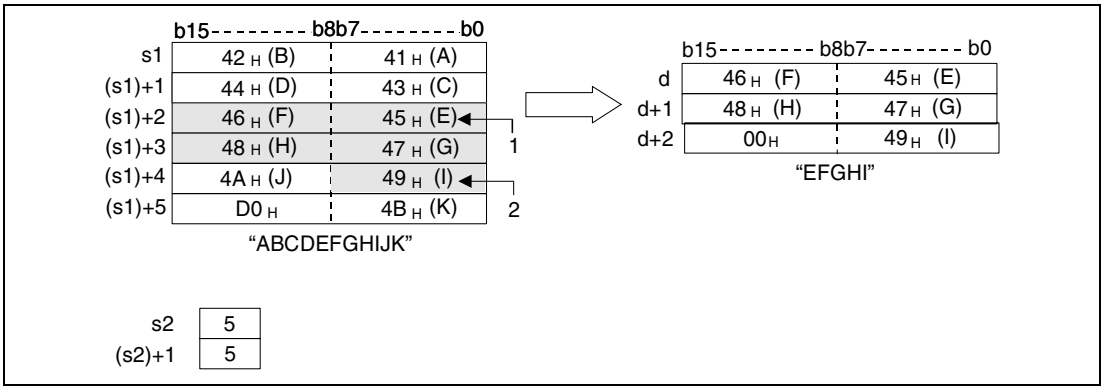
Speichern und Verschieben von Zeichenfolgenteilen

MIDR Speichern definierter Zeichenfolgenteile

Die MIDR-Anweisung speichert einen definierten Teil der ab s1 gespeicherten Zeichenfolge ab d.

Das erste Zeichen des zu speichernden Teils ist in s2 (Array_s2[1]) angegeben und wird vom linken Teil der Zeichenfolge (unteres Byte von s1) beginnend gezählt.

Die Länge des zu speichernden Teils ist in s2+1 (Array_s2[2]) angegeben.

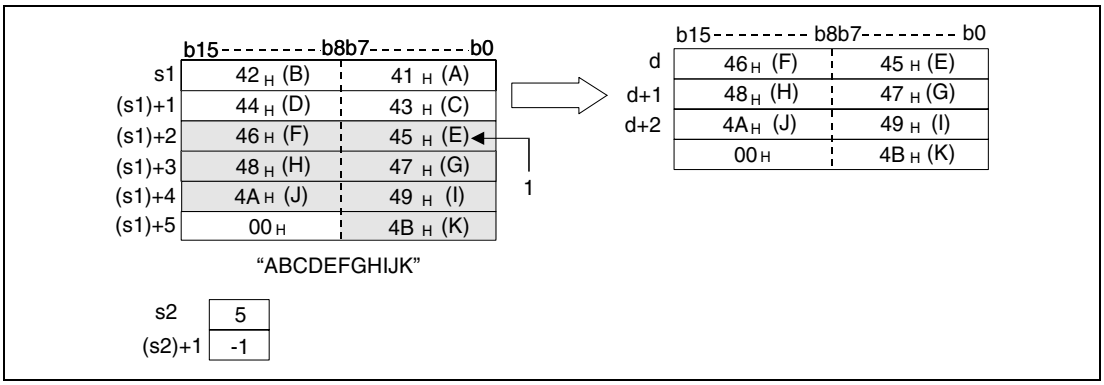


[1] Position des 5.Zeichens (s2)

[2] Position des letzten zu speichernden Zeichens

Es wird keine Verarbeitung durchgeführt, wenn die Anzahl der Zeichen in (s2)+1 (Array_s2[2]) Null beträgt.

Wenn in (s2)+1 (Array_s2[2]) der Wert -1 angegeben wird, erfolgt die Speicherung aller ab dem in s2 (Array_s2[1]) angegebenen Zeichen.



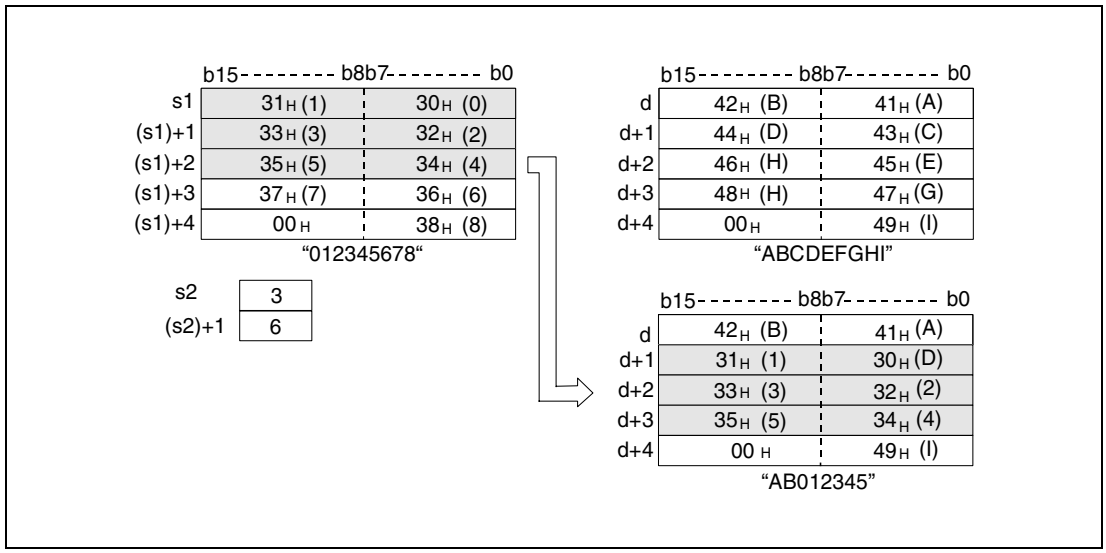
[1] Position des 5.Zeichens (s2)

MIDW Verschieben von Zeichenfolgenteilen in einen definierten Bereich

Die MIDW-Anweisung speichert einen Teil definierter Länge der ab s1 gespeicherten Zeichenfolge in einen definierten Bereich in d bis d+n.

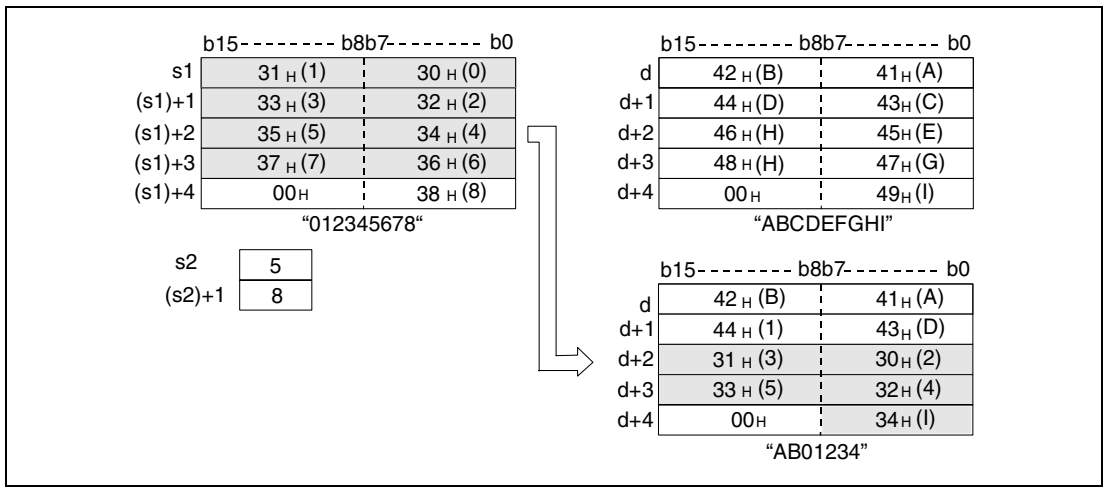
Die erste Adresse des Speicherbereichs in d bis d+n ist in s2 (Array_s2[1]) angegeben und wird vom linken Teil der Zeichenfolge (unteres Byte von d) beginnend gezählt.

Die Länge des zu speichernden Teils ist in s2+1 (Array_s2[2]) angegeben.

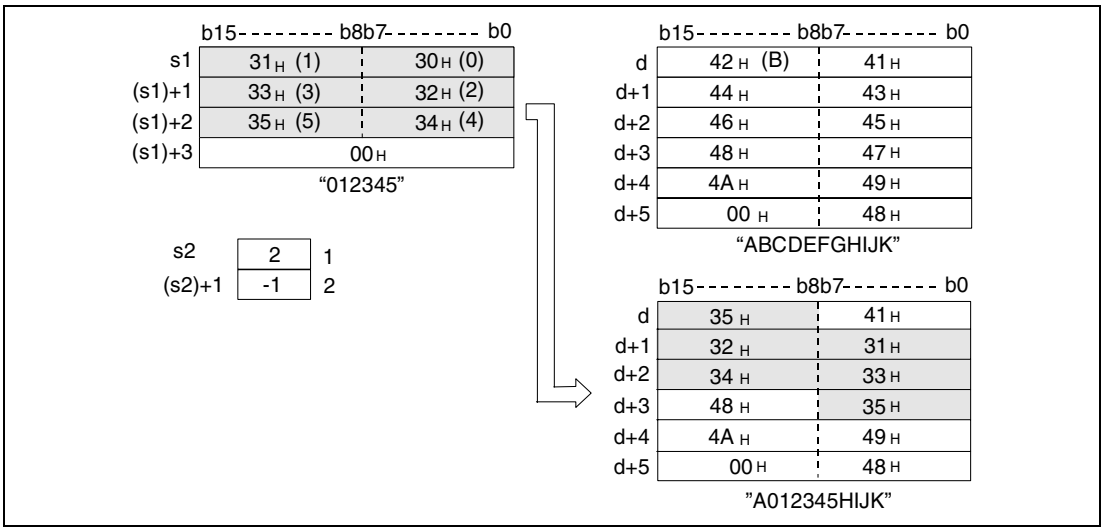


Es wird keine Verarbeitung durchgeführt, wenn die Anzahl der Zeichen in (s2)+1 (Array_s2[2]) Null beträgt.

Wenn die Anzahl der Zeichen, die in (s2)+1 (Array_s2[2]) angegeben ist, außerhalb des ab d angegebenen Speicherbereichs liegt, wird der Rest der Zeichenfolge abgeschnitten. In der folgenden Abbildung werden die Zeichen "35H" bis "37H" nicht gespeichert.



Wenn in (s2)+1 (Array_s2[2]) der Wert -1 angegeben wird, erfolgt die Speicherung aller Zeichen der Zeichenfolge ab s1.



Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

Für die MIDR-Anweisung

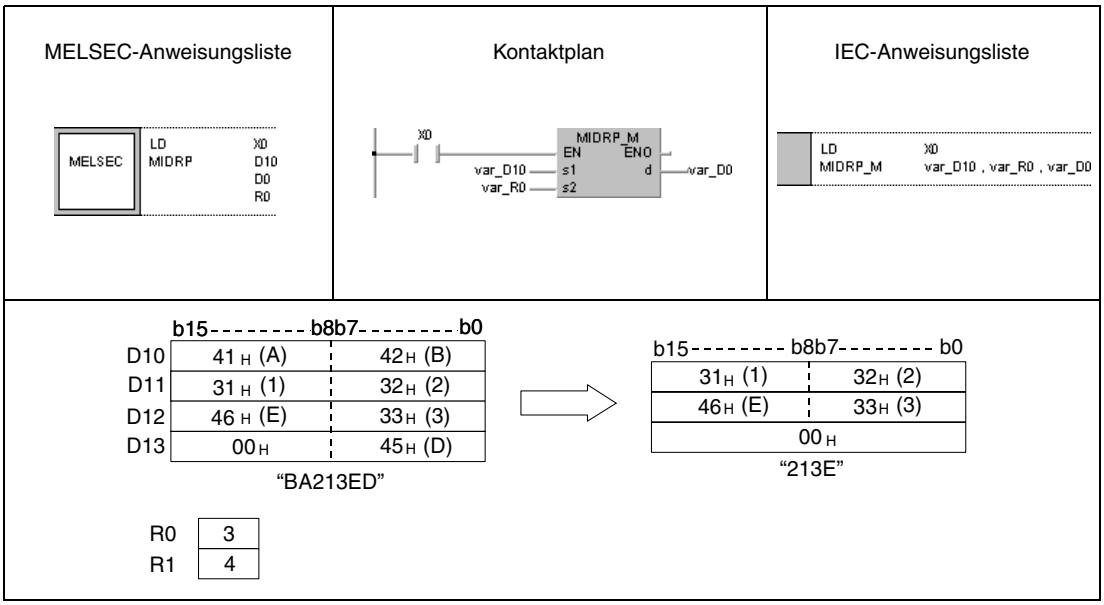
- Die in s2 (Array_s2[1]) angegebene Startadresse der zu speichernden Zeichen liegt außerhalb des Bereichs s1 bis (s1)+n (Fehlercode 4101).
- Die in (s2)+1 (Array_s2[2]) angegebene Anzahl von zu speichernden Zeichen liegt außerhalb des Speicherbereichs d bis d+n (Fehlercode 4101).

Für die MIDW-Anweisung

- Die in s2 (Array_s2[1]) angegebene Startadresse des Speicherbereichs liegt außerhalb des Bereichs d bis d+n (Fehlercode 4101).
- Die in (s2)+1 (Array_s2[2]) angegebene Anzahl von zu speichernden Zeichen liegt außerhalb des Speicherbereichs in s1 bis (s1)+n (Fehlercode 4101).

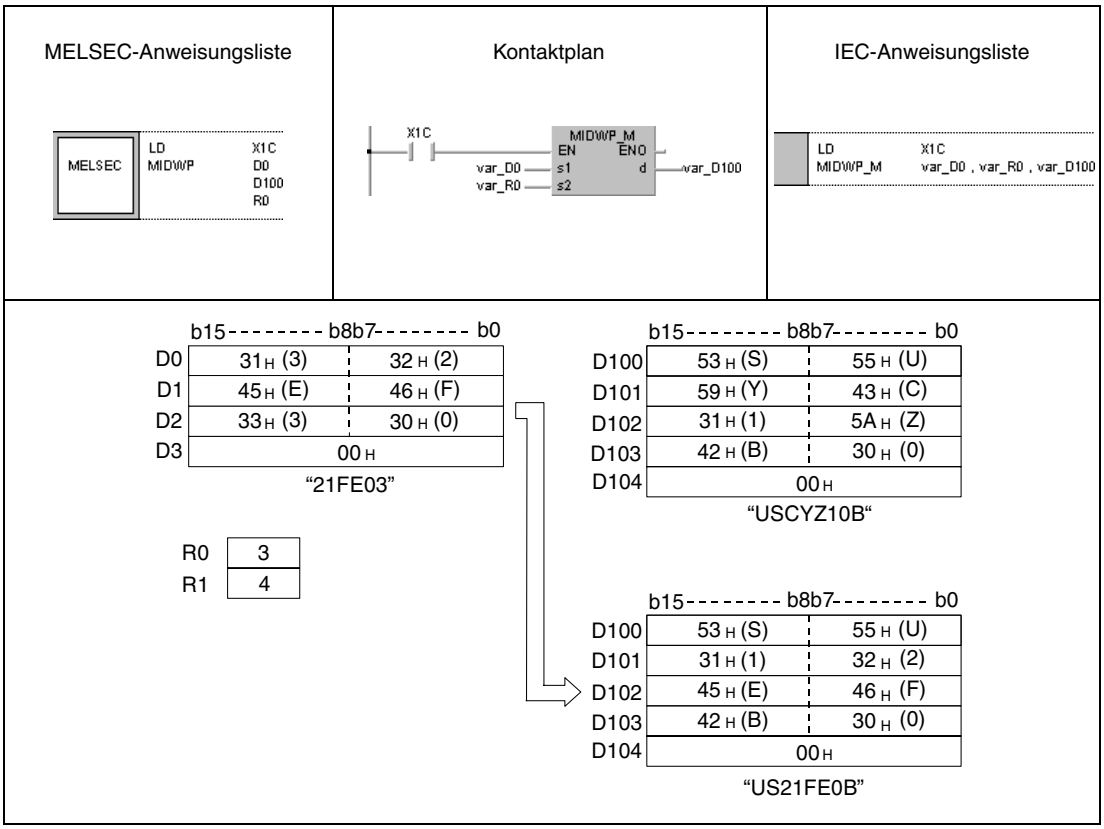
Beispiel 1 MIDRP

Das folgende Programm speichert mit positiver Flanke von X0 die in R1 (var_R0 Array [2]) angegebene Anzahl von Zeichen ab der in R0 (var_R0 Array [1]) angegebenen Position aus D10 bis D13 in D0 bis D2.



Beispiel 2 MIDWP

Das folgende Programm speichert mit positiver Flanke von X1C die ersten mit R1 (var_R0 Array [2]) angegebenen Zeichen aus D0 bis D3 ab der in R0 (var_R0 Array [1]) angegebenen Position in D100 bis D104.



HINWEIS *Diese Programmbeispiele sind ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.11.18 INSTR, INSTRP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●[1]

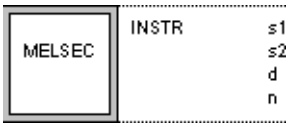
[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

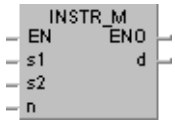
	Operanden										Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sonder-module U□\G□	Index-Register Zn	Konstanten		Andere		
	Bit	Wort		Bit	Wort			K, H (16#)	$	U		
s1	—	●	●	—	—	—	—	—	●	—	SM0	5
s2	—	●	●	—	—	—	—	—	●	—		
d	●	●	●	●	●	●	●	—	—	—		
n	●	●	●	●	●	●	●	●	—	—		

GX IEC Developer

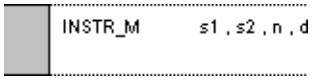
MELSEC-Anweisungsliste



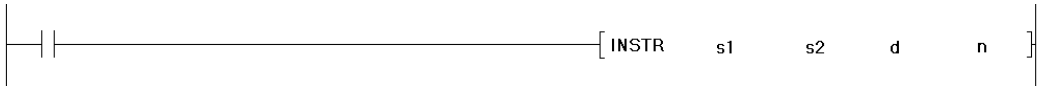
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

Operand	Befehlswert	Datentyp
s1	Erste Adresse der Operanden, in dem die zu suchende Zeichenfolge gespeichert ist.	Zeichenfolge
s2	Erste Adresse der Operanden, in dem die gesuchte Zeichenfolge gespeichert ist.	
d	Erste Adresse, in der das Suchergebnis gespeichert wird.	BIN-16-Bit
n	Startadresse der Suche.	

Funktionsweise

Suche von Zeichenfolgen

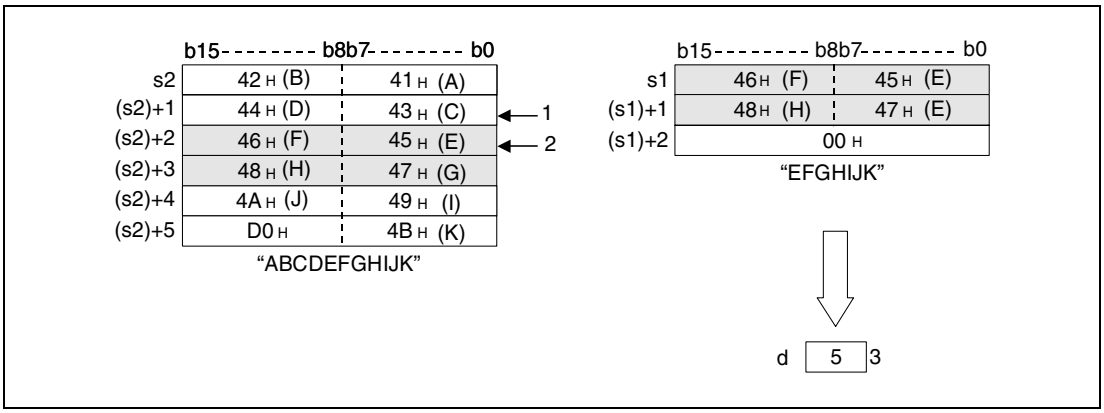
INSTR Zeichenfolgensuche

Die INSTR-Anweisung sucht in der Zeichenfolge in s2 bis (s2)+n die in s1 bis (s1)+n angegebene Zeichenfolge.

Die Suche beginnt mit dem in n angegebenen Zeichen.

Das Zeichen, ab dem die gesuchte Zeichenfolge gefunden wurde, wird in d gespeichert. Das Zeichen wird vom linken Teil der Zeichenfolge (unteres Byte von s2) beginnend gezählt.

Bei n=3



[1] Die Suche beginnt beim 3. Zeichen

[2] Erstes Zeichen der gesuchten Zeichenfolge

[3] Suchergebnis

Wenn keine Zeichenfolge gefunden wurde, wird in d eine Null gespeichert.

Wenn der in n angegebene Wert negativ oder Null ist, wird die Anweisung nicht verarbeitet.

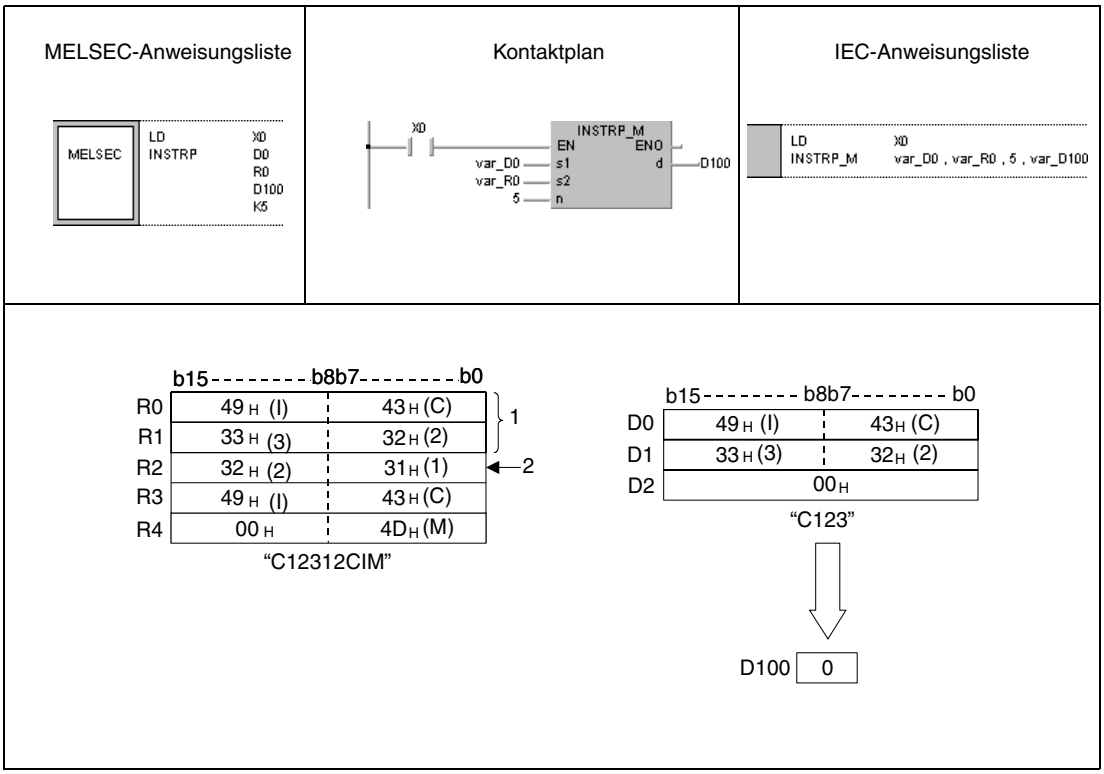
Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die in n angegebene Startadresse der Suche liegt außerhalb des Bereichs s2 bis (s2)+n (Fehlercode 4100).

Beispiel 1 INSTRP

Das folgende Programm sucht mit positiver Flanke von X0 ab R0 nach der in D0 bis D2 angegebenen Zeichenfolge. Die Suche beginnt ab dem 5. Zeichen der Zeichenfolge ab R0. Das Suchergebnis (0) wird in D100 gespeichert.



[1] Dieser Bereich wird nicht durchsucht, da die Suche erst mit dem 5. Zeichen beginnt.

[2] Die Suche beginnt beim 5. Zeichen.

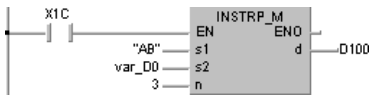
Beispiel 2 INSTRP

Das folgende Programm sucht mit positiver Flanke von X0 ab D0 nach der Zeichenfolge "AB". Die Suche beginnt ab dem 3. Zeichen der Zeichenfolge ab D0. Das Suchergebnis (5) wird in D100 gespeichert.

MELSEC-Anweisungsliste

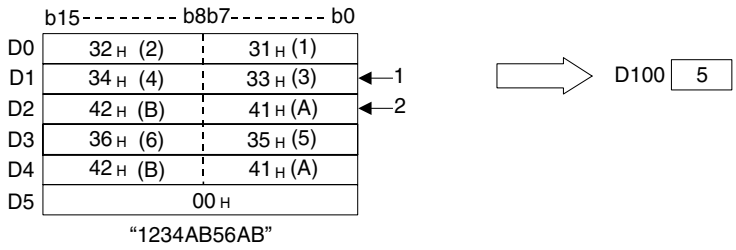
```
MELSEC   LD      X1C
         INSTRP  "AB"
                 D0
                 D100
                 K3
```

Kontaktplan



IEC-Anweisungsliste

```
LD      X1C
INSTRP_M  "AB" , var_D0 , 3 , var_D100
```



[1] Beginn der Suche beim 3. Zeichen.

[2] Die gesuchte Zeichenfolge beginnt beim 5. Zeichen.

HINWEIS *Diese Programmbeispiele sind ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.11.19 EMOD, EMODP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●[1]

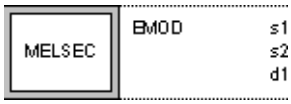
[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

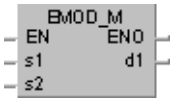
	Operanden										Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sonder-module U□\G□	Index-Register Zn	Konstanten		Andere		
	Bit	Wort		Bit	Wort			K, H (16#)	E			
s1	—	●	●	—	●	●	—	—	●	—	SM0	4
s2	●	●	●	●	●	●	●	●	—	—		
d1	—	●	●	—	—	—	—	—	—	—		

GX IEC Developer

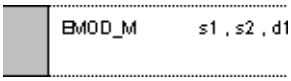
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

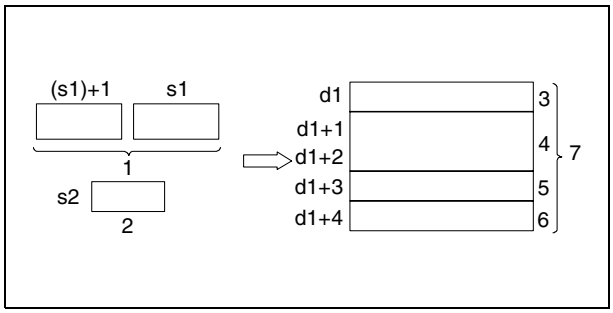
Operand	Befehlswert	Datentyp
s1	Gleitkommazahl (reelle Zahl) oder erste Adresse des Operanden, in dem die Gleitkommazahl gespeichert ist.	reelle Zahl
s2	Anzahl der Stellen, um die das Dezimalkomma nach rechts verschoben wird, oder erste Adresse des Operanden, in dem diese Daten gespeichert sind.	BIN-16-Bit
d1	Erste Adresse des Operanden, in dem die Gleitkommazahl im BCD-Datenformat gespeichert wird.	

Funktions-weise

Gleitkommazahl-Umrechnung in das BCD-Format

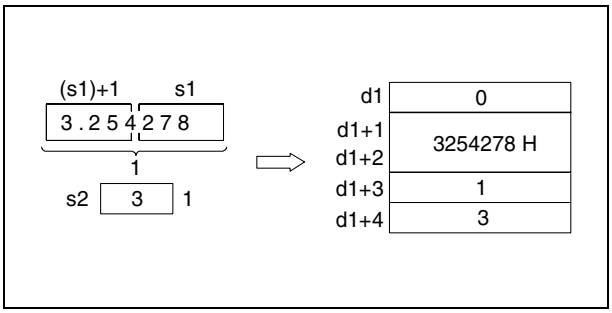
EMOD Umrechnung in das BCD-Format

Die EMOD-Anweisung berechnet aus der Gleitkommazahl (reelle Zahl) in s1 und (s1)+1 unter Berücksichtigung der in s2 angegebenen Kommaverschiebung nach rechts das BCD-Format. Das Ergebnis wird in d1 bis (d1)+4 gespeichert.

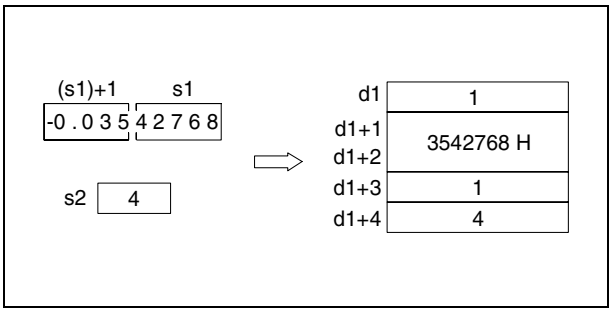


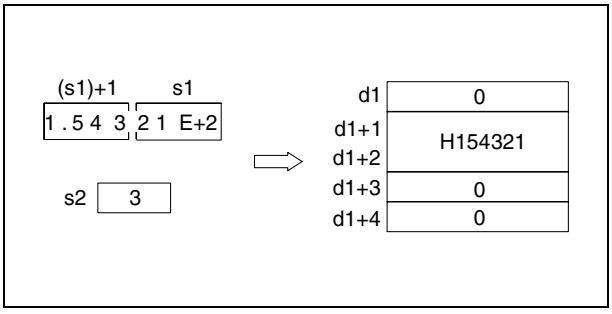
[1] Gleitkommazahl (reelle Zahl)
[2] Verschiebung des Dezimalkommas nach rechts
[3] Vorzeichen (0 = positiv/ 1 = negativ)
[4] 7 BCD-Stellen
[5] Exponentenvorzeichen (0 = positiv/ 1 = negativ)
[6] BCD Exponent (Wertebereich 0 bis 38)
[7] Gleitkommazahl im BCD-Datenformat

In den folgenden Abbildungen sind Umrechnungsbeispiele dargestellt.

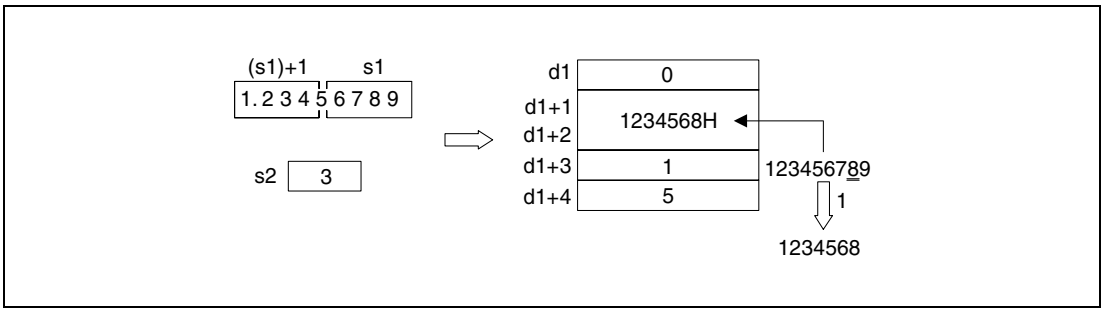


[1]Gleitkommazahl (reelle Zahl)





Die Gleitkommazahl in s1 und (s1)+1 wird auf 7 Stellen gerundet und dann in (d1)+1 und (d1)+2 gespeichert.



[1] aufgerundet

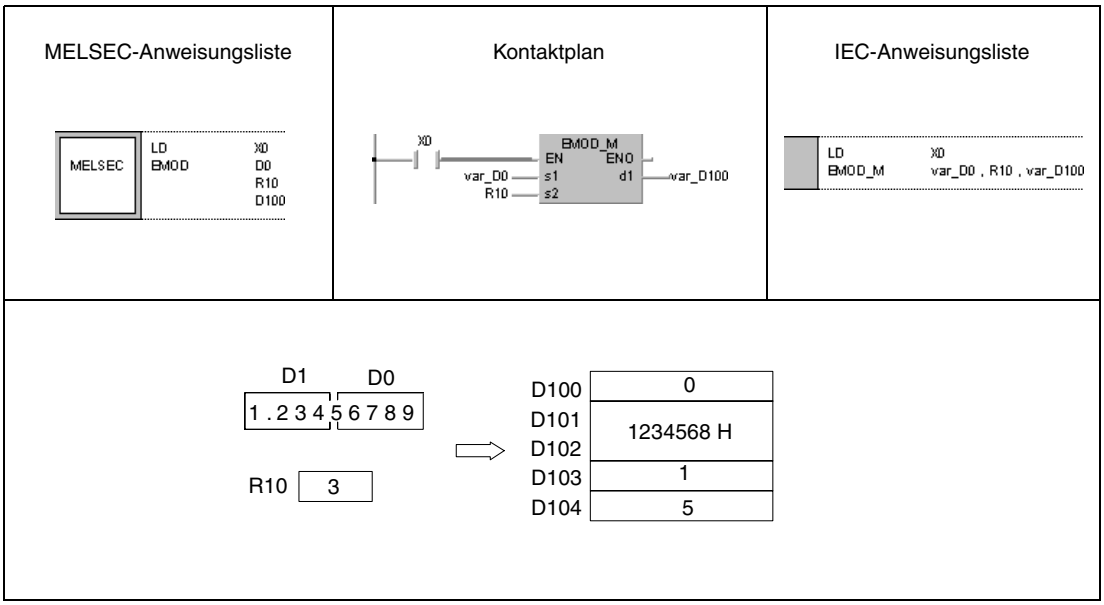
Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die Anzahl der Stellen der Kommaverschiebung s2 liegt außerhalb des Bereichs von 0 bis 7 (Fehlercode 4100).
- Der in d1 bis (d1)+4 eingegebene Wert liegt außerhalb des Speicherbereichs des Operanden (Fehlercode 4101).

Beispiel EMOD

Das folgende Programm rechnet für die Einschaltdauer von X0 die in D0 und D1 angegebene Gleitkommazahl (reelle Zahl) unter Berücksichtigung der in R10 angegebenen Kommaverschiebung um und speichert das Ergebnis in D100 bis D104.



HINWEIS *Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.11.20 EREXP, EREXPP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●[1]

[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

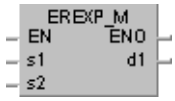
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort				U		
s1	—	●	●	—	—	—	—	—	—	SM0	4
s2	●	●	●	●	●	●	●	●	—		
d1	—	●	●	—	●	●	—	—	—		

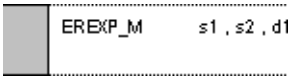
GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

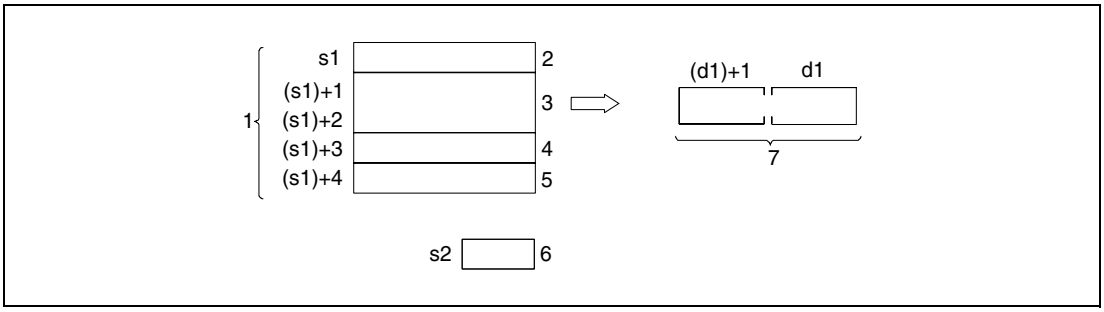
Operand	Befehlswert	Datentyp
s1	Erste Adresse des Operanden, in dem die Gleitkommazahl im BCD-Datenformat gespeichert ist.	BIN-16-Bit
s2	Nachkommastellendaten oder Operand, in dem diese Daten gespeichert sind.	
d1	Operand, in dem die Gleitkommazahl (reelle Zahl) gespeichert wird.	reelle Zahl

Funktions-weise

Gleitkommazahl-Umrechnung in das Dezimal-Format

EREXP Umrechnung in das Dezimal-Format

Die EREXP-Anweisung berechnet aus der Gleitkommazahl im BCD-Format in s1 bis (s1)+4 unter Berücksichtigung der in s2 angegebenen Nachkommastellen das Dezimal-Format der Gleitkommazahl (reelle Zahl). Das Ergebnis wird in d1 und (d1)+1 gespeichert.



[1] Gleitkommazahl im BCD-Datenformat

[2] Vorzeichen (0 = positiv / 1 = negativ)

[3] 7 BCD-Stellen

[4] Exponentenvorzeichen (0 = positiv / 1 = negativ)

[5] BCD Exponent (Wertebereich 0 bis 38)

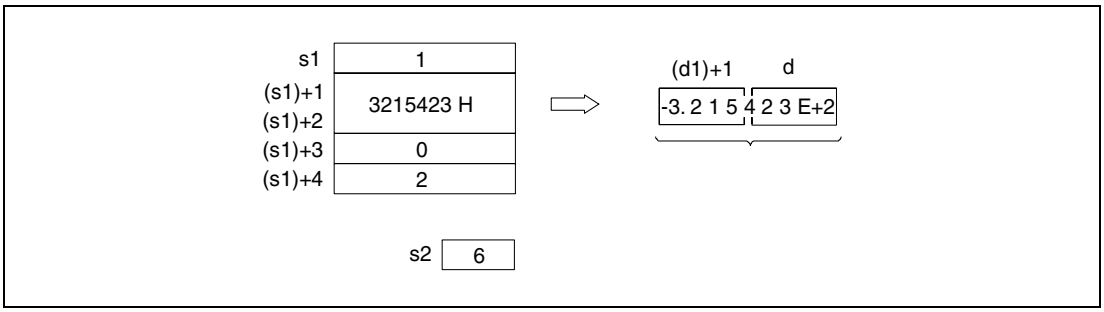
[6] Anzahl der Nachkommastellen (Wertebereich 0 bis 7)

[7] Gleitkommazahl (reelle Zahl)

Das Vorzeichen in s1 und das Vorzeichen des Exponenten in (s1)+3 werden bei positivem Wert als 0 gesetzt. Bei negativem Wert wird eine 1 gesetzt.

Der Wert des BCD-Exponenten (s1)+4 kann zwischen 0 und 38 liegen.

Die Nachkommastellen in s2 können einen Wert zwischen 0 und 7 annehmen.



Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

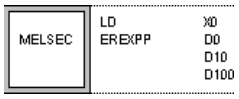
- Die Vorzeichenangabe in s1 hat einen anderen Wert als 0 oder 1 (Fehlercode 4100).
- Die BCD-Daten in (s1)+1 und (s1)+2 haben mehr als 8 Stellen (Fehlercode 4100).
- Das Exponentenvorzeichen in (s1)+3 hat einen anderen Wert als 0 oder 1 (Fehlercode 4100).
- Die Exponentendaten in (s1)+4 liegen außerhalb des Bereichs von 0 bis 38 (Fehlercode 4100).
- Die Anzahl der Nachkommastellen in s2 liegt außerhalb des Bereichs von 0 bis 7 (Fehlercode 4101).

Beispiel

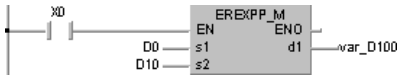
EREXPP

Das folgende Programm rechnet mit positiver Flanke von X0 die in D0 bis D4 angegebene Gleitkommazahl im BCD-Format unter Berücksichtigung der in D10 angegebenen Nachkommastellen in eine Gleitkommazahl (reelle Zahl) im Dezimal-Format um und speichert das Ergebnis in D100 und D101.

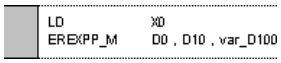
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



7.12 Sonderfunktionen

Funktion	MELSEC-Anweisung im MELSEC-EDITOR	MELSEC-Anweisung im IEC-EDITOR
Sinusberechnung	SIN	SIN_MD
		SIN_E_MD
	SINP	SIN_P_MD
		SIN_P_E_MD
Cosinusberechnung	COS	COS_MD
		COS_E_MD
	COSP	COS_P_MD
		COS_P_E_MD
Tangensberechnung	TAN	TAN_MD
		TAN_E_MD
	TANP	TAN_P_MD
		TAN_P_E_MD
Arcussinusberechnung	ASIN	ASIN_MD
		ASIN_E_MD
	ASINP	ASIN_P_MD
		ASIN_P_E_MD
Arcuscosinusberechnung	ACOS	ACOS_MD
		ACOS_E_MD
	ACOSP	ACOS_P_MD
		ACOS_P_E_MD
Arcustangensberechnung	ATAN	ATAN_MD
		ATAN_E_MD
	ATANP	ATAN_P_MD
		ATAN_P_E_MD
Umrechnung von Grad in Radiant	RAD	RAD_MD
		RAD_E_MD
	RADP	RAD_P_MD
		RAD_P_E_MD
Umrechnung von Radiant in Grad	DEG	DEG_MD
		DEG_E_MD
	DEGP	DEG_P_MD
		DEG_P_E_MD
Quadratwurzelberechnung	SQR	SQR_MD
		SQR_E_MD
	SQRP	SQR_P_MD
		SQR_P_E_MD
Gleitkommazahlen als Exponent von e	EXP	EXP_MD
		EXP_E_MD
	EXPP	EXP_P_MD
		EXP_P_E_MD

Funktion	MELSEC-Anweisung im MELSEC-EDITOR	MELSEC-Anweisung im IEC-EDITOR
Logorithmus-naturalis-Berechnung	LOG	LOG_MD
		LOG_E_MD
	LOGP	LOG_P_MD
		LOG_P_E_MD
Generierung von Zufallszahlen	RND	RND_M
	RNDP	RNDP_M
Aktualisierung von Zufallszahlenserien	SRND	SRND_M
	SRNDP	SRNDP_M
Quadratwurzelberechnung aus 4-stelligen BCD-Daten	BSQR	BSQR_MD
		BSQR_K_MD
	BSQRP	BSQR_P_MD
		BSQR_K_P_MD
Quadratwurzelberechnung aus 8-stelligen BCD-Daten	BDSQR	BDSQR_MD
		BDSQR_K_MD
	BDSQRP	BDSQR_P_MD
		BDSQR_K_P_MD
Sinusberechnung mit BCD-Daten	BSIN	BSIN_MD
		BSIN_K_MD
	BSINP	BSIN_P_MD
		BSIN_K_P_MD
Cosinusberechnung mit BCD-Daten	BCOS	BCOS_MD
		BCOS_K_MD
	BCOSP	BCOS_P_MD
		BCOS_K_P_MD
Tangensberechnung mit BCD-Daten	BTAN	BTAN_MD
		BTAN_K_MD
	BTANP	BTAN_P_MD
		BTAN_K_P_MD
Arcussinusberechnung mit BCD-Daten	BASIN	BASIN_MD
	BASINP	BASIN_P_MD
Arcuscosinusberechnung mit BCD-Daten	BACOS	BACOS_MD
	BACOSP	BACOS_P_MD
Arcustangensberechnung mit BCD-Daten	BATAN	BATAN_MD
	BATANP	BATAN_P_MD

HINWEIS *Nutzen Sie in den IEC-Editoren die IEC-Anweisungen.*

7.12.1 SIN, SINP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●[2]

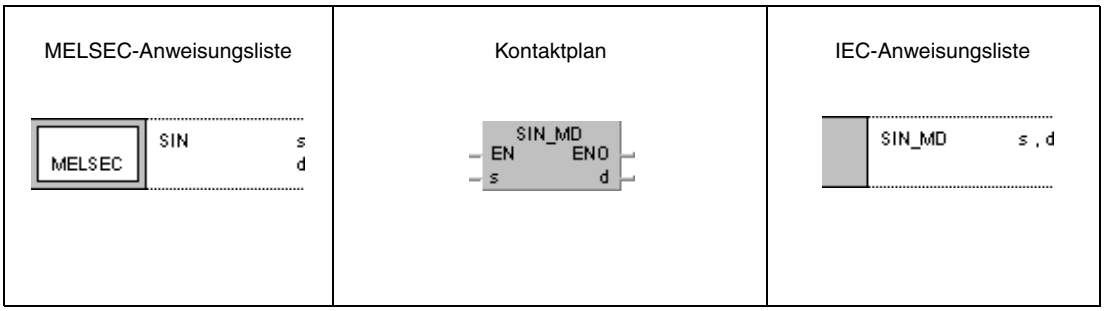
[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

[2] Nicht für Q00JCPU, Q00CPU und Q01CPU

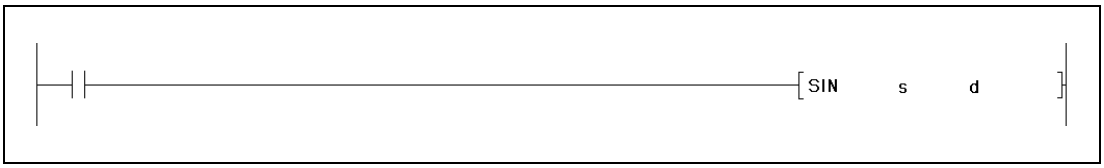
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten E	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	●	●	—	●	—	—	3
d	—	●	●	—	●	●	—	—	—		

GX IEC Developer



GX Developer



Variablen

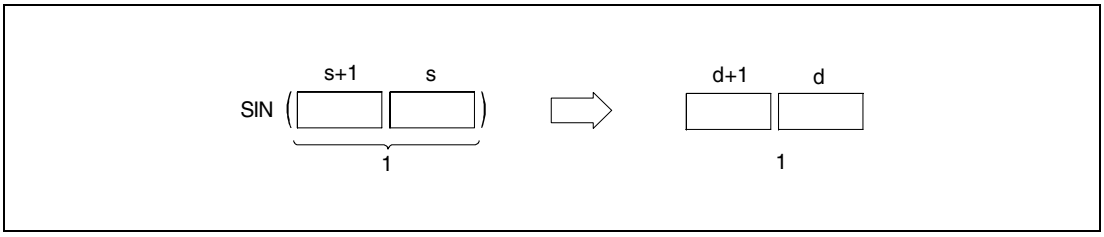
Operand	Befehlswert	Datentyp
s	Erste Adresse des Operanden, in dem die Winkeldaten, die zur Ausführung der SIN-Anweisung (Sinus) notwendig sind, gespeichert sind.	reelle Zahl
d	Erste Adresse des Operanden, in dem das Ergebnis gespeichert wird.	

Funktions-weise

Sinusberechnung

SIN Sinusberechnung mit Gleitkommazahlen

Die SIN-Anweisung berechnet den Sinuswert aus den Winkeldaten in s und s+1 und speichert das Ergebnis in d und d+1.



[1] Gleitkommazahl (reelle Zahl)

Die Winkelangaben in s und s+1 müssen im Bogenmaß (Grad x π/180) eingegeben werden. Die Umwandlung zwischen Grad und Radiant wird in den Anweisungen RAD und DEG näher erläutert.

Fehler-quellen

Wenn bei Verwendung der Q4AR-CPU die unter s angegebene Adresse -0 enthält und der Sondermerker SM707 nicht gesetzt ist, wird der Fehlercode 4100 ausgegeben.

Beispiel SIN

Das folgende Programm berechnet für die Einschaltdauer von SM400 den Sinuswert, basierend auf den 4-stelligen BCD-Winkelangaben an X20 bis X2F, und speichert das Ergebnis als Gleitkommazahl (reelle Zahl) in D0 und D1.

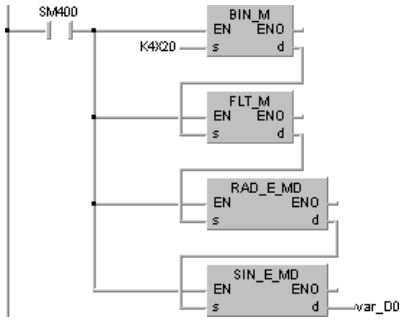
MELSEC-Anweisungsliste

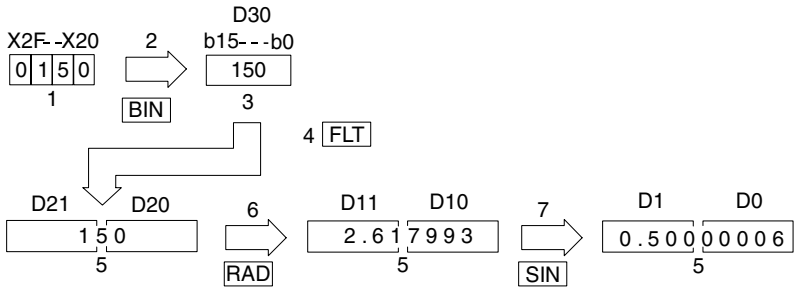
```
MELSEC
        LD      SM400
        BIN     K4X20
                D30
        FLT     D30
                D20
        RAD     D20
                D10
        SIN     D10
                D0
```

IEC-Anweisungsliste

```
LD          SM400
BIN_M       K4X20 , D30
FLT_M       D30 , var_D20
RAD_E_MD    var_D20 , var_D10
SIN_E_MD    var_D10 , var_D0
```

Kontaktplan





[1] BCD-Wert

[2] Umwandlung in das BIN-Format

[3] BIN-Wert

[4] Umwandlung in das Gleitkommaformat

[5] Gleitkommazahl (reelle Zahl)

[6] Umwandlung ins Bogenmaß

[7] Berechnung des Sinuswertes

HINWEIS *Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisa-tionseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.12.2 COS, COSP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●[2]

[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

[2] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

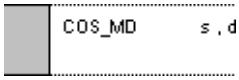
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten E	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	●	●	—	●	—	—	3
d	—	●	●	—	●	●	—	—	—		

GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan

IEC-Anweisungsliste



GX Developer

Variablen

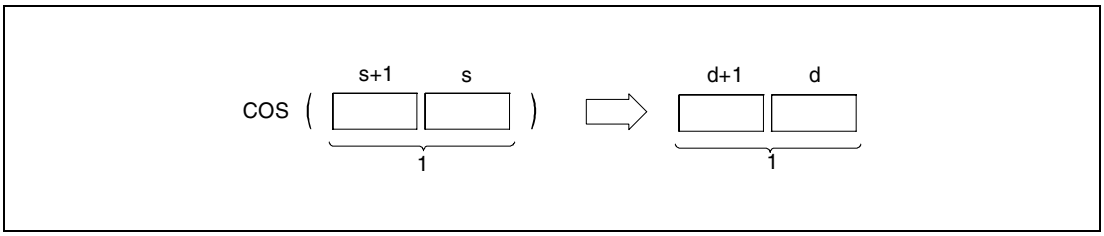
Operand	Befehlswert	Datentyp
s	Erste Adresse des Operanden, in dem die Winkeldaten, die zur Ausführung der COS-Anweisung (Cosinus) notwendig sind, gespeichert sind.	reelle Zahl
d	Erste Adresse des Operanden, in dem das Ergebnis gespeichert wird.	

Funktions-weise

Cosinusberechnung

COS Cosinusberechnung mit Gleitkommazahlen

Die COS-Anweisung berechnet den Cosinuswert aus den Winkeldaten in s und s+1 und speichert das Ergebnis in d und d+1.



[1] Gleitkommazahl (reelle Zahl)

Die Winkelangaben in s und s+1 müssen im Bogenmaß (Grad x π/180) eingegeben werden. Die Umwandlung zwischen Grad und Radiant wird in den Anweisungen RAD und DEG näher erläutert.

Fehler-quellen

Wenn bei Verwendung der Q4AR-CPU die unter s angegebene Adresse -0 enthält und der Sondermerker SM707 nicht gesetzt ist, wird der Fehlercode 4100 ausgegeben.

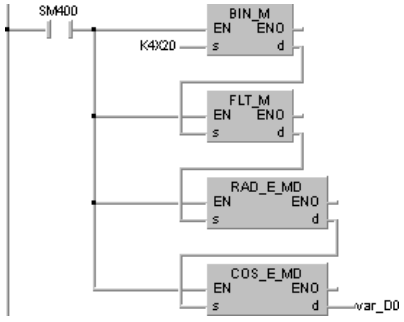
Beispiel COS

Das folgende Programm berechnet für die Einschaltdauer von SM400 den Cosinuswert, basierend auf den 4-stelligen BCD-Winkelangaben an X20 bis X2F, und speichert das Ergebnis als Gleitkommazahl (reelle Zahl) in D0 und D1.

MELSEC-Anweisungsliste

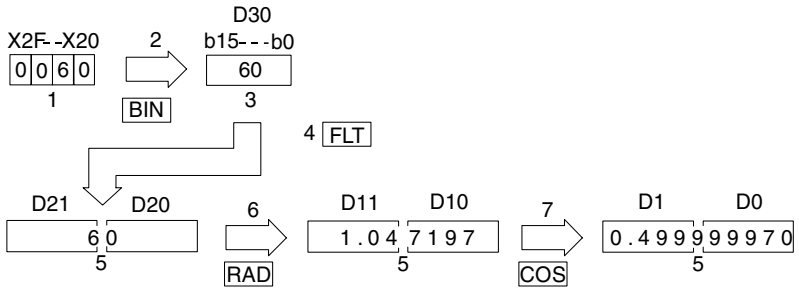
```
MELSEC
        LD      SM400
        BIN     K4X20
                D30
        FLT     D30
                D20
        RAD     D20
                D10
        COS     D10
                D0
```

Kontaktplan



IEC-Anweisungsliste

```
        LD      SM400
        BIN_M   K4X20 , D30
        FLT_M   D30 , var_D20
        RAD_E_MD  var_D20 , var_D10
        COS_E_MD  var_D10 , var_D0
```



[1] BCD-Wert

[2] Umwandlung ins BIN-Format

[3] Binärwert

[4] Umwandlung ins Gleitkommaformat

[5] Gleitkommazahl (reelle Zahl)

[6] Umrechnung in Radianten

[7] Berechnung des Cosinuswertes

HINWEIS *Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.12.3 TAN, TANP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●[2]

[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

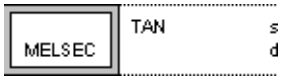
[2] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

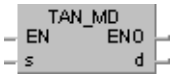
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten E	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	●	●	—	●	—	SM0	3
d	—	●	●	—	●	●	—	—	—		

GX IEC Developer

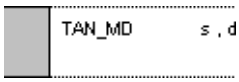
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

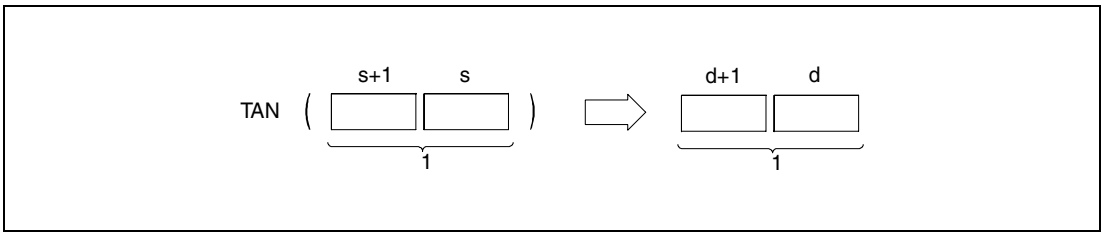
Operand	Befehlswert	Datentyp
s	Erste Adresse des Operanden, in dem die Winkeldaten, die zur Ausführung der TAN-Anweisung (Tangens) notwendig sind, gespeichert sind.	reelle Zahl
d	Erste Adresse des Operanden, in dem das Ergebnis gespeichert wird.	

Funktions-weise

Tangensberechnung

TAN Tangensberechnung mit Gleitkommazahlen

Die TAN-Anweisung berechnet den Tangenswert aus den Winkeldaten in s und s+1 und speichert das Ergebnis in d und d+1.



[1] Gleitkommazahl (reelle Zahl)

Die Winkelangaben in s und s+1 müssen im Bogenmaß (Grad x π/180) eingegeben werden. Die Umwandlung zwischen Grad und Radiant wird in den Anweisungen RAD und DEG näher erläutert.

Wenn die Winkelangaben in s und s+1 die Werte π/2 rad oder (3/2)xπ rad annehmen, wird eine Fehlermeldung in der Bogenmaßberechnung generiert.

Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Das Verarbeitungsergebnis nimmt den Wert Null an oder liegt nicht zwischen $\pm 2^{-127}$ und $\pm 2^{129}$ (Fehlercode 4100).
- Wenn bei Verwendung der Q4AR-CPU die unter s angegebene Adresse -0 enthält und der Sondermerker SM707 nicht gesetzt ist, wird der Fehlercode 4100 ausgegeben.

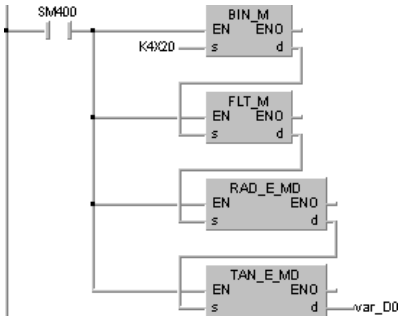
Beispiel TAN

Das folgende Programm berechnet mit positiver Flanke von SM400 den Tangenswert, basierend auf den 4-stelligen BCD-Winkelangaben an X20 bis X2F, und speichert das Ergebnis als Gleitkommazahl (reelle Zahl) in D0 und D1.

MELSEC-Anweisungsliste

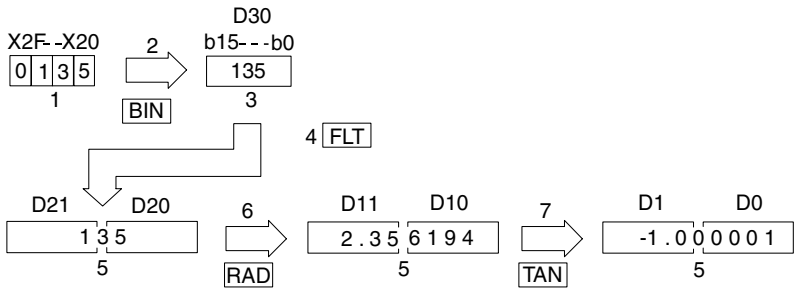
```
MELSEC
        LD      SM400
        BIN     K4X20
                D30
        FLT     D30
                D20
        RAD     D20
                D10
        TAN     D10
                D0
```

Kontaktplan



IEC-Anweisungsliste

```
LD          SM400
BIN_M       K4X20 , D30
FLT_M       D30 , var_D20
RAD_E_MD    var_D20 , var_D10
TAN_E_MD    var_D10 , var_D0
```



[1] BCD-Wert

[2] Umwandlung ins BIN-Format

[3] Binärwert

[4] Umwandlung ins Gleitkommaformat

[5] Gleitkommazahl (reelle Zahl)

[6] Umrechnung in Radianten

[7] Berechnung des Tangenswertes

HINWEIS *Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.12.4 ASIN, ASINP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●[2]

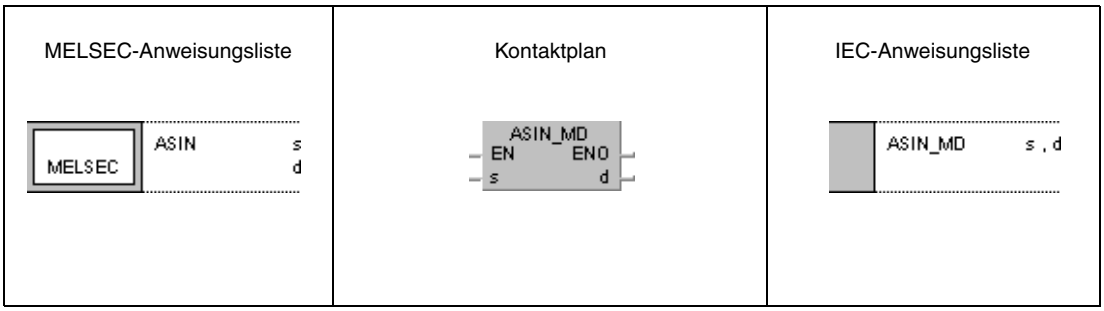
[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

[2] Nicht für Q00JCPU, Q00CPU und Q01CPU

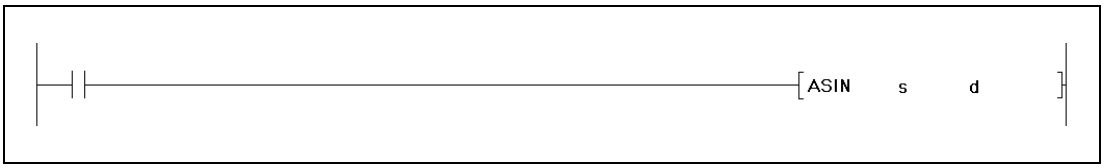
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten E	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	●	●	—	●	—	SM0	3
d	—	●	●	—	●	●	—	—	—		

GX IEC Developer



GX Developer



Variablen

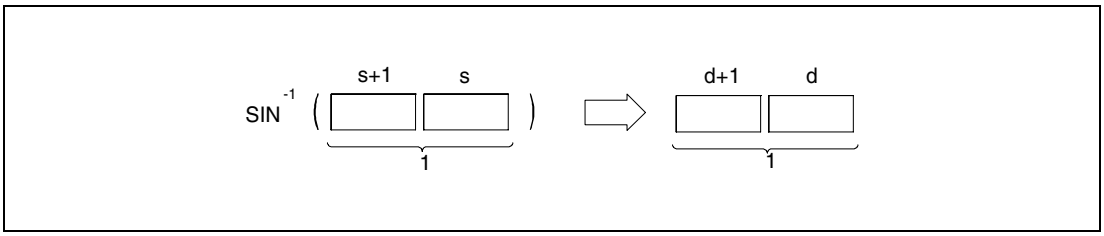
Operand	Befehlswert	Datentyp
s	Erste Adresse des Operanden, in dem der Sinuswert, der zur Berechnung des Arcussinus notwendig ist, gespeichert ist.	reelle Zahl
d	Erste Adresse des Operanden, in dem das Ergebnis gespeichert wird.	

Funktions-weise

Arcussinusberechnung

ASIN Arcussinusberechnung mit Gleitkommazahlen

Die ASIN-Anweisung berechnet aus dem Sinuswert in s und s+1 die Winkeldaten und speichert das Ergebnis in d und d+1.



[1] Gleitkommazahl (reelle Zahl)

Der Sinuswert in s und s+1 kann sich innerhalb des Wertebereichs von -1 bis 1 bewegen.

Die Winkelangaben in d und d+1 müssen im Bogenmaß (Grad x π/180) eingegeben werden. Die Umwandlung zwischen Grad und Radiant wird in den Anweisungen RAD und DEG näher erläutert.

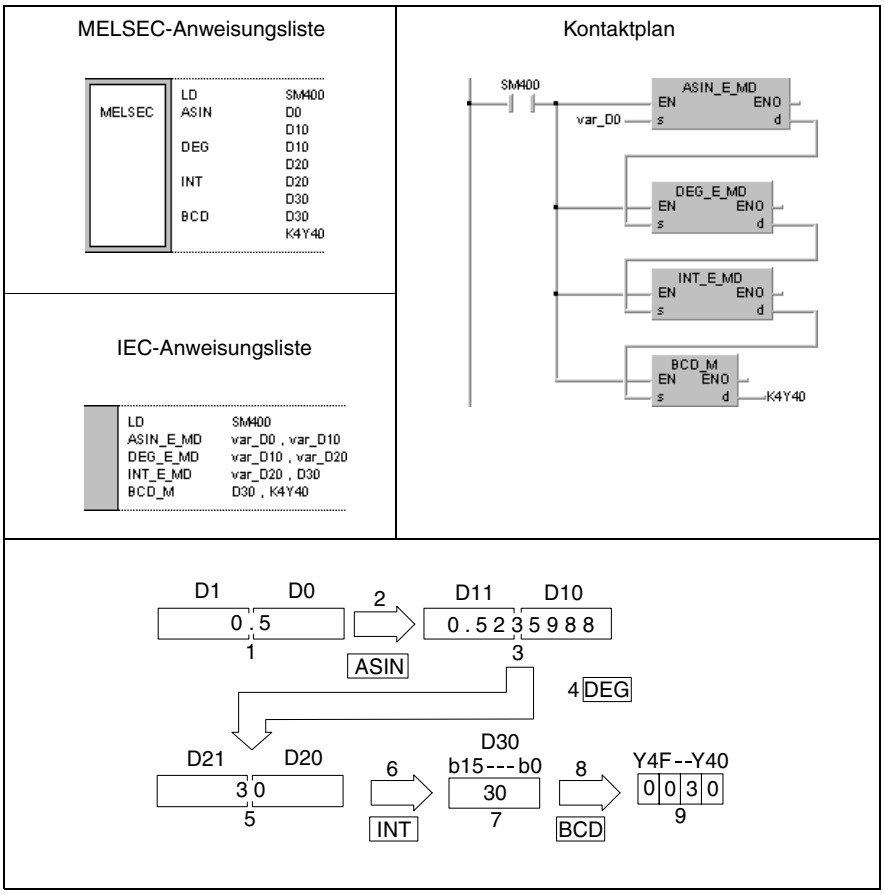
Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Der Wert in s und s+1 liegt außerhalb des Wertebereichs von -1 bis 1 (Fehlercode 4100).
- Wenn bei Verwendung der Q4AR-CPU die unter s angegebene Adresse -0 enthält und der Sondermerker SM707 nicht gesetzt ist, wird der Fehlercode 4100 ausgegeben.

Beispiel ASIN

Das folgende Programm berechnet für die Einschaltdauer von SM400 den Arcussinuswert, basierend auf der Gleitkommazahl (reelle Zahl) in D0 und D1, und gibt das berechnete Bogenmaß an Y20 bis Y4F als 4-stelligen BCD-Wert aus.



[1] Gleitkommazahl (reelle Zahl)
[2] Arcussinus-Berechnung
[3] Gleitkommazahl (reelle Zahl)
[4] Umwandlung der Winkelangaben
[5] Gleitkommazahl (reelle Zahl)
[6] Umwandlung ins BIN-Format
[7] Binärwert
[8] Umwandlung ins BCD-Format
[9] BCD-Wert

HINWEIS *Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.12.5 ACOS, ACOSP

CPU

AnS	AnN	AnA(S)	AnU	Qn(S), Q4AR	System Q
		●[1]	●[1]	●	●[2]

[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

[2] Nicht für Q00JCPU, Q00CPU und Q01CPU

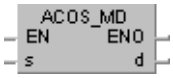
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten E	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	●	●	—	●	—	SM0	3
d	—	●	●	—	●	●	—	—	—		

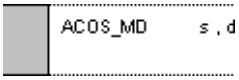
GX IEC Developer

MELSEC-Anweisungsliste

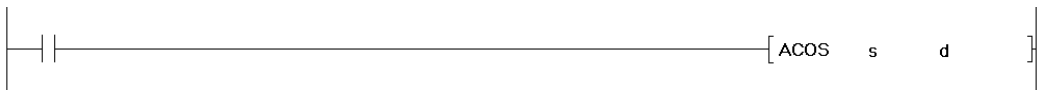
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

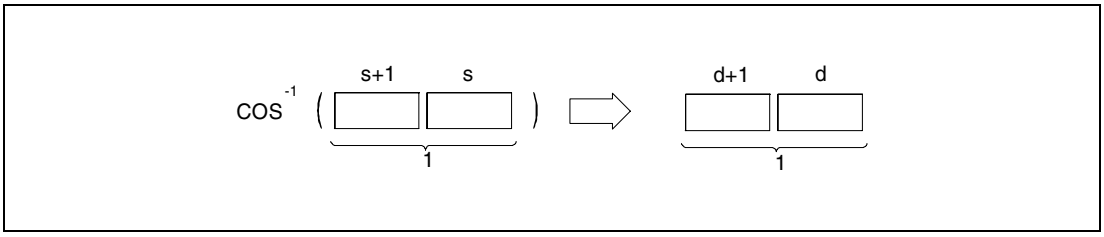
Operand	Befehlswert	Datentyp
s	Erste Adresse des Operanden, in dem der Cosinuswert, der zur Berechnung des Arcuscosinus notwendig ist, gespeichert ist.	reelle Zahl
d	Erste Adresse des Operanden, in dem das Ergebnis gespeichert wird.	

Funktions-weise

Arcuscosinusberechnung

ACOS Arcuscosinusberechnung mit Gleitkommazahlen

Die ACOS-Anweisung berechnet aus dem Cosinuswert in s und s+1 die Winkeldaten und speichert das Ergebnis in d und d+1.



[1] Gleitkommazahl (reelle Zahl)

Der Cosinuswert in s und s+1 kann sich innerhalb des Wertebereichs von -1 bis 1 bewegen.

Die Winkelangaben in d und d+1 müssen im Bogenmaß (Grad x π/180) eingegeben werden. Die Umwandlung zwischen Grad und Radiant wird in den Anweisungen RAD und DEG näher erläutert.

Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Der Wert in s und s+1 liegt außerhalb des Wertebereichs von -1 bis 1 (Fehlercode 4100).
- Wenn bei Verwendung der Q4AR-CPU die unter s angegebene Adresse -0 enthält und der Sondermerker SM707 nicht gesetzt ist, wird der Fehlercode 4100 ausgegeben.

Beispiel ACOS

Das folgende Programm berechnet für die Einschaltdauer von SM400 den Arcuscosinuswert, basierend auf der Gleitkommazahl (reelle Zahl) in D0 und D1, und gibt das berechnete Bogenmaß an Y20 bis Y4F als 4-stelligen BCD-Wert aus.

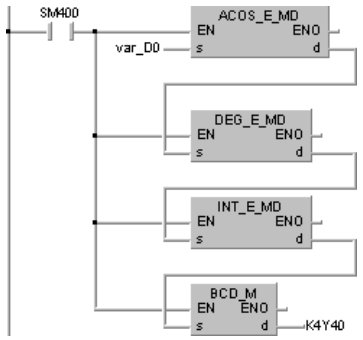
MELSEC-Anweisungsliste

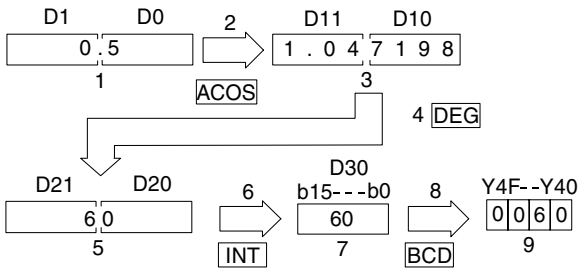
```
MELSEC   LD     SM400
         ACOS   D0
                D10
         DEG    D10
                D20
         INT    D20
                D30
         BCD    D30
                K4Y40
```

IEC-Anweisungsliste

```
LD          SM400
ACOS_E_MD   var_D0 , var_D10
DEG_E_MD    var_D10 , var_D20
INT_E_MD    var_D20 , D30
BCD_M       D30 , K4Y40
```

Kontaktplan





[1] Gleitkommazahl (reelle Zahl)
[2] Arcuscosinus-Berechnung
[3] Gleitkommazahl (reelle Zahl)
[4] Umwandlung der Winkelangaben
[5] Geitkommazahl (reelle Zahl)
[6] Umwandlung ins BIN-Format
[7] Binärwert
[8] Umwandlung ins BCD-Format
[9] BCD-Wert

HINWEIS *Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.12.6 ATAN, ATANP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●[2]

[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

[2] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

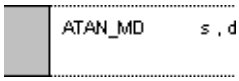
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten E	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	●	●	—	●	—	—	3
d	—	●	●	—	●	●	—	—	—		

GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan

IEC-Anweisungsliste



GX Developer

Variablen

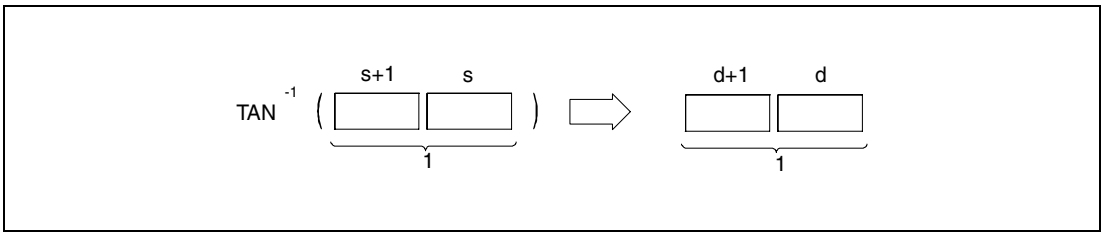
Operand	Befehlswert	Datentyp
s	Erste Adresse des Operanden, in dem der Tangenswert, der zur Berechnung des Arcustangens notwendig ist, gespeichert ist.	reelle Zahl
d	Erste Adresse des Operanden, in dem das Ergebnis gespeichert wird.	

Funktions-weise

Arcustangensberechnung

ATAN Arcustangensberechnung mit Gleitkommazahlen

Die ATAN-Anweisung berechnet aus dem Tangenswert in s und s+1 die Winkeldaten und speichert das Ergebnis in d und d+1.



[1] Gleitkommazahl (reelle Zahl)

Die Winkelangaben in d und d+1 müssen im Bogenmaß (Grad x π/180) eingegeben werden. Die Umwandlung zwischen Grad und Radiant wird in den Anweisungen RAD und DEG näher erläutert.

Fehler-quellen

Wenn bei Verwendung der Q4AR-CPU die unter s angegebene Adresse -0 enthält und der Sondermerker SM707 nicht gesetzt ist, wird der Fehlercode 4100 ausgegeben.

Beispiel ATAN

Das folgende Programm berechnet für die Einschaltdauer von SM400 den Arcustangenswert, basierend auf der Gleitkommazahl (reelle Zahl) in D0 und D1, und gibt das berechnete Bogenmaß an Y20 bis Y4F als 4-stelligen BCD-Wert aus.

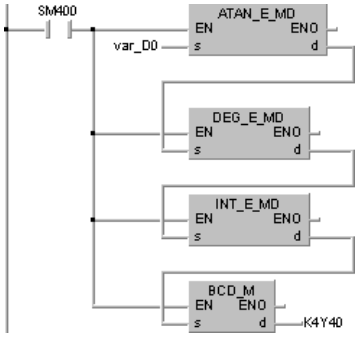
MELSEC-Anweisungsliste

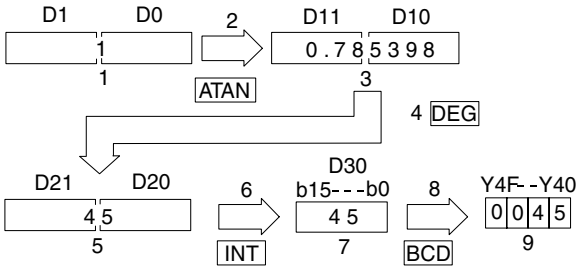
```
MELSEC   LD     SM400
         ATAN   D0
                D10
         DEG    D10
                D20
         INT    D20
                D30
         BCD    D30
                K4Y40
```

IEC-Anweisungsliste

```
LD          SM400
ATAN_E_MD   var_D0 , var_D10
DEG_E_MD    var_D10 , var_D20
INT_E_MD    var_D20 , D30
BCD_M       D30 , K4Y40
```

Kontaktplan





[1] Gleitkommazahl (reelle Zahl)
[2] Arcustangens-Berechnung
[3] Gleitkommazahl (reelle Zahl)
[4] Umwandlung der Winkelangaben
[5] Geitkommazahl (reelle Zahl)
[6] Umwandlung ins BIN-Format
[7] Binärwert
[8] Umwandlung ins BCD-Format
[9] BCD-Wert

HINWEIS *Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.12.7 RAD, RADP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●[2]

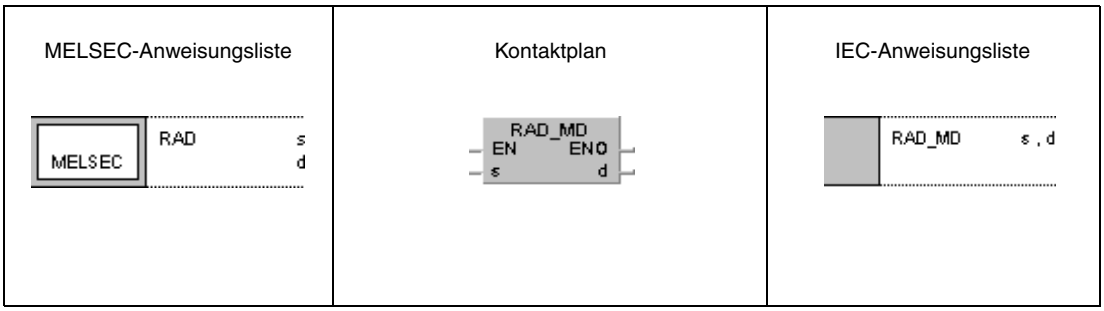
[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

[2] Nicht für Q00JCPU, Q00CPU und Q01CPU

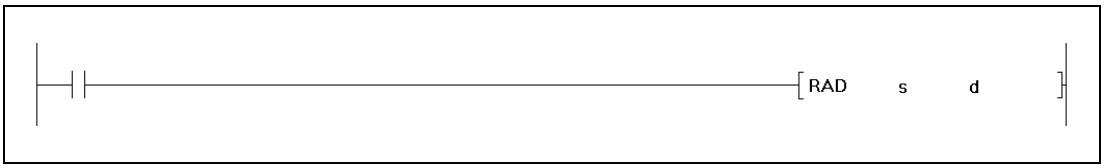
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten E	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	●	●	—	●	—	—	3
d	—	●	●	—	●	●	—	—	—		

GX IEC Developer



GX Developer



Variablen

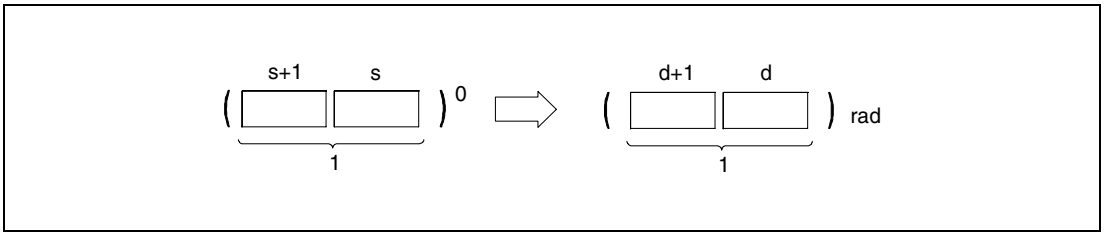
Operand	Befehlswert	Datentyp
s	Erste Adresse des Operanden, in dem die Winkelgrade gespeichert sind, die in Radianten umgerechnet werden soll.	reelle Zahl
d	Erste Adresse des Operanden, in dem das Ergebnis der Umrechnung gespeichert wird.	

Funktions-weise

Umrechnung von Grad in Radiant

RAD Umrechnungsanweisung

Die RAD-Anweisung berechnet aus der Gradangabe (°) in s und s+1 den entsprechenden Radiantenwert (rad) und speichert das Ergebnis in d und d+1.



[1] Gleitkommazahl (reelle Zahl)

Die Umrechnung von Grad in Radiant wird nach folgender Formel durchgeführt:

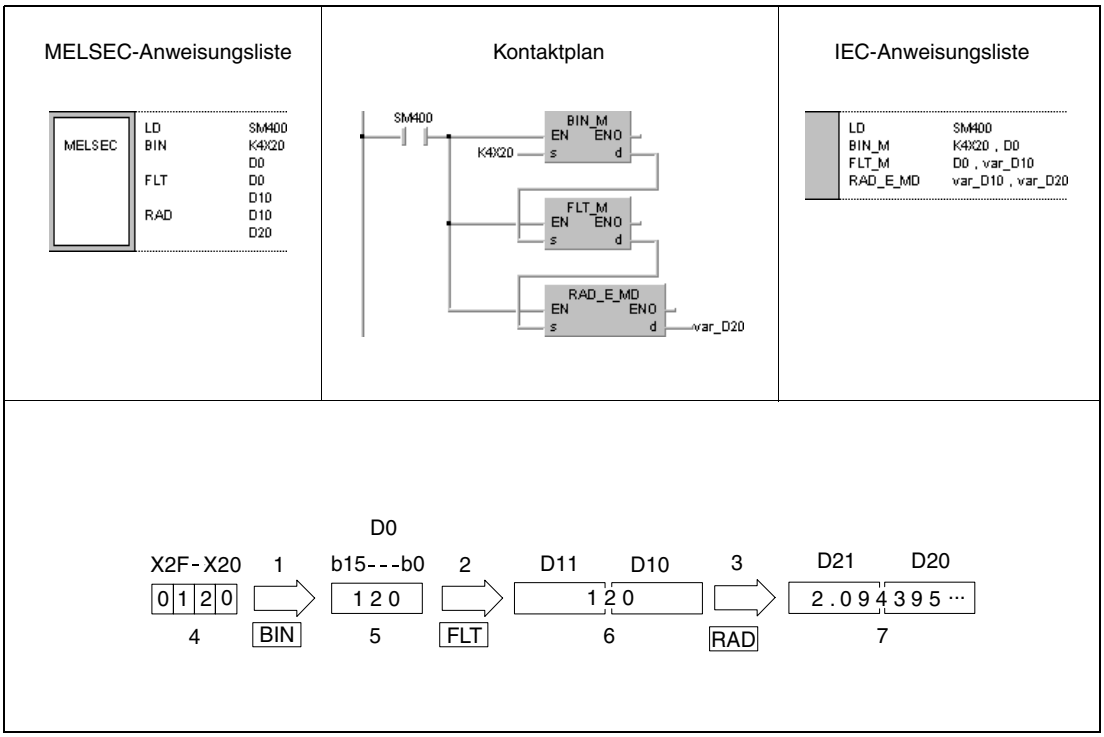
Radiantenwert = Gradwert x π / 180

Fehler-quellen

Wenn bei Verwendung der Q4AR-CPU die unter s angegebene Adresse -0 enthält und der Sondermerker SM707 nicht gesetzt ist, wird der Fehlercode 4100 ausgegeben.

Beispiel RAD

Das folgende Programm berechnet für die Einschaltdauer von SM400 aus dem Gradwert des 4-stelligen BCD-Werts an X20 bis X2F einen Radiantenwert und speichert das Ergebnis als Gleitkommazahl in D20 und D21.



[1] Umwandlung ins BIN-Format

[2] Umwandlung ins Gleitkommaformat

[3] Umwandlung in Radianten

[4] BCD-Wert

[5] Binärwert

[6] Gleitkommazahl (reelle Zahl)

[7] Gleitkommazahl (reelle Zahl)

HINWEIS *Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.12.8 DEG, DEGP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●[2]

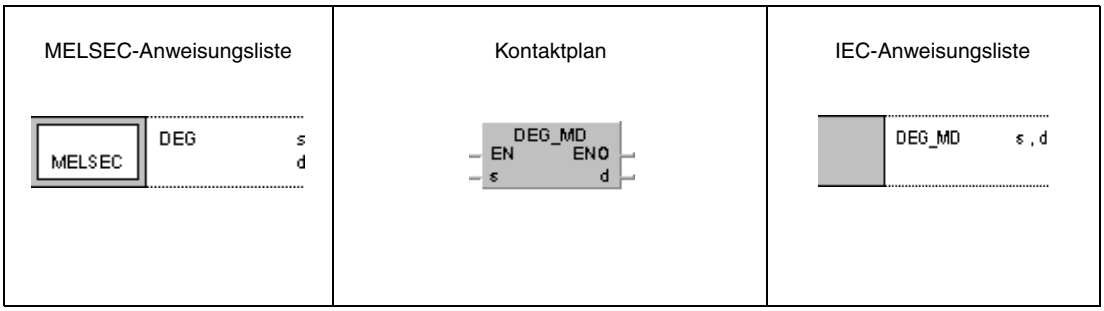
[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

[2] Nicht für Q00JCPU, Q00CPU und Q01CPU

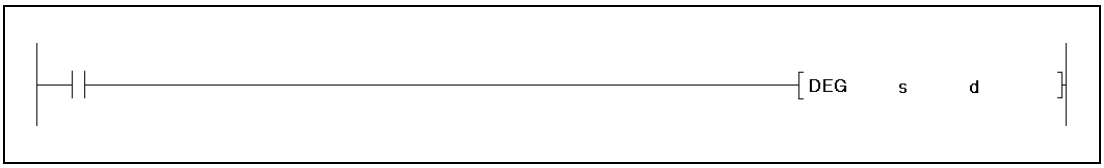
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten E	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	●	●	—	●	—	—	3
d	—	●	●	—	●	●	—	—	—		

GX IEC Developer



GX Developer



Variablen

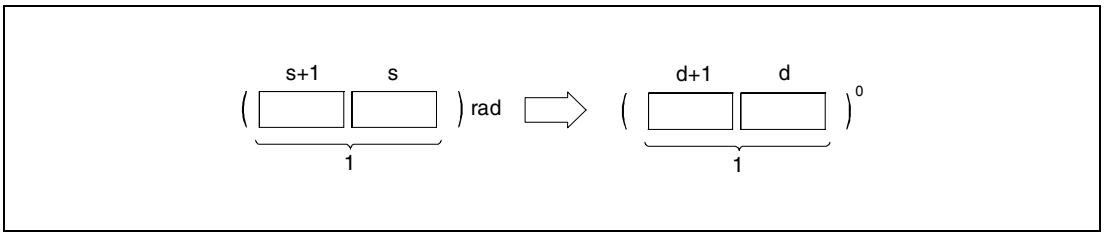
Operand	Befehlswert	Datentyp
s	Erste Adresse des Operanden, in dem der Radiantenwert gespeichert ist, der in Grad umgerechnet werden soll.	reelle Zahl
d	Erste Adresse des Operanden, in dem das Ergebnis der Umrechnung gespeichert wird.	

Funktions-weise

Umrechnung von Radiant in Grad

DEG Umrechnungsanweisung

Die DEG-Anweisung berechnet aus der Radiantenangabe (rad) in s und s+1 den entsprechenden Gradwert (°) und speichert das Ergebnis in d und d+1 ab.



[1] Gleitkommazahl (reelle Zahl)

Die Umrechnung von Radiant in Grad wird nach folgender Formel durchgeführt:

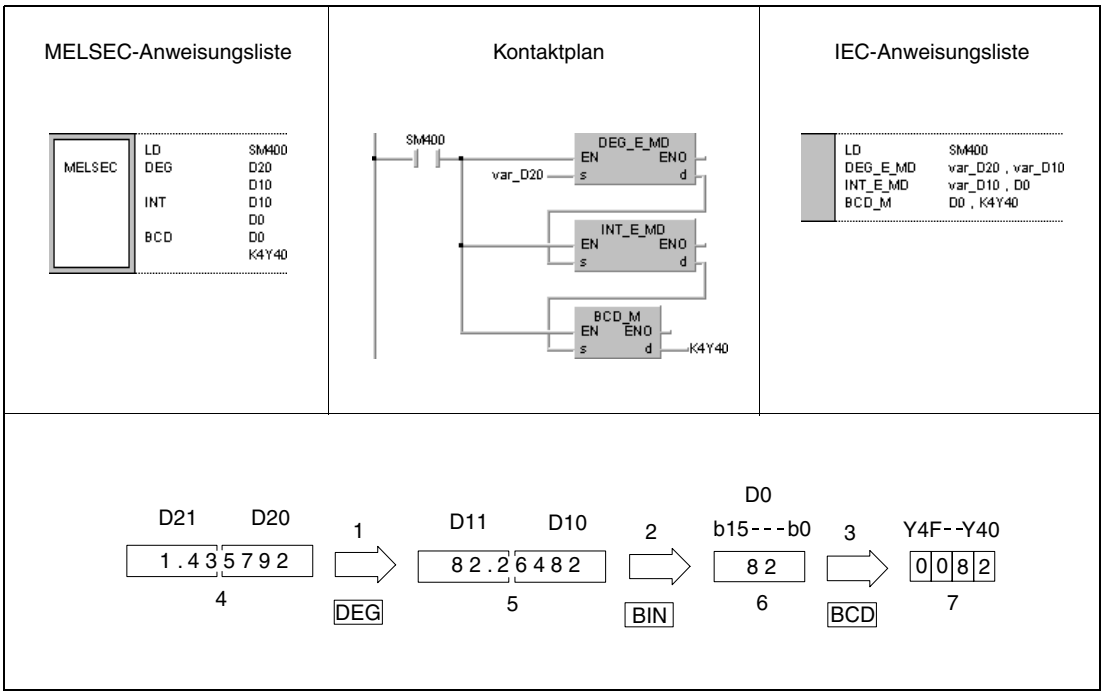
Gradwert = Radiantenwert x 180 / π

Fehler-quellen

Wenn bei Verwendung der Q4AR-CPU die unter s angegebene Adresse -0 enthält und der Sondermerker SM707 nicht gesetzt ist, wird der Fehlercode 4100 ausgegeben.

Beispiel DEG

Das folgende Programm berechnet für die Einschaltdauer von SM400 aus dem Radiantenwert, der in D20 und D21 als 4-stelliger BCD-Wert gespeichert ist, einen Gradwert und gibt das Ergebnis als 4-stelligen BCD-Wert an Y20 bis Y4F aus .



[1] Umwandlung in Grad
[2] Umwandlung ins BIN-Format
[3] Umwandlung ins BCD-Format
[4] Gleitkommazahl (reelle Zahl)
[5] Gleitkommazahl (reelle Zahl)
[6] Binärwert
[7] BCD-Wert

HINWEIS *Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.12.9 SQR, SQRP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●[2]

[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

[2] Nicht für Q00JCPU, Q00CPU und Q01CPU

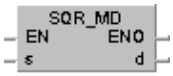
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten E	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	●	●	—	●	—	SM0	3
d	—	●	●	—	●	●	—	—	—		

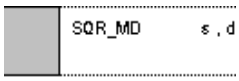
GX IEC Developer

MELSEC-Anweisungsliste

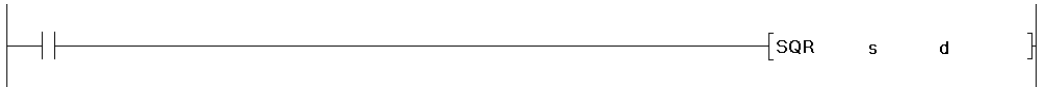
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

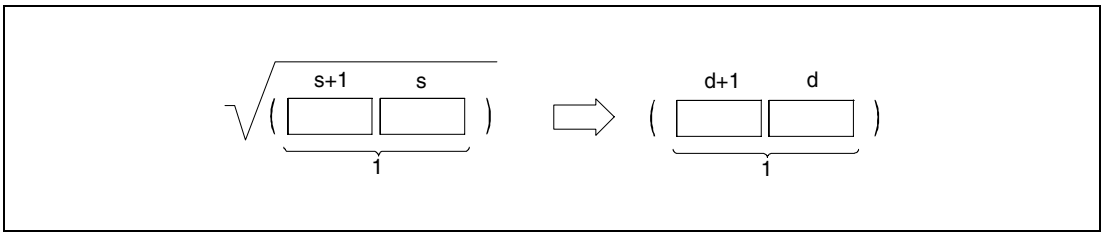
Operand	Befehlswert	Datentyp
s	Erste Adresse des Operanden, in dem der Wert gespeichert ist, aus dem eine Quadratwurzel gezogen werden soll.	reelle Zahl
d	Erste Adresse des Operanden, in dem das Ergebnis der Berechnung gespeichert wird.	

Funktions-weise

Quadratwurzelberechnung

SQR Quadratwurzel aus Gleitkommazahlen

Die SQR-Anweisung berechnet aus der Gleitkommazahl in s und s+1 die Quadratwurzel und speichert das Ergebnis in d und d+1.



[1] Gleitkommazahl (reelle Zahl)

Es können nur positive Werte in s und s+1 gespeichert werden. Die Anweisung kann nicht mit negativen Werten durchgeführt werden.

Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Der Wert, der ab s eingegeben wurde, ist negativ.
- Wenn bei Verwendung der Q4AR-CPU die unter s angegebene Adresse -0 enthält und der Sondermerker SM707 nicht gesetzt ist, wird der Fehlercode 4100 ausgegeben.

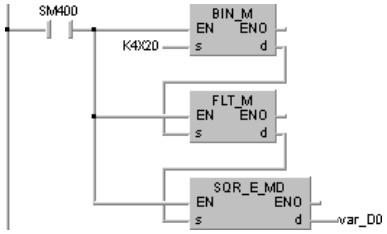
Beispiel SQR

Das folgende Programm berechnet für die Einschaltdauer von SM400 aus dem 4-stelligen BCD-Wert an X20 bis X2F die Quadratwurzel und speichert das Ergebnis als Gleitkommazahl in D0 und D1.

MELSEC-Anweisungsliste

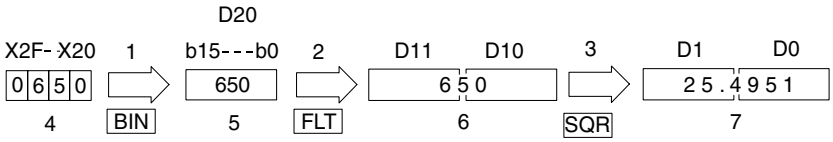
```
MELSEC
LD    SM400
BIN   K4X20
      D20
FLT   D20
      D10
SQR   D10
      D0
```

Kontaktplan



IEC-Anweisungsliste

```
LD      SM400
BIN_M   K4X20 , D20
FLT_M   D20 , var_D10
SQR_E_MD var_D10 , var_D0
```



[1] Umwandlung ins BIN-Format

[2] Umwandlung ins Gleitkommaformat

[3] Quadratwurzelberechnung

[4] BCD-Wert

[5] Binärwert

[6] Gleitkommazahl (reelle Zahl)

[7] Gleitkommazahl (reelle Zahl)

HINWEIS *Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.12.10 EXP, EXPP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●[2]

[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

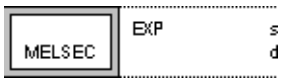
[2] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

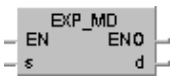
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten E	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	●	●	—	●	—	SM0	3
d	—	●	●	—	●	●	—	—	—		

GX IEC Developer

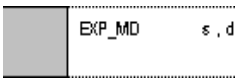
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

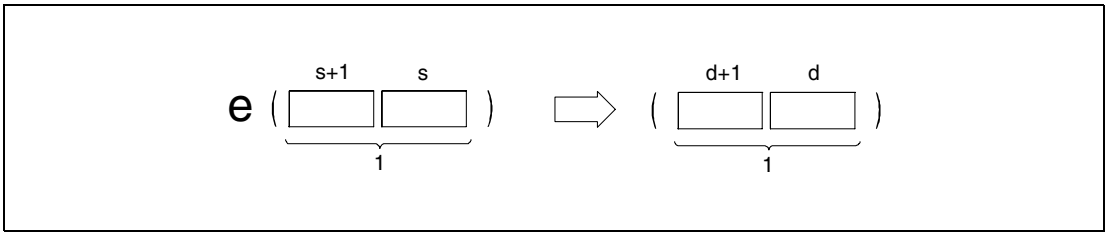
Operand	Befehlswert	Datentyp
s	Erste Adresse des Operanden, in dem der Wert gespeichert ist, mit dem die EXP-Anweisung ausgeführt werden soll.	reelle Zahl
d	Erste Adresse des Operanden, in dem das Ergebnis der Berechnung gespeichert wird.	

Funktions-weise

Gleitkommazahlen als Exponent zur Basis e

EXP Exponent von e

Die EXP-Anweisung führt mit der Gleitkommazahl in s und s+1 die Berechnung des Exponenten zur Basis e durch und speichert das Ergebnis in d und d+1.



[1] Gleitkommazahl (reelle Zahl)

Die Berechnung basiert auf der Eulerschen Zahl (e = 2.718281828).

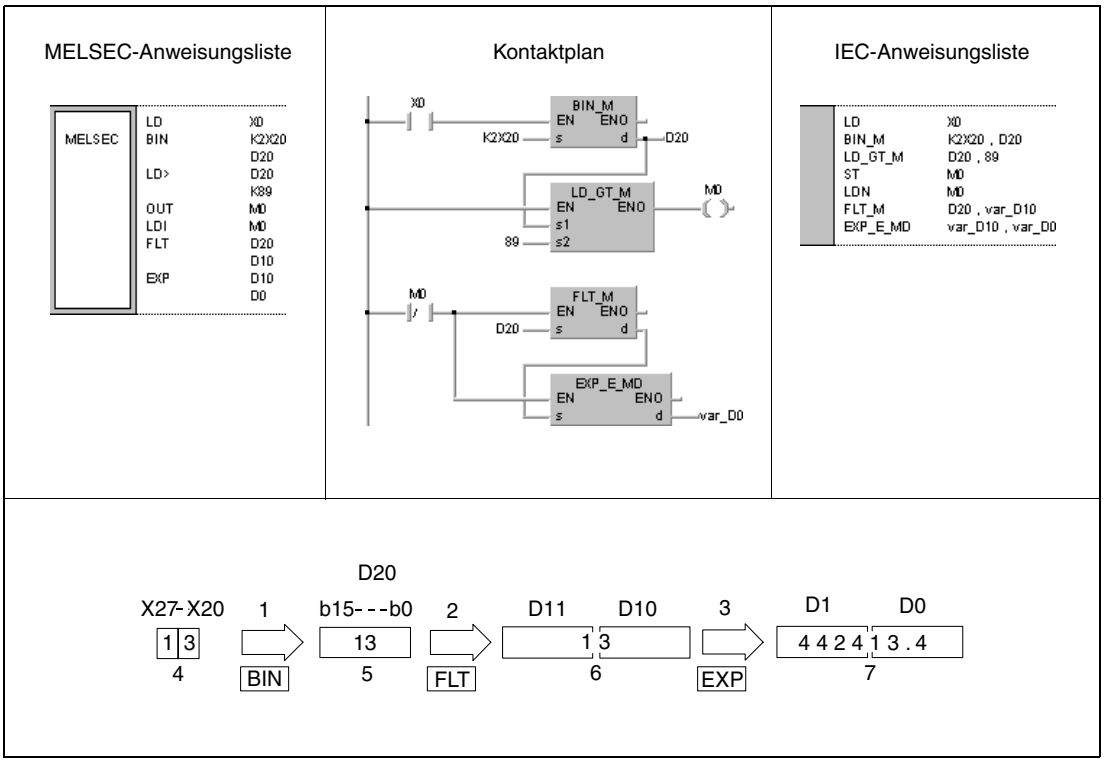
Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Das Berechnungergebnis liegt außerhalb des Wertebereichs von 2^{-127} und 2^{129} (Fehlercode 4100).
- Wenn bei Verwendung der Q4AR-CPU die unter s angegebene Adresse -0 enthält und der Sondermerker SM707 nicht gesetzt ist, wird der Fehlercode 4100 ausgegeben.

Beispiel EXP

Das folgende Programm berechnet das Ergebnis der Exponentialfunktion zur Basis e mit dem 2-stelligen BCD-Wert an X20 bis X27 und speichert das Ergebnis als Gleitkommazahl in D0 und D1.



[1] Umwandlung ins BIN-Format

[2] Umwandlung ins Gleitkommaformat

[3] Exponential Berechnung

[4] BCD-Wert

[5] Binärwert

[6] Gleitkommazahl (reelle Zahl)

[7] Gleitkommazahl (reelle Zahl)

HINWEISE *Das Berechnungsergebnis darf 2^{129} ln = 89.41598 nicht überschreiten. Wenn der BCD-Wert den Wert 90 überschreitet, wird eine Fehlermeldung von SM0 generiert.*

Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.

7.12.11 LOG, LOGP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●[2]

[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

[2] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

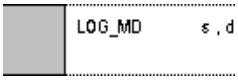
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten E	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	●	●	—	●	—	SM0	3
d	—	●	●	—	●	●	—	—	—		

GX IEC Developer

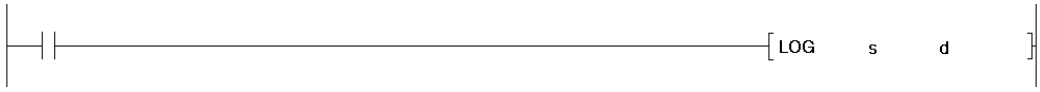
MELSEC-Anweisungsliste

Kontaktplan

IEC-Anweisungsliste



GX Developer



Variablen

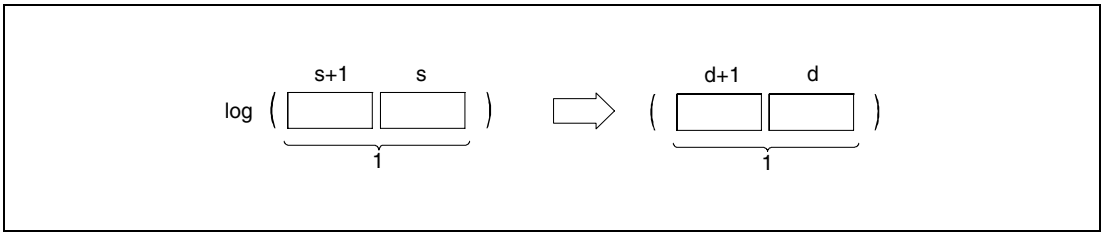
Operand	Befehlswert	Datentyp
s	Erste Adresse des Operanden, in dem der Wert gespeichert ist, mit dem die LOG-Anweisung ausgeführt werden soll.	reelle Zahl
d	Erste Adresse des Operanden, in dem das Ergebnis der Berechnung gespeichert wird.	

Funktions-weise

Logarithmus-naturalis-Berechnung

LOG Logarithmus-naturalis-Berechnung mit Gleitkommazahlen

Die LOG-Anweisung berechnet den Logarithmus naturalis aus der Gleitkommazahl in s und s+1 und speichert das Ergebnis in d und d+1.



[1] Gleitkommazahl (reelle Zahl)

Es können nur positive Werte in s und s+1 eingegeben werden. Die Berechnung kann nicht mit negativen Werten durchgeführt werden.

Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Der ab s eingegebene Wert ist negativ (Fehlercode 4100).
- Das Berechnungsergebnis liegt außerhalb des Wertebereichs von 2^{-127} und 2^{129} (Fehlercode 4100).
- Wenn bei Verwendung der Q4AR-CPU die unter s angegebene Adresse -0 enthält und der Sondermerker SM707 nicht gesetzt ist, wird der Fehlercode 4100 ausgegeben.

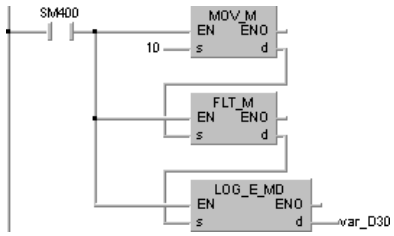
Beispiel 1 LOG

Das folgendene Programm berechnet den Logarithmus naturalis aus dem Wert 10 und speichert das Ergebnis in D30 und D31.

MELSEC-Anweisungsliste

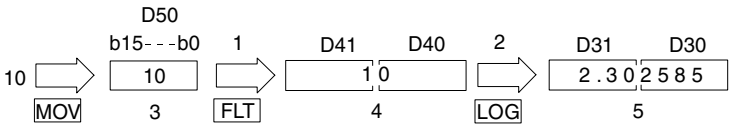
```
MELSEC
        LD      SM400
        MOV     K10
                D50
        FLT     D50
                D40
        LOG     D40
                D30
```

Kontaktplan



IEC-Anweisungsliste

```
LD      SM400
MOV_M   10 , D50
FLT_M   D50 , var_D40
LOG_E_MD var_D40 , var_D30
```



[1] Umwandlung ins Gleitkommaformat

[2] Logarithmusberechnung

[3] Binärwert

[4] Gleitkommazahl (reelle Zahl)

[5] Gleitkommazahl (reelle Zahl)

HINWEIS *Die CPU arbeitet mit dem Logarithmus naturalis mit der Basis e. Um den Wert in den dekadischen Logarithmus mit der Basis 10 umzuwandeln, kann folgende Formel benutzt werden:*

$log_{10} X = 0{,}43429 \times log_e X$

HINWEIS *Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.12.12 RND, RNDP, SRND, SRNDP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●[1]

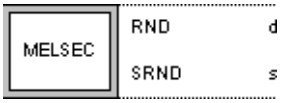
[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

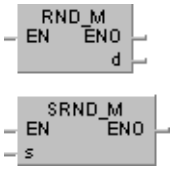
	Operanden								Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sonder-module U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere	
	Bit	Wort		Bit	Wort					
s				●				—	—	—	2

GX IEC Developer

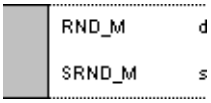
MELSEC-Anweisungsliste



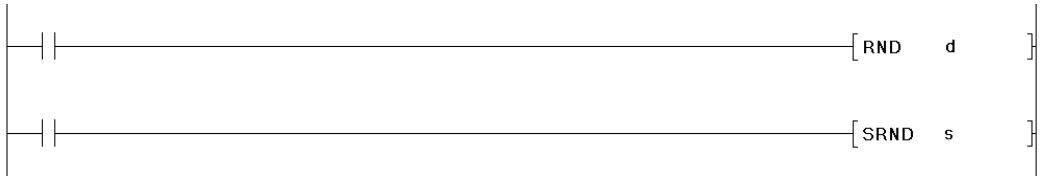
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

Operand	Befehlswert	Datentyp
d	Erste Adresse des Operanden, in dem die Zufallszahl gespeichert wird.	BIN-16-Bit
s	Zufallszahlenserie oder erste Adresse des Operanden, in dem diese Daten gespeichert sind.	

Funktions-weise

Generierung von Zufallszahlen und Serienaktualisierung

RND Generierung von Zufallszahlen

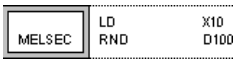
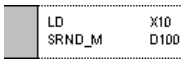
Die RND-Anweisung generiert aus dem Bereich 0 bis 32767 eine Zufallszahl und speichert sie in d.

SRND Aktualisierung von Zufallszahlenserien

Die SRND-Anweisung aktuallisiert die Zufallszahlenserie, die in s gespeichert ist.

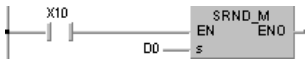
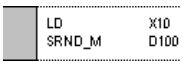
Beispiel 1 RND

Das folgende Programm speichert für die Einschaltdauer von X10 die generierte Zufallszahl in D100.

MELSEC-Anweisungsliste	Kontaktplan	IEC-Anweisungsliste
		

Beispiel 2 SRND

Das folgende Programm aktualisiert für die Einschaltdauer von X10 die Zufallszahlenserie in D0.

MELSEC-Anweisungsliste	Kontaktplan	IEC-Anweisungsliste
		

7.12.13 BSQR, BSQRP, BDSQR, BDSQRP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●[2]

[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

[2] Nicht für Q00JCPU, Q00CPU und Q01CPU

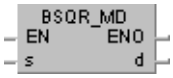
Operanden MELSEC Q

	Operanden								Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere	
	Bit	Wort		Bit	Wort					
s				●				●	—	SM0	3
d				●				—	—		

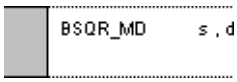
GX IEC Developer

MELSEC-Anweisungsliste

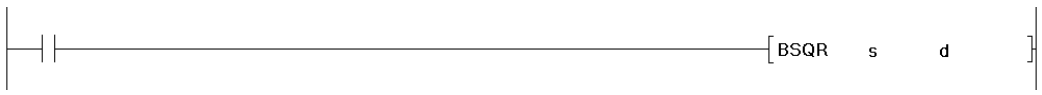
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

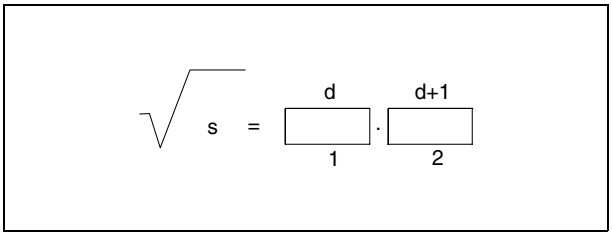
Operand	Befehlswert	Datentyp
s	Daten, mit denen die Quadratwurzelberechnung durchgeführt wird, oder Adresse der Operanden, in denen diese Daten gespeichert sind.	BCD 4-/8-stellig
d	Erste Adresse des Operanden, in dem das Ergebnis der Quadratwurzelberechnung gespeichert wird.	BCD 4-stellig

Funktionsweise

Quadratwurzelberechnung aus 4-/8-stelligen BCD-Daten

BSQR Quadratwurzelberechnung aus 4-stelligen BCD-Daten

Die BSQR-Anweisung berechnet die Quadratwurzel aus s und speichert das Ergebnis in d und d+1.



[1] Integerteil
[2] Nachkommastellen

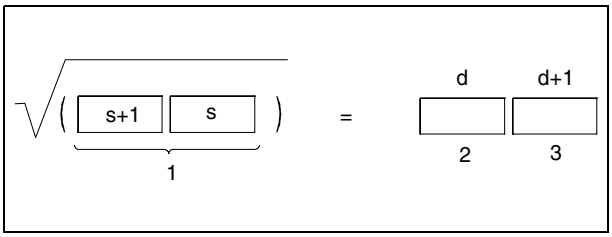
Der Wert in s kann nur ein BCD-Wert mit maximal 4 Stellen sein. Der Wertebereich von 0 bis 9999 darf nicht verlassen werden.

Das Berechnungsergebnis, das in d und d+1 gespeichert wird, darf den Wertebereich von 0 bis 9999 nicht verlassen.

Das Berechnungsergebnis wird auf 5 Stellen genau berechnet und gerundet als 4-stelliger Wert dargestellt.

BDSQR Quadratwurzelberechnung aus 8-stelligen BCD-Daten

Die BDSQR-Anweisung berechnet die Quadratwurzel aus s und s+1 und speichert das Ergebnis in d und d+1.



[1]Zwei-Wort-Daten
[2]Integerteil
[3]Nachkommastellen

Der Wert in s und s+1 kann nur ein BCD-Wert mit maximal 8 Stellen sein. Der Wertebereich von 0 bis 99999999 darf nicht verlassen werden.

Das Berechnungsergebnis, das in d und d+1 gespeichert wird, darf den Wertebereich von 0 bis 9999 nicht verlassen.

Das Berechnungsergebnis wird auf 5 Stellen genau berechnet und gerundet als 4-stelliger Wert dargestellt.

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die ab s gespeicherten Daten sind keine BCD-Daten (Fehlercode 4100).

Beispiel 1 BSQR

Das folgende Programm berechnet für die Einschaltdauer von SM400 die Quadratwurzel aus dem BCD-Wert 1325 und gibt den Integerteil des Ergebnisses als 4-stelligen BCD-Wert an Y50 bis Y5F aus. Die Nachkommastellen werden als 4-stelliger BCD-Wert an Y40 bis Y4F ausgegeben.

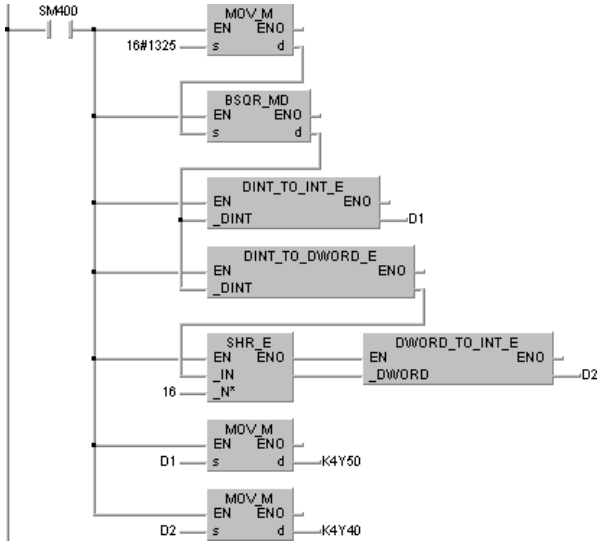
MELSEC-Anweisungsliste

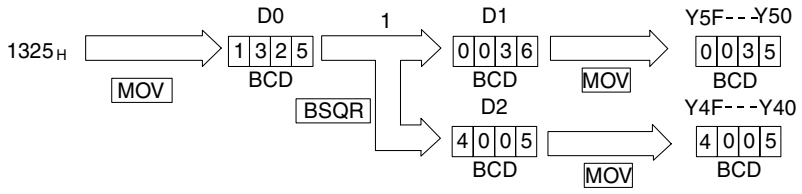
```
MELSEC  LD    SM400
        MOV   H1235
              D0
        BSQR  D0
              D1
        MOV   D1
              K4Y50
        MOV   D2
              K4Y40
```

IEC-Anweisungsliste

```
LD                SM400
MOV_M             H1325 , D0
BSQR_MD           D0 , var_D1
DINT_TO_INT_E     var_D1 , D1
DINT_TO_DWORD_E   var_D1 , var_temp
SHR_E             var_temp , K16 , var_temp
DWORD_TO_INT_E    var_temp , D2
LD                SM400
MOV_M             D1 , K4Y50
MOV_M             D2 , K4Y40
```

Kontaktplan





[1] Quadratwurzel-Berechnung

Beispiel 2 BDSQR

Das folgende Programm berechnet für die Einschaltdauer von SM400 die Quadratwurzel aus dem BCD-Wert 74625813 und gibt den Integerteil des Ergebnisses als 4-stelligen BCD-Wert an Y50 bis Y5F aus. Die Nachkommastellen werden als 4-stelliger BCD-Wert an Y40 bis Y4F ausgegeben.

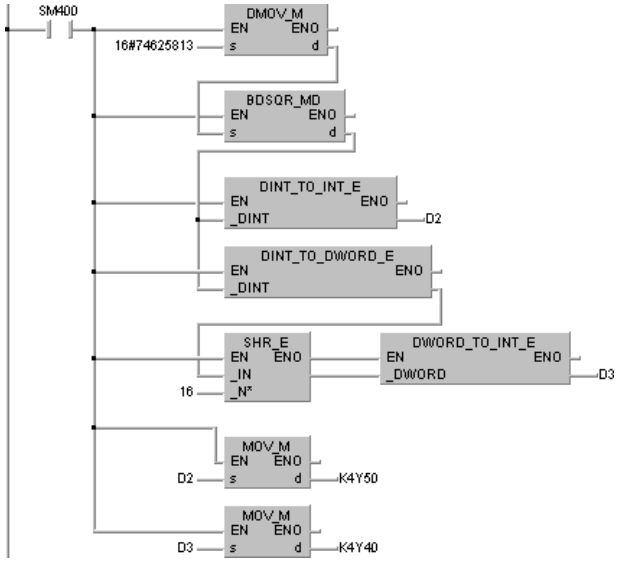
MELSEC-Anweisungsliste

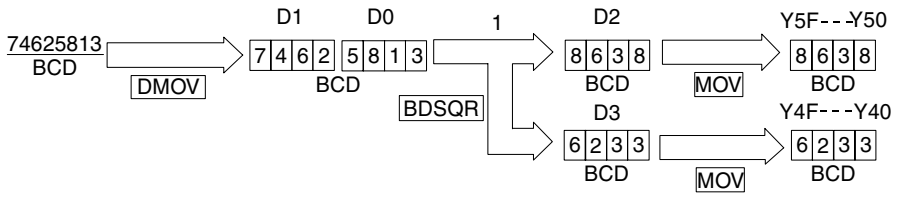
```
MELSEC   LD      SM400
         DMOV    H74625813
                 D0
         BDSQR   D0
                 D2
         MOV     D2
                 K4Y50
         MOV     D3
                 K4Y40
```

IEC-Anweisungsliste

```
LD              SM400
DMOV_M          H74625813 , var_D0
BDSQR_MD        var_D0 , var_D2
DINT_TO_INT_E   var_D2 , D2
DINT_TO_DWORD_E var_D2 , var_temp
SHR_E           var_temp , K16 , var_temp
DWORD_TO_INT_E  var_temp , D3
ST              D3
LD              SM400
MOV_M           D2 , K4Y50
MOV_M           D3 , K4Y40
```

Kontaktplan





[1] Quadratwurzel-Berechnung

HINWEIS *Diese Programmbeispiele sind ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.12.14 BSIN, BSINP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●[2]

[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

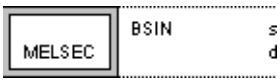
[2] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

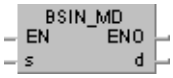
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	●	●	●	●	●	●	●	●	—	SM0	3
d	—	●	●	—	—	—	—	—	—		

GX IEC Developer

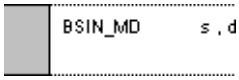
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

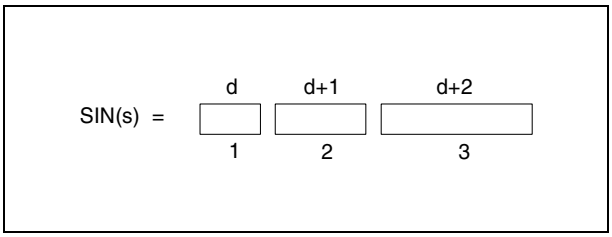
Operand	Befehlswert	Datentyp
s	Erste Adresse des Operanden, in dem die Winkeldaten, die zur Ausführung der SIN-Anweisung (Sinus) notwendig sind, gespeichert sind.	BCD 4-stellig
d	Erste Adresse des Operanden, in dem das Ergebnis gespeichert wird.	

Funktionsweise

Sinusberechnung mit BCD-Daten

BSIN Sinusberechnung

Die BSIN-Anweisung berechnet den Sinuswert aus den Winkeldaten in s. Das Vorzeichen des Ergebnisses wird in d gespeichert. Der Zahlenwert des Ergebnisses wird in d+1 und d+2 gespeichert.



[1] Vorzeichen
[2] Integerteil
[3] Nachkommastellen

Der Wert s muss ein BCD-Wert innerhalb des Wertebereichs von 0° bis 360° sein.

Das Vorzeichen des Ergebnisses in d nimmt bei positivem Wert den Wert 0 und bei negativem Wert den Wert 1 an.

Das Ergebnis in d+1 und d+2 kann ein BCD-Wert zwischen -1.000 und 1.000 sein.

Das Berechnungsergebnis wird von der 5-ten Stelle an gerundet.

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die in s angegebenen Daten sind keine BCD-Daten (Fehlercode 4100).
- Die in s angegebenen Daten liegen außerhalb des Wertebereichs von 0° bis 360° (Fehlercode 4100).

Beispiel BSIN

Das folgende Programm berechnet für die Einschaltdauer von SM400 den Sinuswert aus dem 3-stelligen BCD-Wert an X20 bis X2B. Wenn der Wert an X20 bis X2B größer als 360 ist, wird der Wert in einen Wertebereich zwischen 0° bis 360° korrigiert.

Das Vorzeichen wird an Y60 ausgegeben. Der Integerteil wird als 1-stelliger BCD-Wert an Y50 bis Y53 ausgegeben.

Die Nachkommastellen werden als 4-stelliger BCD-Wert an Y40 bis Y4F ausgegeben.

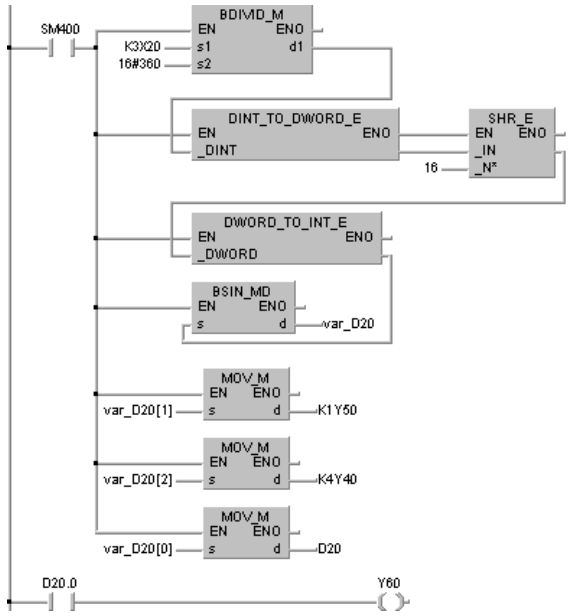
MELSEC-Anweisungsliste

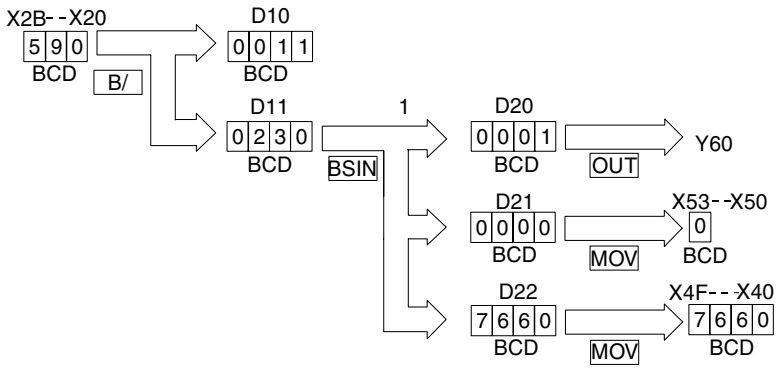
```
MELSEC   LD      SM400
         B/      K3X20
                 H360
                 D10
         BSIN    D11
                 D20
         MOV     D21
                 K1X50
         MOV     D22
                 K4X40
         LD      D20.0
         OUT     Y60
```

IEC-Anweisungsliste

```
LD                  SM400
BDIVMD_M            K3X20 , H360 , var_D10
DINT_TO_DWORD_E     var_D10 , var_Merker
SHR_E               var_Merker , K16 , var_Merker
DWORD_TO_INT_E      var_Merker , D11
LD                  SM400
BSIN_MD             D11 , var_D20
MOV_M               var_D20[1] , K1Y50
MOV_M               var_D20[2] , K4Y40
MOV_M               var_D20[0] , D20
LD                  D20.0
ST                  Y60
```

Kontaktplan





[1] Sinus-Berechnung

HINWEIS *Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.12.15 BCOS, BCOSP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●[2]

[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

[2] Nicht für Q00JCPU, Q00CPU und Q01CPU

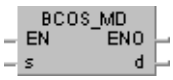
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	●	●	●	●	●	●	●	●	—	SM0	3
d	—	●	●	—	—	—	—	—	—		

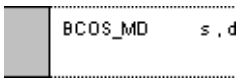
GX IEC Developer

MELSEC-Anweisungsliste

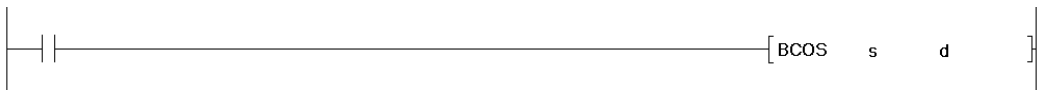
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

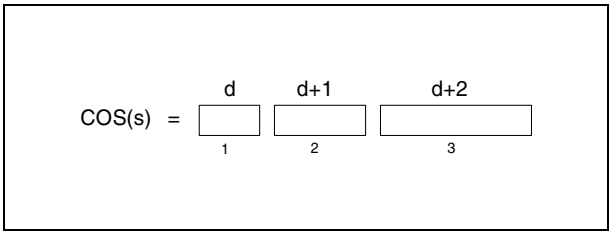
Operand	Befehlswert	Datentyp
s	Erste Adresse des Operanden, in dem die Winkeldaten, die zur Ausführung der BCOS-Anweisung (Cosinus) notwendig sind, gespeichert sind.	BCD 4-stellig
d	Erste Adresse des Operanden, in dem das Ergebnis gespeichert wird.	

Funktions-weise

Cosinusberechnung mit BCD-Daten

BCOS Cosinusberechnung

Die BCOS-Anweisung berechnet den Cosinuswert aus den Winkeldaten in s. Das Vorzeichen des Ergebnisses wird in d gespeichert. Der Zahlenwert des Ergebnisses wird in d+1 und d+2 gespeichert.



[1] Vorzeichen
[2] Integerteil
[3] Nachkommastellen

Der Wert in s muss ein BCD-Wert innerhalb des Wertebereichs von 0° bis 360° sein.

Das Vorzeichen des Ergebnisses in d nimmt bei positivem Wert den Wert 0 und bei negativem Wert den Wert 1 an.

Das Ergebnis in d+1 und d+2 kann ein BCD-Wert zwischen -1.000 und 1.000 sein.

Das Berechnungsergebnis wird von der 5-ten Stelle an gerundet.

Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die in s angegebenen Daten sind keine BCD-Daten (Fehlercode 4100).
- Die in s angegebenen Daten liegen außerhalb des Wertebereichs von 0° bis 360° (Fehlercode 4100).

Beispiel BCOS

Das folgende Programm berechnet für die Einschaltdauer von SM400 den Cosinuswert aus dem 3-stelligen BCD-Wert an X20 bis X2B. Wenn der Wert an X20 bis X2B größer als 360 ist, wird der Wert in einen Wertebereich zwischen 0° bis 360° korrigiert.

Das Vorzeichen wird an Y60 ausgegeben. Der Integerteil wird als 1-stelliger BCD-Wert an Y50 bis Y53 ausgegeben.

Die Nachkommastellen werden als 4-stelliger BCD-Wert an Y40 bis Y4F ausgegeben.

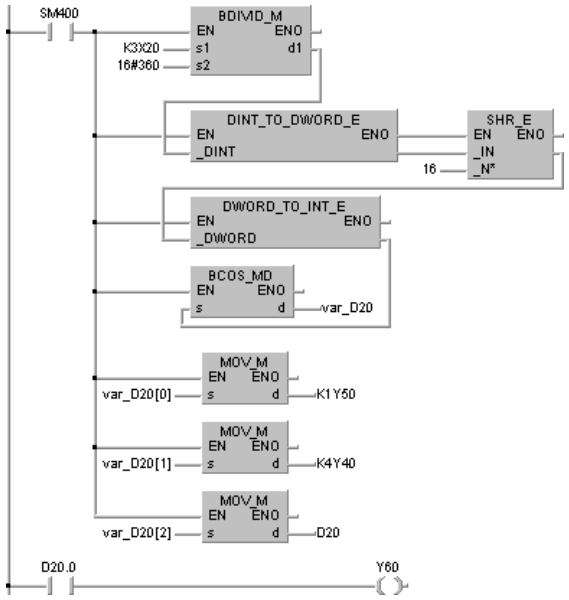
MELSEC-Anweisungsliste

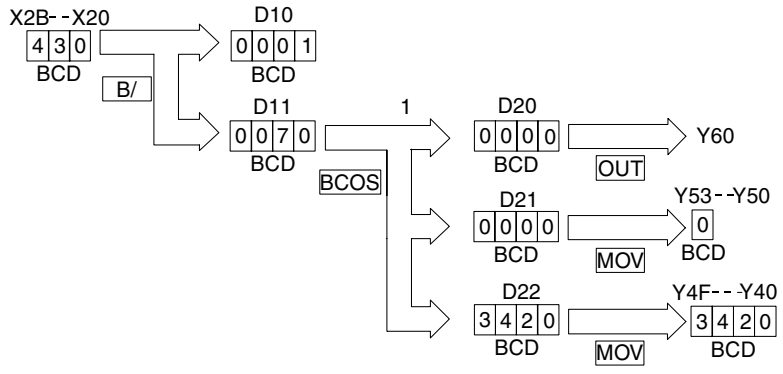
```
MELSEC  LD      SM400
        B/      K3X20
                H360
                D10
        BCOS    D11
                D20
        MOV     D21
                K1Y50
        MOV     D22
                K4Y40
        LD      D20.0
        OUT     Y60
```

IEC-Anweisungsliste

```
LD              SM400
BDIMD_M         K3X20 , H360 , var_D10
DINT_TO_DWORD_E var_D10 , var_Merker
SHR_E           var_Merker , K16 , var_Merker
DWORD_TO_INT_E  var_Merker , D11
LD              SM400
BCOS_MD         D11 , var_D20
MOV_M           var_D20[0] , K1Y50
MOV_M           var_D20[1] , K4Y40
MOV_M           var_D20[2] , D20
LD              D20.0
ST              Y60
```

Kontaktplan





[1] Cosinus-Berechnung

HINWEIS *Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.12.16 BTAN, BTANP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●[2]

[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

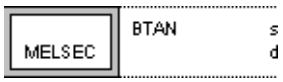
[2] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

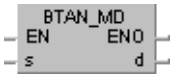
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	●	●	●	●	●	●	●	●	—	SM0	3
d	—	●	●	—	—	—	—	—	—		

GX IEC Developer

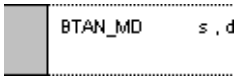
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

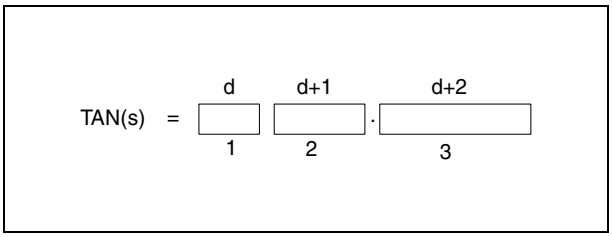
Operand	Befehlswert	Datentyp
s	Erste Adresse des Operanden, in dem die Winkeldaten, die zur Ausführung der BTAN-Anweisung (Tangens) notwendig sind, gespeichert sind.	BCD 4-stellig
d	Erste Adresse des Operanden, in dem das Ergebnis gespeichert wird.	

Funktions-weise

Tangensberechnung mit BCD-Daten

BTAN Tangensberechnung

Die BTAN-Anweisung berechnet den Tangenswert aus den Winkeldaten in s. Das Vorzeichen des Ergebnisses wird in d gespeichert. Der Zahlenwert des Ergebnisses wird in d+1 und d+2 gespeichert.



[1] Vorzeichen
[2] Integerteil
[3] Nachkommastellen

Der Wert in s muss ein BCD-Wert innerhalb des Wertebereichs von 0° bis 360° sein.

Das Vorzeichen des Ergebnisses in d nimmt bei positivem Wert den Wert 0 und bei negativem Wert den Wert 1 an.

Das Ergebnis in d+1 und d+2 kann ein BCD-Wert zwischen -57.2900 und 57.2900 sein.

Das Berechnungsergebnis wird von der 5-ten Stelle an gerundet.

Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die in s angegebenen Daten sind keine BCD-Daten (Fehlercode 4100).
- Die in s angegebenen Daten sind liegen außerhalb des Wertebereichs von 0°bis 360° (Fehlercode 4100).
- Die in s angegebenen Werte betragen 90° oder 270° (Fehlercode 4100).

Beispiel BTAN

Das folgende Programm berechnet für die Einschaltdauer von SM400 den Tangenswert aus dem 3-stelligen BCD-Wert an X20 bis X2B. Wenn der Wert an X20 bis X2B größer als 360 ist, wird der Wert in einen Wertebereich zwischen 0° bis 360° korrigiert.

Das Vorzeichen wird an Y60 ausgegeben. Der Integerteil wird als 1-stelliger BCD-Wert an Y50 bis Y53 ausgegeben.

Die Nachkommastellen werden als 4-stelliger BCD-Wert an Y40 bis Y4F ausgegeben.

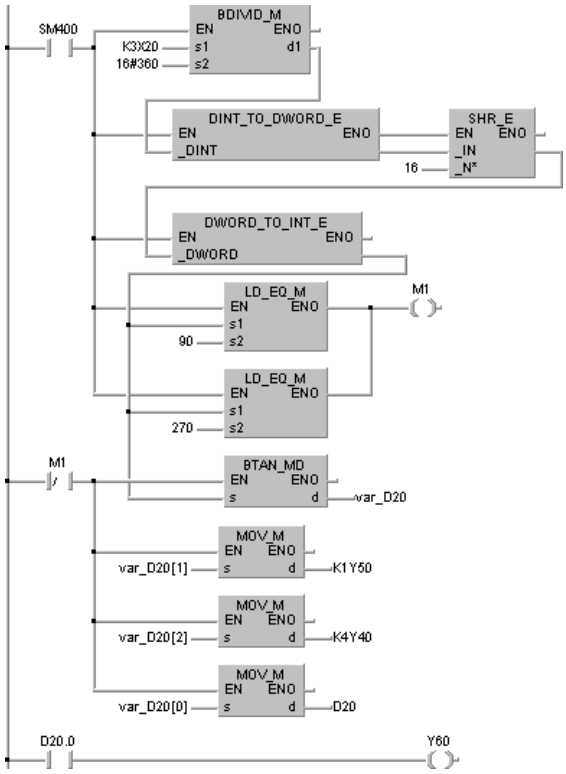
MELSEC-Anweisungsliste

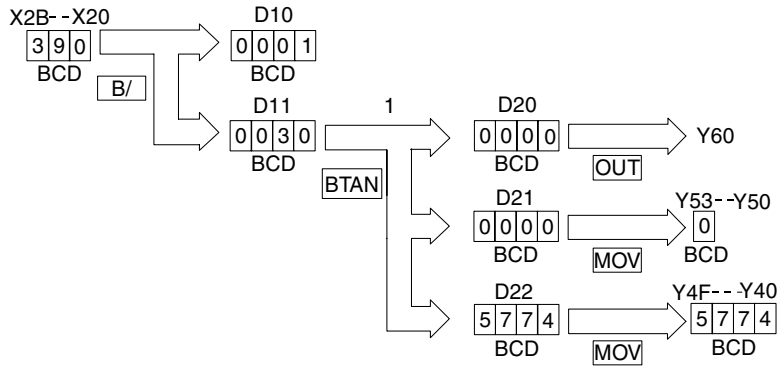
```
MELSEC  LD      SM400
        B/      K3X20
                H360
                D10
        LD=     D11
                K90
        OR=     D11
                K270
        OUT     M1
        LDI     M1
        BTAN    D11
                D20
        MOV     D21
                K1Y50
        MOV     D22
                K4Y40
        LD      D20.0
        OUT     Y60
```

IEC-Anweisungsliste

```
LD              SM400
BDIVD_M         K3X20 , H360 , var_D10
DINT_TO_DWORD_E var_D10 , var_Merker
SHR_E           var_Merker , K16 , var_Merker
DWORD_TO_INT_E  var_Merker , D11
LD              TRUE
LD_EQ_M         D11 , K90
OR_EQ_M         D11 , K270
ST              M1
LDN             M1
BTAN_MD         D11 , var_D20
MOV_M           var_D20[1] , K1Y50
MOV_M           var_D20[2] , K4Y40
MOV_M           var_D20[0] , D20
LD              D20.0
ST              Y60
```

Kontaktplan





[1] Tangens-Berechnung

HINWEIS *Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.12.17 BASIN, BASINP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●[2]

[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

[2] Nicht für Q00JCPU, Q00CPU und Q01CPU

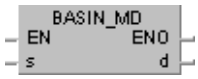
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	—	—	—	—	—	SM0	3
d	●	●	●	●	●	●	●	—	—		

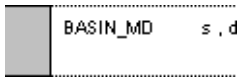
GX IEC Developer

MELSEC-Anweisungsliste

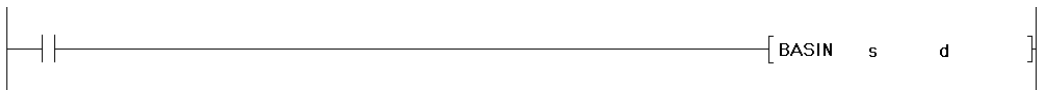
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

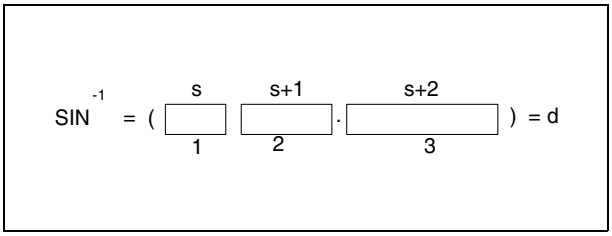
Operand	Befehlswert	Datentyp
s	Erste Adresse des Operanden, in dem der Sinuswert, der zur Ausführung der BASIN-Anweisung (Arcussinus) notwendig ist, gespeichert ist.	BCD 4-stellig
d	Erste Adresse des Operanden, in dem das Ergebnis gespeichert wird.	

Funktions-weise

Arcussinusberechnung mit BCD-Daten

BASIN Arcussinusberechnung

Die BASIN-Anweisung berechnet aus dem Sinuswert in s, s+1 und s+2 die Winkeldaten und speichert das Ergebnis in d.



[1] Vorzeichen
[2] Integerteil
[3] Nachkommastellen

Das Vorzeichen in s nimmt bei positivem Wert den Wert 0 und bei negativem Wert den Wert 1 an.

Der Integerteil vor dem Dezimalkomma und die Nachkommastellen dürfen nur BCD-Werte zwischen 0 und 1.0000 sein.

Der Zahlenwert des Ergebnisses in d kann ein BCD-Wert zwischen 0° und 90° oder zwischen 270° und 360° sein.

Das Berechnungsergebnis wird von der 5-ten Nachkommastelle an gerundet.

Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die in s bis s+2 angegebenen Daten sind keine BCD-Daten (Fehlercode 4100).
- Die in s bis s+2 angegebenen Daten liegen außerhalb des Wertebereichs von -1.0000 bis 1.0000 (Fehlercode 4100).

Beispiel BASIN

Das folgende Programm berechnet für die Einschaltdauer von SM400 den Arcussinuswert aus dem Vorzeichenwert an X0 (positiv = X0 gesetzt, negativ = X0 nicht gesetzt), dem 1-stelligen BCD-Integerteil an X30 bis X33 und den Nachkommastellen des 4-stelligen BCD-Werts an X20 bis X2F. Der errechnete Winkelwert wird als 4-stelliger BCD-Wert an Y40 bis Y4F ausgegeben.

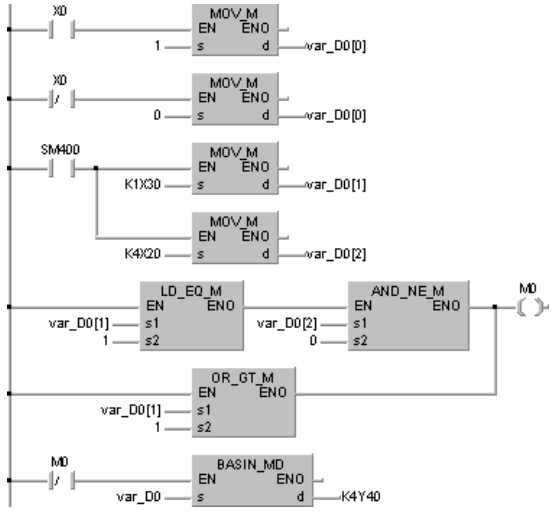
MELSEC-Anweisungsliste

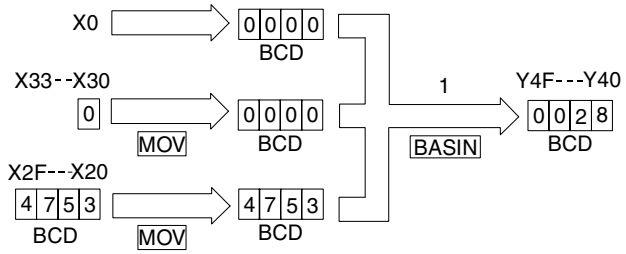
```
MELSEC  LD      X0
        MOV     K1
                D0
        LDI     X0
        MOV     K0
                D0
        LD      SM400
        MOV     K1X30
                D1
        MOV     K4X20
                D2
        LD=     D1
                K1
        AND<>   D2
                K0
        OR>     D1
                K1
        OUT     M0
        LDI     M0
        BASIN   D0
                K4Y40
```

IEC-Anweisungsliste

```
LD          X0
MOV_M       1 , var_D0[0]
LDN         X0
MOV_M       0 , var_D0[0]
LD          SM400
MOV_M       K1X30 , var_D0[1]
MOV_M       K4X20 , var_D0[2]
LD_EQ_M     var_D0[1] , 1
AND_NE_M    var_D0[2] , 0
OR_GT_M     var_D0[1] , 1
ST          M0
LDN         M0
BASIN_MD    var_D0 , K4Y40
```

Kontaktplan





[1] Arcussinus-Berechnung

HINWEIS *Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.12.18 BACOS, BACOSP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●[2]

[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

[2] Nicht für Q00JCPU, Q00CPU und Q01CPU

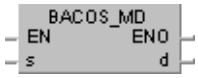
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	—	—	—	—	—	SM0	3
d	●	●	●	●	●	●	●	—	—		

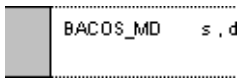
GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

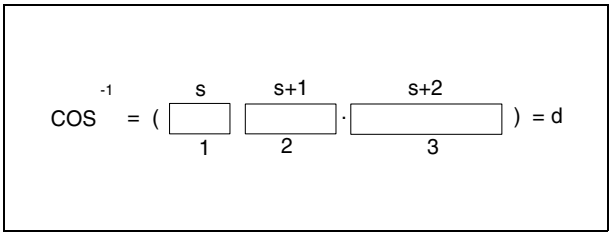
Operand	Befehlswert	Datentyp
s	Erste Adresse des Operanden, in dem der Cosinuswert, der zur Ausführung der BACOS-Anweisung (Arcuscosinus) notwendig ist, gespeichert ist.	BCD 4-stellig
d	Erste Adresse des Operanden, in dem das Ergebnis gespeichert wird.	

Funktions-weise

Arcuscosinusberechnung mit BCD-Daten

BACOS Arcuscosinusberechnung

Die BACOS-Anweisung berechnet aus dem Cosinuswert in s, s+1 und s+2 die Winkeldaten und speichert das Ergebnis in d.



[1] Vorzeichen
[2] Integerteil
[3] Nachkommastellen

Das Vorzeichen in s nimmt bei positivem Wert den Wert 0 und bei negativem Wert den Wert 1 an.

Der Integerteil vor dem Dezimalkomma und die Nachkommastellen dürfen nur BCD-Werte zwischen 0 und 1.0000 sein.

Der Zahlenwert des Ergebnisses in d kann ein BCD-Wert zwischen 0° und 180° sein.

Das Berechnungsergebnis wird von der 5-ten Nachkommastelle an gerundet.

Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die in s bis s+2 angegebenen Daten sind keine BCD-Daten (Fehlercode 4100).
- Die in s bis s+2 angegebenen Daten liegen außerhalb des Wertebereichs von -1.000 bis 1.000 (Fehlercode 4100).

Beispiel BACOS

Das folgende Programm berechnet für die Einschaltdauer von SM400 den Arcuscosinuswert aus dem Vorzeichenwert an X0 (positiv = X0 gesetzt, negativ = X0 nicht gesetzt), dem 1-stelligen BCD-Integerteil an X30 bis X33 und den Nachkommastellen des 4-stelligen BCD-Werts an X20 bis X2F. Der errechnete Winkelwert wird als 4-stelliger BCD-Wert an Y40 bis Y4F ausgegeben.

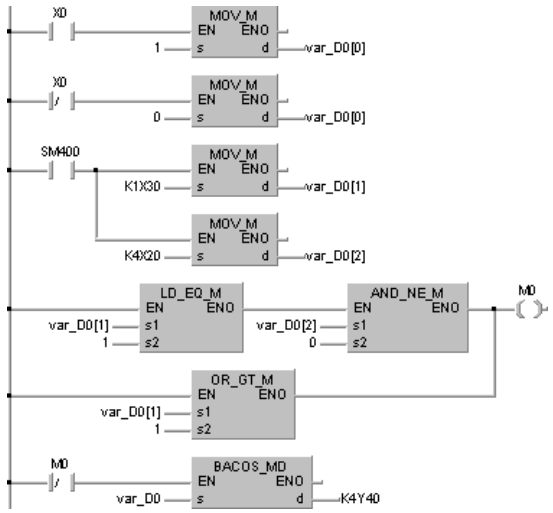
MELSEC-Anweisungsliste

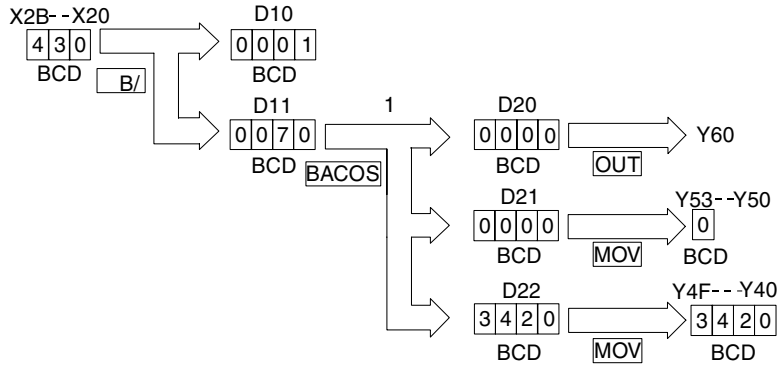
```
MELSEC  LD      X0
        MOV     K1
                D0
        LDI     X0
        MOV     K0
                D0
        LD      SM400
        MOV     K1X30
                D1
        MOV     K4X20
                D2
        LD=     D1
                K1
        AND<>   D2
                K0
        OR>     D1
                K1
        OUT     M0
        LDI     M0
        BACOS   D0
                K4Y40
```

IEC-Anweisungsliste

```
LD          X0
MOV_M       1 , var_D0[0]
LDN         X0
MOV_M       0 , var_D0[0]
LD          SM400
MOV_M       K1X30 , var_D0[1]
MOV_M       K4X20 , var_D0[2]
LD_EQ_M     var_D0[1] , 1
AND_NE_M    var_D0[2] , 0
OR_GT_M     var_D0[1] , 1
ST          M0
LDN         M0
BACOS_MD    var_D0 , K4Y40
```

Kontaktplan





[1] Arcuscosinus-Berechnung

HINWEIS *Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.12.19 BATAN, BATANP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●[2]

[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

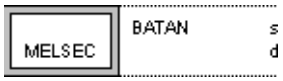
[2] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	—	—	—	—	—	SM0	3
d	●	●	●	●	●	●	●	—	—		

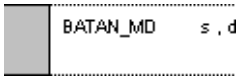
GX IEC Developer

MELSEC-Anweisungsliste



Kontaktplan

IEC-Anweisungsliste



GX Developer

Variablen

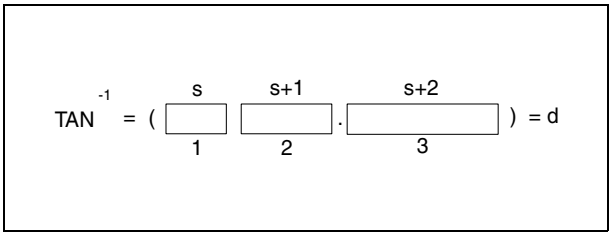
Operand	Befehlswert	Datentyp
s	Erste Adresse des Operanden, in dem der Tangenswert, der zur Ausführung der BATAN-Anweisung (Arcustangens) notwendig ist, gespeichert ist.	BCD 4-stellig
d	Erste Adresse des Operanden, in dem das Ergebnis gespeichert wird.	

Funktions-weise

Arcustangensberechnung mit BCD-Daten

BATAN Arcustangensberechnung

Die BATAN-Anweisung berechnet aus dem Tangenswert in s, s+1 und s+2 die Winkeldaten und speichert das Ergebnis in d.



[1] Vorzeichen
[2] Integerteil
[3] Nachkommastellen

Das Vorzeichen in s nimmt bei positivem Wert den Wert 0 und bei negativem Wert den Wert 1 an.

Der Integerteil vor dem Dezimalkomma und die Nachkommastellen dürfen nur BCD-Werte zwischen 0 und 9999.9999 sein.

Der Zahlenwert des Ergebnisses in d kann ein BCD-Wert zwischen 0° und 90° oder zwischen 270° und 360° sein.

Das Berechnungsergebnis wird von der 5-ten Nachkommastelle an gerundet.

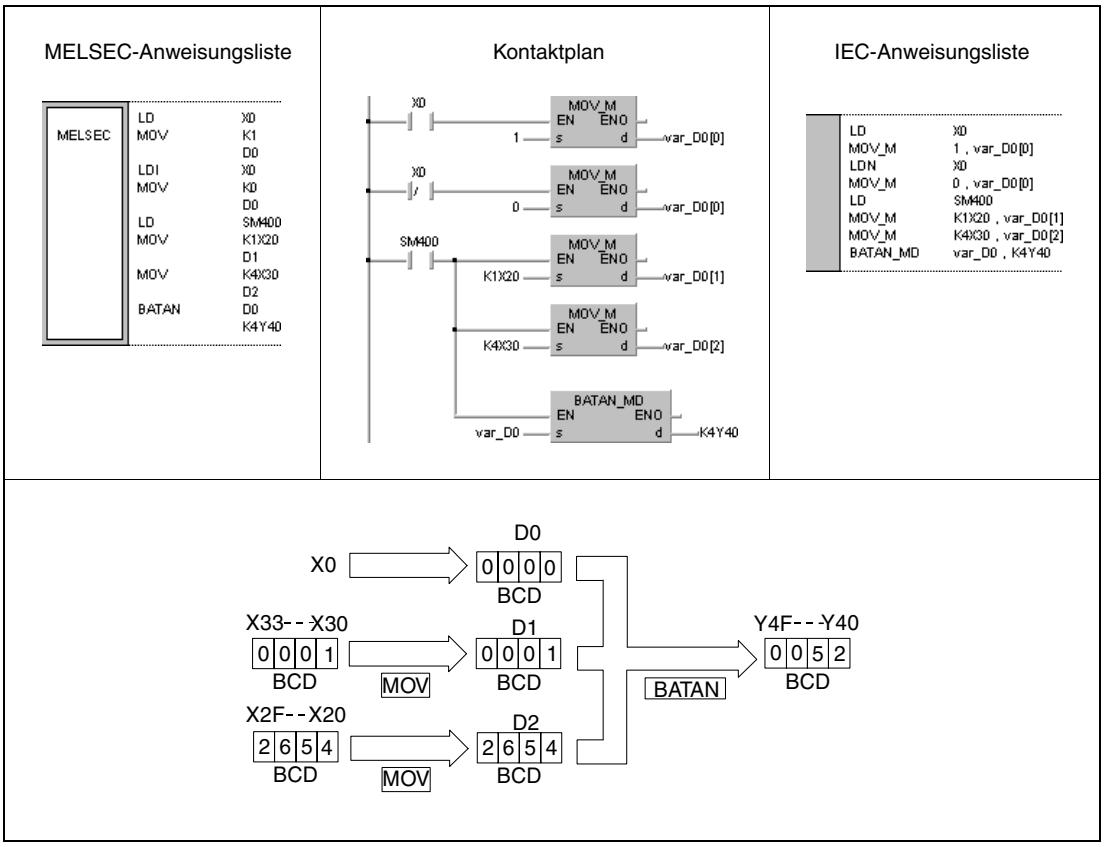
Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die in s bis s+2 angegebenen Daten sind keine BCD-Daten (Fehlercode 4100).

Beispiel BATAN

Das folgende Programm berechnet für die Einschaltdauer von SM400 den Arcustangenswert aus dem Vorzeichenwert an X0 (positiv = X0 gesetzt, negativ = X0 nicht gesetzt), dem 1-stelligen BCD-Integerteil an X20 bis X23 und den Nachkommastellen des 4-stelligen BCD-Werts an X30 bis X3F. Der errechnete Winkelwert wird als 4-stelliger BCD-Wert an Y40 bis Y4F ausgegeben.



[1] Arcustangens-Berechnung

HINWEIS *Diese Programmbeispiele sind ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- oder Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.13 Datenkontrollanweisungen

Die in der Tabelle angegebenen Datenkontrollanweisungen besitzen Ein- und Ausgangsoperanden. Die 16- und 32-Bit-Binärdaten der Eingangsoperanden werden nach einer Ausgangswertbegrenzung oder der Bildung eines Eingangs-/Ausgangsoffsets wieder an die Ausgangsoperanden ausgegeben.

Funktion	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor
Begrenzung des Ausgangswertebereichs von 16-/32-Bit-Binärdaten	LIMIT	LIMIT_MD
	LIMITP	LIMIT_P_MD
	DLIMIT	DLIMIT_MD
	DLIMITP	DLIMIT_P_MD
Eingangsoffsetwert von 16-/32-Bit-Binärdaten	BAND	BAND_MD
	BANDP	BAND_P_MD
	DBAND	DBAND_MD
	DBANDP	DBAND_P_MD
Ausgangsoffsetwert von 16-/32-Bit-Binärdaten	ZONE	ZONE_MD
	ZONEP	ZONE_P_MD
	DZONE	DZONE_MD
	DZONEP	DZONE_P_MD

HINWEIS *Nutzen Sie in den IEC-Editoren die IEC-Anweisungen.*

7.13.1 LIMIT, LIMITP, DLIMIT, DLIMITP

CPU

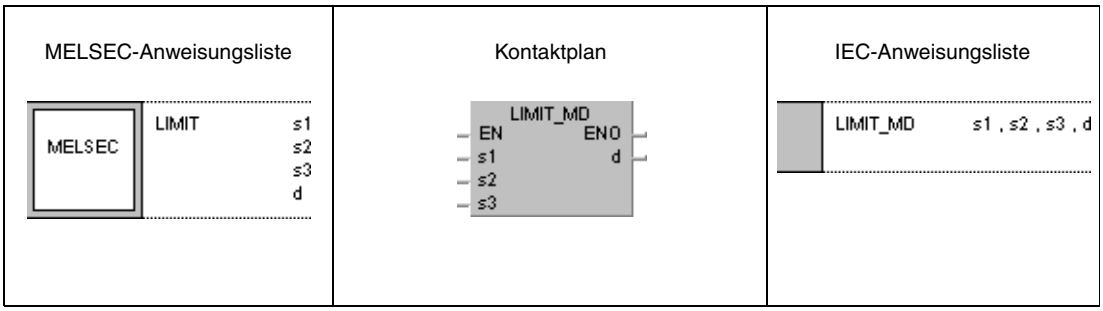
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●

[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

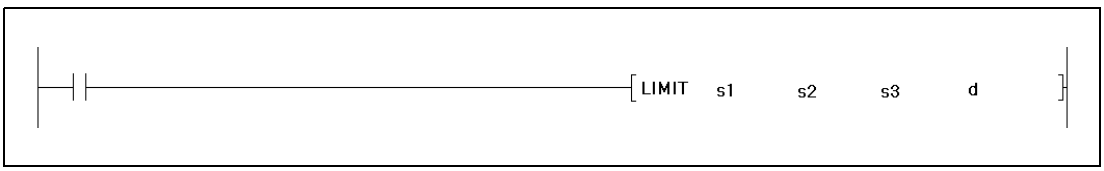
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	●	●	●	●	●	●	●	●	—	SM0	5
s2	●	●	●	●	●	●	●	●	—		
s3	●	●	●	●	●	●	●	●	—		
d	●	●	●	●	●	●	●	—	—		

GX IEC Developer



GX Developer



Variablen

Operand	Befehlswert	Datentyp
s1	Unterer Grenzwert (kleinster Ausgangsschwellwert) des Ausgangswertes.	BIN-16-Bit
s2	Oberer Grenzwert (größter Ausgangsschwellwert) des Ausgangswertes.	
s3	Zu begrenzender Eingangswert.	
d	Adresse des Operanden, in welchem der begrenzte Ausgangswert gespeichert wird.	

Funktions-weise

Begrenzung des Ausgangswertebereichs von 16- und 32-Bit-Binärdaten

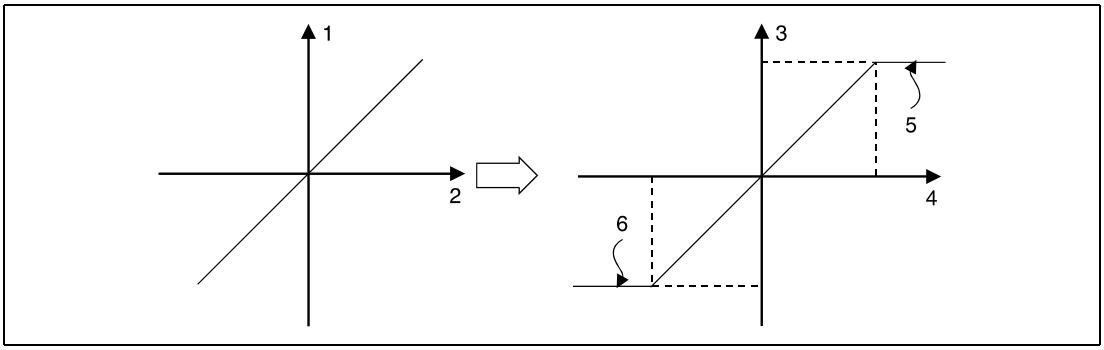
LIMIT Begrenzungsanweisung für 16-Bit-Binärdaten

Die LIMIT-Anweisung überprüft die im Operanden s3 angegebenen Daten darauf, ob sie innerhalb des in s1 angegebenen unteren Grenzwertes bzw. des in s2 angegebenen oberen Grenzwertes liegen. Abhängig von dem Prüfergebnis werden die Werte wie folgt in dem in d angegeben Operanden gespeichert:

Wenn der in s3 angegebene Datenwert kleiner als der in s1 angegebene untere Grenzwert ist, wird der untere Grenzwert in dem in d angegebenen Operanden gespeichert.

Wenn der in s3 angegebene Datenwert größer als der in s2 angegebene obere Grenzwert ist, wird der obere Grenzwert in dem in d angegebenen Operanden gespeichert.

Wenn der in s3 angegebene Datenwert zwischen dem unteren und oberen Grenzwert liegt, wird der entsprechende Datenwert in dem in d angegeben Operanden gespeichert.



[1] Ausgangswert

[2] Eingangswert

[3] Ausgangswert (d)

[4] Eingangswert (s3)

[5] Oberer Grenzwert (s2)

[6] Unterer Grenzwert (s1)

Der in s1, s2 und s3 angegebene Wert kann zwischen -32768 und 32767 liegen.

Wenn nur der obere Grenzwert kontrolliert werden soll, muss für den unteren Grenzwert in s1 der Wert -32768 eingetragen werden.

Wenn nur der untere Grenzwert kontrolliert werden soll, muss für den oberen Grenzwert in s2 der Wert 32767 eingetragen werden.

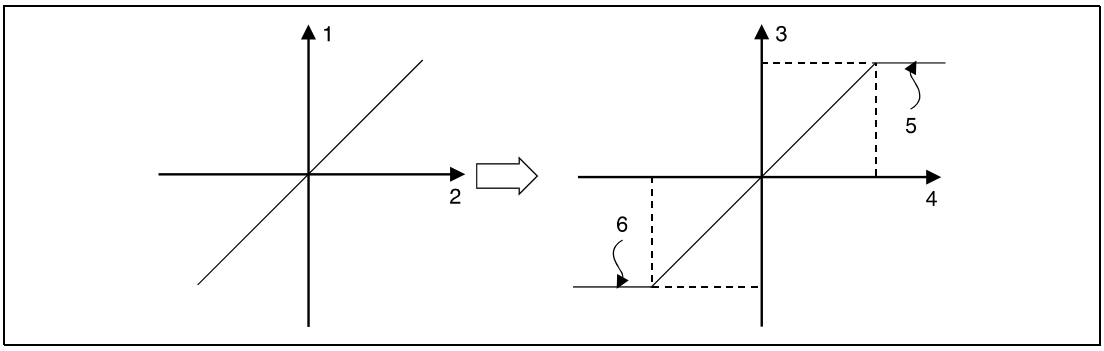
DLIMIT Begrenzungsanweisung für 32-Bit-Binärdaten

Die DLIMIT-Anweisung überprüft die in den Operanden s3 und (s3)+1 angegebenen Daten darauf, ob sie innerhalb des in s1 und (s1)+1 angegebenen unteren Grenzwertes bzw. des in s2 und (s2)+1 angegebenen oberen Grenzwertes liegen. Abhängig von dem Prüfergebnis werden die Werte wie folgt in den in d und d+1 angegeben Operanden gespeichert:

Wenn der in s3 und (s3)+1 angegebene Datenwert kleiner als der in s1 und (s1)+1 angegebene untere Grenzwert ist, wird der untere Grenzwert in den in d und d+1 angegebenen Operanden gespeichert.

Wenn der in s3 und (s3)+1 angegebene Datenwert größer als der in s2 und (s2)+1 angegebene obere Grenzwert ist, wird der obere Grenzwert in den in d und d+1 angegebenen Operanden gespeichert.

Wenn der in s3 und (s3)+1 angegebene Datenwert zwischen dem unteren und oberen Grenzwert liegt, wird der entsprechende Datenwert in den in d und d+1 angegebenen Operanden gespeichert.



[1] Ausgangswert

[2] Eingangswert

[3] Ausgangswert (d+1, d)

[4] Eingangswert ((s3)+1, s3)

[5] Oberer Grenzwert ((s2)+1, s2)

[6] Unterer Grenzwert ((s1)+1, s1)

Der in s1 und (s1)+1 und s2, (s2)+1, s3 und (s3)+1 angegebene Wert kann zwischen -2147483648 und 2147483647 liegen.

Wenn nur der obere Grenzwert kontrolliert werden soll, muss für den unteren Grenzwert in s1 und (s1)+1 der Wert -2147483648 eingetragen werden.

Wenn nur der untere Grenzwert kontrolliert werden soll, muss für den oberen Grenzwert in s2 und (s2)+1 der Wert 2147483647 eingetragen werden.

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Der in s1 ((s1)+1) angegebene Wert ist größer als der in s2 ((s2)+1) angegebene Wert (Fehlercode 4100).

Beispiel 1 LIMITP

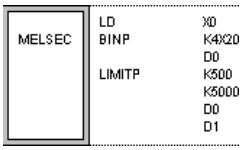
Im folgenden Programm wird mit positiver Flanke von X0 der Wert der BCD-Daten an X20 bis X2F daraufhin überprüft, ob er sich zwischen der Untergrenze 500 und der Obergrenze 5000 befindet. Das Ergebnis der Überprüfung wird in D1 gespeichert.

Ist der Wert in D0 größer als 5000, wird in D1 der Wert 5000 eingetragen.

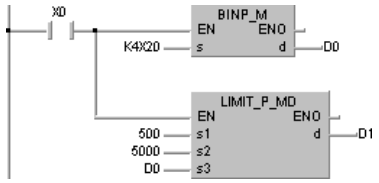
Ist der Wert in D0 kleiner als 500, wird in D1 der Wert 500 eingetragen.

Liegt der Wert in D0 zwischen 500 und 5000, wird der entsprechende Wert in D1 gespeichert.

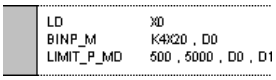
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



Beispiel 2 DLIMIT

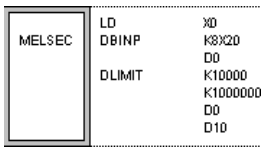
Im folgenden Programm wird mit positiver Flanke von X0 der Wert der BCD-Daten an X20 bis X3F daraufhin überprüft, ob er sich zwischen der Untergrenze 10000 und der Obergrenze 1000000 befindet. Das Ergebnis der Überprüfung wird in D10 und D11 gespeichert.

Ist der Wert in D0 und D1 größer als 1000000, wird in D10 und D11 der Wert 1000000 eingetragen.

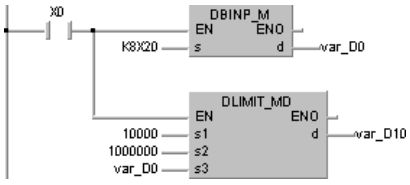
Ist der Wert in D0 und D1 kleiner als 10000, wird in D10 und D11 der Wert 10000 eingetragen.

Liegt der Wert in D0 zwischen 10000 und 1000000, wird der entsprechende Wert in D10 und D11 gespeichert.

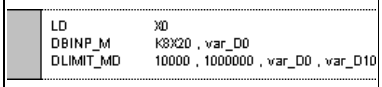
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



7.13.2 BAND, BANDP, DBAND, DBANDP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●

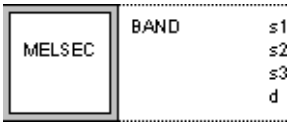
[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

Operanden MELSEC Q

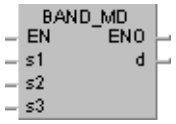
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	●	●	●	●	●	●	●	●	—	SM0	5
s2	●	●	●	●	●	●	●	●	—		
s3	●	●	●	●	●	●	●	●	—		
d	●	●	●	●	●	●	●	—	—		

GX IEC Developer

MELSEC-Anweisungsliste

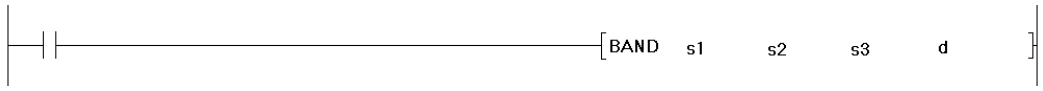


Kontaktplan



IEC-Anweisungsliste

GX Developer



Variablen

Operand	Befehlswert	Datentyp
s1	Unterer Eingangsoffsetwert (in diesem Bereich ist der Ausgabewert 0).	BIN-16-Bit
s2	Oberer Eingangsoffsetwert (in diesem Bereich ist der Ausgabewert 0).	
s3	Eingangswert, auf den der Offsetwert angewendet wird.	
d	Adresse des Operanden, in welchem das Subtraktionsergebnis aus Eingangswert und Offsetwert gespeichert wird.	

Funktions-weise

Eingangsoffset von 16- und 32-Bit-Binärdaten

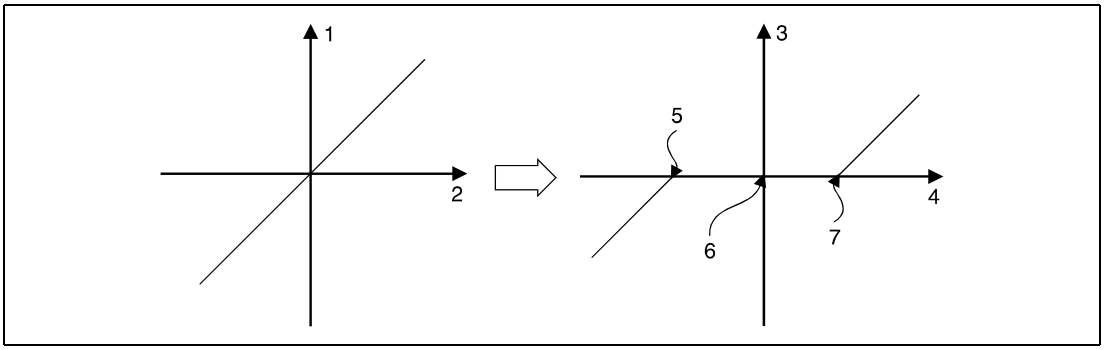
BAND Eingangsoffset von 16-Bit-Binärdaten

Die BAND-Anweisung subtrahiert von der in dem in s3 angegebenen Operanden gespeicherten 16-Bit-Binärzahl einen unteren (negativen) und oberen (positiven) Offsetwert. Der untere Offsetwert ist in s1 und der obere Offsetwert in s2 gespeichert. Das Ergebnis wird in Abhängigkeit von dem Eingangswert in dem in d angegebenen Operanden wie folgt gespeichert:

Wenn der Datenwert in s3 kleiner als der untere Offsetwert in s1 ist, wird das Ergebnis aus der Subtraktion s3 - s1 in dem in d angegebenen Operanden gespeichert.

Wenn der Datenwert in s3 größer als der obere Offsetwert in s2 ist, wird das Ergebnis der Subtraktion s3 - s2 in dem in d angegebenen Operanden gespeichert.

Wenn sich der Datenwert in s3 innerhalb der beiden Offsetbereiche befindet, wird der Wert 0 in dem in d angegebenen Operanden gespeichert.



[1] Ausgangswert

[2] Eingangswert

[3] Ausgangswert (d)

[4] Eingangswert (s3)

[5] Unterer (negativer) Eingangsoffset (s1)

[6] Ausgangswert = 0

[7] Oberer (positiver) Eingangsoffset (s2)

Der in s1, s2 und s3 angegebene Wert kann zwischen -32768 und 32767 liegen.

Wenn das Subtraktionsergebnis den Bereich zwischen -32768 und 32767 verläßt, findet folgender Vorgang statt:

Bei Unterschreiten des Wertes -32768 wird der Rest der Subtraktion bei 32767 beginnend durchgeführt. Wenn in s3 z.B. der Wert -32768 gespeichert ist und s1 mit dem Wert 10 davon subtrahiert wird, ergibt sich

$$-32768 - 10 = 8000_H - A_H = 7FF6_H = 32758.$$

Bei Überschreiten des Wertes 32760 wird der Rest der Subtraktion beginnend bei -32768 durchgeführt.

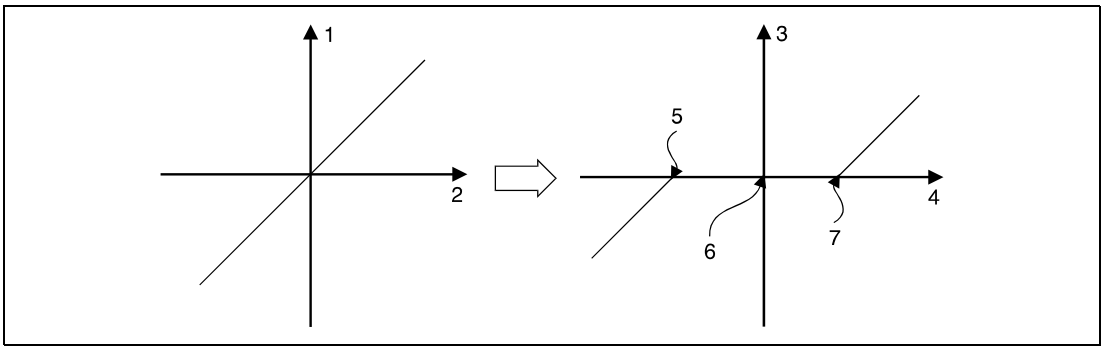
DBAND Eingangsoffset von 32-Bit-Binärdaten

Die DBAND-Anweisung subtrahiert von der in den in s3 und (s3)+1 angegebenen Operanden gespeicherten 32-Bit-Binärzahl einen unteren (negativen) und oberen (positiven) Offsetwert. Der untere Offsetwert ist in s1 und (s1)+1 und der obere Offsetwert in s2 und (s2)+1 gespeichert. Das Ergebnis wird in Abhängigkeit von dem Eingangswert in den in d und d+1 angegebenen Operanden wie folgt gespeichert:

Wenn der Datenwert in s3 und (s3)+1 kleiner als der untere Offsetwert in s1 und (s1)+1 ist, wird das Ergebnis aus der Subtraktion s3, (s3)+1 - s1, (s1)+1 in dem in d und d+1 angegebenen Operanden gespeichert.

Wenn der Datenwert in s3 und (s3)+1 größer als der obere Offsetwert in s2 und (s2)+1 ist, wird das Ergebnis der Subtraktion s3, (s3)+1 - s2, (s2)+1 in dem in d und d+1 angegebenen Operanden gespeichert.

Wenn sich der Datenwert in s3 und (s3)+1 innerhalb der beiden Offsetbereiche befindet, wird der Wert 0 in dem in d und d+1 angegebenen Operanden gespeichert.



[1] Ausgangswert

[2] Eingangswert

[3] Ausgangswert (d+1, d)

[4] Eingangswert ((s3)+1, s3)

[5] Unterer (negativer) Eingangsoffset ((s1)+1, s1)

[6] Ausgangswert = 0

[7] Oberer (positiver) Eingangsoffset ((s2)+1, s2)

Der in s1 und (s1)+1, s2 und (s2)+1, s3 und (s3)+1 angegebene Wert kann zwischen -2147483648 und 2147483647 liegen.

Wenn das Subtraktionsergebnis den Bereich zwischen -2147483648 und 2147483647 verläßt, findet folgender Vorgang statt:

Bei Unterschreiten des Wertes -2147483648 wird der Rest der Subtraktion bei 2147483647 beginnend durchgeführt. Wenn in s3 und (s3)+1 z.B. der Wert -2147483648 gespeichert ist und s1 und (s1)+1 mit dem Wert 1000 davon subtrahiert werden, ergibt sich

$$-2147483648 - 1000 = 80000000_H - 3E8_H = 7FFFFC18_H = 2147482648.$$

Bei Überschreiten des Wertes 2147483647 wird der Rest der Subtraktion beginnend bei dem Wert -2147483648 durchgeführt.

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Der in s1 ((s1)+1) angegebene Wert ist größer als der in s2 ((s2)+1) angegebene Wert (Fehlercode 4100).

Beispiel 1 BANDP

Im folgenden Programm werden mit positiver Flanke von X0 von den BCD-Daten an X20 bis X2F der untere (negative) Offsetwert -1000 und der obere (positive) Offsetwert 1000 subtrahiert. Das Ergebnis wird in D1 gespeichert.

Ist der Wert in D0 größer als 1000, wird in D1 der Wert D0 - 1000 eingetragen.

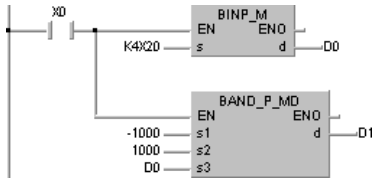
Ist der Wert in D0 kleiner als -1000, wird in D1 der Wert D0 - (-1000) eingetragen.

Liegt der Wert in D0 zwischen -1000 und 1000, wird der Wert 0 in D1 gespeichert.

MELSEC-Anweisungsliste

```
MELSEC
        LD      X0
        BINP    K4X20
                D0
        BANDP   K-1000
                K1000
                D0
                D1
```

Kontaktplan



IEC-Anweisungsliste

```
LD          X0
BINP_M      K4X20 , D0
BAND_P_MD   -1000 , 1000 , D0 , D1
```

Beispiel 2 DBANDP

Im folgenden Programm werden mit positiver Flanke von X0 von den BCD-Daten an X20 bis X3F der untere (negative) Offsetwert -10000 und der obere (positive) Offsetwert 10000 subtrahiert. Das Ergebnis wird in D10 und D11 gespeichert.

Ist der Wert in D0 und D1 größer als 10000, wird in D10 und D11 der Wert D0, D1 - 10000 eingetragen.

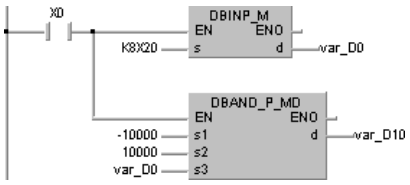
Ist der Wert in D0 und D1 kleiner als -10000, wird in D10 und D11 der Wert D0, D1 - (-10000) eingetragen.

Liegt der Wert in D0 und D1 zwischen -10000 und 10000, wird der Wert 0 in D10 und D11 gespeichert.

MELSEC-Anweisungsliste

```
MELSEC
        LD      X0
        DBINP   K8X20
                D0
        DBANDP  K-10000
                K10000
                D0
                D10
```

Kontaktplan



IEC-Anweisungsliste

```
LD          X0
DBINP_M     K8X20 , var_D0
DBAND_P_MD  -10000 , 10000 , var_D0 , var_D10
```

7.13.3 ZONE, ZONEP, DZONE, DZONEP

CPU

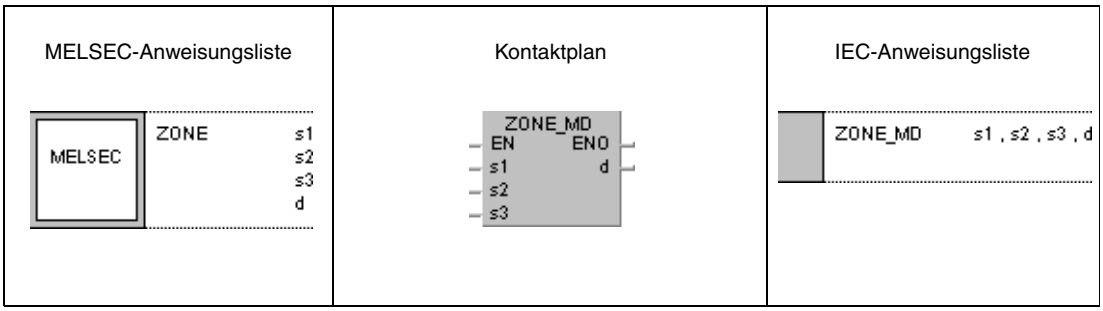
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●

[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

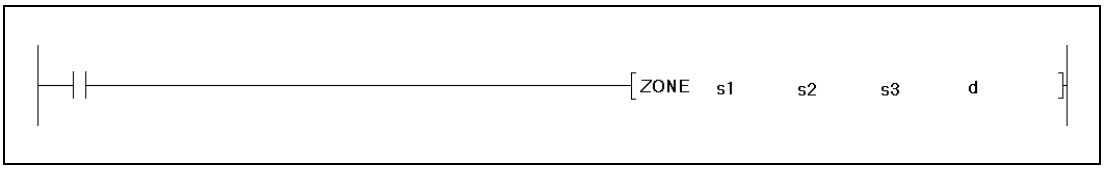
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	●	●	●	●	●	●	●	●	—	SM0	5
s2	●	●	●	●	●	●	●	●	—		
s3	●	●	●	●	●	●	●	●	—		
d	●	●	●	●	●	●	●	—	—		

GX IEC Developer



GX Developer



Variablen

Operand	Befehlswert	Datentyp
s1	Negativer Offsetwert, der zu dem Eingangswert addiert wird.	BIN-16-Bit
s2	Positiver Offsetwert, der zu dem Eingangswert addiert wird.	
s3	Eingangswert, auf den der Offsetwert angewendet wird.	
d	Adresse des Operanden, in welchem die Summe aus Eingangswert und Offsetwert gespeichert wird.	

Funktions-weise

Ausgangsoffset von 16- und 32-Bit-Binärdaten

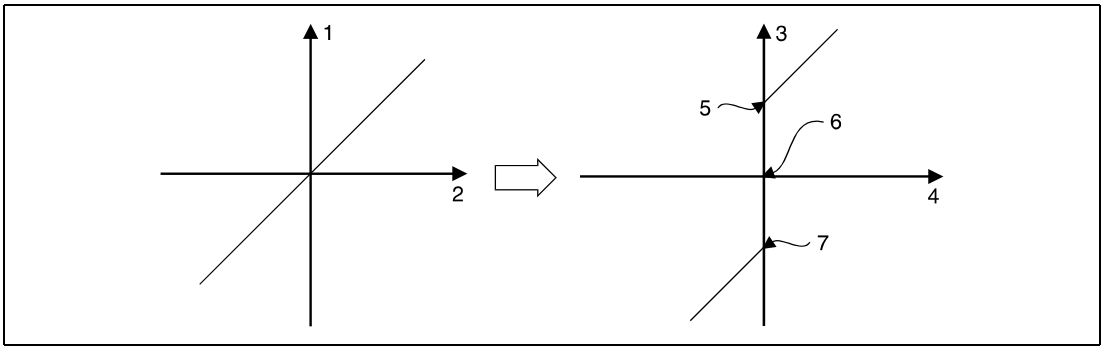
ZONE Ausgangsoffset von 16-Bit-Binärdaten

Die ZONE-Anweisung addiert zu der 16-Bit-Binärzahl in dem in s3 angegebenen Operanden einen negativen und positiven Offsetwert. Der negative Offsetwert ist in s1 und der positive Offsetwert in s2 gespeichert. Das Ergebnis wird in Abhängigkeit von dem Eingangswert in dem in d angegebenen Operanden wie folgt gespeichert:

Wenn der Datenwert in s3 kleiner als 0 ist, wird das Ergebnis aus der Addition s3 + s1 in dem in d angegebenen Operanden gespeichert.

Wenn der Datenwert in s3 größer als 0 ist, wird das Ergebnis der Addition s3 + s2 in dem in d angegebenen Operanden gespeichert.

Wenn der Datenwert in s3 gleich 0 ist, wird der Wert 0 in dem in d angegebenen Operanden gespeichert.



[1] Ausgangswert

[2] Eingangswert

[3] Ausgangswert (d)

[4] Eingangswert (s3)

[5] Oberer (positiver) Ausgangsoffset (s2)

[6] Eingangswert = 0

[7] Unterer (negativer) Ausgangsoffset (s1)

Der in s1, s2 und s3 angegebene Wert kann zwischen -32768 und 32767 liegen.

Wenn das Additionsergebnis den Bereich zwischen -32768 und 32767 verläßt, findet folgender Vorgang statt:

Bei Unterschreiten des Wertes -32768 wird der Rest der Operation bei 32767 beginnend durchgeführt. Wenn in s3 z.B. der Wert -32768 gespeichert ist und s1 mit dem Wert -100 addiert wird, ergibt sich

$$-32768 + (-100) = 8000_H + FF9C_H = 7F9C_H = 32668.$$

Bei Überschreiten des Wertes 32760 wird der Rest der Addition beginnend bei -32768 durchgeführt.

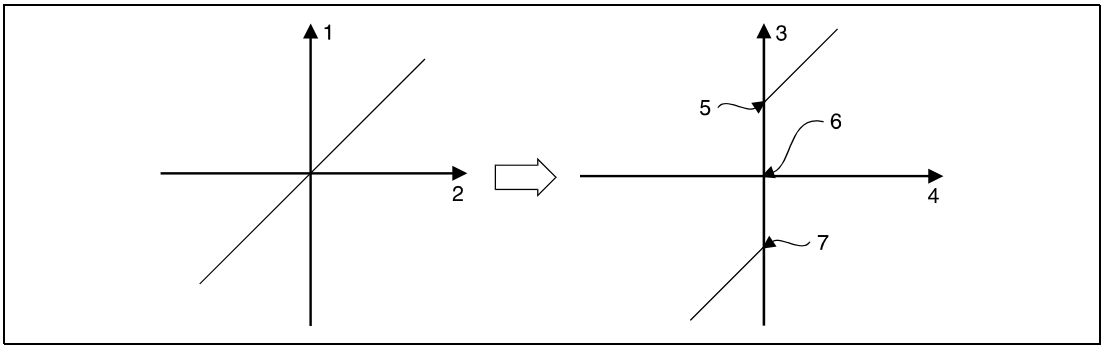
DZONE Ausgangsoffset von 32-Bit-Binärdaten

Die DZONE-Anweisung addiert zu der 32-Bit-Binärzahl in den in s3 und (s3)+1 angegebenen Operanden einen negativen und positiven Offsetwert. Der negative Offsetwert ist in s1 und (s1)+1 und der positive Offsetwert in s2 und (s2)+1 gespeichert. Das Ergebnis wird in Abhängigkeit von dem Eingangswert in den in d und d+1 angegebenen Operanden wie folgt gespeichert:

Wenn der Datenwert in s3 und (s3)+1 kleiner als 0 ist, wird das Ergebnis aus der Addition s3, (s3)+1 + s1, (s1)+1 in dem in d und d+1 angegebenen Operanden gespeichert.

Wenn der Datenwert in s3 und (s3)+1 größer als 0 ist, wird das Ergebnis der Addition s3, (s3)+1 + s2, (s2)+1 in dem in d und d+1 angegebenen Operanden gespeichert.

Wenn der Datenwert in s3 und (s3)+1 gleich 0 ist, wird der Wert 0 in dem in d und d+1 angegebenen Operanden gespeichert.



[1] Ausgangswert

[2] Eingangswert

[3] Ausgangswert (d+1, d)

[4] Eingangswert ((s3)+1, s3)

[5] Oberer (positiver) Ausgangsoffset ((s2)+1, s2)

[6] Eingangswert = 0

[7] Unterer (negativer) Ausgangsoffset ((s1)+1, s1)

Der in s1 und (s1)+1, s2 und (s2)+1, s3 und (s3)+1 angegebene Wert kann zwischen -2147483648 und 2147483647 liegen.

Bei Unterschreiten des Wertes -2147483648 wird der Rest der Operation bei 2147483647 beginnend durchgeführt. Wenn in s3 und (s3)+1 z.b. der Wert -2147483648 gespeichert ist und s1 und (s1)+1 mit dem Wert -1000 addiert werden, ergibt sich

$$-2147483648 + (-1000) = 80000000_H + FFFFFC18_H = 7FFFFC18_H = 2147482648.$$

Bei Überschreiten des Wertes 2147483647 wird der Rest der Addition beginnend bei dem Wert -2147483648 durchgeführt.

Beispiel 1 ZONEP

Im folgenden Programm werden mit positiver Flanke von X0 zu den BCD-Daten an X20 bis X2F der negative Offsetwert -100 und der positive Offsetwert 100 addiert. Das Ergebnis wird in D1 gespeichert.

Ist der Wert in D0 größer als 0, wird in D1 der Wert D0 + 100 eingetragen.

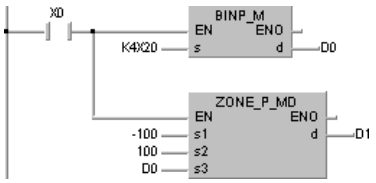
Ist der Wert in D0 kleiner als 0, wird in D1 der Wert D0 + (-100) eingetragen.

Ist der Wert in D0 gleich 0, wird der Wert 0 in D1 gespeichert.

MELSEC-Anweisungsliste

```
MELSEC
        LD      X0
        BINP    K4X20
                D0
        ZONEP   K-100
                K100
                D0
                D1
```

Kontaktplan



IEC-Anweisungsliste

```
LD              X0
BINP_M          K4X20 , D0
ZONE_P_MD       -100 , 100 , D0 , D1
```

Beispiel 2 DZONEP

Im folgenden Programm werden mit positiver Flanke von X0 zu den BCD-Daten an X20 bis X3F der negative Offsetwert -10000 und der positive Offsetwert 10000 addiert. Das Ergebnis wird in D10 und D11 gespeichert.

Ist der Wert in D0 und D1 größer als 0, wird in D10 und D11 der Wert D0, D1 + 10000 eingetragen.

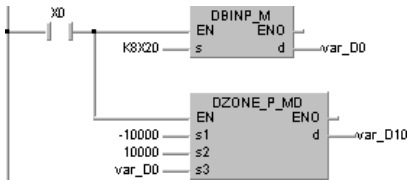
Ist der Wert in D0 und D1 kleiner als 0, wird in D10 und D11 der Wert D0, D1 + (-10000) eingetragen.

Ist der Wert in D0 und D1 gleich 0, wird der Wert 0 in D10 und D11 gespeichert.

MELSEC-Anweisungsliste

```
MELSEC
        LD      X0
        DBINP   K8X20
                D0
        DZONEP  K-10000
                K10000
                D0
                D10
```

Kontaktplan



IEC-Anweisungsliste

```
LD              X0
DBINP_M         K8X20 , var_D0
DZONE_P_MD      -10000 , 10000 , var_D0 , var_D10
```

7.14 Umschaltanweisungen für File-Registerblöcke

Die Umschaltanweisungen ermöglichen das Umschalten zwischen File-Registerblöcken und zwischen Dateinamen in File-Registern. Die Tabelle liefert eine Übersicht der Anweisungen.

Funktion	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor
Umschaltung zwischen File-Registerblöcken	RSET	RSET_MD
		RSET_K_MD
	RSETP	RSET_P_MD
		RSET_K_P_MD
Umschaltung zwischen Dateien in File-Registern	QDRSET	QDRSET_M
	QDRSETP	QDRSET_P_MD
Umschaltung zwischen Dateien für Kommentardaten in File-Registern	QCDSET	QCDSET_M
	QCDSETP	QCDSET_P_MD

7.14.1 RSET, RSETP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●

[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

Operanden MELSEC Q

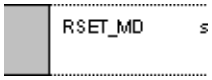
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	●	●	●	●	●	●	●	●	—	SM0	2

GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan

IEC-Anweisungsliste



GX Developer

Variablen

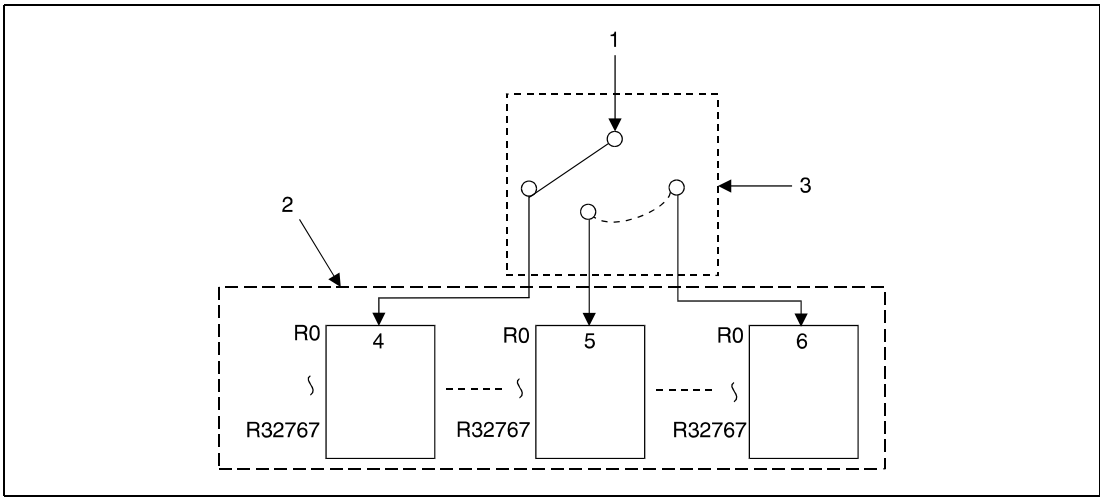
Operand	Befehlswert	Datentyp
s	Adresse des File-Registerblocks oder Operand, in dem diese Adresse gespeichert ist.	BIN-16-Bit

Funktions-weise

Umschaltung zwischen File-Registerblöcken

RSET Umschaltanweisung für File-Registerblöcke

Die RSET-Anweisung schaltet von einem File-Registerblock, der im Programm benutzt wird, zu dem File-Registerblock mit der in s angegebenen Adresse um. Nach der Umschaltung werden alle vom Ablaufprogramm benutzten File-Register (R0 - R32767) aus dem Block benutzt, in den umgeschaltet wurde.



[1] Verarbeitung mit File-Registerzugriff

[2] Vom Programm benutzte Datei

[3] Auswahl des File-Registerblocks (s)

[4] Block 0

[5] Block 1

[6] Block n

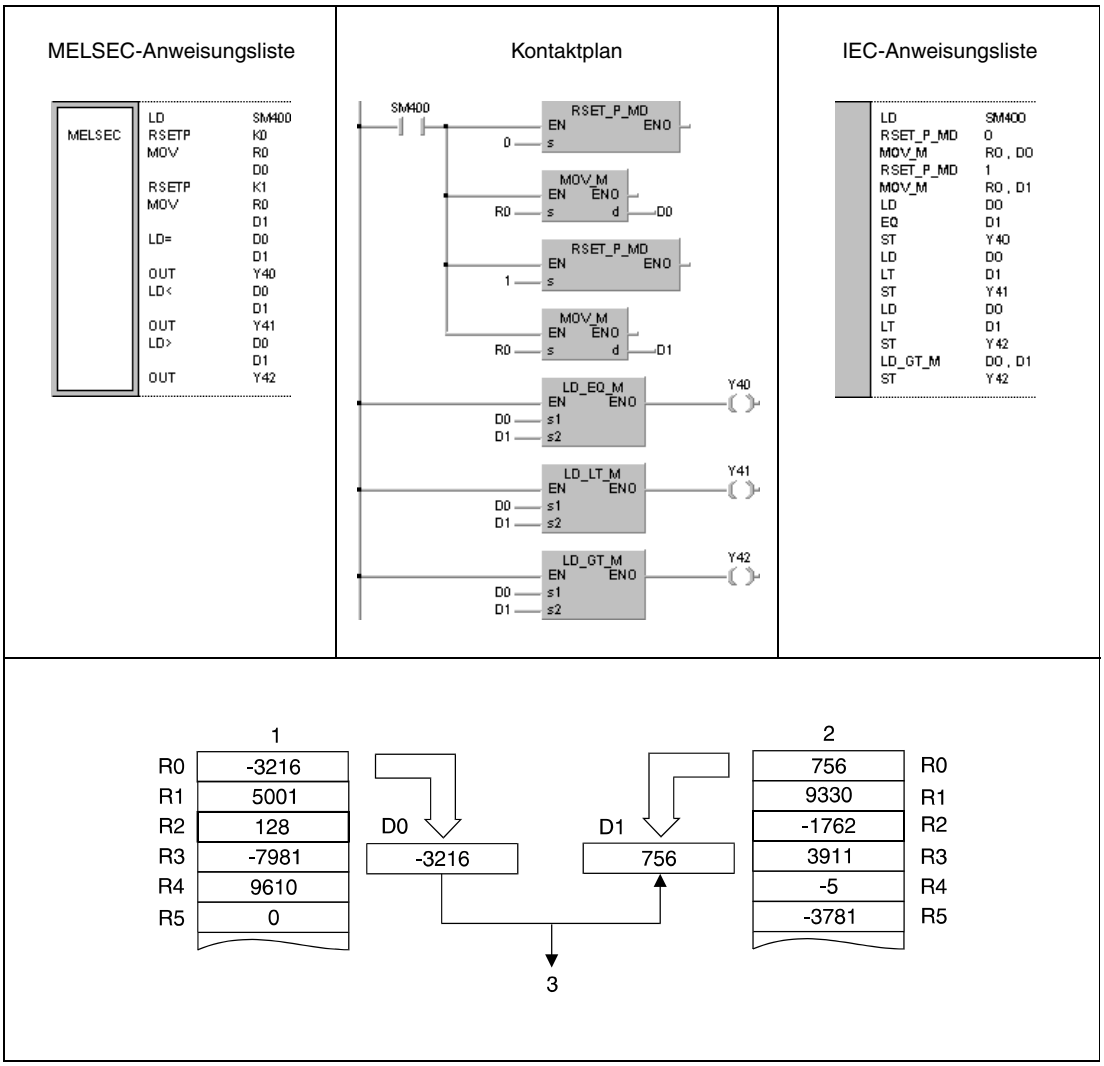
Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- An der in s angegebenen Adresse befindet sich kein File-Registerblock (Fehlercode 4100).
- In dem in s adressierten Registerblock befinden sich keine File-Register (Fehlercode 4101).

Beispiel RSETP

Im folgenden Programm werden mit positiver Flanke von SM400 die File-Register R0 in File-Registerblock 0 und 1 verglichen. Die File-Registerblöcke 0 und 1 werden über die RSET-Anweisung adressiert. Die beiden File-Register R0 werden über die MOV-Anweisung ausgelesen. Ist der Wert von R0 (Block 0) gleich dem von R0 (Block1), wird der Ausgang Y40 gesetzt. Ist der Wert von R0 (Block 0) kleiner als der Wert von R0 (Block 1), wird der Ausgang Y41 gesetzt. Ist der Wert von R0 (Block 0) größer als der Wert von R0 (Block 1), wird der Ausgang Y42 gesetzt.



[1] Block 0

[2] Block 1

[3] Y41 wird gesetzt, da D0 kleiner D1 ist

7.14.2 QDRSET, QDRSETP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●[1]

[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten $	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	—	—	—	●	—	SM0	2 + n[1)]

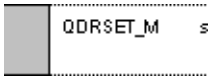
[1] n = (Anzahl Zeichen im Dateinamen/2) = Anzahl zusätzlicher Schritte. (Nachkommastellen werden aufgerundet).

GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan

IEC-Anweisungsliste



GX Developer

Variablen

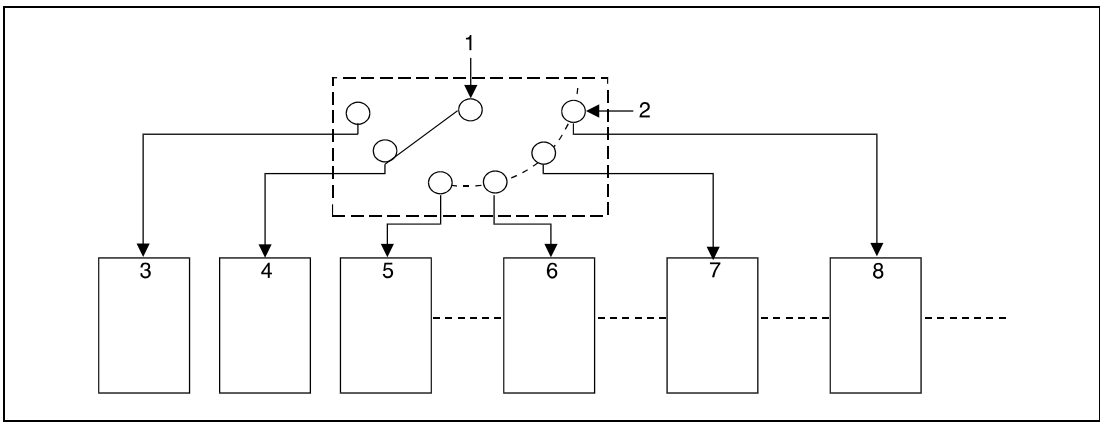
Operand	Befehlswert	Datentyp
s	Laufwerksadresse und Name der Datei, in welcher sich die File-Registerdatei befindet, in die umgeschaltet wird, oder Operand, in dem diese Daten gespeichert sind.	Zeichenfolge

Funktionsweise

Umschaltung zwischen File-Registerdateien

QDRSET Umschaltanweisung für File-Registerdateien

Die QDRSET-Anweisung schaltet von einer im Programm benutzten File-Registerdatei in die in s angegebene File-Registerdatei um. Nach der Umschaltung werden alle vom Ablaufprogramm benutzten File-Register (R0 - R32767) aus dem Block 0 der File-Registerdatei benutzt, in die umgeschaltet wurde. Die Umschaltung zwischen den Blöcken erfolgt mittels der RSET-Anweisung.



[1] Verarbeitung mit File-Registerzugriff

[2] Auswahl des Laufwerks und der Datei (s)

[3] Laufwerk 1, Datei A

[4] Laufwerk 1, Datei B

[5] Laufwerk 1, Datei C

[6] Laufwerk 2, Datei A

[7] Laufwerk 3, Datei A

[8] Laufwerk 4, Datei A

Es können 4 Laufwerke adressiert werden (1 - 4). Die Laufwerksadresse 0 kann nicht vergeben werden, da dieser Bereich für den internen Speicher reserviert ist.

Die Erweiterung .QDR muss bei der Angabe der File-Registerdatei nicht angegeben werden.

Die Auswahl der File-Registerdatei wird durch Angabe der 0 im Hexadezimalcode (00H) als File-Registerdateiname wieder gelöscht.

Die File-Registerdateien, auf die mit der QDRSET-Anweisung umgeschaltet wurde, haben auch in den Fällen Verarbeitungspriorität, in denen die Laufwerksadressen und File-Registerdateinamen in den Parametern angegeben werden.

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

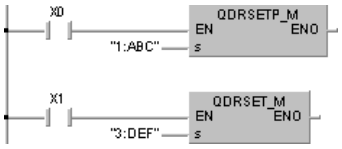
- Die File-Registerdatei befindet sich nicht in dem in s angegebenen Laufwerk (Fehlercode 2410).

Beispiel QDRSET/QDRSETP

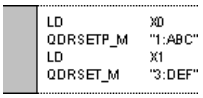
Im folgenden Programm wird mit positiver Flanke von X0 auf die File-Registerdatei ABC.QDR in dem Laufwerk 1 umgeschaltet. Für die Einschaltdauer von X1 wird auf die File-Registerdatei DEF.QDR in dem Laufwerk 3 umgeschaltet.

MELSEC-Anweisungsliste

Kontaktplan



IEC-Anweisungsliste



7.14.3 QCDSET, QCDSETP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●[1]

[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

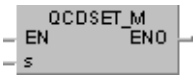
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten $	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	—	—	—	●	—	SM0	2 +n[1)]

[1] n = (Anzahl Zeichen im Dateinamen/2) = Anzahl zusätzlicher Schritte. (Nachkommastellen werden aufgerundet).

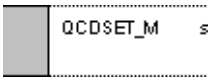
GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

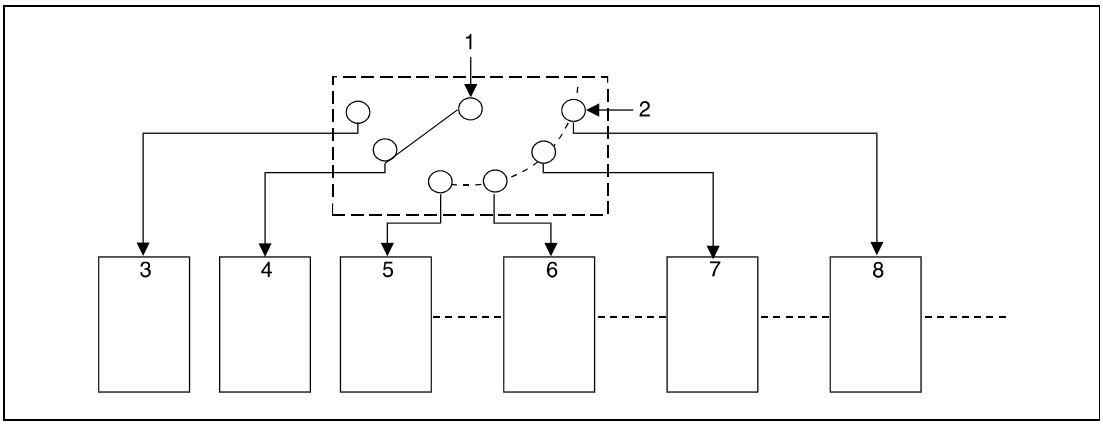
Operand	Befehlswert	Datentyp
s	Laufwerksadresse und Name der Datei, in der sich die Kommentardatei befindet, in die umgeschaltet wird, oder Operand, in dem diese Daten gespeichert sind.	Zeichenfolge

Funktionsweise

Umschaltung zwischen Kommentardateien

QCDSET Umschaltanweisung für Kommentardateien

Die QCDSET-Anweisung schaltet von einer im Programm benutzten Kommentardatei in die in s angegebene Kommentardatei um. Nach der Umschaltung werden alle vom Ablaufprogramm benutzten Kommentardaten aus der Kommentardatei benutzt, in die umgeschaltet wurde.



[1] Verarbeitung mit Kommentardatenzugriff

[2] Auswahl des Laufwerks und der Kommentardatei (s)

[3] Laufwerk 1, Kommentardatei A

[4] Laufwerk 1, Kommentardatei B

[5] Laufwerk 1, Kommentardatei C

[6] Laufwerk 2, Kommentardatei A

[7] Laufwerk 3, Kommentardatei A

[8] Laufwerk 4, Kommentardatei A

Es können 4 Laufwerke adressiert werden (1 - 4). Die Laufwerksadresse 0 kann nicht vergeben werden, da dieser Bereich für den internen Speicher reserviert ist.

Die Erweiterung .QCD muss bei der Angabe der Kommentardatei nicht angegeben werden.

Die Auswahl der Kommentardatei wird durch Angabe der 0 im Hexadezimalcode (00H) als File-Registerdateiname wieder gelöscht.

Die Kommentardateien, auf die mit der QCDSET-Anweisung umgeschaltet wurde, haben auch in den Fällen Verarbeitungspriorität, in denen die Laufwerksadressen und Kommentardateinamen in den Parametern angegeben werden.

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die Kommentardatei befindet sich nicht in dem in s angegebenen Laufwerk (Fehlercode 2410).

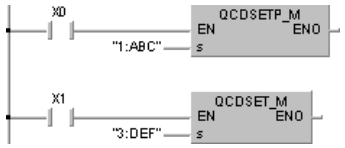
Beispiel QCDSET/QCDSETP

Im folgenden Programm wird mit positiver Flanke von X0 auf die Kommentardatei ABC.QCD in dem Laufwerk 1 umgeschaltet. Für die Einschaltdauer von X1 wird auf die Kommentardatei DEF.QCD in dem Laufwerk 3 umgeschaltet.

MELSEC-Anweisungsliste

```
MELSEC   LD       X0
         QCDSETP  "1:ABC"
         LD       X1
         QCDSET   "3:DEF"
```

Kontaktplan



IEC-Anweisungsliste

```
         LD       X0
         QCDSETP_M "1:ABC"
         LD       X1
         QCDSET_M "3:DEF"
```

7.15 Uhr-Anweisungen

Mit den Uhr-Anweisungen ist es möglich, Uhr-Daten zu lesen und zu schreiben, Rechenoperationen mit Uhr-Daten durchzuführen und das Datenformat der Uhr-Daten zu ändern. Die Tabelle gibt eine Übersicht der Anweisungen.

Funktion	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor
Lesen von Uhr-Daten	DATERD	DATERD_MD
	DATERDP	DATERD_P_MD
Schreiben von Uhr-Daten	DATEWR	DATEWR_MD
	DATEWRP	DATEWR_P_MD
Addition von Uhr-Daten	DATE+	DATEPLUS_M
	DATE+P	DATEPLUSP_M
Subtraktion von Uhr-Daten	DATE-	DATEMINUS_M
	DATE-P	DATEMINUSP_M
Wechsel des Datenformats der Uhr-Daten von Std., Min. und Sek. in Sekunden	SECOND	SECOND_M
	SECONDP	SECONDP_M
Wechsel des Datenformats der Uhr-Daten von Sekunden in Std., Min. und Sek.	HOUR	HOUR_M
	HOURP	HOURP_M

7.15.1 DATERD, DATERDP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●

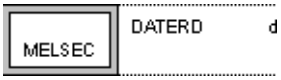
[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

Operanden MELSEC Q

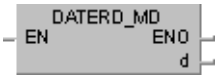
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten	Andere		
	Bit	Wort		Bit	Wort						
d	—	●	●	—	—	—	—	—	—	—	2

GX IEC Developer

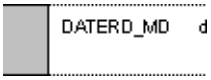
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

Operand	Befehlswert	Datentyp	
		MELSEC	IEC
d	Erste Adresse des Operanden, in dem die gelesenen Uhr-Daten gespeichert werden.	BIN-16-Bit	Array [0..6] of ANY16

Funktionsweise

Lesen von Uhr-Daten

DATERD Leseanweisung

Die DATERD-Anweisung liest Jahr, Monat, Tag, Stunde, Minute, Sekunde und Wochentag aus der CPU-internen Uhr, und speichert diese Uhr-Daten im Binärdatenformat in den in d+0 (Array_d[1]) bis d+6 (Array_d[7]) angegebenen Operanden. Die Zuordnung der Register zu den Uhr-Daten ist in der folgenden Abbildung dargestellt und lautet wie folgt:

d+0, Array_d[1] = Jahr (1)

d+1, Array_d[2] = Monat (Januar = 1, Dezember = 12)(2)

d+2, Array_d[3] = Tag (3)

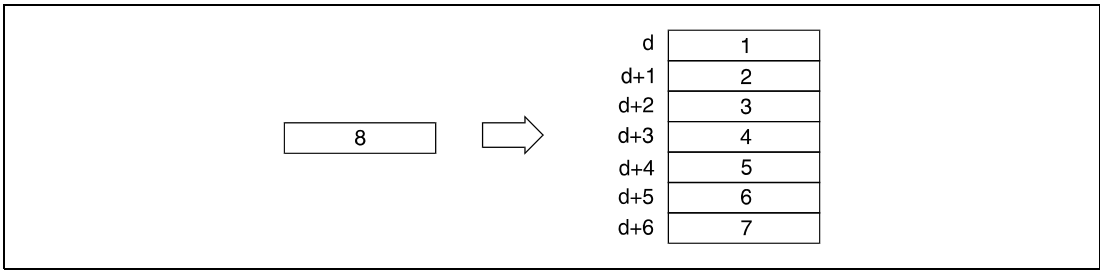
d+3, Array_d[4] = Stunde (24-Stunden-Uhr)(4)

d+4, Array_d[5] = Minute (5)

d+5, Array_d[6] = Sekunde (6)

d+6, Array_d[7] = Wochentag (7)

Die CPU-Uhr ist in der Abbildung mit 8 bezeichnet.



Die folgende Tabelle enthält die Wertebereiche der in d+0 (Array_d[1]) bis d+6 (Array_d[7]) gespeicherten Uhr-Daten.

Uhr-Daten	Jahr	Monat	Tag	Stunde	Minute	Sekunde	Wochentag
Eingabebereich	1)	1 - 12	1 - 31	0 - 23	0 - 59	0 - 59	0 - 6
Operanden	d+0 (Array_d[0])	d+1 (Array_d[1])	d+2 (Array_d[2])	d+3 (Array_d[3])	d+4 (Array_d[4])	d+5 (Array_d[5])	d+6 (Array_d[6])

1 Wertebereich 0 bis 99 bei QnA-CPU, 1980 bis 2079 bei einer CPU des System Q

Bei der Speicherung der Jahresangabe werden in der QnA-CPU nur die Einer- und Zehnerstellen gespeichert (z.B. 1995 = 95).
Bei einer CPU aus dem System Q wird die Jahreszahl als vierstellige Zahl gespeichert.

Der in D+6 (Array_d[7]) gespeicherte Wochentag wird zwischen 0 und 6 angegeben. Die Zuordnung der Wochentage enthält folgende Tabelle.

Wochentag	Sonntag	Montag	Dienstag	Mittwoch	Donnerstag	Freitag	Samstag
Speicherwerte	0	1	2	3	4	5	6

Schaltjahre werden von der CPU-Uhr automatisch eingerechnet.

Beispiel 1 DATERD (QnA-CPU)

Im folgenden Programm werden für die Einschaltdauer von SM400 die Uhr-Daten aus der CPU-internen Uhr ausgelesen und über die Ausgänge Y47 - Y67 als BCD-Daten wie folgt aufgeteilt ausgegeben.

Y60 - Y67 = Monat

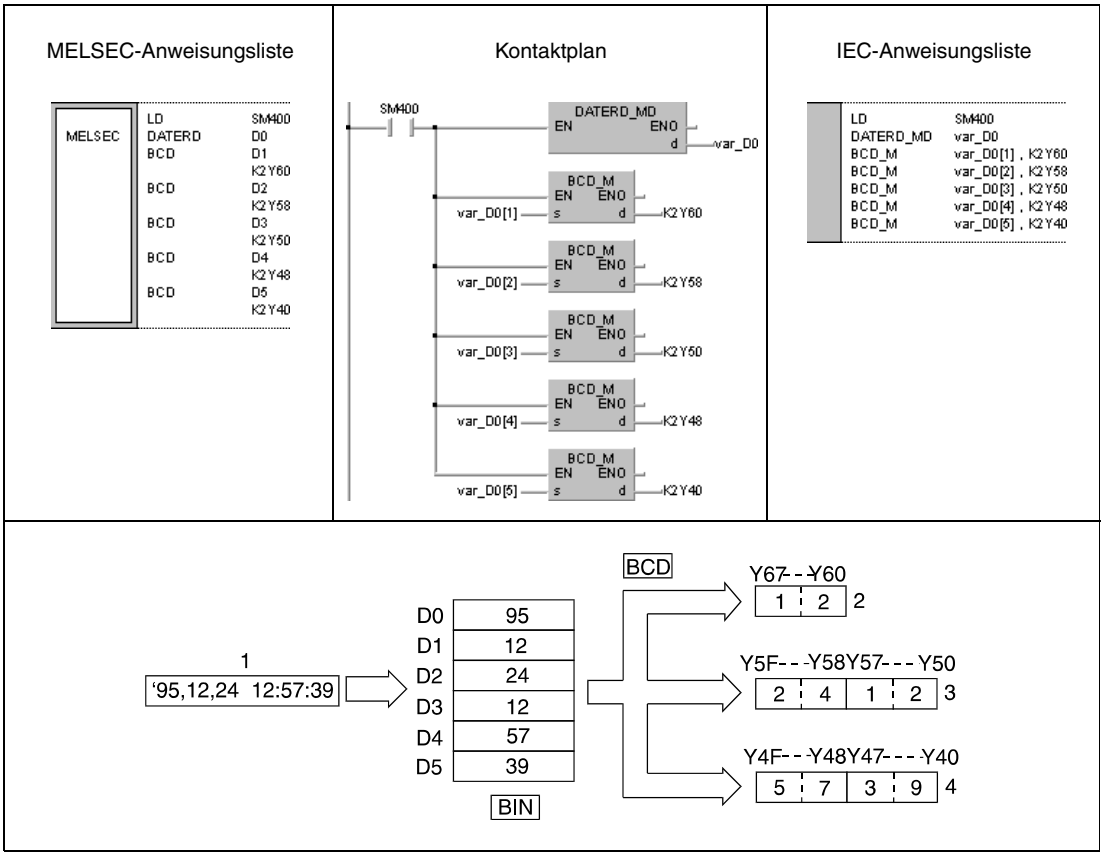
Y58 - Y5F = Tag

Y50 - Y57 = Stunde

Y48 - Y4F = Minute

Y40 - Y47 = Sekunde

Uhr-Daten	Jahr	Monat	Tag	Stunde	Minute	Sekunde	Wochentag
Operanden	D0 (var_D0[0])	D1 (var_D0[1])	D2 (var_D0[2])	D3 (var_D0[3])	D4 (var_D0[4])	D5 (var_D0[5])	D6 (var_D0[6])



[1] Uhr-Daten

[2] Monat

[3] Tag, Stunde

[4] Minute, Sekunde

HINWEIS *Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- und Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

Beispiel 2 DATERD (CPU aus dem System Q)

Im folgenden Programm werden für die Einschaltdauer von SM400 die Uhr-Daten aus der CPU-internen Uhr ausgelesen und über die Ausgänge Y47 - Y67 als BCD-Daten wie folgt aufgeteilt ausgegeben.

Y70 - Y7F =Jahr Y68 - Y6F = Monat

Y60 - Y67 = Tag Y58 - Y5F = Stunde

Y50 - Y57 = Minute Y48 - Y4F = Sekunde

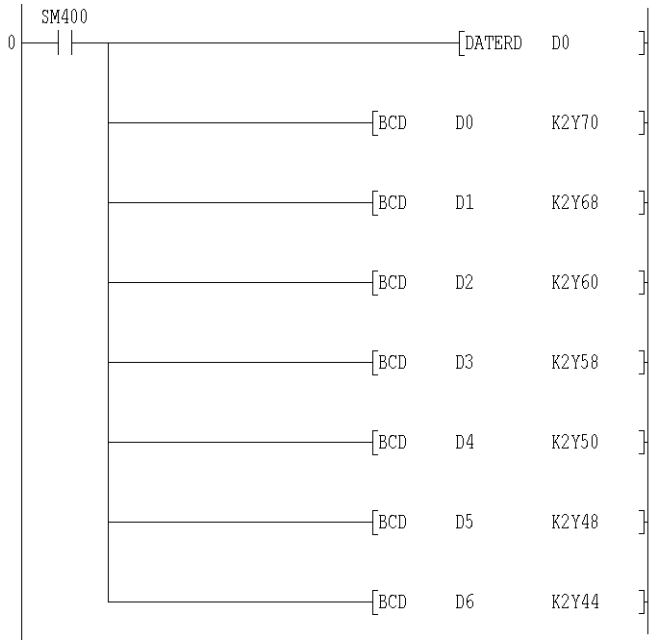
Y44 - Y47 = Wochentag

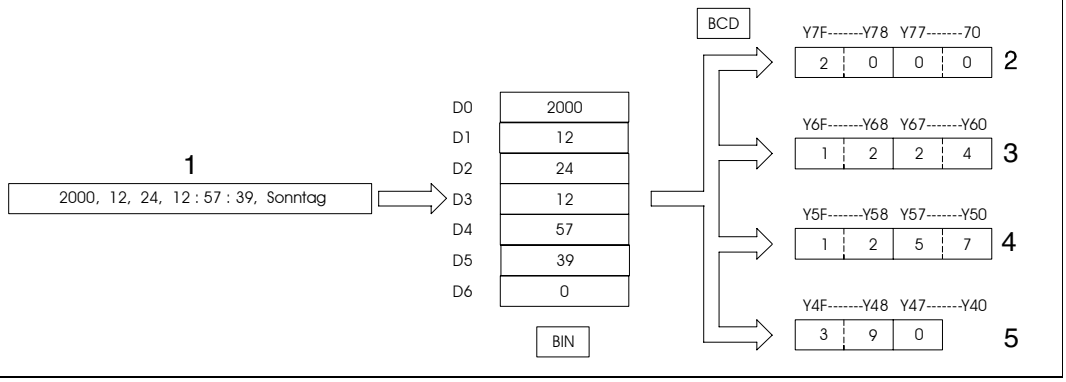
Uhr-Daten	Jahr	Monat	Tag	Stunde	Minute	Sekunde	Wochentag
Operanden	D0	D1	D2	D3	D4	D5	D6

Anweisungsliste

```
 0   LD       SM400
 1   DATERD   D0
 3   BCD      D0      K2Y70
 6   BCD      D1      K2Y68
 9   BCD      D2      K2Y60
12   BCD      D3      K2Y58
15   BCD      D4      K2Y50
18   BCD      D5      K2Y48
21   BCD      D6      K2Y44
24   END
```

Kontaktplan





[1] Uhr-Daten

[2] Jahr

[3] Monat, Tag

[4] Stunde, Minute,

[5] Sekunde, Wochentag

7.15.2 DATEWR, DATEWRP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●

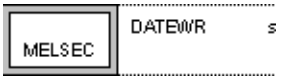
[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

Operanden MELSEC Q

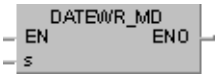
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort				DY		
s	—	●	●	—	—	—	—	—	—	SM0	2

GX IEC Developer

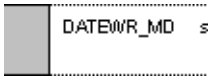
MELSEC-Anweisungsliste



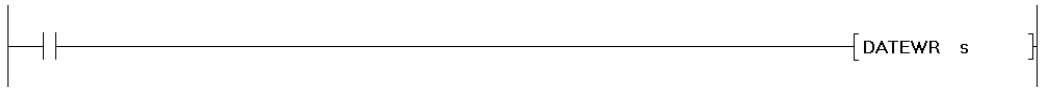
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

Operand	Befehlswert	Datentyp	
		MELSEC	IEC
s	Erste Adresse des Operanden, in dem die in die CPU-interne Uhr zu schreibenden Daten gespeichert sind.	BIN-16-Bit	Array [0..6] of ANY16

Funktionsweise

Schreiben von Uhr-Daten

DATEWR Schreibanweisung

Die DATEWR-Anweisung schreibt die in den in s+0 (Array_s[1]) bis s+6 (Array_s[7]) angegebenen Operanden gespeicherten Uhr-Daten Jahr, Monat, Tag, Stunde, Minute, Sekunde und Wochentag in die CPU-interne Uhr. Die Uhr-Daten sind im Binärdatenformat gespeichert. Die Zuordnung der Register zu den Uhr-Daten, die in die CPU-interne Uhr geschrieben werden, ist in der folgenden Abbildung dargestellt und lautet wie folgt:

s+0, Array_s[1] = Jahr (1)

s+1, Array_s[2] = Monat (Januar = 1, Dezember = 12)(2)

s+2, Array_s[3] = Tag (3)

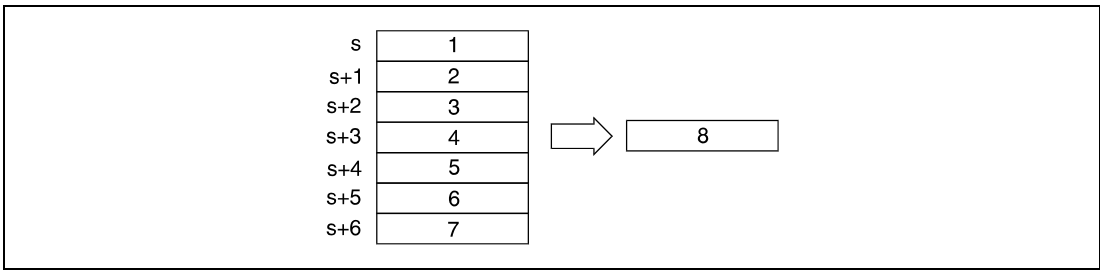
s+3, Array_s[4] = Stunde (24-Stunden-Uhr, 0 bis 23 Uhr)(4)

s+4, Array_s[5] = Minute (5)

s+5, Array_s[6] = Sekunde (6)

s+6, Array_s[7] = Wochentag (7)

Die CPU-Uhr ist in der Abbildung mit 8 bezeichnet.



Die folgende Tabelle enthält die Wertebereiche der in s+0 (Array_s[1]) bis s+6 (Array_s[7]) gespeicherten Uhr-Daten.

Uhr-Daten	Jahr	Monat	Tag	Stunde	Minute	Sekunde	Wochentag
Eingabebereich	1)	1 - 12	1 - 31	0 - 23	0 - 59	0 - 59	0 - 6
Operanden	s+0 (Array_s[0])	s+1 (Array_s[1])	s+2 (Array_s[2])	s+3 (Array_s[3])	s+4 (Array_s[4])	s+5 (Array_s[5])	s+6 (Array_s[6])

[1] Wertebereich 0 bis 99 bei QnA-CPU, 1980 bis 2079 bei einer System Q-CPU

Von der QnA-CPU werden bei der Jahresangabe nur die Einer- und Zehnerstellen gespeichert (z. B. 1995 = 95). Bei einer System Q-CPU wird die eine vierstellige Jahreszahl gespeichert.

Der in s+6 (Array_s[7]) gespeicherte Wochentag wird zwischen 0 und 6 angegeben. Die Zuordnung der Wochentage enthält folgende Tabelle.

Wochentag	Sonntag	Montag	Dienstag	Mittwoch	Donnerstag	Freitag	Samstag
Speicherwerte	0	1	2	3	4	5	6

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die in s+0 (Array_s[1]) bis s+6 (Array_s[7]) angegebenen Uhr-Daten liegen außerhalb des Wertebereichs (Fehlercode 4100).

Beispiel 1 DATEWRP (QnA-CPU)

Im folgenden Programm werden mit positiver Flanke von X40 die an den Eingängen X0 bis X2F im BCD-Datenformat anstehenden Uhr-Daten in die CPU-interne Uhr geschrieben. Die Eingänge sind den Uhr-Daten wie folgt zugeordnet.

X28 - X2F = Jahr

X20 - X27 = Monat

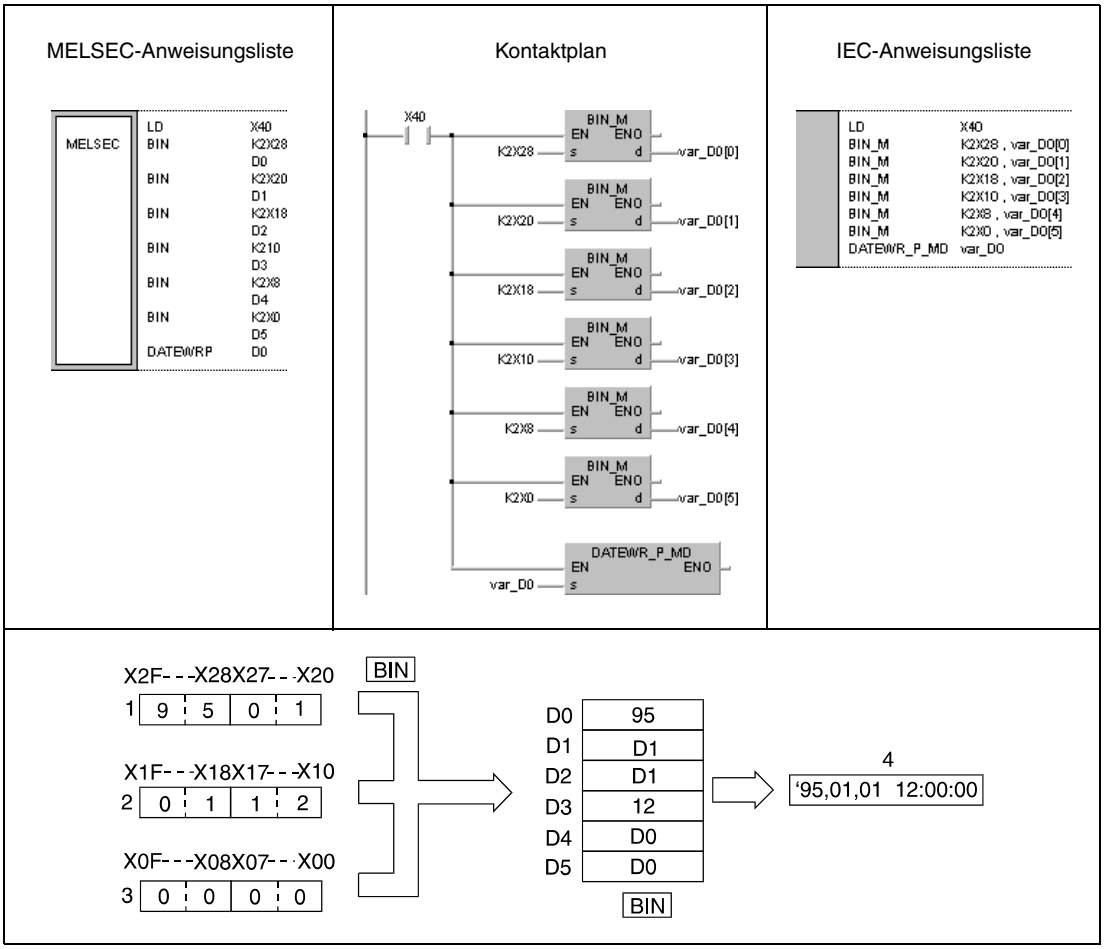
X18 - X1F = Tag

X10 - X17 = Stunde

X8 - XF = Minute

X0 - X7 = Sekunde

Uhr-Daten	Jahr	Monat	Tag	Stunde	Minute	Sekunde	Wochentag
Operanden	D0 (var_D0[0])	D1 (var_D0[1])	D2 (var_D0[2])	D3 (var_D0[3])	D4 (var_D0[4])	D5 (var_D0[5])	D6 (var_D0[6])



[1] Jahr,Monat

[2] Tag, Stunde

[3] Minute, Sekunde

[4] Uhr-Daten

HINWEIS *Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- und Checker-Fehlermeldungen können die Folge sein.Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

Beispiel 2 DATEWRP (CPU aus dem System Q)

Im folgenden Programm werden mit positiver Flanke von X40 die an den Eingängen X0 bis X2F im BCD-Datenformat anstehenden Uhr-Daten in die CPU-interne Uhr geschrieben. Die Eingänge sind den Uhr-Daten wie folgt zugeordnet.

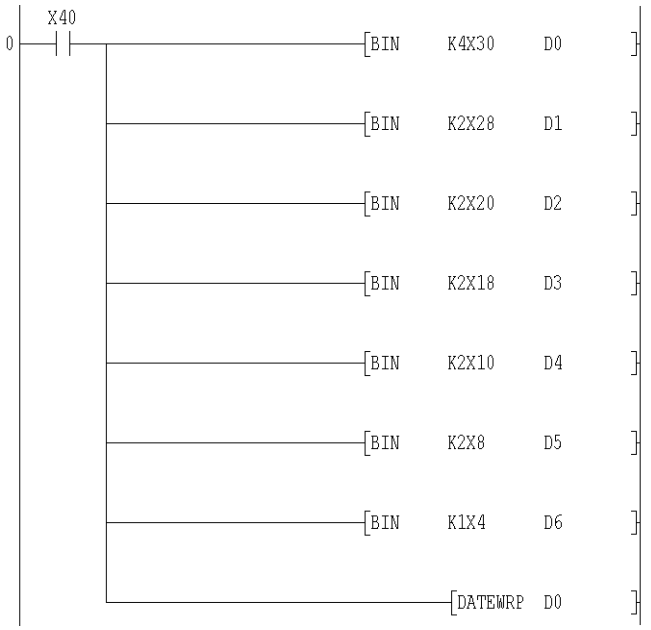
X30 - X3F = Jahr
X28 - X2F = Monat
X20 - X27 = Tag
X18 - X1F = Stunde
X10 - X17= Minute
X8 - XF = Sekunde

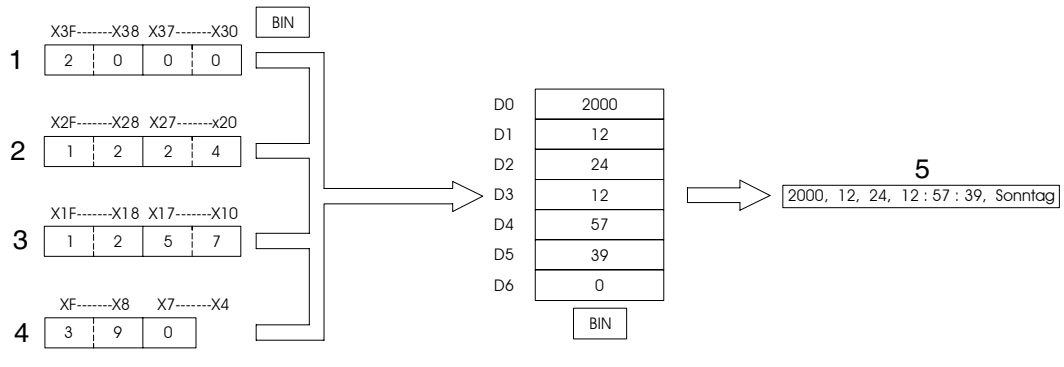
Uhr-Daten	Jahr	Monat	Tag	Stunde	Minute	Sekunde	Wochentag
Operanden	D0	D1	D2	D3	D4	D5	D6

MELSEC-Anweisungsliste

```
 0   LD       X40
 1   BIN      K4X30    D0
 4   BIN      K2X28    D1
 7   BIN      K2X20    D2
10   BIN      K2X18    D3
13   BIN      K2X10    D4
16   BIN      K2X8     D5
19   BIN      K1X4     D6
22   DATEWRP  D0
24   END
```

Kontaktplan





[1] Jahr
[2] Monat, Tag
[3] Stunde, Minute
[4] Sekunde, Wochentag
[5] Uhr-Daten

7.15.3 DATE+, DATE+P

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

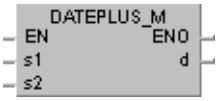
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten	Andere		
	Bit	Wort		Bit	Wort						
s1	—	●	●	—	—	—	—	—	—	SM0	4
s2	—	●	●	—	—	—	—	—	—		
d	—	●	●	—	—	—	—	—	—		

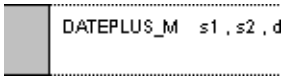
GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

Operand	Befehlswert	Datentyp	
		MELSEC	IEC
s1	Uhr-Daten, zu denen hinzuaddiert wird.	BIN-16-Bit	Array [0..2] of ANY16
s2	Uhr-Daten, die addiert werden.		
d	Erste Adresse des Operanden, in dem die Uhr-Daten des Additionsergebnisses gespeichert werden.		

Funktions-weise

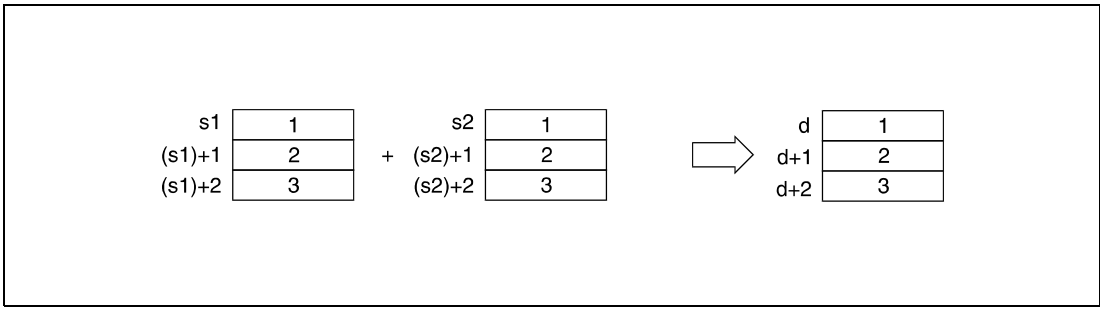
Addition von Uhr-Daten

DATE+ Additionsanweisung

Die DATE+-Anweisung addiert zu den Uhr-Daten, die sich in den Operanden ab s1 befinden, die sich in den Operanden ab s2 befinden. Die Uhr-Daten des Additionsergebnisses werden in den Operanden ab d gespeichert.

Die folgende Tabelle enthält die Wertebereiche der in s1+0 bis s1+2 (Array_s1[1] – Array_s1[3]), s2+0 bis s2+2 (Array_s2[1] – Array_s2[3]) und d+0 bis d+2 (Array_d[1] – Array_d[3]) speicherbaren Uhr-Daten.

Uhr-Daten	Jahr	Monat	Tag	Stunde	Minute	Sekunde	Wochentag
Eingabebereich	—	—	—	0 - 23	0 - 59	0 - 59	—
Operanden	—	—	—	s1+0 (Array_s1[0])	s1+1 (Array_s1[1])	s1+2 (Array_s1[2])	—
Operanden	—	—	—	s2+0 (Array_s2[0])	s2+1 (Array_s2[1])	s2+2 (Array_s2[2])	—
Operanden	—	—	—	d+0 (Array_d[0])	d+1 (Array_d[1])	d+2 (Array_d[2])	—

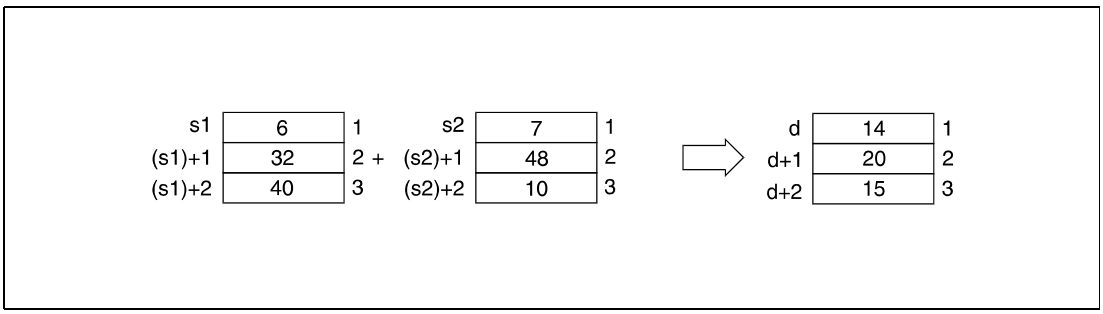


[1] Stunde

[2] Minute

[3] Sekunde

In der folgenden Abbildung wird zu der Uhrzeit
6 Stunden, 32 Minuten und 40 Sekunden ((s1)+0 – (s1)+2, (Array_s1[1] – Array_s1[3])) die Uhrzeit
7 Stunden, 48 Minuten und 10 Sekunden ((s2)+0 – (s2)+2, (Array_s2[1] – Array_s2[3])) addiert und das Ergebnis
14 Stunden, 20 Minuten und 50 Sekunden in d+0 bis d+2 ((Array_d[1] – Array_d[3])) gespeichert.



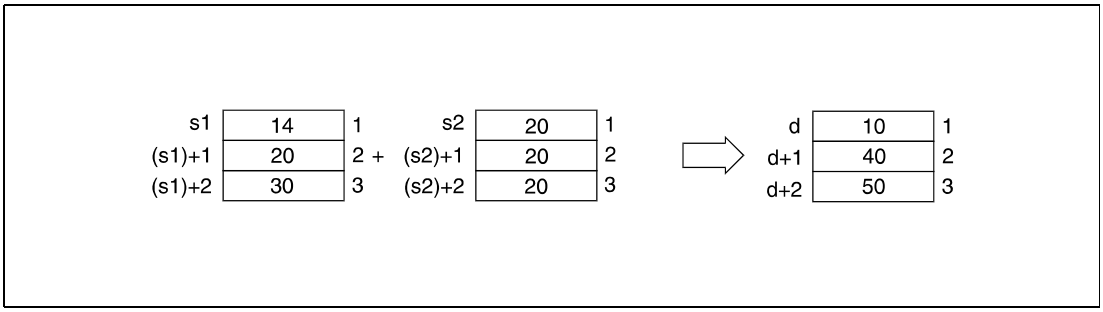
[1] Stunde

[2] Minute

[3] Sekunde

Wird das Ergebnis der Addition zweier Uhr-Daten größer als 24 Stunden, werden von dem Ergebnis automatisch 24 Stunden subtrahiert, um eine korrekte Zeitangabe zu erhalten.

Das Ergebnis der in der folgenden Abbildung dargestellten Addition der Uhrzeit 14 Stunden, 20 Minuten und 30 Sekunden und der Uhrzeit 20 Stunden, 20 Minuten und 20 Sekunden ist 34 Stunden, 40 Minuten und 50 Sekunden. Dieses Ergebnis ist als Uhrzeit nicht darstellbar. Durch die Subtraktion von 24 Stunden lautet das Ergebnis dann 10 Stunden, 40 Minuten und 50 Sekunden (10:40:50 am Folgetag).



[1] Stunde

[2] Minute

[3] Sekunde

HINWEIS *Weitere Informationen über die Eingabe von Stunden, Minuten und Sekunden sind dem Abschnitt „Schreiben von Uhr-Daten“ zu entnehmen.*

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

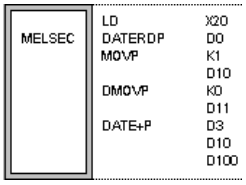
- Die in (s1)+0 bis (s1)+2 ((Array_s1[1] – Array_s1[3])) und (s2)+0 bis (s2)+2 ((Array_s2[1] – Array_s2[3])) angegebenen Uhr-Daten liegen außerhalb des Wertebereichs.

Beispiel DATE+P

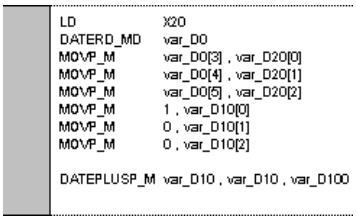
Im folgenden Programm werden mit positiver Flanke von X20 die Uhr-Daten aus der CPU-internen Uhr mittels der DATERDP-Anweisung ausgelesen und in den Registern D0 bis D6 gespeichert (erste Abbildung nach dem Programmbeispiel). Zu den Stunden, Minuten und Sekunden dieser Uhr-Daten wird eine Stunde (D10, D11, D12) mittels der DATE+P-Anweisung addiert. Das Ergebnis wird in D100 bis D102 gespeichert (zweite Abbildung nach dem Programmbeispiel).

Uhr-Daten	Jahr	Monat	Tag	Stunde	Minute	Sekunde	Wochentag
Operanden	D0 (var_D0[0])	D1 (var_D0[1])	D2 (var_D0[2])	D3 (var_D0[3])	D4 (var_D0[4])	D5 (var_D0[5])	D6 (var_D0[6])
Operanden	—	—	—	D20 (var_D20[0])	D21 (var_D20[1])	D22 (var_D20[2])	—
Operanden	—	—	—	D10 (var_D10[0])	D11 (var_D10[1])	D12 (var_D10[2])	—
Operanden	—	—	—	D100 (var_D100[0])	D101 (var_D100[1])	D102 (var_D100[2])	—

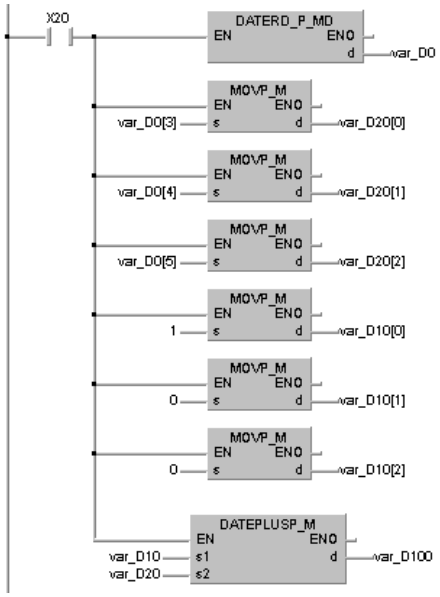
MELSEC-Anweisungsliste



IEC-Anweisungsliste

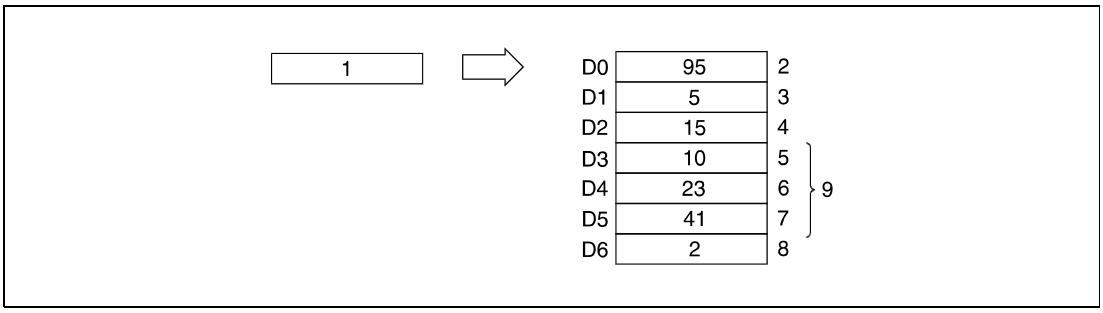


Kontaktplan



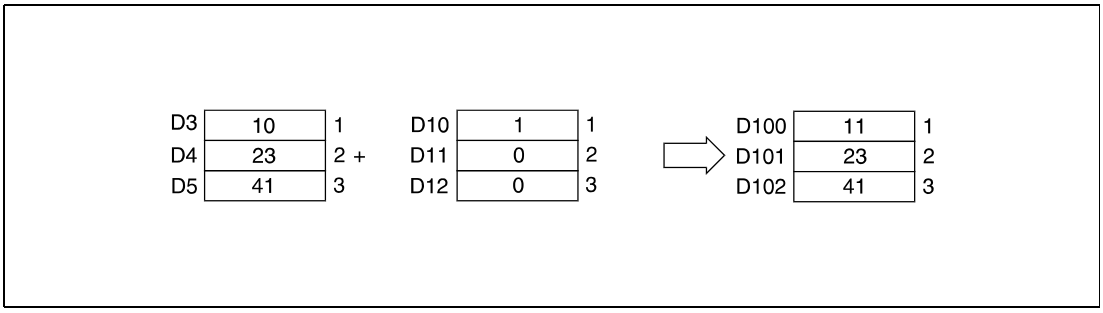
HINWEIS *Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen. Compiler- und Checker-Fehlermeldungen können die Folge sein.*

Die folgende Abbildung zeigt das Lesen der Uhr-Daten mittels der DATERDP-Anweisung.



[1] QnA CPU-Uhr
[2] Jahr
[3] Monat (Januar = 1, Dezember = 12)
[4] Tag
[5] Stunde (24-Stunden-Uhr)
[6] Minute
[7] Sekunde
[8] Wochentag
[9] Uhrzeitdaten

Die folgende Abbildung zeigt die mittels der DATE+P-Anweisung ausgeführte Addition.



[1] Stunde
[2] Minute
[3] Sekunde

7.15.4 DATE-, DATE-P

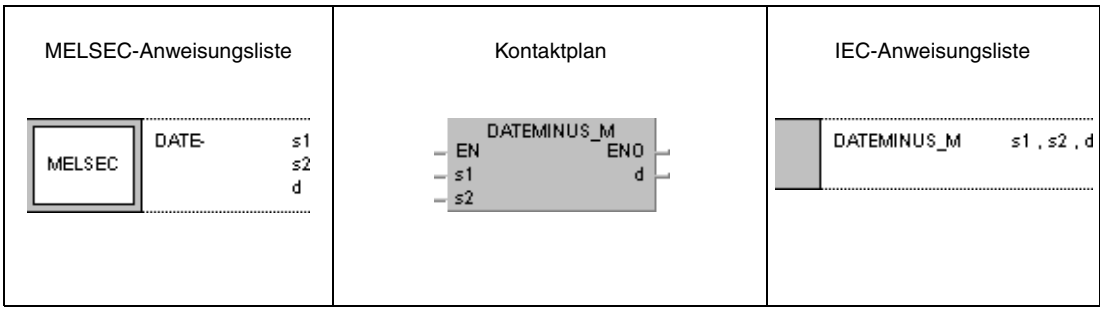
CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort				DY		
s1	—	●	●	—	—	—	—	—	—	SM0	4
s2	—	●	●	—	—	—	—	—	—		
d	—	●	●	—	—	—	—	—	—		

GX IEC Developer



GX Developer

Variablen

Operand	Befehlswert	Datentyp	
		MELSEC	IEC
s1	Erste Adresse des Operanden, in dem die Uhr-Daten, von denen subtrahiert wird, gespeichert sind.	BIN-16-Bit	Array [0..2] of ANY16
s2	Erste Adresse des Operanden, in dem die Uhr-Daten, die subtrahiert werden, gespeichert sind.		
d	Erste Adresse des Operanden, in dem die Uhr-Daten des Subtraktionsergebnisses gespeichert werden.		

Funktions-weise

Subtraktion von Uhr-Daten

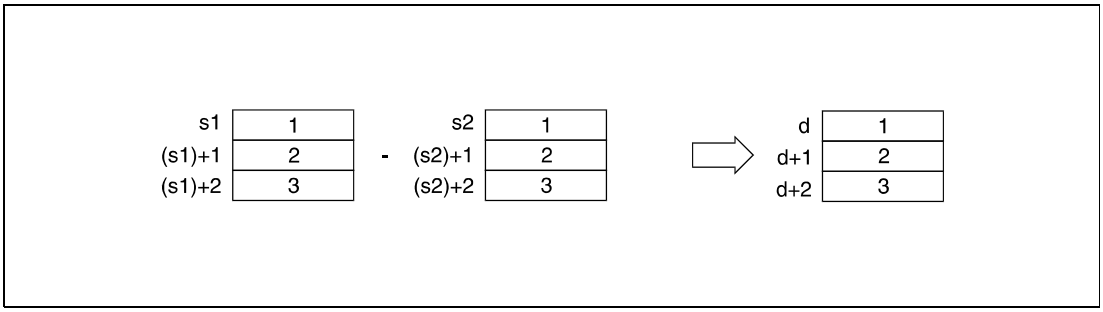
DATE- Subtraktionsanweisung

Die DATE-Anweisung subtrahiert von den Uhr-Daten, die sich in den Operanden ab s2 befinden, die Uhr-Daten, die sich in den Operanden ab s1 befinden. Die Uhrendaten des Subtraktionsergebnisses werden in den Operanden ab d gespeichert.

Die folgende Tabelle enthält die Wertebereiche der in (s1)+0 bis (s1)+2 (Array_s1[1] – Array_s1[3]), (s2)+0 bis (s2)+2 (Array_s2[1] – Array_s2[3]) und d+0 bis d+2 (Array_d[1] – Array_d[3]) speicherbaren Uhr-Daten.

Uhr-Daten	Jahr	Monat	Tag	Stunde	Minute	Sekunde	Wochentag
Eingabebereich	—	—	—	0 - 23	0 - 59	0 - 59	—

Uhr-Daten	Jahr	Monat	Tag	Stunde	Minute	Sekunde	Wochentag
Operanden	—	—	—	s1+0 (Array_s1[0])	s1+1 (Array_s1[1])	s1+2 (Array_s1[2])	—
Operanden	—	—	—	s2+0 (Array_s2[0])	s2+1 (Array_s2[1])	s2+2 (Array_s2[2])	—
Operanden	—	—	—	d+0 (Array_d[0])	d+1 (Array_d[1])	d+2 (Array_d[2])	—

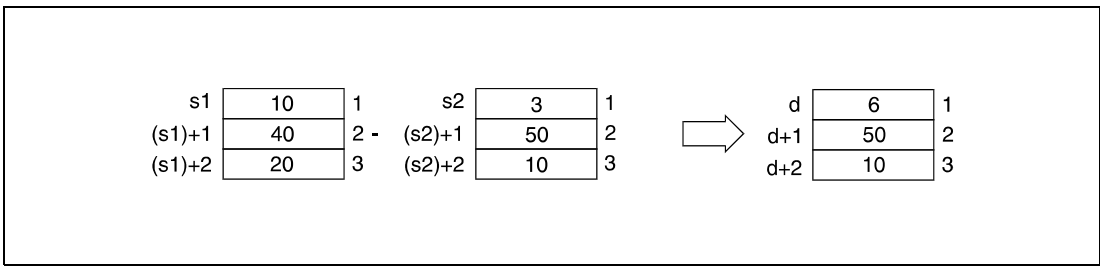


[1] Stunde

[2] Minute

[3] Sekunde

In der folgenden Abbildung wird die Uhrzeit 3 Stunden, 50 Minuten und 10 Sekunden ((s2)+0 - (s2)+2 (Array_s2[1] – Array_s2[3])) von der Uhrzeit 10 Stunden, 40 Minuten und 20 Sekunden ((s1)+0 - (s1)+2 (Array_s1[1] – Array_s1[3])) subtrahiert und das Ergebnis 6 Stunden, 50 Minuten und 10 Sekunden in d+0 bis d+2 (Array_d[1] – Array_d[3]) gespeichert.



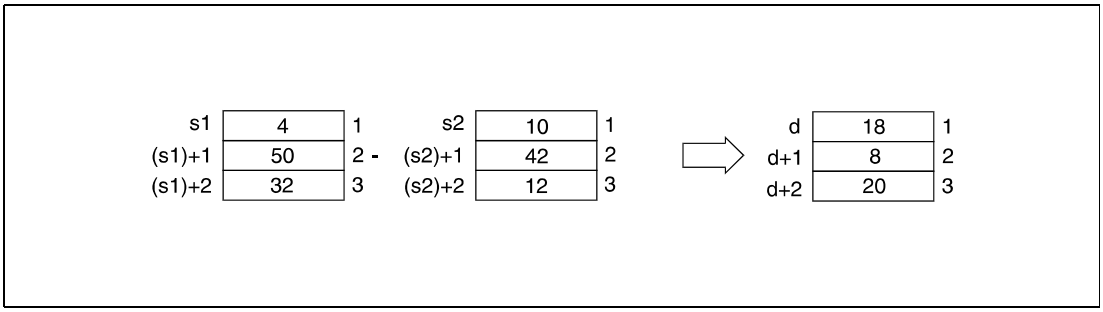
[1] Stunde

[2] Minute

[3] Sekunde

Wird das Ergebnis der Subtraktion zweier Uhr-Daten negativ, werden zu dem Ergebnis automatisch 24 Stunden addiert, um eine korrekte Zeitangabe zu erhalten.

Das Ergebnis der in der folgenden Abbildung dargestellten Subtraktion der Uhrzeit 10 Stunden, 42 Minuten und 12 Sekunden von der Uhrzeit 4 Stunden, 50 Minuten und 32 Sekunden ist -6 Stunden, 8 Minuten und 20 Sekunden. Dieses Ergebnis ist als Uhrzeit nicht darstellbar. Durch die Addition von 24 Stunden lautet das Ergebnis dann 18 Stunden, 8 Minuten und 20 Sekunden (18:08:20 am Vortag).



[1] Stunde

[2] Minute

[3] Sekunde

HINWEIS *Weitere Informationen über die Eingabe von Stunden, Minuten und Sekunden sind dem Abschnitt „Schreiben von Uhr-Daten“ zu entnehmen.*

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

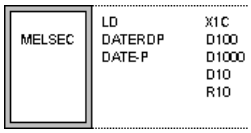
- Die in (s1)+0 bis (s1)+2 (Array_s1[1] – Array_s1[3]) und (s2)+0 bis (s2)+2 (Array_s2[1] – Array_s2[3]) angegebenen Uhrendaten liegen außerhalb des Wertebereichs.

Beispiel DATE-P

Im folgenden Programm werden mit positiver Flanke von X1C die Uhr-Daten aus der CPU-internen Uhr mittels der DATERDP-Anweisung ausgelesen, und in den Registern D100 bis D106 gespeichert (erste Abbildung nach dem Programmbeispiel). Von den Stunden, Minuten und Sekunden dieser Uhr-Daten werden mittels der DATE-P-Anweisung 10 Stunden (D10), 40 Minuten (D11) und 10 Sekunden (D12) subtrahiert. Zu dem negativen Subtraktionsergebnis -8 Stunden, 41 Minuten und 10 Sekunden werden 24 Stunden addiert. Das Ergebnis 16 Stunden, 41 Minuten und 10 Sekunden (16:41:10 des Vortags) wird in R10 bis R12 gespeichert (zweite Abbildung nach dem Programmbeispiel).

Uhr-Daten	Jahr	Monat	Tag	Stunde	Minute	Sekunde	Wochentag
Operanden	D100 (var_D100[0])	D101 (var_D100[1])	D102 (var_D100[2])	D103 (var_D100[3])	D104 (var_D100[4])	D105 (var_D100[5])	D106 (var_D100[6])
Operanden	—	—	—	D1000 (var_D1000[0])	D1001 (var_D1000[1])	D1002 (var_D1000[2])	—
Operanden	—	—	—	D10 (var_D10[0])	D11 (var_D10[1])	D12 (var_D10[2])	—
Operanden	—	—	—	R10 (var_R10[0])	R11 (var_R10[1])	R12 (var_R10[2])	—

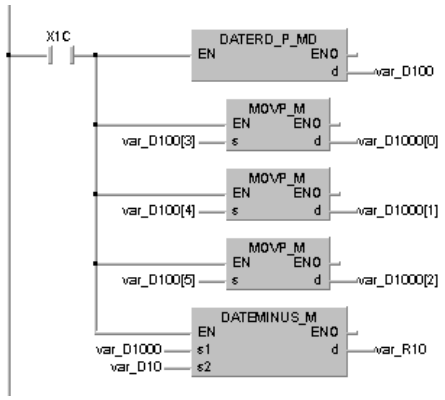
MELSEC-Anweisungsliste



IEC-Anweisungsliste

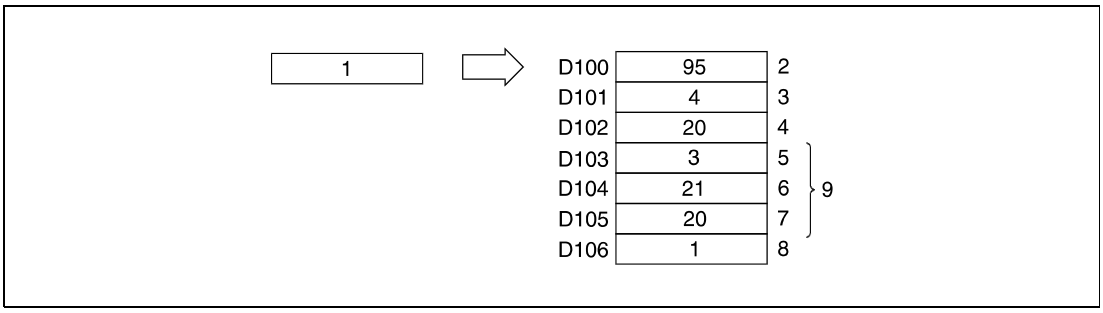
```
LD              X1C
DATERD_P_MD     var_D100
MOVP_M          var_D100[3] , var_D1000[0]
MOVP_M          var_D100[4] , var_D1000[1]
MOVP_M          var_D100[5] , var_D1000[2]
DATEMINUSP_M    var_D1000, var_D10 , var_R10
```

Kontaktplan



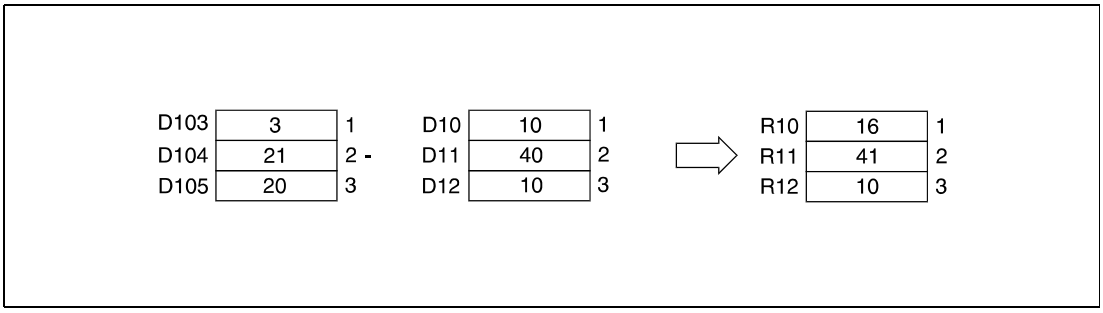
HINWEIS *Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- und Checker-Fehlermeldungen können die Folge sein.Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

Die folgende Abbildung zeigt das Lesen der Uhr-Daten mittels der DATERDP-Anweisung.



[1] QnA CPU-Uhr
[2] Jahr
[3] Monat (Januar = 1, Dezember = 12)
[4] Tag
[5] Stunde (24-Stunden-Uhr)
[6] Minute
[7] Sekunde
[8] Wochentag
[9] Uhrzeitdaten

Die folgende Abbildung zeigt die mittels der DATE-P-Anweisung ausgeführte Subtraktion.



[1] Stunde
[2] Minute
[3] Sekunde

7.15.5 SECOND, SECONDP, HOUR, HOURP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

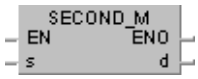
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
SECOND											
s	—	●	●	—	—	—	—	—	—	SM0	3
d	●	●	●	●	●	●	●	—	—		
HOUR											
s	●	●	●	●	●	●	●	●	—	SM0	3
d	—	●	●	—	—	—	—	—	—		

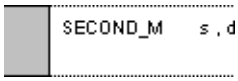
GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

Operand	Befehlswert	Datentyp	
		MELSEC	IEC
SECOND			
s	Stunden, Minuten, Sekunden	BIN-16-/32-Bit	Array [0..2] of ANY16
d	Sekunden		ANY32
HOUR			
s	Sekunden	BIN-16-/32-Bit	ANY32
d	Stunden, Minuten, Sekunden		Array [0..2] of ANY16

Funktionsweise

Umwandeln von Uhr-Datenformaten

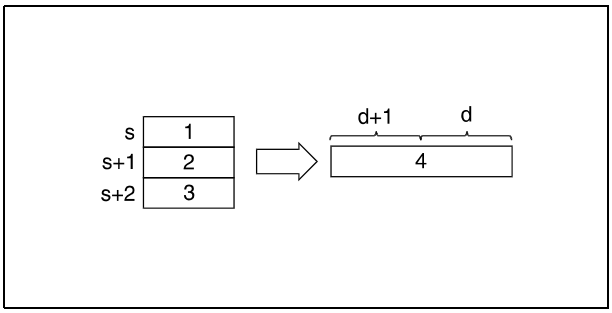
SECOND Umwandeln von Stunden, Minuten und Sekunden in Sekunden

Die SECOND-Anweisung wandelt Uhr-Daten im Format Stunden, die sich in den Operanden s+0 (Array_s[1]) bis s+2 (Array_s[3]) befinden, in das Format Sekunden um, und speichert das Ergebnis in den in d und d+1 angegebenen Operanden.

Die folgende Tabelle enthält die Wertebereiche der in den Operanden s+0 (Array_s[1]) bis s+2 (Array_s[3]) speicherbaren Uhr-Daten.

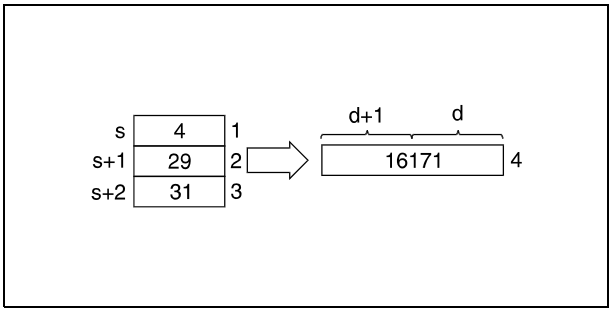
Uhr-Daten	Jahr	Monat	Tag	Stunde	Minute	Sekunde	Wochentag
Eingabebereich	—	—	—	0 - 23	0 - 59	0 - 59	—

Uhr-Daten	Jahr	Monat	Tag	Stunde	Minute	Sekunde	Wochentag
Operanden	—	—	—	s+0 (Array_s[0])	s+1 (Array_s[1])	s+2 (Array_s[2])	—
Operanden	—	—	—	—	—	d+0 (Array_d[0]) bis d+1 (Array_d[1])	—



[1] Stunde
[2] Minute
[3] Sekunde
[4] In Sekunden umgewandelter Wert

In der folgenden Abbildung werden 4 Stunden , 29 Minuten und 31 Sekunden in 16171 Sekunden umgewandelt.



[1] Stunde
[2] Minute
[3] Sekunde
[4] In Sekunden umgewandelter Wert

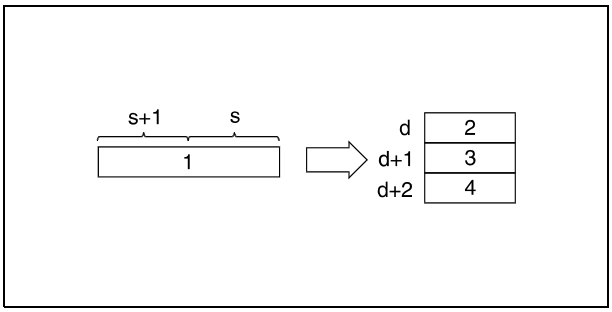
HOUR Umwandeln von Sekunden in Stunden, Minuten und Sekunden

Die HOUR-Anweisung wandelt die Uhr-Daten im Format Sekunden, die sich in den Operanden s+0 bis s+1 befinden, in das Format Stunden um.

Die folgende Tabelle enthält die Wertebereiche der in d+0 (Array_d[1]) bis d+2 (Array_d[3]) speicherbaren Uhr-Daten.

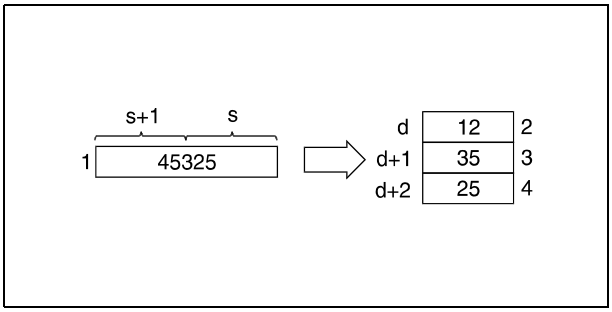
Uhr-Daten	Jahr	Monat	Tag	Stunde	Minute	Sekunde	Wochentag
Eingabebereich	—	—	—	0 - 23	0 - 59	0 - 59	—

Uhr-Daten	Jahr	Monat	Tag	Stunde	Minute	Sekunde	Wochentag
Operanden	—	—	—	d+0 (Array_d[0])	d+1 (Array_d[1])	d+2 (Array_d[2])	—
Operanden	—	—	—	—	—	s+0 (Array_s[0]) bis s+1 (Array_s[1]	—



[1] Umzuwandelnder Wert in Sekunden
[2] Stunde
[3] Minute
[4] Sekunde

In der folgenden Abbildung werden 45325 Sekunden in 12 Stunden, 35 Minuten und 25 Sekunden umgewandelt.



[1] Umzuwandelnder Wert in Sekunden
[2] Stunde
[3] Minute
[4] Sekunde

Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

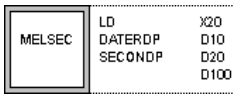
- Die in s+0 (Array_s[1]) bis s+2 (Array_s[3]) bei der SECOND-Anweisung bzw. die in s+0 und s+1 bei der HOUR-Anweisung angegebenen Uhren-Daten liegen außerhalb des Wertebereichs (Fehlercode 4100).

Beispiel 1 SECONDP

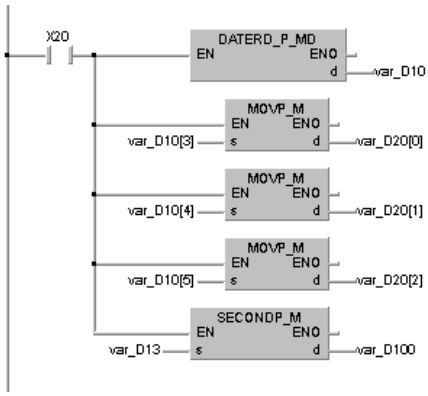
Im folgenden Programm werden mit positiver Flanke von X20 die Uhr-Daten aus der CPU-internen Uhr mittels der DATERDP-Anweisung ausgelesen, und in den Registern D10 bis D16 gespeichert (erster Teil der Abbildung nach dem Programmbeispiel). Die Stunden D20, Minuten D21 und Sekunden D22 dieser Uhr-Daten werden mittels der SECONDP-Anweisung in Sekunden umgewandelt. Das Ergebnis wird in D100 und D101 gespeichert (zweiter Teil der Abbildung nach dem Programmbeispiel).

Uhr-Daten	Jahr	Monat	Tag	Stunde	Minute	Sekunde	Wochentag
Operanden	D10 (var_D10[0])	D11 (var_D10[1])	D12 (var_D10[2])	D13 (var_D10[3])	D14 (var_D10[4])	D15 (var_D10[5])	D16 (var_D10[6])
Operanden	—	—	—	D20 (var_D20[0])	D21 (var_D20[1])	D22 (var_D20[2])	—
Operanden	—	—	—	—	—	D100 (var_D10[0]) bis D101 (var_D10[1])	—

MELSEC-Anweisungsliste



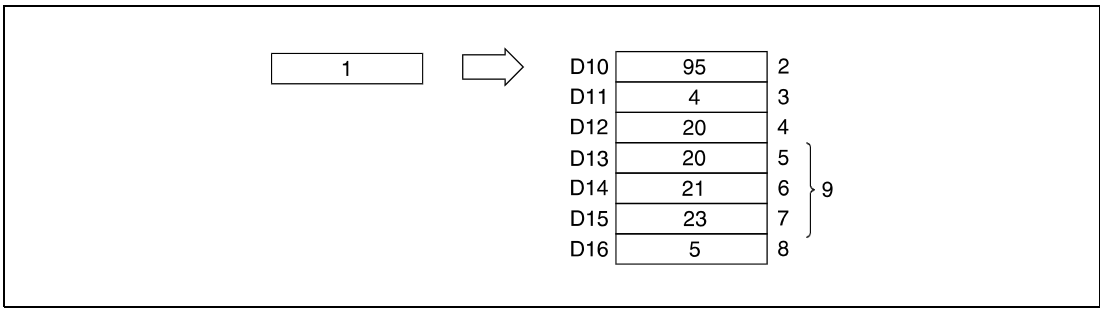
Kontaktplan



IEC-Anweisungsliste

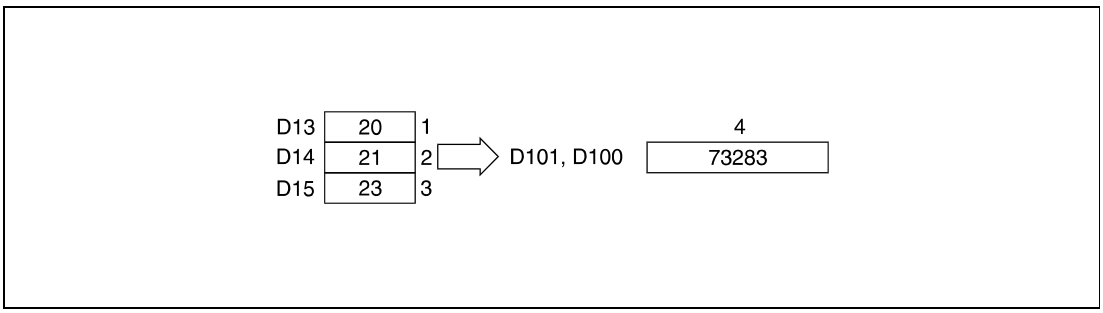
```
LD          X20
DATERD_P_MD var_D10
MOVP_M      var_D10[3] , var_D20[0]
MOVP_M      var_D10[4] , var_D20[1]
MOVP_M      var_D10[5] , var_D20[2]
SECONDP_M   var_D20 , var_D100
```

Die folgende Abbildung zeigt das Lesen der Uhr-Daten mittels der DATERDP-Anweisung.



[1] QnA CPU-Uhr
[2] Jahr
[3] Monat (Januar = 1, Dezember = 12)
[4] Tag
[5] Stunde (24-Stunden-Uhr)
[6] Minute
[7] Sekunde
[8] Wochentag
[9] Uhrzeitdaten

Die folgende Abbildung zeigt die mittels der SECONDP-Anweisung ausgeführte Umwandlung in Sekunden.

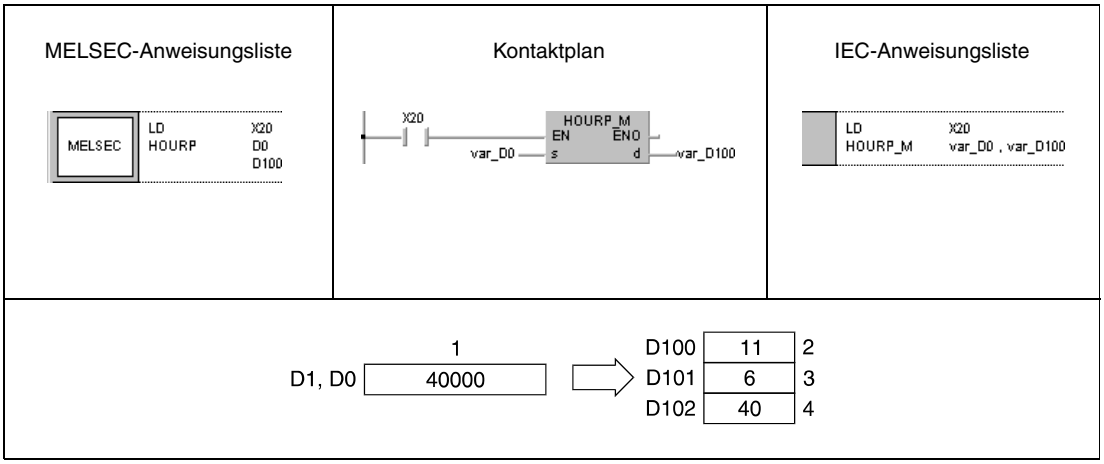


[1] Stunde
[2] Minute
[3] Sekunde
[4] In Sekunden umgewandelter Wert

Beispiel 2 HOURP

Im folgenden Programm werden mit positiver Flanke von X20 die in D0 und D1 gespeicherten Sekunden in Stunden, Minuten und Sekunden umgewandelt. Das Ergebnis wird in den in Klammern angegebenen Operanden gespeichert.

Uhr-Daten	Jahr	Monat	Tag	Stunde	Minute	Sekunde	Wochentag
Operanden	—	—	—	D0 (var_D0[1])	D1 (var_D0[2])	D2 (var_D0[3])	—
Operanden	—	—	—	—	—	D100 (var_D100[0]) bis D101 (var_D100[1])	—



[1] Umzuwandelnder Wert in Sekunden

[2] Stunde

[3] Minute

[4] Sekunde

HINWEIS *Diese Programmbeispiele sind ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- und Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

7.16 Anweisungen für Peripheriegeräte

Die Anweisungen für Peripheriegeräte ermöglichen die Ausgabe von Meldungen an Peripheriegeräten und die Eingabe von Daten über Tastaturen an Peripheriegeräten.

Funktion	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor
Ausgabe von Meldungen an Peripheriegeräten	MSG	MSG_M
Tastatureingabe von Daten an Peripheriegeräten	PKEY	PKEY_M

7.16.1 MSG

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

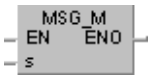
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten $	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	—	—	—	●	—	—	2

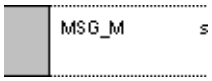
GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

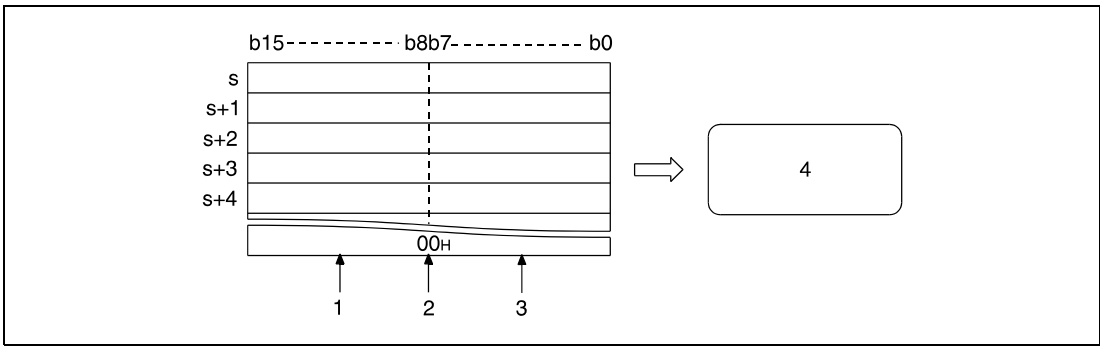
Operand	Befehlswert	Datentyp
s	Zeichenfolgendaten, die an dem Peripheriegerät dargestellt werden oder erste Adresse des Operanden, in dem diese Daten gespeichert sind.	Zeichenfolge

Funktionsweise

Ausgabe von Meldungen an Peripheriegeräten

MSG Ausgabeanweisung

Die MSG-Anweisung gibt die Zeichenfolge, die sich ab den Operanden ab s befinden, als Meldung an ein im Terminal-Modus angegebenes Peripheriegerät aus. Das Ende der Zeichenfolge wird mit dem Zeichencode "00H" gekennzeichnet.



[1] 2., 4.,..., (n+1)-tes Zeichen

[2] Der Zeichencode "00H" kennzeichnet das Ende der Zeichenfolge

[3] 1., 3., ..., n-tes Zeichen

[4] Darstellung der Zeichenfolgen (Meldungen) an einem Peripheriegerät

Es können maximal 64 Zeichen auf dem Display des Peripheriegerätes dargestellt werden.

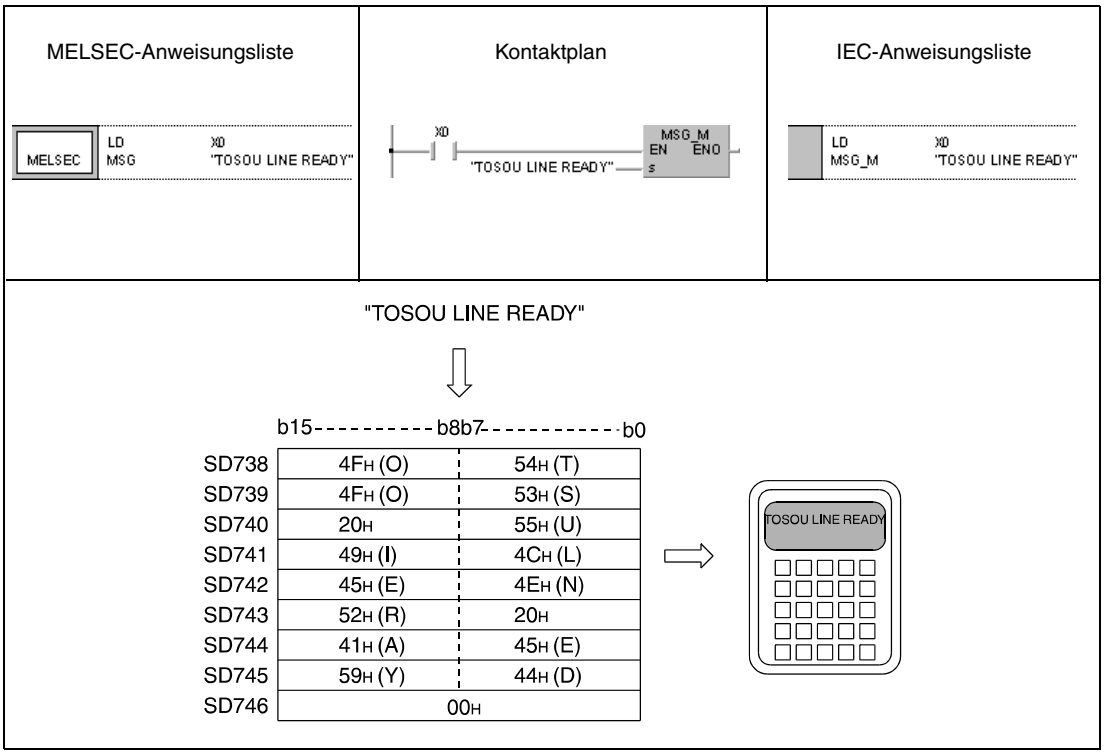
Die in s angegebenen Zeichenfolgendaten werden in den Diagnoseregistern SD738 bis SD773 gespeichert (Speicherbereich für Meldungen).

Während der Ausführung der MSG-Anweisung wird der Diagnosemerker SM738 (Ausführungssignal der MSG-Anweisung) gesetzt. Wenn der Merker SM738 gesetzt ist, wird keine andere MSG-Anweisung ausgeführt.

Nach der vollständigen Ausführung der MSG-Anweisung, d.h. nach Darstellung aller Zeichen auf dem Display des Peripheriegerätes wird der Diagnosemerker SM738 zurückgesetzt und der Inhalt (Zeichenfolge) der Diagnoseregister SD738 bis SD773 gelöscht (mit dem Zeichencode "00H" überschrieben).

Beispiel MSG

Im folgenden Programm wird mit Einschalten von X0 die Zeichenfolge "TOSOU LINE READY" als Meldung an das Display eines Peripheriegerätes gesendet und dargestellt.



7.16.2 PKEY

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Operanden MELSEC Q

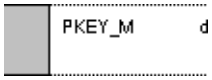
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort				U		
d	—	●	●	—	—	—	—	●	—	SM0	2

GX IEC Developer

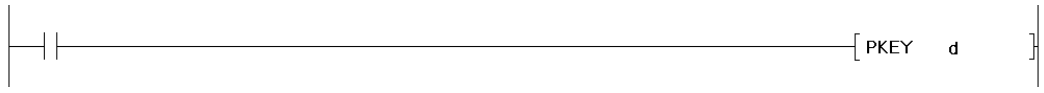
MELSEC-Anweisungsliste

Kontaktplan

IEC-Anweisungsliste



GX Developer



Variablen

Operand	Befehlswert	Datentyp
d	Erste Adresse des Operanden, in dem die eingegebene Zeichenfolge gespeichert wird.	BIN-16-Bit

Funktions-weise

Tastatureingabe von Daten an Peripheriegeräten

PKEY Eingabeanweisung

Die PKEY-Anweisung löscht die Datenwörter in den in d+0 bis d+17 angegebenen Operanden, und der Diagnosemerker SM736 (Ausführungssignal der PKEY-Anweisung) wird gesetzt. Zusätzlich wird der Diagnosemerker SM737 (Tastaturdatenempfangssignal) gesetzt. Wenn die Ausführung der PKEY-Anweisung beendet ist, werden die eingegebenen Tastaturdaten (Zeichen) aus dem im Terminal-Modus bestimmten Peripheriegerät gelesen und im ASCII-Format in die in d+0 bis d+17 angegebenen Operanden geschrieben.

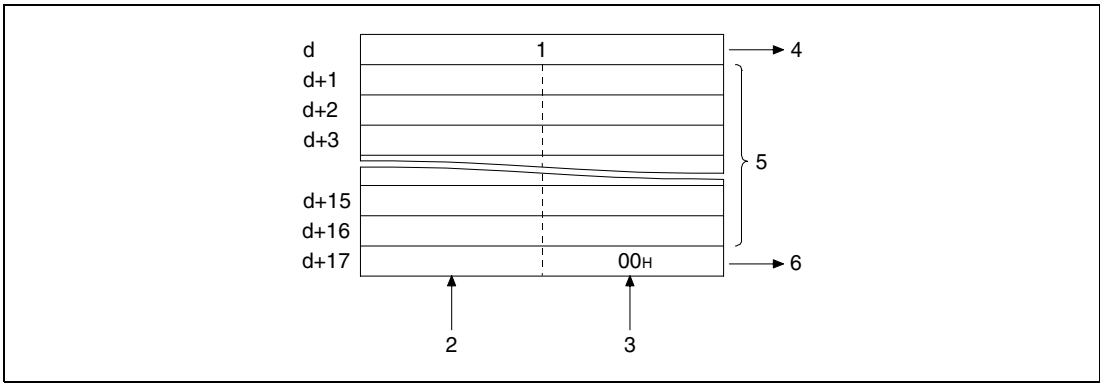
Mit Zurücksetzen der Ausführungsbedingung der PKEY-Anweisung werden auch die Merker SM736 und SM737 zurückgesetzt.

Der Merker SM737 wird gesetzt, wenn ein durch die Tastatur eingegebenes Zeichen von dem Peripheriegerät empfangen wird, und zurückgesetzt, wenn die QnA CPU die Tastatureingabedaten speichert. Solange der Merker SM737 gesetzt ist, können keine Tastatureingabedaten vom Peripheriegerät empfangen werden.

Die Tastatureingabe am Peripheriegerät ist beendet, wenn die Zeichenfolge "CR" empfangen wird.

Es können insgesamt 32 Zeichen eingegeben werden. Nach der Eingabe von 32 Zeichen wird der Empfang von Tastatureingabedaten von dem Peripheriegerät abgebrochen, ohne dass die Zeichenfolge "CR" empfangen wurde.

Die Speicherung der Tastatureingabedaten (Zeichen) in den in d+1 bis d+17 angegebenen Operanden erfolgt gemäß nachstehender Abbildung.



1 Zähler

2 2. bis 32. Zeichen

3 1. bis 31. Zeichen

4 Anzahl der eingegebenen Zeichen (Binärdatenwert)

5 Maximal 16 Zeichen

6 Der Zeichencode "00H" kennzeichnet das Ende der eingegebenen Zeichenfolge (Anzahl der eingegebenen Zeichen: ungerade = höherwertiges Byte, gerade = niedrigwertiges Byte)

Die PKEY-Anweisung kann nicht von zwei oder mehr Stellen gleichzeitig ausgeführt werden. Soll die PKEY-Anweisung von zwei oder mehr Stellen ausgeführt werden, muss mit dem Merker SM736 (Ausführungssignal der PKEY-Anweisung) ein Interlock realisiert werden, um eine gleichzeitige Ausführung zu verhindern.

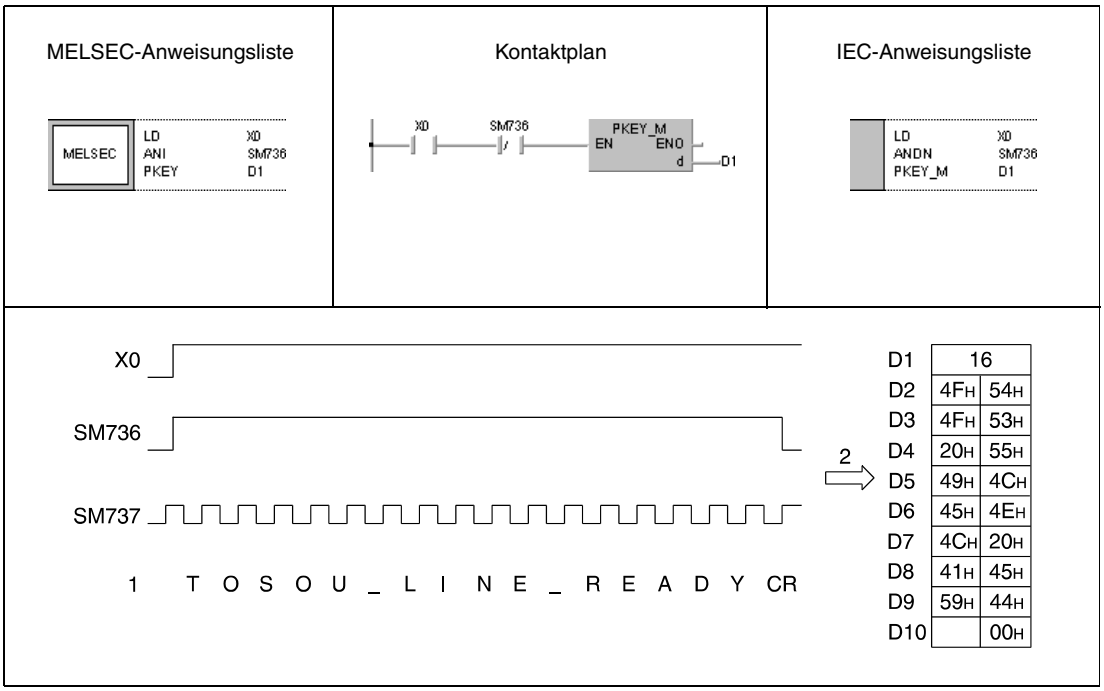
Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Es wurde der Versuch unternommen, Tastatureingabedaten zu speichern, die außerhalb des für die Speicherung vorgesehenen Bereichs der in d+0 bis d+17 angegebenen Operanden liegen (Fehlercode 4101).

Beispiel PKEY

Im folgenden Programm wird mit Einschalten von X0 die mittels Tastatur in das Peripheriegerät eingegebene Zeichenfolge "TOSOU LINE READY" in den Registern D1 bis D10 gespeichert.



[1] Eingabedaten

[2] Speicherung der eingegebenen Daten

7.17 Programmanweisungen

Die Programmanweisungen ermöglichen das Umschalten von Programmen in unterschiedliche Programmmodi. Die Tabelle liefert eine Übersicht der Anweisungen.

Funktion	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor
Umschaltung von Programmen in den Standby-Modus	PSTOP	PSTOP_M
	PSTOPP	PSTOPP_M
Umschaltung von Programmen in den Standby-Modus mit Rücksetzen der Ausgänge	POFF	POFF_M
	POFFP	POFFP_M
Umschaltung von Programmen in den Modus einer Programmausführung pro Zyklus	PSCAN	PSCAN_M
	PSCANP	PSCANP_M
Umschaltung von Programmen in den Modus niedriger Verarbeitungsgeschwindigkeit	PLOW	PLOW_M
	PLOWP	PLOWP_M

HINWEIS *„Bitte prüfen Sie in Ihrer vorliegenden Version des GX IEC Developers, ob diese Anweisungen unterstützt werden“.*

7.17.1 PSTOP, PSTOPP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●[1]

[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sonder-module U□\G□	Index-Register Zn	Konstanten $	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	—	—	—	●	—	SM0	2 + n[1)]

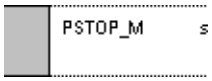
[1] n = (Anzahl Zeichen im Programmnamen/2) = Anzahl zusätzlicher Schritte. (Nachkommastellen werden aufgerundet).

GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan

IEC-Anweisungsliste



GX Developer

Variablen

Operand	Befehlswert	Datentyp
s	Zeichenfolge mit dem Namen der Programmdatei, die in den Standby-Modus umgeschaltet wird, oder erste Adresse des Operanden, in dem diese Daten gespeichert sind.	Zeichenfolge

Funktionsweise

Programmumschaltung in den Standby-Modus

PSTOP Umschaltanweisung für den Standby-Modus

Die PSTOP-Anweisung schaltet die Programmdatei, die in s benannt wird, in den Standby-Modus um. In diesem Modus wird die Programmdatei erst nach Aufruf ausgeführt.

Die Umschaltung in diesen Modus ist nur bei Programmdateien möglich, die in dem internen Speicher (Laufwerk 0) gespeichert sind.

Die Umschaltung in den Standby-Modus wird erst nach Verarbeiten der END-Anweisung wirksam.

Diese Anweisung hat auch in den Fällen Verarbeitungspriorität, in denen der Ausführungsmodus in den Parametern angegeben wird.

Bei der Benennung der Programmdatei muss die Erweiterung .QPG nicht angegeben werden, da dieser Dateityp automatisch angesprochen wird.

Fehlerquellen

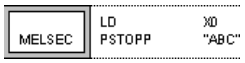
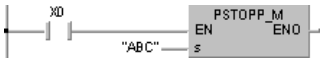
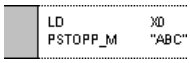
In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die durch den Dateinamen angegebene Programmdatei existiert nicht (Fehlercode 2410).

Beispiel

PSTOPP

Im folgenden Programm wird mit der positiven Flanke von X0 die Programmdatei "ABC" in den Standby-Modus umgeschaltet.

MELSEC-Anweisungsliste	Kontaktplan	IEC-Anweisungsliste
		

7.17.2 POFF, POFFP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●[1]

[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten $	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	—	—	—	●	—	SM0	2 + n[1)]

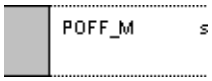
[1] n = (Anzahl Zeichen im Programmnamen/2) = Anzahl zusätzlicher Schritte. (Nachkommastellen werden aufgerundet).

GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan

IEC-Anweisungsliste



GX Developer

Variablen

Operand	Befehlswert	Datentyp
s	Erste Adresse des Operanden, in dem der Name der Programmdatei angegeben ist, die in den Standby-Modus mit Rücksetzen der Ausgänge umgeschaltet wird.	Zeichenfolge

Funktionsweise

Programmumschaltung in den Standby-Modus mit Rücksetzen der Ausgänge

POFF, Umschaltanweisung für den Standby-Modus mit zurückgesetzten Ausgängen

Die POFF-Anweisung schaltet die Programmdatei, die in s benannt wird, in den Standby-Modus mit Rücksetzen der Ausgänge um. In diesem Modus werden die von der Programmdatei angesprochenen Ausgänge und Spulen zunächst in den Zustand versetzt, als ob die Ausführungsbedingungen für die Anweisungen, die diese Kontakte und Spulen ansprechen, nicht gesetzt sind. Anschließend erfolgt die Umschaltung in den Standby-Modus.

Die Umschaltung in diesen Programmmodus ist nur bei Programmdateien möglich, die in dem internen Speicher (Laufwerk 0) gespeichert sind.

Die Umschaltung in den Standby-Modus mit Rücksetzen der Ausgänge wird erst nach Verarbeiten der END-Anweisung wirksam.

Diese Anweisung hat auch in den Fällen Verarbeitungspriorität, in denen der Ausführungsmodus in den Parametern angegeben wird.

Bei der Benennung der Programmdatei muss die Erweiterung .QPG nicht angegeben werden, da dieser Dateityp automatisch angesprochen wird.

HINWEIS *Bei Verwendung der POFF-Anweisung werden die von einer OUT-Anweisung angesprochenen Spulen zurückgesetzt (siehe Funktionsweise).*

Fehlerquellen

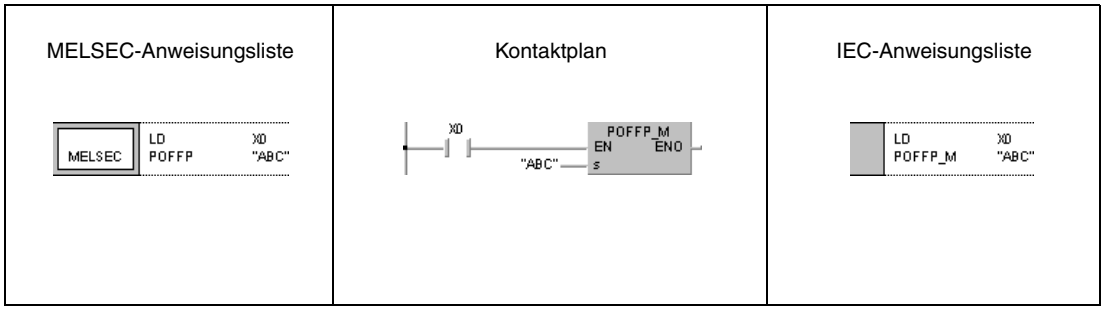
In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die durch den Dateinamen angegebene Programmdatei existiert nicht (Fehlercode 2410).

Beispiel

POFFP

Im folgenden Programm wird mit der positiven Flanke von X0 die Programmdatei "ABC" in den Standby-Modus mit Rücksetzen der Ausgänge umgeschaltet. In diesem Modus werden die von der Programmdatei "ABC" angesprochenen Ausgänge und Spulen zunächst in den Zustand versetzt, als ob die Ausführungsbedingungen für die Anweisungen, die diese Kontakte und Spulen ansprechen, nicht gesetzt sind. Anschließend wird die Programmdatei "ABC" in den Standby-Modus umgeschaltet.



7.17.3 PSCAN, PSCANP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●[1]

[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

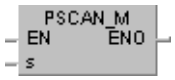
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten $	Andere		
	Bit	Wort		Bit	Wort				DY		
s	—	●	●	—	—	—	—	●	—	SM0	2 + n[1)]

[1] n = (Anzahl Zeichen im Programmnamen/2) = Anzahl zusätzlicher Schritte. (Nachkommastellen werden aufgerundet).

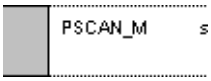
GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

Operand	Befehlswert	Datentyp
s	Zeichenfolge mit dem Namen der Programmdatei, die in den Modus einer Programmausführung pro Zyklus umgeschaltet wird, oder erste Adresse des Operanden, in dem diese Daten gespeichert sind.	Zeichenfolge

Funktionsweise

Programmumschaltung in den Modus einer Programmausführung pro Zyklus

PSCAN Umschaltanweisung für den Modus einer Programmausführung pro Zyklus

Die PSCAN-Anweisung schaltet die Programmdatei, die in s benannt wird, in den Modus einer Programmausführung pro Zyklus um. In diesem Modus wird das Programm einmal in jedem Programmzyklus ausgeführt.

Die Umschaltung in diesen Modus ist nur bei Programmdateien möglich, die in dem internen Speicher (Laufwerk 0) gespeichert sind.

Die Umschaltung in den Modus einer Programmausführung pro Zyklus wird erst nach Verarbeiten der END-Anweisung wirksam.

Diese Anweisung hat auch in den Fällen Verarbeitungspriorität, in denen der Ausführungsmodus in den Parametern angegeben wird.

Bei der Benennung der Programmdatei muss die Erweiterung .QPG nicht angegeben werden, da dieser Dateityp automatisch angesprochen wird.

Fehlerquellen

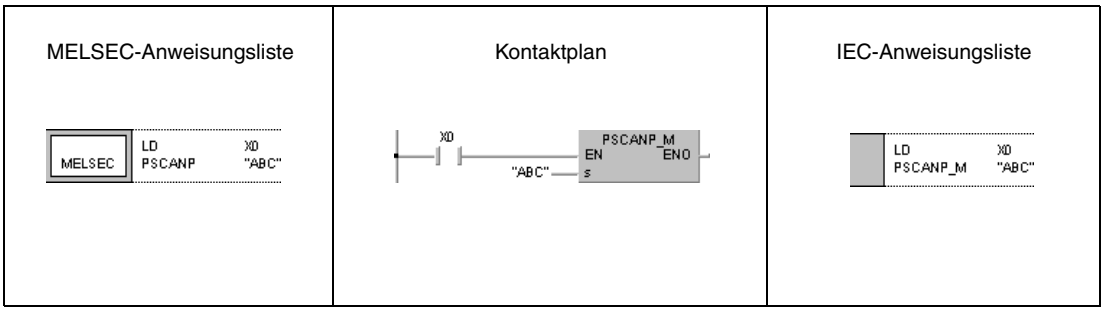
In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die durch den Dateinamen angegebene Programmdatei existiert nicht (Fehlercode 2410).

Beispiel

PSCANP

Im folgenden Programm wird mit der positiven Flanke von X0 die Programmdatei "ABC" in den Modus einer Programmausführung pro Zyklus umgeschaltet.



7.17.4 PLOW, PLOWP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●[1]

[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

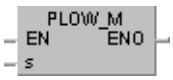
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten $	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	—	—	—	●	—	SM0	2 + n[1)]

[1] n = (Anzahl Zeichen im Programmnamen/2) = Anzahl zusätzlicher Schritte. (Nachkommastellen werden aufgerundet).

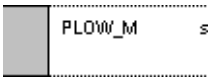
GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

Operand	Befehlswert	Datentyp
s	Zeichenfolge mit dem Namen der Programmdatei, die in den Modus niedriger Verarbeitungsgeschwindigkeit umgeschaltet wird oder erste Adresse des Operanden, in dem diese Daten gespeichert sind.	Zeichenfolge

Funktionsweise

Programmumschaltung in den Modus niedriger Verarbeitungsgeschwindigkeit

PLOW Umschaltanweisung für den Modus niedriger Verarbeitungsgeschwindigkeit

Die PLOW-Anweisung schaltet die Programmdatei, die in s benannt wird, in den Modus niedriger Verarbeitungsgeschwindigkeit um. In diesem Modus wird das Programm mit niedriger Verarbeitungsgeschwindigkeit ausgeführt.

Die Umschaltung in diesen Modus ist nur bei Programmdateien möglich, die in dem internen Speicher (Laufwerk 0) gespeichert sind.

Die Umschaltung in den Modus niedriger Verarbeitungsgeschwindigkeit wird erst nach Verarbeiten der END-Anweisung wirksam.

Diese Anweisung hat auch in den Fällen Verarbeitungspriorität, in denen der Ausführungsmodus in den Parametern angegeben wird.

Bei der Benennung der Programmdatei muss die Erweiterung .QPG nicht angegeben werden, da dieser Dateityp automatisch angesprochen wird.

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die durch den Dateinamen angegebene Programmdatei existiert nicht (Fehlercode 2410).
- Innerhalb der angegebenen Programmdatei befindet sich eine CHK-Anweisung (Fehlercode 4235).

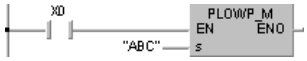
Beispiel

PLOWP

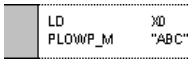
Im folgenden Programm wird mit der positiven Flanke von X0 die Programmdatei "ABC" in den Modus niedriger Verarbeitungsgeschwindigkeit umgeschaltet.

MELSEC-Anweisungsliste

Kontaktplan



IEC-Anweisungsliste



7.18 Weitere Anweisungen

In dem Abschnitt befinden sich Anweisungen zum Setzen und Rücksetzen von WDT und Übertragstellen, zur Vorgabe von Ausführungszyklen, Anweisungen zum Schreiben, Zuweisen einer Adresse zur indirekten Adressierung, Lesen und Eingeben unterschiedlicher Daten in und aus verschiedenen Speichern. Die Tabelle liefert eine Übersicht der Anweisungen.

Funktion	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor
Watch-Dog-Timer zurücksetzen	WDT	WDT_M
Übertragstelle setzen und zurücksetzen	STC	STC_M
	CLC	CLC_M
Vorgabe von Ausführungszyklen	DUTY	DUTY_M
Direktes Lesen eines Bytes	ZRRDB	ZRRDB_M
	ZRRDBP	ZRRDBP_M
Direktes Schreiben eines Bytes	ZRWRB	ZRWRB_M
	ZRWRBP	ZRWRBP_M
Operand, der indirekt angesprochen werden soll, speichern	ADRSET	ADRSET_M
	ADRESETP	ADRSETP_M
Tastatureingabe numerischer Werte	KEY	KEY_MD
Sichern der Indexregisterinhalte in ein Register	ZPUSH	ZPUSH_M
	ZPUSHP	ZPUSHP_M
Wiederherstellen der Indexregisterinhalte aus einem Register	ZPOP	ZPOP_M
	ZPOPP	ZPOPP_M
Schreiben von Daten in ein EEPROM-Register	EROMWR	EROMWR_M
	EROMWRP	EROMWRP_M

HINWEIS *Die Anweisungen ADRSET und ADRSETP können nur bei der Programmierung mit dem GX Developer verwendet werden.*

7.18.1 WDT, WDTP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●[1]	●	●

[1] AnA + MELSECNET/10

Operanden MELSEC A

	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
																							1 ●[1]			

[1] Die Anzahl der Schritte bei der Verwendung einer AnA, AnAS oder AnU CPU (AnA + MELSECNET/10) ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

Operanden MELSEC Q

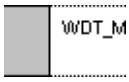
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten	Andere		
	Bit	Wort		Bit	Wort						
—	—	—	—	—	—	—	—	—	—	—	1

GX IEC Developer

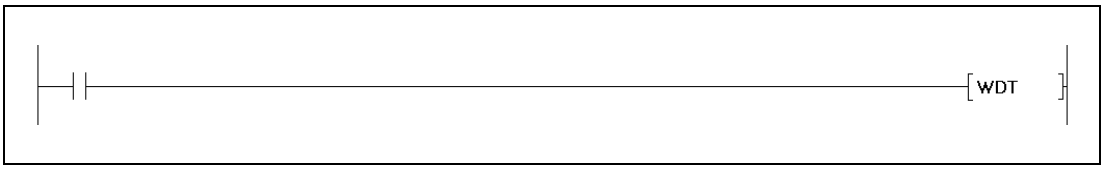
MELSEC-Anweisungsliste

Kontaktplan

IEC-Anweisungsliste



GX Developer



Variablen

Operand	Befehlswert	Datentyp
—	—	—

Funktionsweise

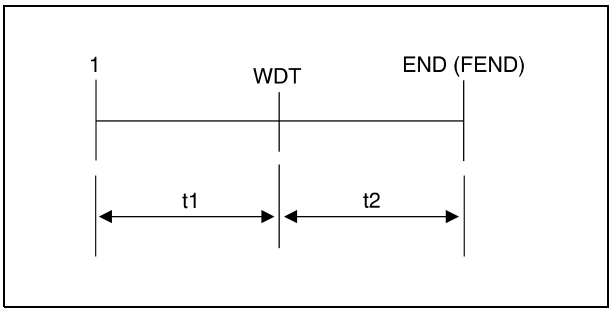
Rücksetzen von Watch-Dog-Timern

WDT Rücksetzen

Die WDT-Anweisung setzt den Watch-Dog-Timer (WDT) in einem Ablaufprogramm zurück.

Die Anweisung ist nur dann zu programmieren, wenn die Zykluszeit des Ablaufprogramms zwischen Programmschritt 0 und END-/FEND-Anweisung die vorgegebene Zeitspanne des Watch-Dog-Timers unter bestimmten Bedingungen überschreitet. Überschreitet die Zykluszeit den vorgegebenen Wert des Watch-Dog-Timers bei jedem Programmzyklus, ist die Parametereinstellung des Watch-Dog-Timers zu ändern.

Der Sollwert des Watch-Dog-Timers muss so gewählt werden, dass die Zeitspanne t1 (zwischen Schritt 0 und WDT-Anweisung) und t2 (zwischen WDT- und END-/FEND-Anweisung) den Sollwert nicht übersteigt.



[1]Schritt 0

Die WDT-Anweisung kann beliebig oft in einem Programmzyklus gesetzt werden. Bei der Programmerstellung ist aber zu berücksichtigen, dass die Ausgänge im Fehlerfall nicht unmittelbar abgeschaltet werden.

Die Werte der Programmzykluszeit, die in den Registern gespeichert sind, werden durch Ausführung der WDT-Anweisung nicht gelöscht. Die in den Registern gespeicherten Werte können daher unter Umständen größer als die über Parameter gesetzten WDT-Werte sein.

HINWEIS *CPUs der Serien A3H, A3M, AnA, AnAS und AnU verwenden Festwerte für Watch-Dog-Timer (A-Serie).*

7.18.2 STC, CLC

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●		

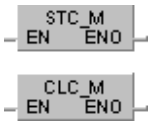
Operanden MELSEC A

	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)								Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
																							1			

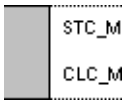
GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan



IEC-Anweisungsliste



Variablen

Operand	Befehlswert	Datentyp
—	—	—

Funktionsweise

Carry Flag setzen und zurücksetzen

STC Carry Flag setzen

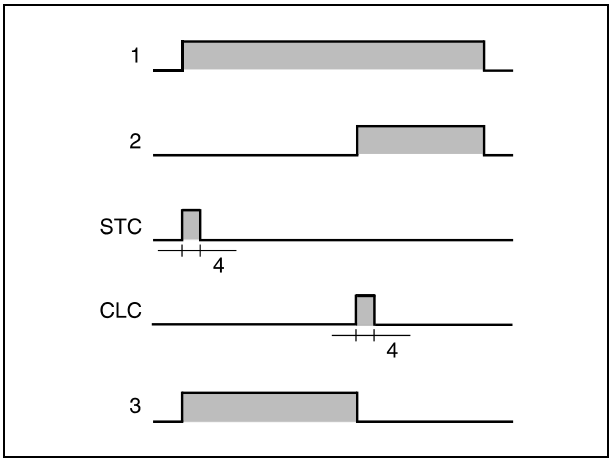
Im Carry Flag wird bei Rotations- und Schiebevorgängen der Übertrag (0 oder 1) gespeichert. Der Übertrag wird als Kontakt im Programm durch den Sondermerker M9012 wiedergegeben. M9012 ist gesetzt, wenn das Carry Flag gleich 1 ist, und ist nicht gesetzt, wenn das Carry Flag gleich 0 ist.

Mit Ausführung der STC-Anweisung wird das Carry Flag (M9012) zwangsweise gesetzt.

CLC Carry Flag rücksetzen

Das Rücksetzen des Carry Flags erfolgt nach Ausführung der CLC-Anweisung. Gleichzeitig wird der Sondermerker M9012 zurückgesetzt.

Die STC-/CLC-Anweisung wird einmalig bei ansteigender Signalflanke der Eingangsbedingung ausgeführt.



[1]Ausführungsbedingung der STC-Anweisung
[2]Ausführungsbedingung der CLC-Anweisung
[3]Carry Flag (M9012)
[4]einmalige Ausführung

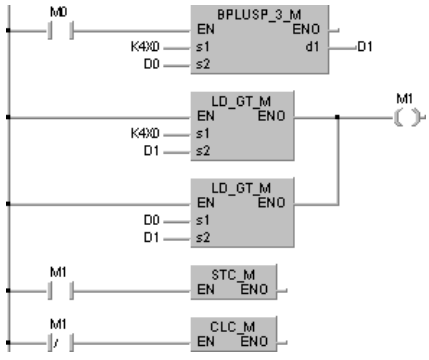
Beispiel STC, CLC

Im folgenden Programm werden mit positiver Flanke von M0 die BCD-Daten aus X0 bis XF zu den BCD-Daten aus D0 addiert und das Ergebnis in D1 gespeichert. Ist das Ergebnis der Addition größer als 9999, wird M1 gesetzt und die STC-Anweisung ausgeführt (M9012 wird gesetzt). Das Carry Flag wird nicht gesetzt, wenn das Ergebnis kleiner oder gleich 9999 ist.

MELSEC-Anweisungsliste

```
MELSEC   LD     M0
         B+P    K4X0
                D0
                D1
         LD>    K4X0
                D1
         OR>    D0
                D1
         OUT    M1
         LD     M1
         STC
         LDI    M1
         CLC
```

Kontaktplan



IEC-Anweisungsliste

```
LD       M0
BPLUSP_3_M  K4X0 , D0 , D1
LD_GT_M  K4X0 , D1
OR_GT_M  D0 , D1
ST       M1
LD       M1
STC_M
LDN      M1
CLC_M
```

7.18.3 DUTY

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Operanden MELSEC A

	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
n1																	●	●								
n2																	●	●					7 ●[1]	●[2]		●
d			●																							

[1] Die Anzahl der Schritte bei der Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

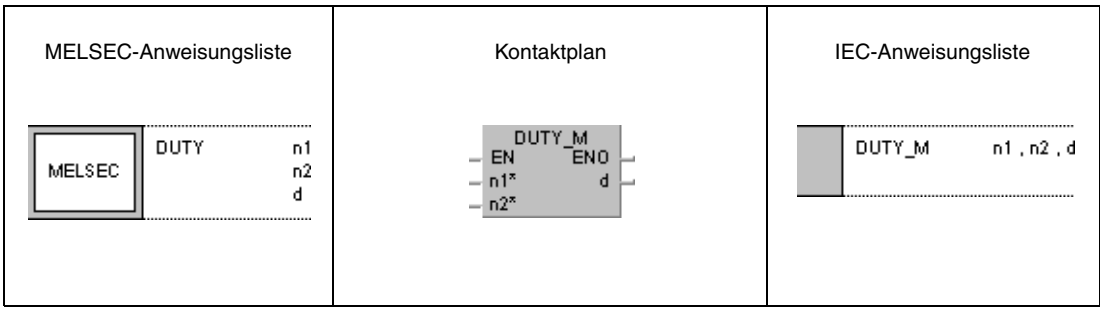
[2] Index-Vergabe ist nur in Verbindung mit einer A3H, A3M, AnA, AnAS und AnU CPU möglich.

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sonder-module U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
n1	●	●	●	●	●	●	●	●	●	SM0	4
n2	●	●	●	●	●	●	●	●	●		
d	●[1]	—	—	—	—	—	—	—	—		

[1] Nur SM420 bis SM424 und SM430 bis SM434

GX IEC Developer



GX Developer

Variablen

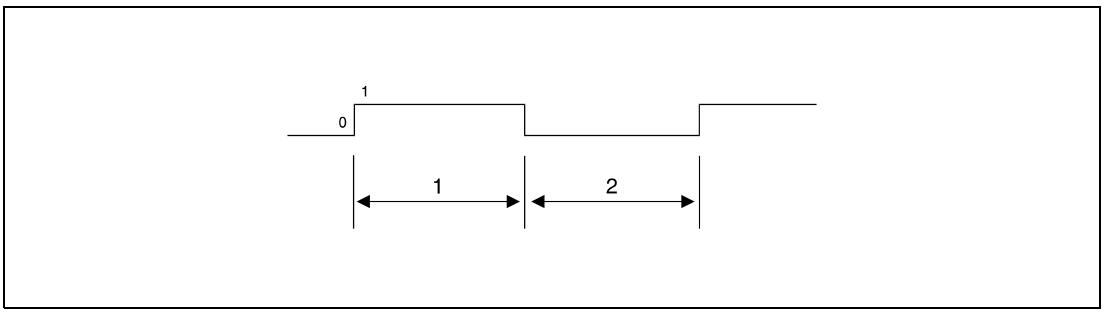
Operand	Befehlswert	Datentyp
n1	Anzahl der Zyklen, in denen die Sonder-/Diagnosemerker gesetzt sind.	BIN-16-Bit
n2	Anzahl der Zyklen, in denen die Sonder-/Diagnosemerker zurückgesetzt sind.	
d	Sonder-/Diagnosemerkeradresse (A-Serie = M9020 - M9024, Q-Serie /System Q = SM420 - SM424 und SM430 - SM434).	Bit

Funktions-weise

Vorgabe von Ausführungszyklen eines Operanden

DUTY Vorgabe der Ausführungszyklen

Die DUTY-Anweisung schaltet die in d angegebenen Operanden (A-Serie = M9020 bis M9024, Q-Serie/System Q = SM420 bis SM424 und SM430 bis SM434) für die in n1 angegebene Anzahl von Programmyklen ein und für die in n2 angegebene Anzahl von Programmzyklen aus. Der oder die entsprechenden Merker (A-Serie = Sondermerker, Q-Serie/System Q = Diagnosemerker) können dann als Eingangsbedingung für Folgeoperationen dienen.



[1] Anzahl der Programmzyklen mit Ausführung

[2] Anzahl der Programmzyklen ohne Ausführung

Programme, die einmal pro Programmzyklus ausgeführt werden, verwenden die Merker SM420 bis SM424 (Q-Serie und System Q).

Programme mit niedriger Verarbeitungsgeschwindigkeit verwenden die Merker SM430 bis SM434 (Q-Serie und System Q).

Zu Beginn der Verarbeitung (Initialisierungsstatus) sind die Merker (A-Serie = M9020 bis M9024, Q-Serie/System Q = SM420 bis SM424 und SM430 bis SM434) ausgeschaltet.

Ist der Wert in n1 = 0, bleiben die Merker ausgeschaltet.

Ist der Wert in n2 = 0 und der Wert in n1 größer als 0, werden die Merker gesetzt und bleiben eingeschaltet.

Die Werte in n1, n2 und d werden bei Aufruf der DUTY-Anweisung gesetzt. Der Zyklusimpuls (Merker) wird bei Erreichen der END-Anweisung ein- oder ausgeschaltet.

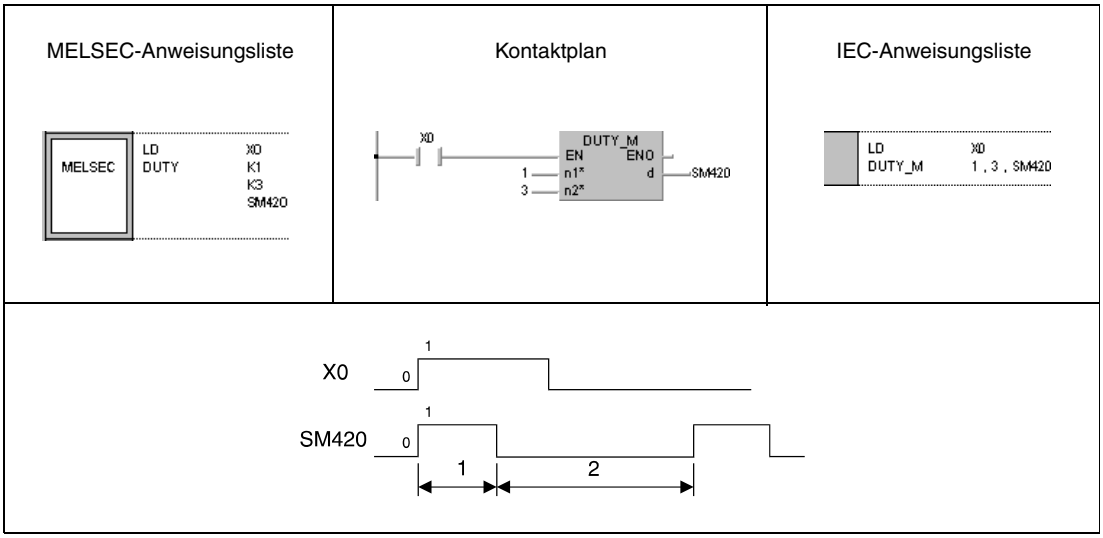
Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Der in d angegebene Operand ist keiner der A-Serie-Merker M9020 bis M9024 bzw. der Q-Serie/System Q-Merker SM420 bis SM424 und SM430 bis SM434 (Fehlercode 4101).
- Die Werte in n1 und n2 sind negativ (Fehlercode 4100).

Beispiel DUTY (Q-Serie)

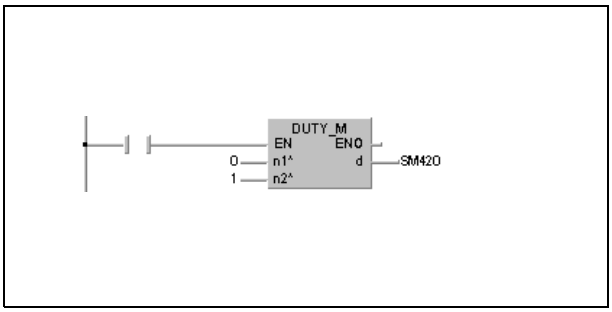
Im folgenden Programm wird der Diagnosemerker mit positiver Flanke von X0 für einen Programmzyklus gesetzt und für 3 Zyklen zurückgesetzt. Der Vorgang wiederholt sich, solange die Programmverarbeitung andauert (siehe Hinweis unten).



[1] Ein Programmzyklus mit Ausführung

[2] Drei Programmzyklen ohne Ausführung

HINWEIS *Nach dem Wegfall der Ausführungsbedingung (X0 = AUS) dauert der Einschaltimpuls der DUTY-Anweisung und somit das zyklische Setzen/Rücksetzen des angegebenen Merkers weiter an. Um die weitere Ausgabe des Einschaltimpulses zu stoppen, muss der folgende Programmteil eingefügt werden.*



7.18.4 ZRRDB, ZRRDBP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

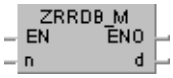
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
n	●	●	●	●	●	●	●	●	—	SM0	3
d	●	●	●	●	●	●	●	—	—		

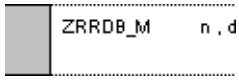
GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

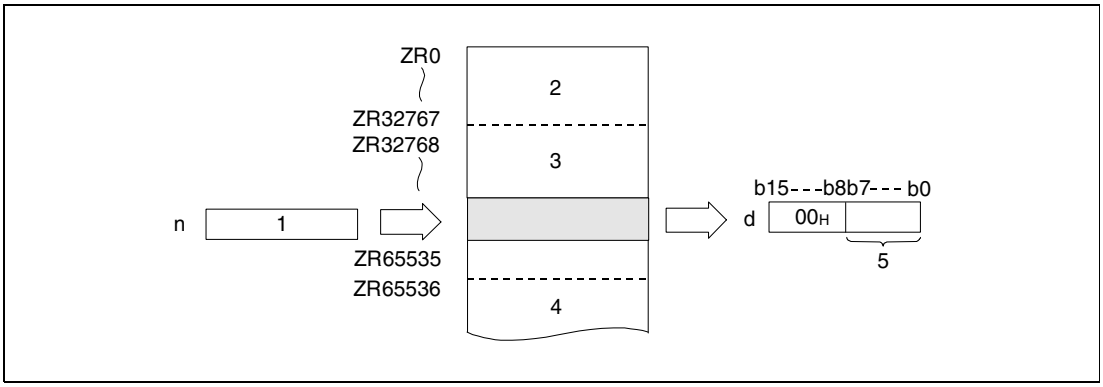
Operand	Befehlswert	Datentyp
n	Serielle Byte-Adresse, die aus dem File-Register ausgelesen wird.	BIN-32-Bit
d	Operand, in dem die gelesenen Bytes gespeichert werden.	BIN-16-Bit

Funktionsweise

Direktes Auslesen eines Bytes aus einem File-Register

ZRRDB Auslesen eines Bytes

Die ZRRDB-Anweisung liest ein Byte der in n angegebenen Byte-Adresse aus einem File-Register aus. Die Byte-Adresse bezeichnet keine Blockadresse. Das gelesene Byte wird in dem niedrigwertigen Byte des in d angegebenen Operanden gespeichert. Das höherwertige Byte des in d angegebenen Operanden wird mit dem Wert "00_H" beschrieben.



[1] Serielle Byte-Adresse

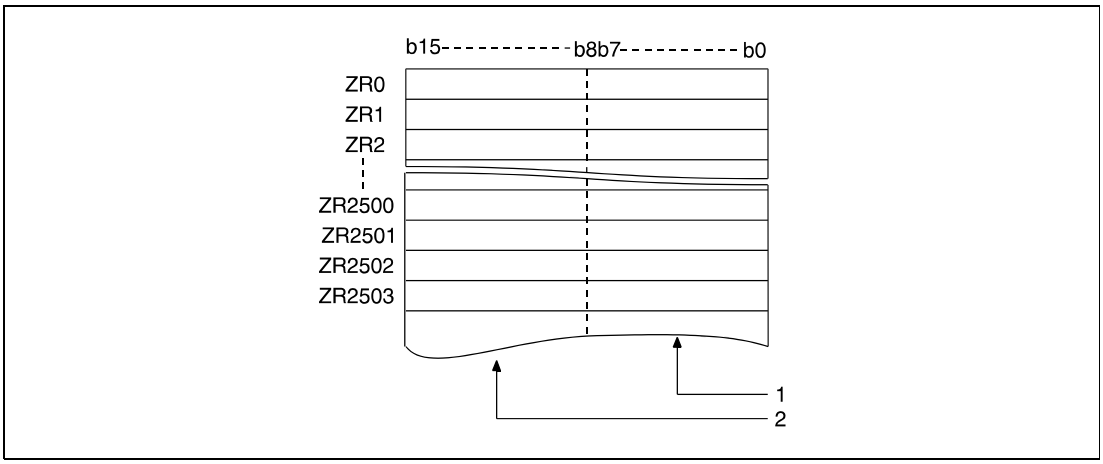
[2] File-Registerbereich für Block 0

[3] File-Registerbereich für Block 1

[4] File-Registerbereich für Block 2

[5] Ausgelesenes Byte

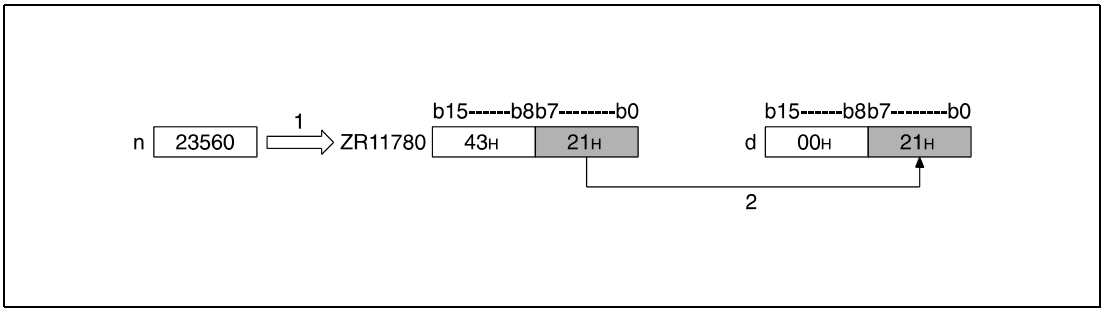
Eine Zuordnung der File-Registeradressen zu den entsprechenden seriellen Byte-Adressen liefert folgende Abbildung.



[1] Speicherbereich für gerade Byte-Adressen (hier Adresse 0 bis Adresse 5006)

[2] Speicherbereich für ungerade Byte-Adressen (hier Adresse 1 bis Adresse 5007)

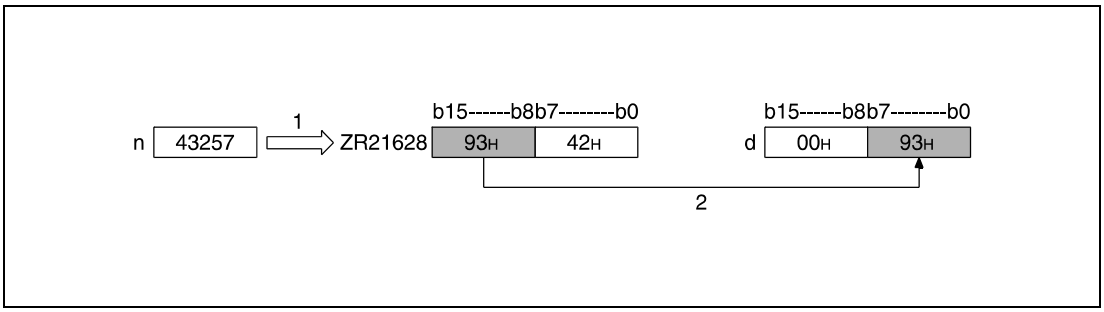
Bei Angabe der Byte-Adresse 23560 wird beispielsweise das niedrigwertige Byte des File-Registers ZR11780 ausgelesen.



[1] Adressierung

[2] Speicherung

Bei Angabe der Byte-Adresse 43257 wird beispielsweise das niedrigwertige Byte des File-Registers ZR21628 ausgelesen.



[1] Adressierung

[2] Speicherung

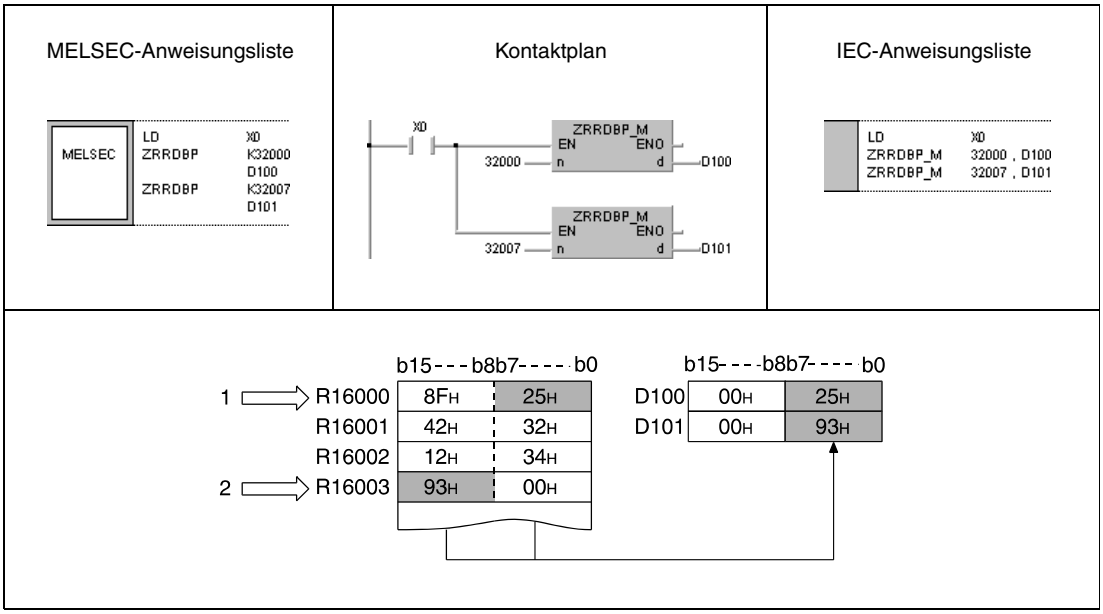
Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die in n angegebene Operandenadresse (serielle Byte-Adresse) liegt außerhalb des für die Speicherung vorgesehenen Bereichs (Fehlercode 4101).

Beispiel ZRRDBP

Im folgenden Programm werden mit positiver Flanke von X0 das niedrigwertige Byte des File-Registers R16000 (Byte-Adresse 32000) und das höherwertige Byte des File-Registers R16003 (Byte-Adresse 32007) ausgelesen und in D100 und D101 gespeichert.



[1] Serielle Byte-Adresse 32000 (niedrigwertiges Byte von File-Register R16000)

[2] Serielle Byte-Adresse 32007 (höherwertiges Byte von File-Register R16003)

7.18.5 ZRWRB, ZRWRBP

CPU

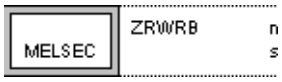
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Operanden MELSEC Q

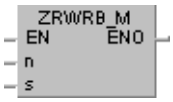
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
n	●	●	●	●	●	●	●	●	—	SM0	3
s	●	●	●	●	●	●	●	●	—		

GX IEC Developer

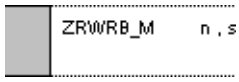
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

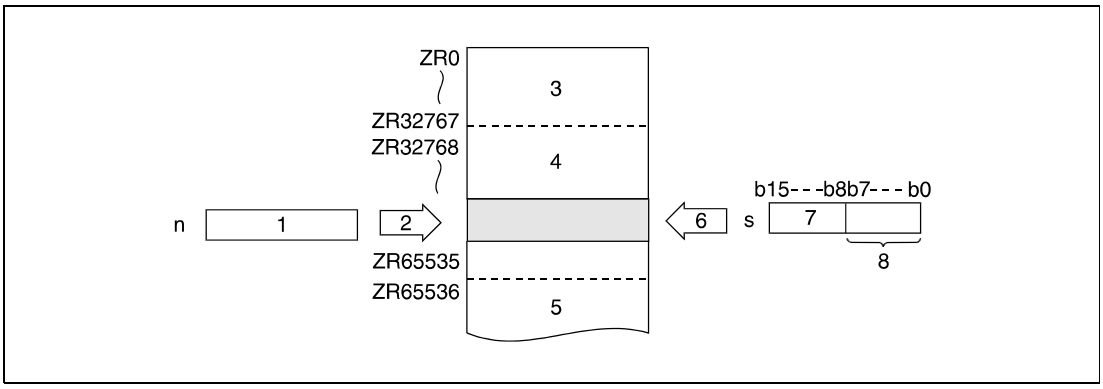
Operand	Befehlswert	Datentyp
n	Serielle Byte-Adresse, die in dem File-Register beschrieben wird.	BIN-32-Bit
s	Operand, in dem die zu schreibenden Daten gespeichert werden.	BIN-16-Bit

Funktionsweise

Direktes Schreiben eines Bytes in ein File-Register

ZRWRB Schreiben eines Bytes

Die ZRRDB-Anweisung schreibt den Inhalt des niedrigwertigen Bytes des in s angegebenen Operanden, das keine Blockadresse bezeichnet, in die in n angegebene serielle Byte-Adresse eines File-Registers. Das höherwertige Byte des in s angegebenen Operanden wird ignoriert.



[1] Serielle Byte-Adresse

[2] Adressierung

[3] File-Registerbereich für Block 0

[4] File-Registerbereich für Block 1

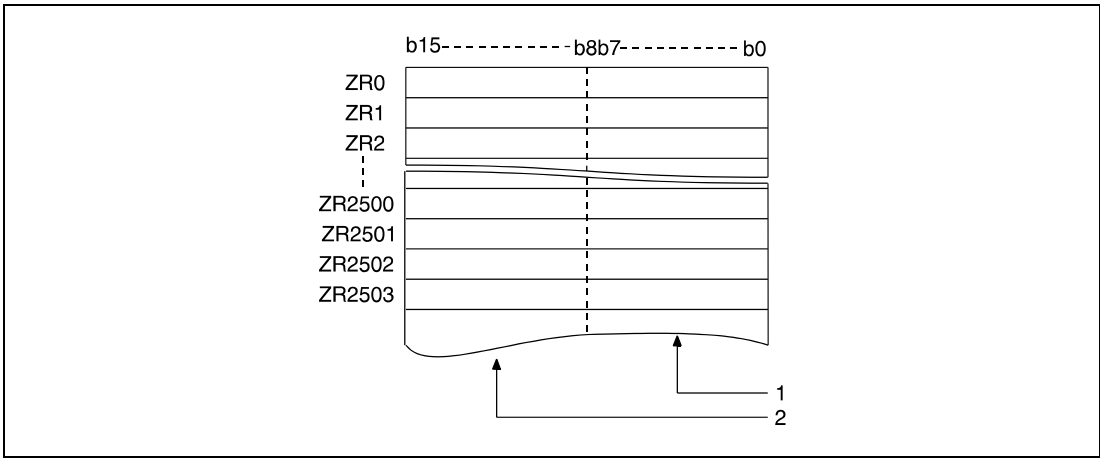
[5] File-Registerbereich für Block 2

[6] Schreiben der Daten

[7] Dieses Byte wird bei der Programmverarbeitung nicht berücksichtigt

[8] Zu schreibendes Byte

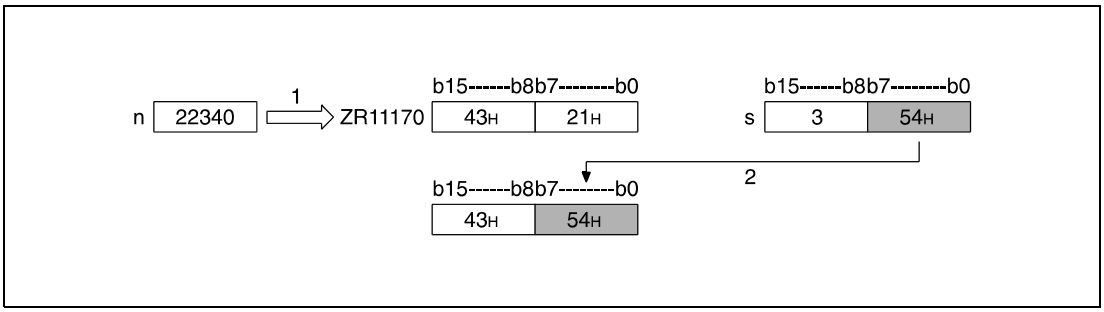
Eine Zuordnung der File-Registeradressen zu den entsprechenden seriellen Byte-Adressen liefert folgende Abbildung.



[1] Speicherbereich für gerade Byte-Adressen (hier Adresse 0 bis Adresse 5006)

[2] Speicherbereich für ungerade Byte-Adressen (hier Adresse 1 bis Adresse 5007)

Bei Angabe der Byte-Adresse 22340 wird beispielsweise das niedrigwertige Byte des File-Registers ZR 11170 mit dem Inhalt des niedrigwertigen Bytes des in s angegebenen Operanden beschrieben.

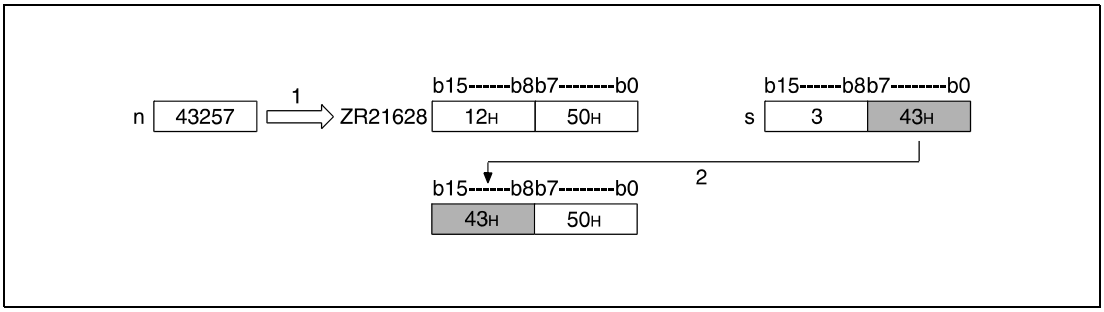


[1] Adressierung

[2] Schreiben des Bytes

[3] Dieses Byte wird bei der Programmverarbeitung nicht berücksichtigt

Bei Angabe der Byte-Adresse 43257 wird beispielsweise das höherwertige Byte des File-Registers ZR21628 mit dem Inhalt des niedrigwertigen Bytes des in s angegebenen Operanden beschrieben.



[1] Adressierung

[2] Schreiben des Bytes

[3] Dieses Byte wird bei der Programmverarbeitung nicht berücksichtigt

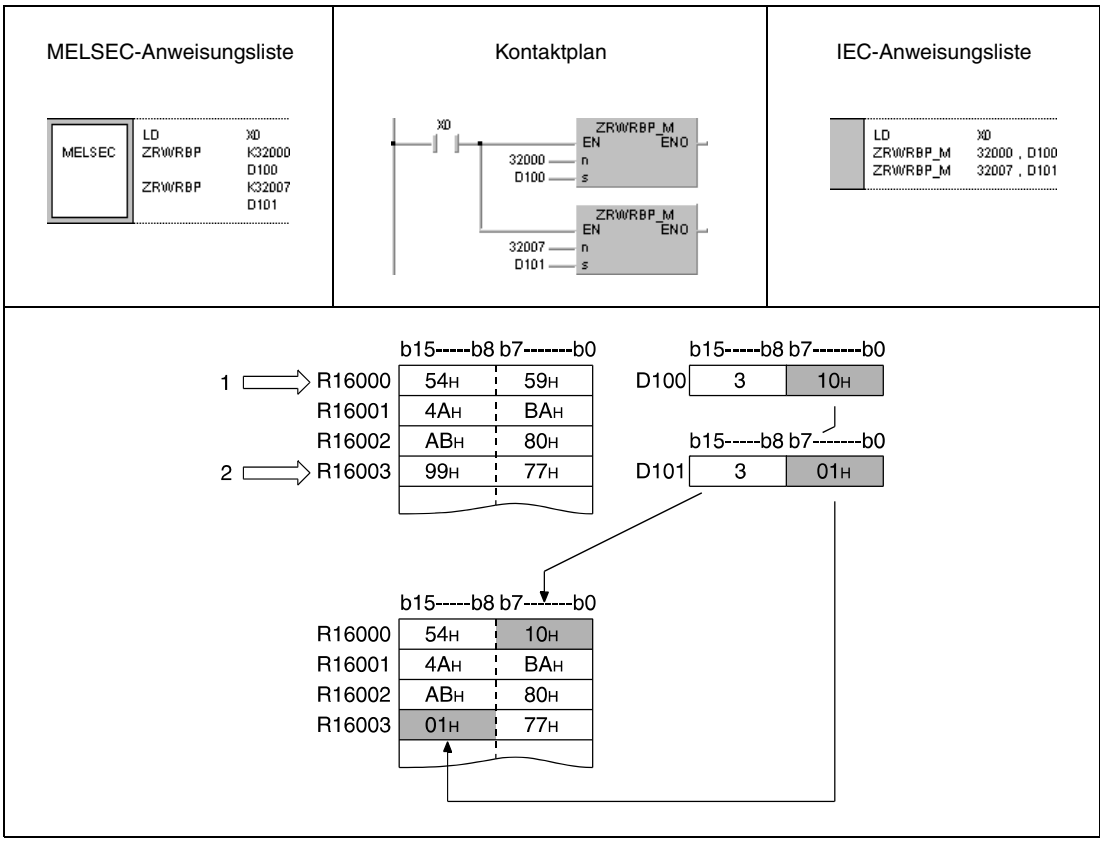
Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die in n angegebene Operandenadresse (serielle Byte-Adresse) liegt außerhalb des für die Speicherung vorgesehenen Bereichs (Fehlercode 4101).

Beispiel ZRWRBP

Im folgenden Programm werden mit positiver Flanke von X0 die Inhalte der niedrigwertigen Bytes der Register D100 und D101 in das niedrigwertige Byte des File-Registers R16000 (Byte-Adresse 32000) und in das höherwertige Byte des File-Registers R16003 (Byte-Adresse 32007) geschrieben.



[1] Serielle Byte-Adresse 32000 (niedrigwertiges Byte von File-Register R16000)

[2] Serielle Byte-Adresse 32007 (höherwertiges Byte von File-Register R16003)

[3] Diese Bytes werden bei der Verarbeitung nicht berücksichtigt

7.18.6 ADRSET, ADRSETP

CPU

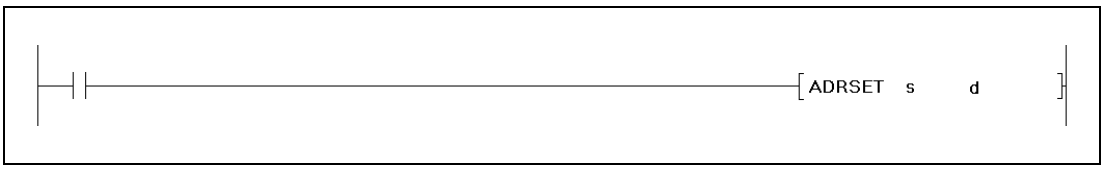
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	●	●	●	—	—	—	—	—	—	SM0	3
d	●	●	●	—	—	—	—	—	—		

HINWEIS *Diese Anweisungen können nicht im GX IEC Developer programmiert werden.*

GX Developer



Variablen

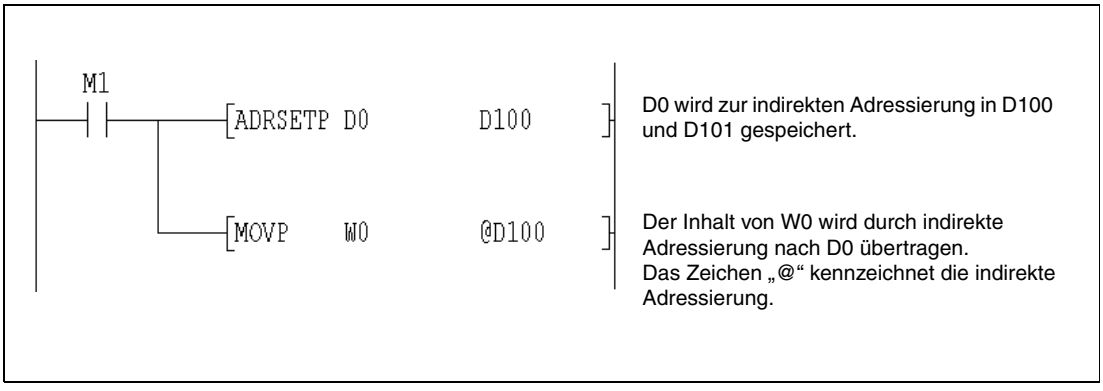
Operand	Befehlswert	Datentyp
s	Operand, der durch indirekte Adressierung angesprochen werden soll	Operanden-name
d	Operand, in dem die Adresse des Operanden gespeichert wird, der durch indirekte Adressierung angesprochen wird	BIN-32-Bit

Funktions-weise

Zuweisung eines Operanden zur indirekten Adressierung

ADRSET Operandenadresse speichern

Der in s angegebene Operand, wird zur indirekten Adressierung in den Operanden d und d+1 gespeichert. In s kann kein Bit-Operand angegeben werden.



Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Ein unzulässiger Operand wurde angegeben (Fehlercode 4100).

7.18.7 KEY

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●[1]	●[1]	●	●[2]

[1] Diese erweiterte Anweisung kann bei der AnA- und AnU-CPU im IEC-Editor als normaler Baustein (Funktion) und im MELSEC-Editor in Verbindung mit den LEDA, -C und -R-Anweisungen programmiert werden.

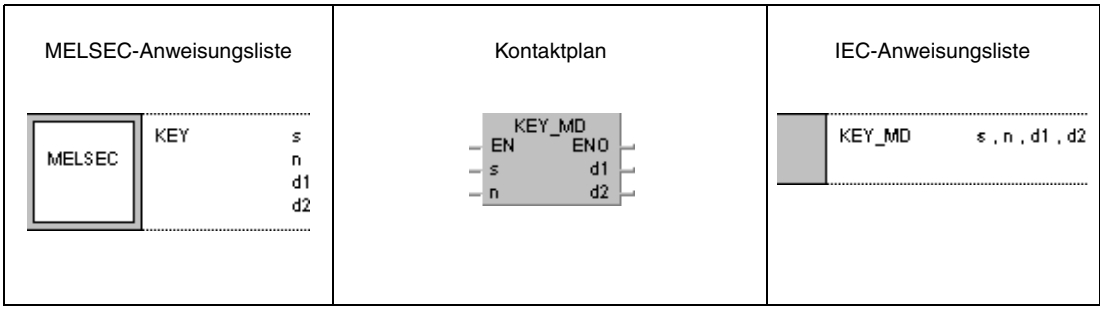
[2] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

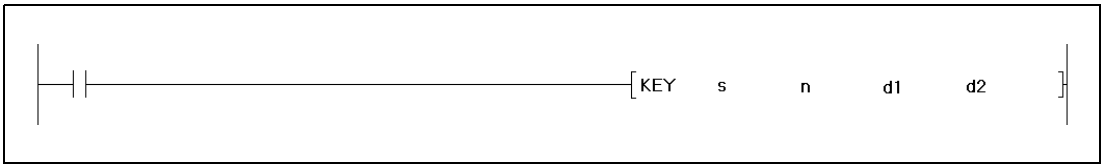
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sonder-module U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	●[1]	—	—	—	—	—	—	—	—	SM0	5
n	●	●	●	●	●	●	●	●	—		
d1	—	●	●	—	—	—	—	—	—		
d2	●	●	●	●	●	●	—	—	—		

[1] Nur X

GX IEC Developer



GX Developer



Variablen

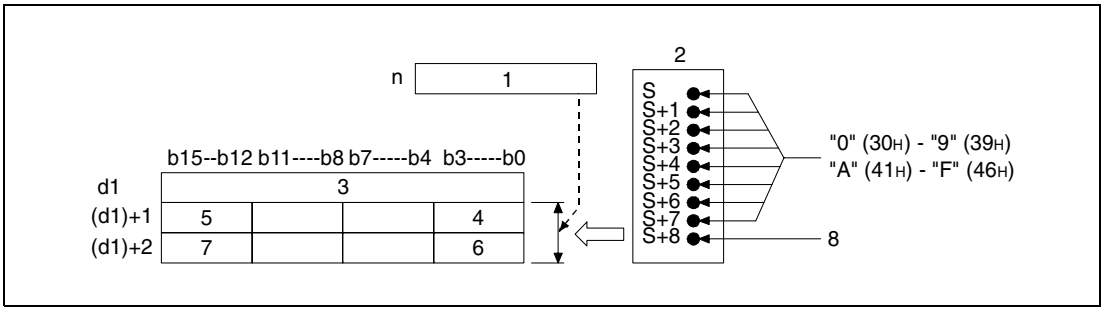
Operand	Befehlswert	Datentyp	
		MELSEC	IEC
s	Erste Adresse der Eingangsoperanden (X), an denen der numerische Wert eingegeben wird.	Bit	Array [1..9] of BOOL
n	Anzahl der numerischen Werte, die eingegeben werden.	BIN-16-Bit	ANY16
d1	Erste Adresse des Operanden, in dem die eingegebenen Werte gespeichert werden.	BIN-16-Bit	Array [1..3] of ANY16
d2	Operand, der nach Abschluss der Eingabe gesetzt wird.	Bit	BOOL

Funktionsweise

Tastatureingabe numerischer Werte

KEY Eingabeanweisung

Die KEY-Anweisung ermöglicht die Tastatureingabe der ASCII-Zeichen 0 (30H) bis 9 (39H) und A (41H) bis F (46H) an den in s+0 (Array_s[1]) bis s+7 (Array_s[8]) angegebenen Eingängen (X). Die an den Eingängen eingegebenen Werte werden hexadezimal codiert und in den in (d1)+0 (Array_d1[1]) bis (d1)+2 (Array_d1[3]) angegebenen Operanden gespeichert. Die Anzahl der einzugebenden Zeichen wird in n angegeben.



[1] Anzahl der einzugebenden Werte

[2] Eingangsmodul

[3] Anzahl der eingegebenen Werte

[4] 8. eingegebenes Zeichen

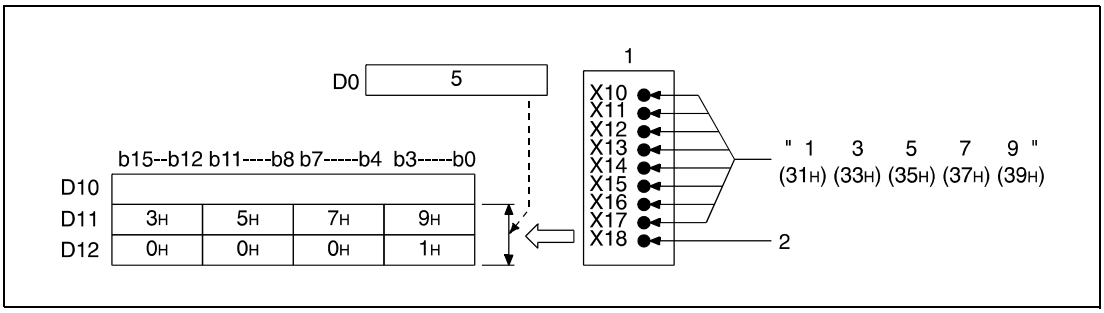
[5] 5. eingegebenes Zeichen

[6] 4. eingegebenes Zeichen

[7] 1. eingegebenes Zeichen

[8] Strobe-Signal

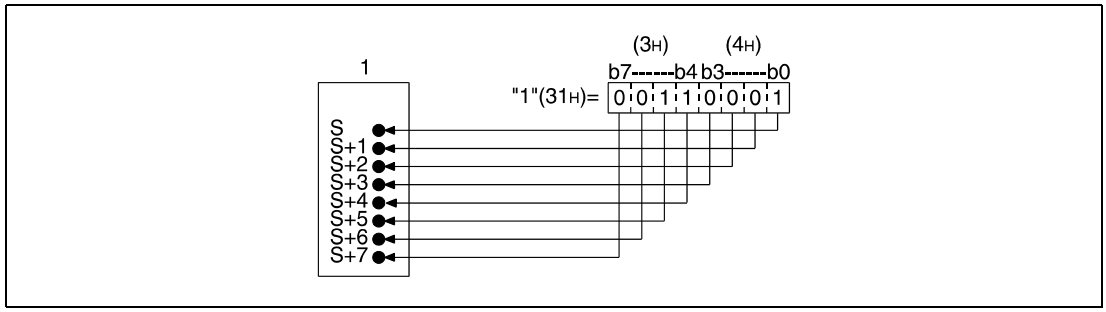
In der folgenden Abbildung wird n beispielsweise mit 5 angegeben und die Werte 1 (31H) bis 5 (35H) an den Eingängen X10 bis X18 des Eingangsmoduls eingegeben.



[1] Eingangsmodul

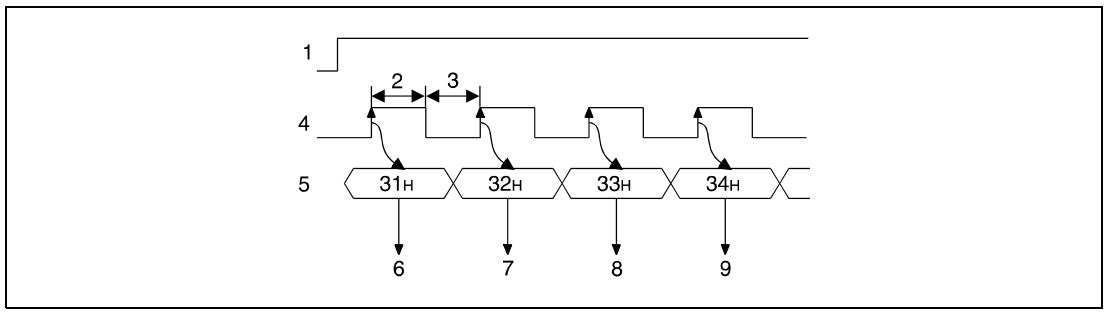
[2] Strobe-Signal

Bei der Eingabe der ASCII-Zeichen an den in s+0 (Array_s[1]) bis s+7 (Array_s[8]) angegebenen Eingängen (X) werden die Zeichen mit 8 Bits gemäß folgender Abbildung binärcodiert.



1 Eingangsmodul

Nach der Eingabe eines ASCII-Zeichens an s+0 (Array_s[1]) bis s+7 (Array_s[8]) wird das Strobe-Signal (s+8, Array_s[9]) gesetzt, um die Eingabedaten intern zu verbinden. Die Zeit, die das Strobe-Signal gesetzt oder zurückgesetzt ist, sollte länger als ein Programmzyklus sein. Ist diese Zeit kürzer als ein Programmzyklus, ist ein einwandfreies Verbinden der Eingabedaten nicht gewährleistet.



1 Ausführungsbedingung für die KEY-Anweisung

2 Für mehr als einen Zyklus eingeschaltet

3 Für mehr als einen Zyklus ausgeschaltet

4 Strobe-Signal (s+8, Array_s[9])

5 ASCII-Eingabedaten (s+0 bis s+7, Array_s[1] bis Array_s[8])

6 "1" einlesen

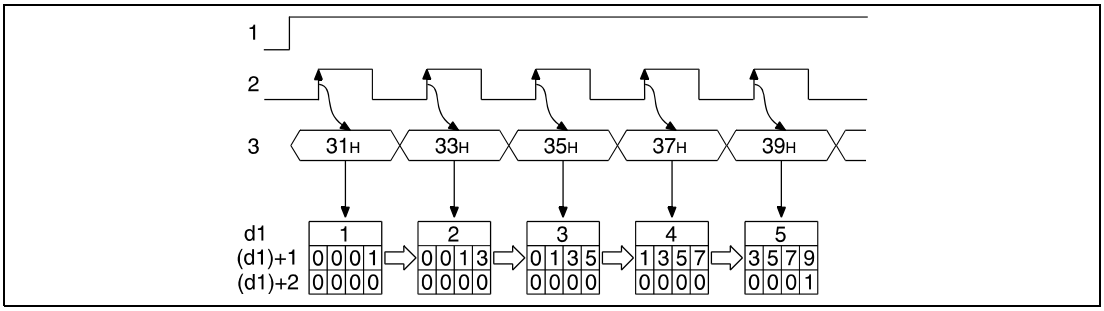
7 "2" einlesen

8 "3" einlesen

9 "4" einlesen

Die KEY-Anweisung kann nur bei gesetzter Ausführungsbedingung ausgeführt werden. Die Ausführungsbedingung muss so lange gesetzt bleiben, bis die Eingabe der in n angegebenen Anzahl von Zeichen abgeschlossen ist.

Die Anzahl der eingegebenen Werte wird in (d1)+0 (Array_d[1]) gespeichert. Die eigentliche Speicherung der eingegebenen ASCII-Zeichen in den in (d1)+1 (Array_d[2]) und (d1)+2 (Array_d[3]) angegebenen Operanden erfolgt in Form von hexadezimalen Binärwerten, d.h. es stehen pro Zeichen 4 Bits zur Verfügung. Die hexadezimalen Binärwerte der Zeichen 0H bis FH lauten von "0000" bis "1111".



[1] Ausführungsbedingung für die KEY-Anweisung

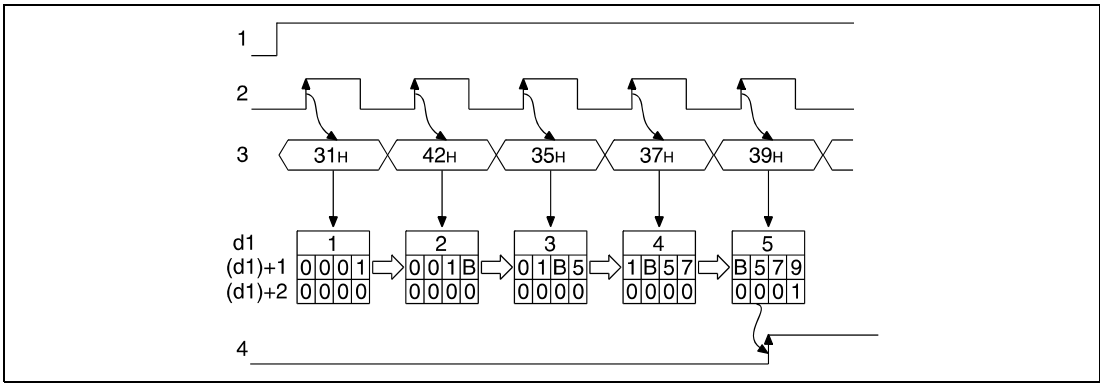
[2] Strobe-Signal (s+8, Array_s[9])

[3] ASCII-Eingabedaten (s+0 bis s+7, Array_s[1] bis Array_s[8])

Die in n angegebene Anzahl von einzugebenden Zeichen muss zwischen 1 und 8 liegen.

Das interne Verbinden der Eingabedaten ist abgeschlossen und der in d2 angegebene Operand wird gesetzt, wenn die in n angegebene Anzahl von Zeichen eingegeben ist oder der Zeichencode "00H" eingegeben wird. Die folgenden Abbildungen verdeutlichen diesen Sachverhalt. Für n wird in diesem Fall 5 angegeben.

In der folgenden Abbildung ist die Eingabe nach 5 Zeichen beendet. In der übernächsten Abbildung ist die Eingabe nach dem Zeichencode "00H" beendet.

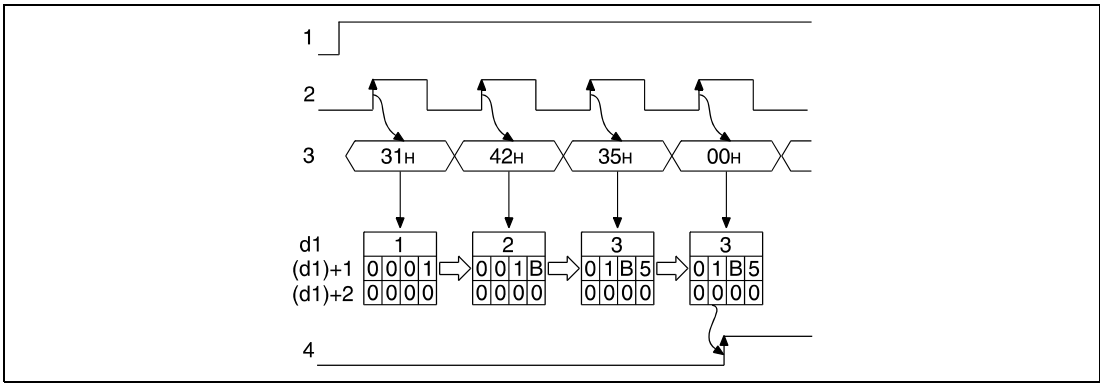


[1] Ausführungsbedingung für die KEY-Anweisung

[2] Strobe-Signal (s+8, Array_s[9])

[3] ASCII-Eingabedaten (s+0 bis s+7, Array_s[1] bis Array_s[8])

[4] Zeicheneingabe abgeschlossen (der in d2 angegebene Operand wird gesetzt)



[1] Ausführungsbedingung für die KEY-Anweisung

[2] Strobe-Signal (s+8, Array_s[9])

[3] ASCII-Eingabedaten (s+0 bis s+7, Array_s[1] bis Array_s[8])

[4] Zeicheneingabe abgeschlossen (der in d2 angegebene Operand wird gesetzt)

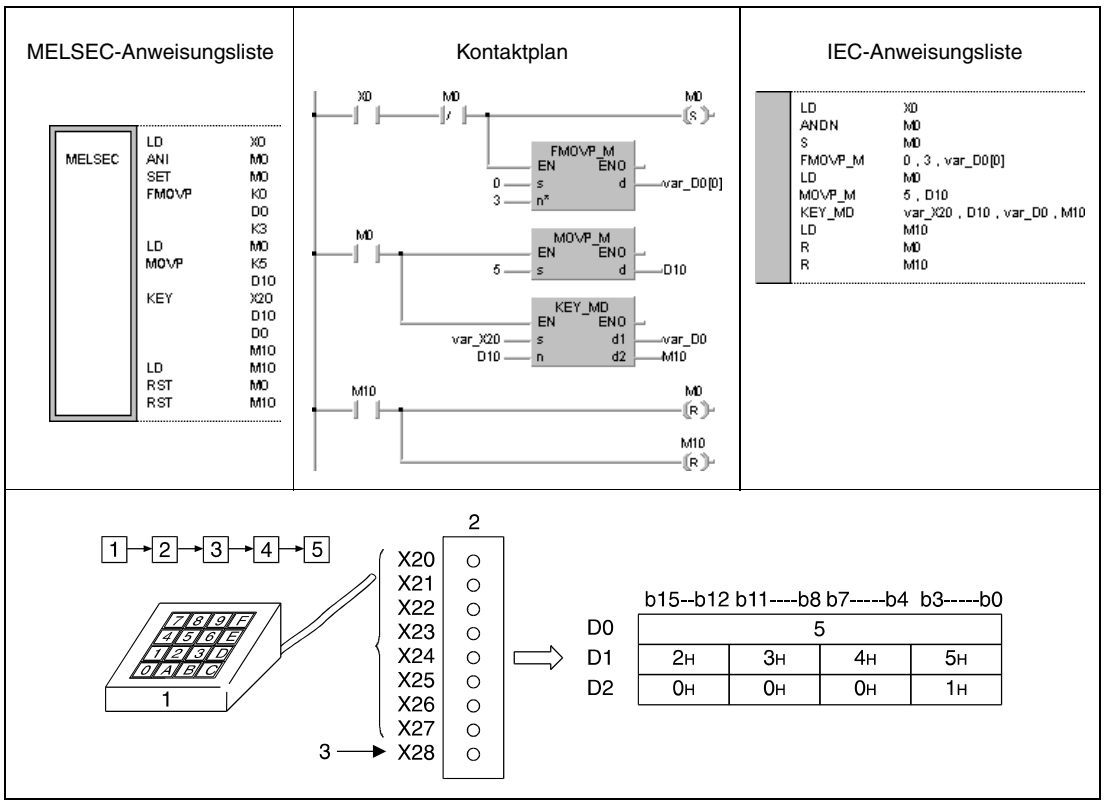
Vor einer erneuten Eingabe von Zeichen muss der Inhalt der in (d1)+0 (Array_d1[1]) bis (d1)+2 (Array_d[3]) angegebenen Operanden gelöscht werden und der in d2 angegebene Operand zurückgesetzt werden. Werden diese Operanden nicht gelöscht bzw. zurückgesetzt, ist eine erneute Eingabe von Zeichen nicht möglich.

Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Der in s angegebene Operand ist kein Eingang (X) (Fehlercode 4100).
- Die in n angegebene Anzahl von einzugebenden Zeichen liegt nicht zwischen 1 und 8.

Beispiel Im folgenden Programm können über die Eingänge X20 (var_X20[0]) bis X27 (var_X20[7]) mittels Tastatur bis zu 5 numerische Werte eingegeben werden. Diese Werte werden hexadezimal binärcodiert in den Registern D1 (var_D0[1]) und D2 (var_D0[2]) gespeichert. Die Anzahl der bereits eingegebenen Werte wird in D0 (var_D0[0]) gespeichert. Vor der Ausführung der KEY-Anweisung werden die Register D0 (var_D0[0]) bis D2 (var_D0[2]) gelöscht und für die Anzahl der Eingabedaten der Wert 5 eingetragen. Nach der Ausführung der KEY-Anweisung wird der Merker M10 (Abschluss der Eingabe) zurückgesetzt. Das Strobe-Signal liegt am Eingang X28 (var_X20[8]) an.



[1] Numerische Tastatur

[2] Eingangsmodul

[3] Strobe-Signal

7.18.8 ZPUSH, ZPUSHP, ZPOP, ZPOPP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

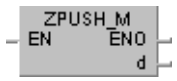
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
d	—	●	●	—	—	—	—	—	—	SM0	2

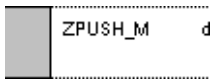
GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

Operand	Befehlswert	Datentyp
d	Erste Adresse des Operanden, in dem die Index-Registerinhalte gesichert werden.	BIN-16-Bit

Funktions-weise

Sichern und Wiederherstellen von Index-Registerinhalten

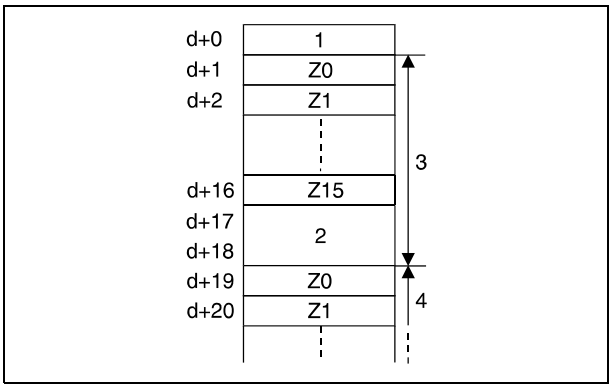
ZPUSH Sichern von Index-Registerinhalten

Die ZPUSH-Anweisung sichert die Inhalte der Index-Register Z0 bis Z15 in den ab d angegebenen Operanden.

Diese Daten können mit der ZPOP-Anweisung wiederhergestellt werden. Die Anweisungen können auf unterschiedliche Ebenen angewendet werden (Nesting), die sich innerhalb einer ZPUSH-/ZPOP-Schleife befinden.

Bei der Anwendung der Anweisungen auf unterschiedlichen Ebenen (Nesting) wird für jede Ausführung der ZPUSH-Anweisung ein Raum von 18 Registern mit 16 Bits in den ab d angegebenen Operanden benötigt. Aus diesem Grund ist ab d ein den Ausführungen der ZPUSH-Anweisung äquivalenter Speicherbereich vorzusehen.

Die folgende Abbildung gibt die Aufteilung des Speicherbereichs ab d wieder.



[1]Anzahl der vorgenommenen Sicherungen
[2]Zwei Datenwörter (systemintern)
[3]Erste Nesting-Ebene (max. 18 Datenwörter)
[4]Zweite Nesting-Ebene

ZPOP Wiederherstellen von Index-Registerinhalten

Die ZPOP-Anweisung stellt die mittels ZPUSH-Anweisung gesicherten Index-Registerinhalte wieder her. Hierbei werden die Daten aus dem ab d angegebenen Speicherbereich ausgelesen und wieder in die entsprechenden Index-Register geschrieben.

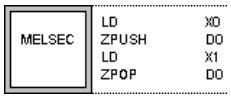
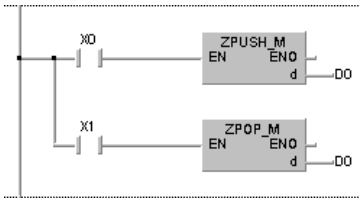
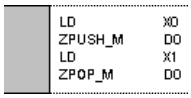
Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Der Speicherbereich der ab d angegebenen Operanden liest außerhalb des für die Speicherung vorgesehenen Bereichs (Fehlercode 4101).
- Der Inhalt des in d+0 angegebenen Operanden (Anzahl der vorgenommenen Sicherungen) ist 0 (Fehlercode 4100).

Beispiel ZPUSH/ZPOP

Im folgenden Programm werden mit X0 die Index-Register gesichert und mit X1 wieder gelesen.

MELSEC-Anweisungsliste	Kontaktplan	IEC-Anweisungsliste
		

7.18.9 EROMWR, EROMWRP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	

Operanden MELSEC Q

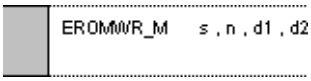
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort				U		
s	—	●	—	—	—	—	—	—	—	SM0	5
d1	—	—	●	—	—	—	—	—	—		
n	●	●	●	●	●	●	●	●	—		
d2	●	—	—	—	—	—	—	—	—		

GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan

IEC-Anweisungsliste



Variablen

Operand	Befehlswert	Datentyp
s	Erste Adresse des Operanden, in dem die zu schreibenden Daten gespeichert sind.	BIN-16-Bit
d1	Erste Adresse des EEPROM-File-Registers, in das geschrieben wird.	
n	Anzahl der zu schreibenden Datenwörter.	
d2	Operand, der nach dem Beenden der Schreiboperation gesetzt wird.	Bit

Funktionsweise

Schreiben von Daten in ein EEPROM-File-Register

EROMWR/EROMWRP Schreibanweisung

Die EROMWR-Anweisung schreibt die mit n angegebene Anzahl der Datenwörter in s in das in d1 angegebene EEPROM-File-Register.

Nach dem Beenden der Schreiboperation wird der in d2 angegebene Operand gesetzt und nach einem Zyklus automatisch wieder zurückgesetzt.

Die EROMWR-Anweisung wird bis zum Erreichen der END-Anweisung ausgeführt. Während dieses Zeitraums können in jedem Verarbeitungszyklus 64 Datenwörter geschrieben werden. Die Anzahl der Verabeitungszyklen ergibt sich aus dem aufgerundeten Quotienten der Division der in n angegebenen Anzahl von Datenwörtern durch den Wert 64. Für die Bestimmung der Verarbeitungszeit ist eine Zykluszeit von ca. 10 ms bei der Verarbeitung zu Grunde zu legen.

Die in s angegebenen Daten dürfen während des Schreibvorgangs nicht aktualisiert werden. In diesem Fall können einige Daten verloren gehen.

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Der Speicherbereich der in n angegebenen Anzahl von Datenwörtern liegt außerhalb des für die Speicherung vorgesehenen Bereichs der in s und d1 angegebenen Operanden (Fehlercode 4101).
- Das in d1 angegebene File-Register existiert nicht oder ist kein EEPROM-File-Register (Fehlercode 4101).

8 Data-Link-Anweisungen

8.1 Grundlagen

Eine QnA(S) CPU kann in den Netzwerksystemen MELSECNET(II)/B/10 verwendet werden. Das System Q unterstützt MELSECNET/10 und MELSECNET/H.

HINWEISE *Die hier verwendete Bezeichnungen MELSECNET/10 und MELSECNET/H gelten für die Netzwerke MELSECNET/10 und MELSECNET/H.*

Die hier verwendete Bezeichnung MELSECNET gilt für die Netzwerksysteme MELSECNET(I), MELSECNET(II) und MELSECNET/B.

Mit Hilfe der Data-Link-Anweisungen liest die CPU Daten aus anderen an das MELSECNET, MELSECNET/10 oder MELSECNET/H angeschlossenen Stationen in die Host-Station.

8.2 Anweisungstypen

Die Data-Link-Anweisungen werden in folgende vier Gruppen eingeteilt:

1. Datenaktualisierungs-Anweisungen (Refresh-Anweisungen)

   Mit diesen Anweisungen werden Daten in den angesprochenen Netzwerkmodulen aktualisiert.

2. Erweiterte Data-Link-Anweisungen

   Bei diesen Anweisungen handelt sich um neue Data-Link-Anweisungen, die in Verbindung mit einer QnA CPU oder einer CPU aus dem System Q verwendet werden. Für die Kommunikation können mehrere Kanäle des Netzwerkmoduls verwendet werden.

3. Zur A-Serie kompatible Data-Link-Anweisungen

   Diese Anweisungen sind mit den erweiterten Data-Link-Anweisungen der ACPUs identisch.

4. Lesen und Schreiben von Routing-Informationen

   Mit diesen Anweisungen ist das Lesen und Schreiben von Routing-Parametern in und aus Relais- und Routing-Stationen möglich.

Für die Systeme MELSECNET und MELSECNET/10 können nur bestimmte Data-Link-Anweisungen verwendet werden. Welche Anweisungen im MELSECNET/10 verwendet werden können, hängt ferner davon ab, ob die CPU in der Ziel-Station eine CPU der A- oder QnA-Serie, ein System Q oder eine ausgelagerte Ein-/Ausgabe-Station ist.

Die folgende Tabelle liefert eine Übersicht über die Data-Link-Anweisungen.

Einteilung	Bedeutung
Netzwerk-Datenaktualisierungs-Anweisungen (Refresh-Anweisungen)	Anweisungen für Datenaktualisierungen in Netzwerkmodulen.
Erweiterte Data-Link-Anweisungen	Lesen und Schreiben von CPU- Daten in und aus Ziel-Stationen in Ziel-Netzwerken, Senden von Daten an Netzwerkmodule in Ziel-Stationen in Ziel-Netzwerken, Lesen mittels SEND-Anweisung gesendeter CPU-Daten, Datenanforderung an andere Stationen (Schreib-/Lese-Operationen mit Uhrdaten, RUN-/STOP-Operation), Lesen und Schreiben von Daten in und aus Sondermodulen in ausgelagerten Ein-/Ausgabe-Stationen.
Zur A-Serie kompatible Data-Link-Anweisungen	Lesen und Schreiben von CPU-Daten in und aus Ziel-Stationen in Ziel-Netzwerken, Lesen und Schreiben von Daten in und aus lokalen Stationen (nur an Master-Stationen), Lesen und Schreiben von Daten in und aus Sondermodulen in ausgelagerten Ein-/Ausgabe-Stationen.
Lesen und Schreiben von Routing-Informationen	Lesen und Schreiben der Routing-Parameter (Netzwerks- und Stationsnummer der Relais-Station, Stationsnummer der Routing-Station).

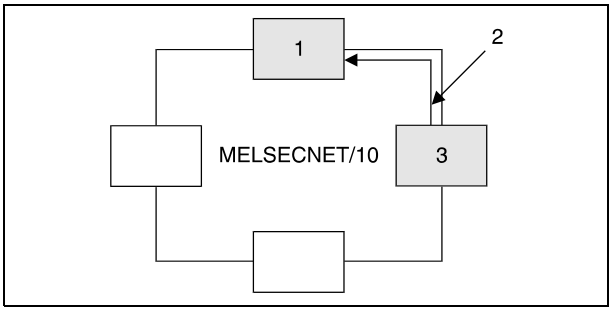
8.3 Schreib- und Lesebereiche der Daten

8.3.1 MELSECNET/10

Im MELSECNET/10 ist eine HOST-Station in der Lage, Schreib- und Lese-Operationen mit Stationen im eigenenen Netzwerk und bei entsprechender Adressierung (Routing-Parameter) auch mit Stationen anderer Netzwerke durchzuführen.

Schreib-/ Leseoperationen mit Stationen im eigenen Netzwerk

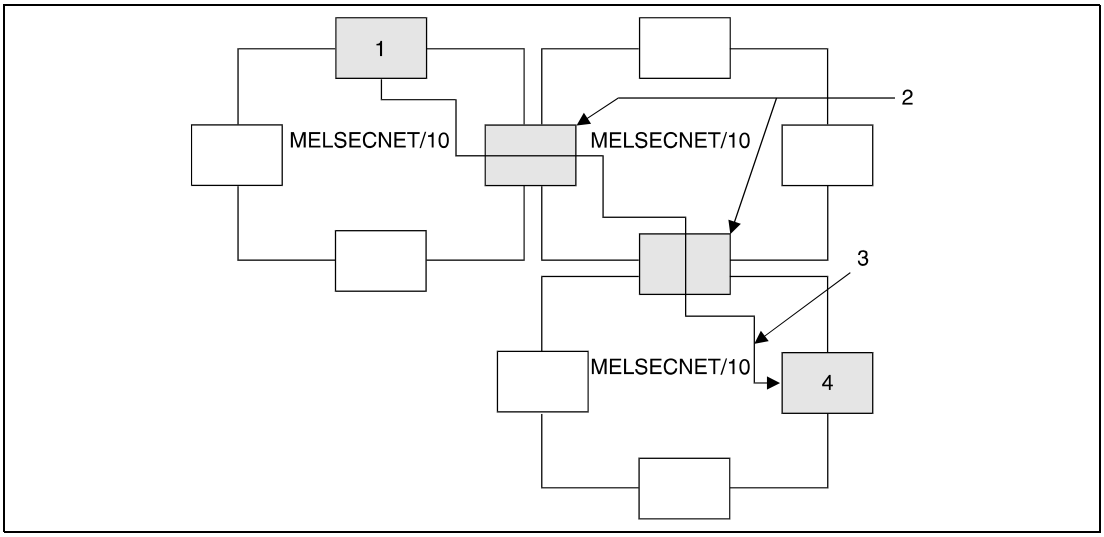
Um Schreib-/Lese-Operationen mit Stationen im eigenen Netzwerk durchzuführen muss die Netzwerknummer der Ziel-Station mit der Nummer des Netzwerkmoduls der HOST-Station identisch sein. Diese Funktion wird verwendet, um Daten in und aus jeder Station zu lesen und zu schreiben, die sich in dem Ziel-Netzwerk befindet.



[1] Station, die die Anweisung ausführt
[2] Lese-Operation
[3] Ziel-Station

Schreib-/ Leseoperationen mit Stationen im anderen Netzwerk

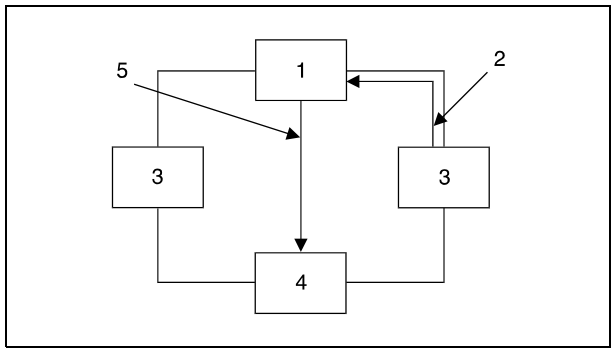
Um Schreib-/Lese-Operationen mit Stationen in anderen Netzwerken durchzuführen, muss die Netzwerknummer der angesprochenen Station eine andere als die des Netzwerkmoduls der HOST-Station sein. Eine Station im Netzwerk der HOST-Station übernimmt die Funktion einer Relais-Station, die die Schreib-/Lese-Operationen an die angesprochene Station im anderen Netzwerk weiterleitet.



[1] Station, die die Anweisung ausführt
[2] Relais-Stationen (die Einstellung von Routing-Parametern ist erforderlich)
[3] Lese-Operation
[4] Ziel-Station

8.3.2 MELSECNET

Im MELSECNET(I/II/B) kann eine Master-Station Schreib-/Lese-Operationen mit lokalen Stationen und ausgelagerten Ein-/Ausgabe-Stationen durchführen.



[1] Master-Station
[2] Schreib-/Lese-Operation
[3] Lokale Station
[4] Ausgelagerte Ein-/Ausgabe-Station
[5] Schreib-/Lese-Operation mit Sondermodulen

8.4 Erweiterte Data-Link-Anweisungen

Im folgenden werden einige Punkte für die Verwendung der erweiterten Data-Link-Anweisung für QnA CPUs und CPUs des System Q beschrieben.

8.4.1 Gleichzeitige Ausführung

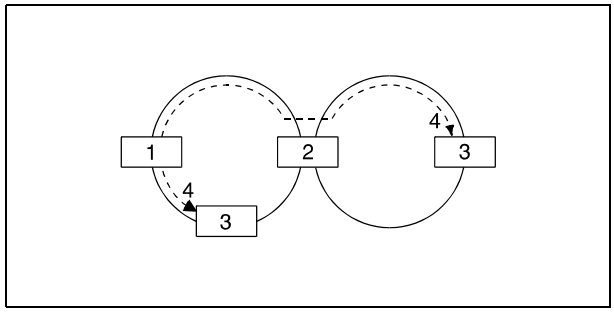
Netzwerkmodule für das System MELSECNET/10 verfügen über 8 Kommunikationsbereiche, die von den Data-Link-Anweisungen genutzt werden können. Bei der Verwendung dieser Netzwerkmodule ist die gleichzeitige Ausführung mehrerer Data-Link-Anweisungen in einem Kommunikationsbereich nicht möglich. Sollen in einem Kommunikationsbereich der CPU mehrere Data-Link-Anweisungen ausgeführt werden, muss durch eine Verriegelung mit den Operanden, die nach vollständiger Ausführung der jeweiligen Schreib-/Leseanweisung gesetzt werden, ein aufeinanderfolgendes Verarbeiten der Anweisungen sichergestellt werden.

8.4.2 Datenübertragungsende

Bei Verwendung der erweiterten Data-Link-Anweisungen kann angegebenen werden, ob das Verarbeitungs- bzw. Datenübertragungsende bestätigt wird oder nicht.

Bestätigung des Datenübertragungsendes

Die folgende Abbildung zeigt den Modus, in dem das Verarbeitungs- bzw. Datenübertragungsende bestätigt wird, wenn die Daten in den adressierten Kanal der Ziel-Station geschrieben wurden (bei Lese-Operationen kann nur dieser Modus gewählt werden).



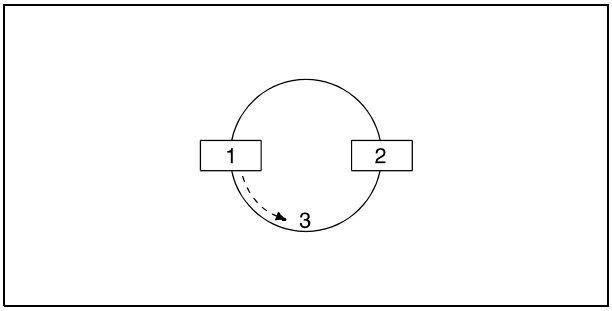
[1] Ausführungsquelle
[2] Relais-Station
[3] Ziel-Station
[4] Verarbeitungs-/Übertragungsende

Keine Bestätigung des Datenübertragungsendes

Die folgenden Abbildungen zeigen den Modus, in dem das Verarbeitungs- bzw. Übertragungsende nicht bestätigt wird.

Innerhalb eines Netzwerks:

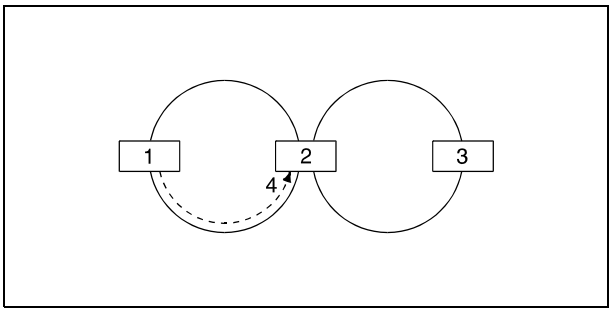
Das Verarbeitungs- bzw. Übertragungsende ist dann erreicht, wenn die HOST-Station alle Daten gesendet hat.



[1] Ausführungsquelle
[2] Ziel-Station
[3] Verarbeitungs-/Übertragungsende

Zwischen verschiedenen Netzwerken:

Das Verarbeitungs- bzw. Übertragungsende ist dann erreicht, wenn die gesendeten Daten die Relais-Station im Netzwerk des HOST erreicht haben.



[1] Ausführungsquelle
[2] Relais-Station
[3] Ziel-Station
[4] Verarbeitungs-/Übertragungsende

HINWEISE *Es wird empfohlen, den Modus mit Bestätigung des Datenübertragungsendes anzugeben und auszuführen, um eine korrekte Verarbeitung der Daten zu gewährleisten.*

Wird der Modus ohne Bestätigung des Datenübertragungsendes angegeben, wird der sendenden Station nach dem Senden der Daten das Übertragungsende gemeldet, unabhängig davon, ob während der Übertragung Fehler aufgetreten sind. Ferner wird von der Ziel-Station der Fehler "Eingangspufferspeicher voll" gemeldet, wenn die Data-Link-Anweisungen von mehreren Stationen ausgeführt werden. Das gilt auch in den Fällen, in denen die Daten korrekt übertragen worden sind. Trotzdem schließt die sendende Station in diesem Fall die Datenübertragung ab.

8.5 Datenaktualisierungs-Anweisungen

Mit den folgenden Datenaktualisierungs-Anweisungen ist das Aktualisieren (Refreshen) von Daten in Netzwerkmodulen durchzuführen. Die folgende Tabelle liefert eine Übersicht der Anweisungen.

Funktion	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor	Im MELSECNET/10 angesprochene Station			MELSECNET
			QnA-CPU System Q	ACPU	Ausgelagerte Ein-/Ausgabe-Station	
Datenaktualisierungs-Anweisungen (Refresh-Anweisungen)	ZCOM	ZCOM_J_M	—	—	—	—
		ZCOM_JP_M				
		ZCOM_U_M				
		ZCOM_UP_M				

8.5.1 ZCOM

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

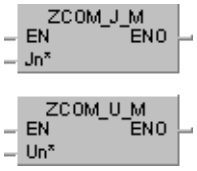
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
—	—	—	—	—	—	—	—	—	—	SM0	5

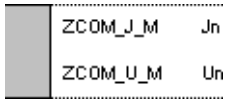
GX IEC Developer

MELSEC-Anweisungsliste

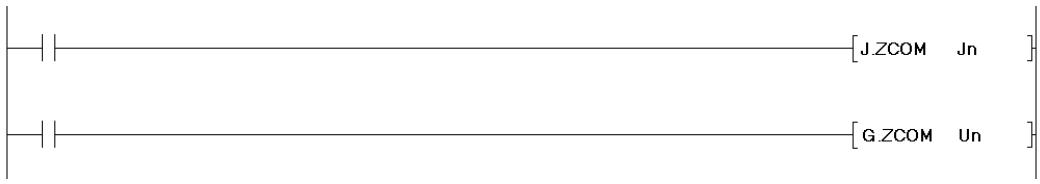
Kontaktplan



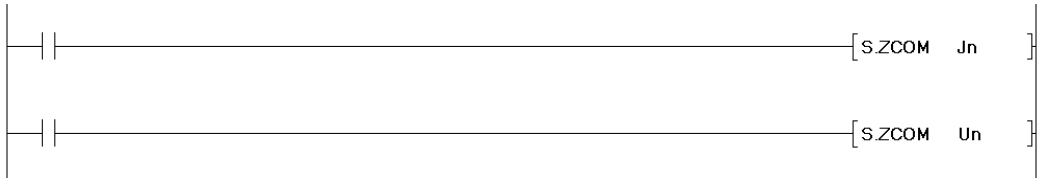
IEC-Anweisungsliste



GX Developer (QnA-CPU)



GX Developer (System Q)



Variablen

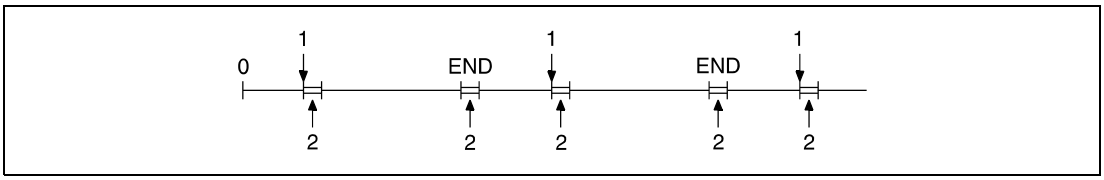
Operand	Befehlswert	Datentyp
Jn	Netzwerknummer der HOST-Station.	BIN-16-Bit
Un	Kopf-E/A-Adresse des Netzwerks der HOST-Station.	

Funktionsweise

Netzwerk-Datenaktualisierung

ZCOM Datenaktualisierung in Netzwerkmodulen

Wenn die ZCOM-Anweisung ausgeführt wird, unterbricht die CPU die Ausführung des Ablaufprogrogramms und führt eine Datenaktualisierung (Refresh) in den mit Jn und Un angegebenen Netzwerkmodulen durch.

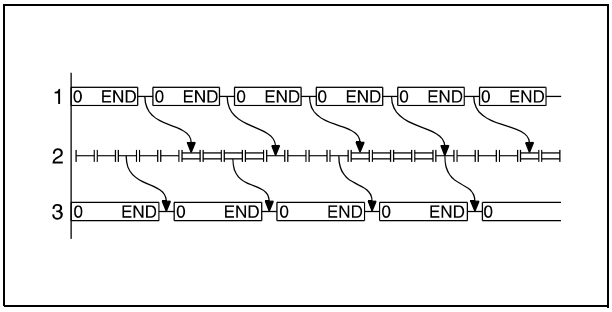


[1] Ausführung der ZCOM-Anweisung

[2] Datenaktualisierung

In Fällen, in denen die Zykluszeit des Ablaufprogramms der HOST-Station länger als die Zykluszeit der anderen Stationen ist, stellt die Verwendung der ZCOM-Anweisung sicher, dass die Daten der anderen Stationen korrekt verbunden werden.

Die folgende Abbildung zeigt ein Beispiel der Datenkommunikation ohne Verwendung der ZCOM-Anweisung.

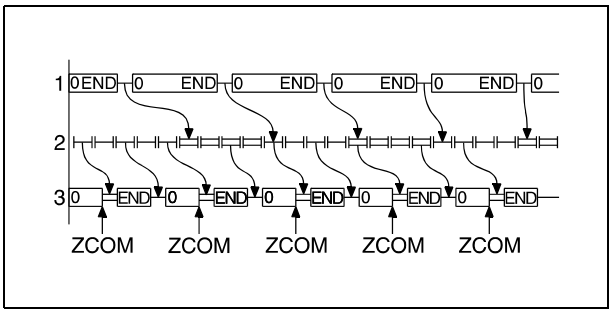


[1] Programm der Kontrollstation

[2] Zyklus der Ziel-Station

[3] Programm der normalen Station

Die folgende Abbildung zeigt eine Datenkommunikation mit Verwendung der ZCOM-Anweisung.

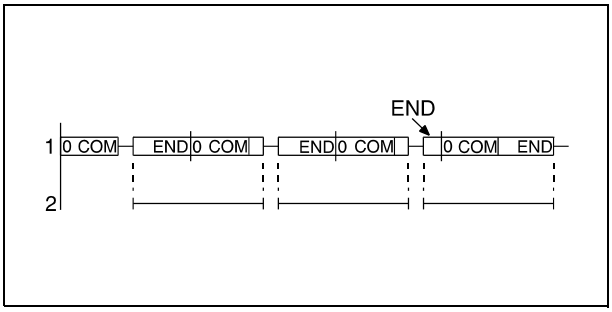


[1] Programm der Kontrollstation

[2] Zyklus der Ziel-Station

[3] Programm der normalen Station

In Fällen, in denen die Zykluszeit der Ziel-Station größer als die Zykluszeit des Ablaufprogramms ist, verbessert auch die Verwendung der ZCOM-Anweisung die Datenkommunikation nicht.



[1] Ablaufprogramm
[2] Zykluszeit der Ziel-Station

Die ZCOM-Anweisung kann in einem Ablaufprogramm beliebig oft angewendet werden. Es ist jedoch zu beachten, dass sich bei jeder Ausführung einer Aktualisierung die Zykluszeit des Ablaufprogramms um die Ausführungszeit der Datenaktualisierung verlängert.

Bei folgenden Operationen kann die ZCOM-Anweisung nicht angewendet werden:

- Kommunikation zwischen der CPU und den peripheren Baugruppen.
- Überwachung und Kontrolle der übrigen Stationen.
- Auslesen des Pufferspeichers der übrigen Sondermodule über ein Computer-Link-Modul.

HINWEIS *Durch Eintrag von "Un" können nicht nur Netzwerkmodule, sondern auch Sondermodule angesprochen werden. Da in diesem Fall der Pufferspeicher des Sondermoduls aktualisiert wird, können so die FROM-/TO-Anweisungen ersetzt werden.*

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

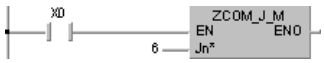
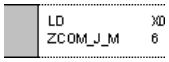
- Die angegebene Netzwerknummer ist nicht mit der HOST-Station verbunden (Fehlercode 4102).
- Das Modul an der angegebenen Ein-/Ausgangsadresse ist kein Netzwerk- oder Link-Modul (Fehlercode 2111).

HINWEISE *Um ausschließlich eine Gesamtdatenverarbeitung auszuführen, ist die COM-Anweisung zu verwenden.*

Es ist zu beachten, dass nicht-konsistente Daten auftreten können. D.h., ein Operand kann sich während eines Programmzyklus verändern.

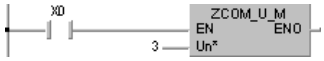
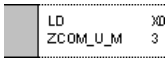
Beispiel 1 J.ZCOM

Das folgende Programm führt für die Einschaltdauer von X0 eine Datenaktualisierung (Refresh) in dem Netzwerkmodul mit der Netzwerknummer 6 durch.

MELSEC-Anweisungsliste	Kontaktplan	IEC-Anweisungsliste
		

Beispiel 2 G.ZCOM

Das folgende Programm führt für die Einschaltdauer von X0 eine Datenaktualisierung in dem Netzwerk-Modul durch, das sich an den Ein-/Ausgangsadressen X/Y30 bis X/Y4F befindet.

MELSEC-Anweisungsliste	Kontaktplan	IEC-Anweisungsliste
		

8.6 Erweiterte Data-Link-Anweisungen der QnA-Serie

Mit diesen Anweisungen ist die Datenkommunikation zwischen Stationen mit QnA-CPUs sowie zwischen QnA-CPUs und ausgelagerten Ein-/Ausgabe-Stationen im MELSECNET/10 möglich. Die folgende Tabelle liefert eine Übersicht der Anweisungen.

Funktion	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor	Im MELSECNET/10 angesprochene Station			MELSECNET
			QnA-CPU	ACPU	Ausgelagerte Ein-/Ausgabe-Station	
Lesen von QnA-CPU-Daten aus Ziel-Stationen in Ziel-Netzwerken	READ	READ_M	●	—	—	—
		READP_M				
		READ_JP_M				
		READ_UP_M				
	SREAD	SREAD_JP_M	●	—	—	—
		SREAD_UP_M				
Schreiben von QnA-CPU-Daten in Ziel-Stationen in Ziel-Netzwerken	WRITE	WRITE_JP_M	●	—	—	—
		WRITE_UP_M				
	SWRITE	SWRITE_M	●	—	—	—
		SWRITE_JP_M				
		SWRITE_UP_M				
Senden von Daten an Netzwerkmodule in Ziel-Stationen in Ziel-Netzwerken	SEND	SEND_M	●	—	—	—
		SEND_4_M				
		SEND_4_P_M				
		SEND_JP_M				
		SEND_UP_M				
Lesen mittels SEND-Anweisung gesendeter QnA-CPU-Daten	RECV	RECV_M	●	—	—	—
		RECVP_M				
		RECV_JP_M				
		RECV_UP_M				
Datenanforderung anderer Stationen (Schreib-/Lese-Operationen mit Uhr-Daten, RUN-/STOP-Abfrage)	REQ	REQ_M	●	—	—	—
		REQP_M				
		REQ_JP_M				
		REQ_UP_M				
Lesen von Daten aus Sonder-Modulen in ausgelagerten Ein-/Ausgabe-Stationen.	ZNFR	ZNFR_JP_M	—	—	●	—
		ZNFR_UP_M				
Schreiben von Daten in Sondermodule in ausgelagerten Ein-/Ausgabe-Stationen.	ZNTO	ZNTO_J_M	—	—	●	—
		ZNTO_U_M				
		ZNTO_JP_M				
		ZNTO_UP_M				

8.6.1 READ

CPU

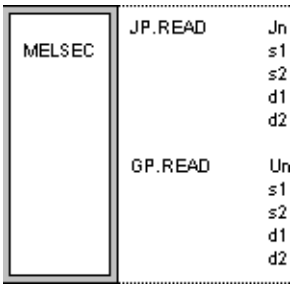
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	

Operanden MELSEC Q

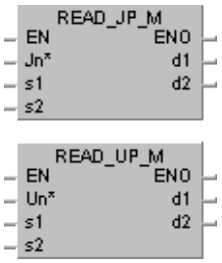
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	—	●	●	—	—	—	—	●	—	SM0	9
s2	—	●	●	—	—	—	—	●	—		
d1	—	●	●	—	—	—	—	—	—		
d2	●	●	●	—	—	—	—	—	—		

GX IEC Developer

MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste

```
READ_JP_M   Jn , s1 , s2 , d1 , d2
READ_UP_M   Un , s1 , s2 , d1 , d2
```

Variablen

Operand	Befehlswert	Datentyp	
		MELSEC	IEC
Jn	Netzwerknummer der HOST-Station. ●[1]	BIN-16-Bit	ANY16
Un	Kopf-Ein-/Ausgangsadresse des Netzwerkmoduls der HOST-Station. ●[2]		
s1	Erster Operand der HOST-Station, in dem die Kontrolldaten gespeichert sind.	Adresse	Array [1..18] of ANY16
s2	Erster Operand der Station, in dem die zu lesenden Daten gespeichert sind.		ANY16
d1	Erster Operand der HOST-Station, in dem die gelesenen Daten gespeichert werden.		
d2	Operand, der nach vollständiger Verarbeitung der Anweisung für einen Programmzyklus gesetzt wird.	Bit	BOOL

HINWEISE

- *[1] Die Netzwerknummer der HOST-Station muss zwischen 1 und 239 liegen. Das Netzwerk mit der Nummer 254 wird über Einstellungen so konfiguriert, dass andere Stationen auf die aktive Station zugreifen können.*
- *[2] Die Kopf-Ein-/Ausgangsadresse des Netzwerkmoduls der HOST-Station muss zwischen 0 und FE$_H$ liegen. Es ist zu beachten, dass der Compiler für Un eine hexadezimale Adressangabe erwartet. Eine Dezimalangabe wird automatisch in eine Hexadezimalzahl umgewandelt.*

Die READ-Anweisung kann nur ausgeführt werden, wenn sich in der Ziel-Station eine QnA-CPU befindet.

Die READ-Anweisung kann bei der Verwendung einer ACPU im MELSECNET/10 nicht angewendet werden.

Es können nur Stationsnummern für QnA-CPUs als Nummer der Ziel-Station angegeben werden.

Operandenübersicht der Kontrolldaten

Operand	Bedeutung	Funktion	Werte-bereich	Festlegung durch
(s1)+0 Array_s1[1]	Ausführungs-Modus	Die Eingangsbestätigung ist gesetzt: (Bit 0 (b0) = 1, fest eingestellt)	0001H 0081H	Benutzer
	Abschluss-Modus der Fehlerverar-beitung	Speichern der Uhr-Daten des Zeitpunktes, an dem die Fehlerverabeitung abgeschlossen ist: - Keine Speicherung der Uhr-Daten, Bit 7 (b7) = 0 - Speicherung der Uhr-Daten, Bit 7 (b7) = 1 (Uhr-Daten ab (s1)+11 (Array_s1[12]) aufwärts)		
(s1)+1 Array_s1[2]	Abschluss-Status der Anweisungs-verarbeitung	Statusanzeige nach Abschluss der Anweisungs-verarbeitung: 0 = Keine Fehler (normaler Verarbeitungsabschluss) ungleich 0 = Fehlercode ●3	—	System
(s1)+2 Array_s1[3]	Von der HOST-Station verwendeter Kanal	Angabe des Kanals, den die HOST-Station verwenden wird.	1 bis 8	Benutzer
(s1)+3 Array_s1[4]	Dummy	Wird nicht verwendet.	0	—
(s1)+4 Array_s1[5]	Netzwerknummer der Ziel-Station	Angabe der Netzwerknummer der Station, aus der gelesen wird.	1 bis 239 254 ●4	Benutzer
(s1)+5 Array_s1[6]	Nummer der Ziel-Station	Angabe der Stationsnummer der Ziel-Station.	1 bis 64	Benutzer
(s1)+6 Array_s1[7]	Dummy	Wird nicht verwendet.	—	—
(s1)+7 Array_s1[8]	Anzahl der Über-tragungsversuche	Angabe der Anzahl der Versuche, die unternom-men werden, um die vollständige Verarbeitung der READ-Anweisung innerhalb der in (s1)+8 (Array_s1[9]) gespeicherten Überwachungszeit durchzuführen.	1 bis15	Benutzer
	Anzahl der durch-geführten Übertra-gungsversuche	Speicherung der durchgeführten Übertragungs-versuche.		System
(s1)+8 Array_s1[9]	Zeiteinstellung des WDT zur Empfangs-zeitüberwachung	Angabe der Überwachungszeit für READ-Lese-Operationen in Sekunden. Erfolgt der Abschluss der Operation nicht innerhalb der angegebenen Zeit, wird die Übertragung mir der in (s1)+7 (Array_s1[8]) angegebenen Anzahl wiederholt.	1 bis 32767 0 = 10 s (fest eingestellt)	Benutzer
(s1)+9 Array_s1[10]	Empfangsdaten-länge	Angabe der Anzahl der auszulesenden Datenblöcke.	1 bis 480	Benutzer
(s1)+10 Array_s1[11]	Dummy	Wird nicht verwendet.	—	—
(s1)+11 Array_s1[12]	Flag der Uhr-Daten-speicherung (wird nur im Fehlerfall gesetzt)	Dieses Flag speichert den Status der in (s1)+0 (Array_s1[1]) angegebenen Speicheroption der Uhr-Daten: - Keine Speicherung der Uhr-Daten = 0 - Speicherung der Uhr-Daten = 1	—	System
(s1)+12 Array_s1[13]	Uhr-Daten (werden nur im Fehlerfall gesetzt)	Höchstwertiges Byte = Jahr (0 bis 99) Niedrigstwertiges Byte = Monat (1 bis 12)	—	System
(s1)+13 Array_s1[14]		Höchstwertiges Byte = Tag (1 bis 31) Niedrigstwertiges Byte = Stunde (0 bis 23)		
(s1)+14 Array_s1[15]		Höchstwertiges Byte = Minute (0 bis 59) Niedrigstwertiges Byte = Sekunde (0 bis 59)		
(s1)+15 Array_s1[16]		Höchstwertiges Byte = 00H Niedrigstwertiges Byte = Wochentag (0 bis 6) (Sonntag = 0, Samstag = 6)		
(s1)+16 Array_s1[17]	Nummer des Netz-werks, in dem der Fehler aufgetreten ist	Speicherung der Netzwerknummer der Station, in der der Fehler aufgetreten ist. Die Netzwerk-nummer liegt zwischen 1 und 239.	—	System

Operand	Bedeutung	Funktion	Werte-bereich	Festlegung durch
(s1)+17 Array_s1[18]	Nummer der Station, in der der Fehler aufgetreten ist	Speicherung der Nummer der Station, in der der Fehler aufgetreten ist. Die Speicherung wird nicht durchgeführt, wenn der Abschluss-Status der Anweisungsausführung "Kanal in Betrieb" lautet. Die Stationsnummer liegt zwischen 1 und 64.	—	System

●[3] Weitere Informationen über Fehlercodes enthält das Handbuch "MELSECNET/10 für QnA-Netzwerk-Systeme".

●[4] Die Netzwerknummer 254 ist ausgewählt, wenn für Jn der Wert 254 angegeben wird.

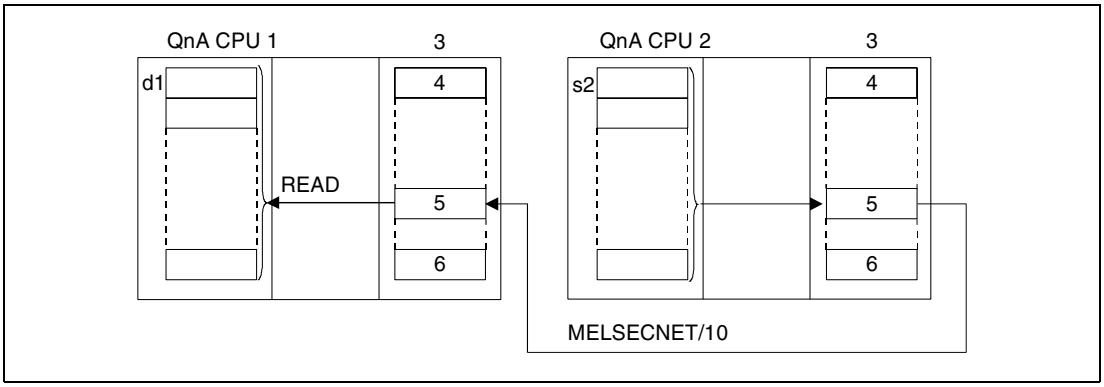
Funktionsweise

Lesen von Daten aus Wortoperanden anderer Stationen

READ Leseanweisung

Die READ-Anweisung liest die ab s2 gespeicherten Daten einer im MELSECNET/10 angeschlossenen Station. Die Stations- und Netzwerknummer sind in den Kontrolldaten angegeben. Die aus der Station gelesenen Daten werden ab d1 in der HOST-Station gespeichert.

Nach dem Abschluss der Lese-Operation wird der Operand d2 in der Ziel-Station gesetzt.



[1] HOST-Station
[2] Ziel-Station
[3] Netzwerkmodul
[4] Kanal 1
[5] Kanal n
[6] Kanal 8

Über eine Relais-Station und gesetzte Routing-Parameter kann auch auf Stationen in anderen Netzwerken zugegriffen werden.

Die Data-Link-Anweisungen können nicht von mehreren Stellen gleichzeitig mit gemeinsamem Zugriff auf die gleichen Kanäle ausgeführt werden. Bei gleichzeitiger Ausführung der Anweisungen von zwei oder mehr Stellen wird durch den Handshake-Betrieb der beiden aktiven Stationen eine Ausführung weiterer Data-Link-Anweisungen verhindert.

Der Ausführungs-Status und der Abschluss-Status (normal, nicht normal) der READ-Anweisung können mittels

- dem Kommunikations-Kanal-Flag (●[5]) des verwendeten Kanals,
- dem Operanden, der nach Abschluss der Lese-Operation gesetzt wird (d2) und
- der Status-Anzeige des Operationsabschlusses (Abschluss einer fehlerfreien oder fehlerhaften Übertragung) ((d2)+1)

wie folgt überprüft werden:

Kommunikations-Kanal-Flag

Dieses Flag wird während der Ausführung der READ-Anweisung gesetzt. Das Flag wird mit der Verarbeitung der END-Anweisung in dem Zyklus zurückgesetzt, in dem die Lese-Operation abgeschlossen wurde.

Operand der HOST-Station, der den Lese-Operationsabschluss anzeigt

Dieser Operand wird mit der Verarbeitung der END-Anweisung in dem Zyklus gesetzt, in dem die Lese-Operation abgeschlossen wurde. Bei der Verarbeitung der darauffolgenden END-Anweisung wird dieser Operand wieder zurückgesetzt.

Status-Anzeige des Lese-Operationsabschlusses

Dieser Operand wird in Abhängigkeit des Abschlussergebnisses der Lese-Operation gesetzt.

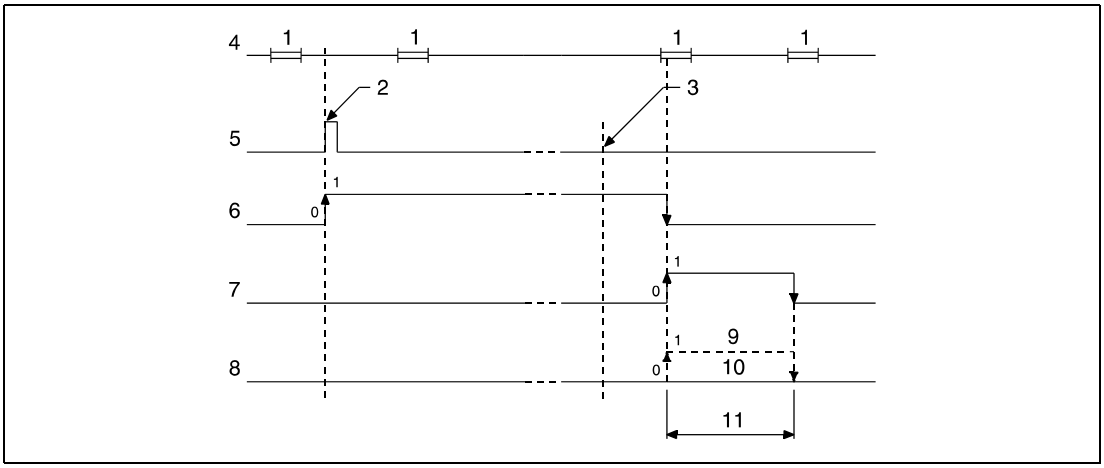
Beim Abschluss einer normalen (fehlerfreien) Übertragung bleibt der Operand unverändert zurückgesetzt.

Beim Abschluss einer nicht normalen (fehlerhaften) Übertragung wird dieser Operand mit der Verarbeitung der END-Anweisung in dem Zyklus gesetzt, in dem die Verarbeitung der READ-Anweisung abgeschlossen wurde. Bei der Verarbeitung der darauffolgenden END-Anweisung wird dieser Operand wieder zurückgesetzt.

HINWEIS ●[5] *Die folgende Tabelle ordnet den Kanalnummern die entsprechenden Kommunikations-Kanal-Flags zu.*

Kanalnummer	1	2	3	4	5	6	7	8
Kommunikations-Kanal-Flag	SB30	SB32	SB34	SB36	SB38	SB3A	SB3C	SB3E

Die folgende Abbildung zeigt die Operationen der HOST-Station während der Ausführung der READ-Anweisung.



[1] END-Verarbeitung
[2] Ausführung der READ-Anweisung
[3] Abschluss der Operation
[4] Programm der HOST-Station
[5] READ-Anweisung
[6] Kommunikations-Kanal-Flag
[7] Operand der HOST-Station, der nach Abschluss der Operation gesetzt wird (d2)
[8] Status-Anzeige des Operationsabschlusses ((d2)+1)
[9] Abschluss einer nicht normalen (fehlerhaften) Übertragung
[10] Abschluss einer normalen (fehlerfreien) Übertragung
[11] Ein Zyklus

Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die Kontrolldateninhalte liegen nicht im gültigen gültigen Wertebereich (Fehlercode 4100).
- Das Netzwerk mit der mit Jn angegebenen Nummer ist nicht mit der Station verbunden (Fehlercode 4102).
- Das Modul an der mit Un angegebenen Ein-/Ausgangsadresse ist kein Netzwerkmodul (Fehlercode 2111).

8.6.2 SREAD

CPU

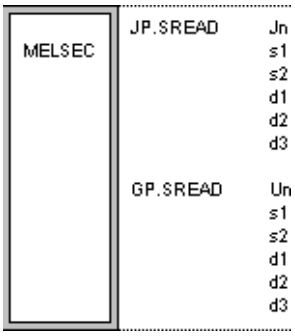
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	

Operanden MELSEC Q

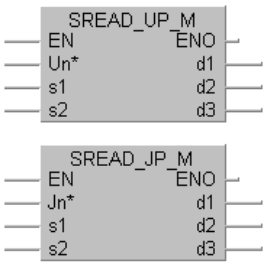
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□N□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	—	●	●	—	—	—	—	●	—	SM0	10
s2	—	●	●	—	—	—	—	●	—		
d1	—	●	●	—	—	—	—	—	—		
d2	●	●	●	—	—	—	—	—	—		
d3	●	●	●	—	—	—	—	—	—		

GX IEC Developer

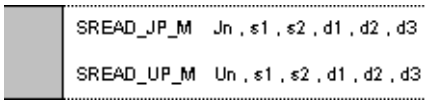
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



Variablen

Operand	Befehlswert	Datentyp	
		MELSEC	IEC
Jn	Netzwerknummer der HOST-Station. ●[1]	BIN-16-Bit	ANY16
Un	Kopf-Ein-/Ausgangsadresse des Netzwerkmoduls der HOST-Station. ●[2]		
s1	Erster Operand der HOST-Station, in dem die Kontrolldaten gespeichert sind.	Adresse	Array [1..18] of ANY16
s2	Erster Operand der Station, in dem die zu lesenden Daten gespeichert sind.		ANY16
d1	Erster Operand der HOST-Station, in dem die gelesenen Daten gespeichert werden.		
d2	Operand der HOST-Station, der nach vollständiger Verarbeitung der Anweisung für einen Programmzyklus gesetzt wird.	Bit	BOOL
d3	Operand der Ziel-Station, der nach vollständiger Verarbeitung der Anweisung für einen Programmzyklus gesetzt wird.		

HINWEISE

- *[1] Die Netzwerknummer der HOST-Station muss zwischen 1 und 239 liegen. Das Netzwerk mit der Nummer 254 wird über Einstellungen so konfiguriert, dass andere Stationen auf die aktive Station zugreifen können.*
- *[2] Die Kopf-Ein-/Ausgangsadresse des Netzwerkmoduls der HOST-Station muss zwischen 0 und FE$_H$ liegen. Es ist zu beachten, dass der Compiler für Un eine hexadezimale Adressangabe erwartet. Eine Dezimalangabe wird automatisch in eine Hexadezimalzahl umgewandelt.*

Die SREAD-Anweisung kann nur ausgeführt werden, wenn sich in der Ziel-Station eine QnA CPU befindet.

Die SREAD-Anweisung kann bei der Verwendung einer ACPU im MELSECNET/10 nicht angewendet werden.

Es können nur Stationsnummern für QnA CPUs als Nummer der Ziel-Station angegeben werden.

Operandenübersicht der Kontrolldaten

Operand	Bedeutung	Funktion	Werte-bereich	Festlegung durch
(s1)+0 Array_s1[1]	Ausführungs-Modus	Die Eingangsbestätigung ist gesetzt: (Bit 0 (b0) = 1, fest eingestellt)	0001H 0081H	Benutzer
	Abschluss-Modus der Fehlerverar-beitung	Speichern der Uhr-Daten des Zeitpunktes, an dem die Fehlerverabeitung abgeschlossen ist: - Keine Speicherung der Uhr-Daten, Bit 7 (b7) = 0 - Speicherung der Uhr-Daten, Bit 7 (b7) = 1 (Uhr-Daten ab (s1)+11 (Array_s1[12]) aufwärts)		
(s1)+1 Array_s1[2]	Abschluss-Status der Anweisungs-verarbeitung	Statusanzeige nach Abschluss der Anweisungs-verarbeitung: - 0 = Keine Fehler (normaler Verarbeitungs-abschluss)- ungleich 0 = Fehlercode ●3	—	System
(s1)+2 Array_s1[3]	Von der HOST-Station verwendeter Kanal	Angabe des Kanals, den die HOST-Station verwenden wird.	1 bis 8	Benutzer
(s1)+3 Array_s1[4]	Dummy	Wird nicht verwendet.	0	—
(s1)+4 Array_s1[5]	Netzwerknummer der Ziel-Station	Angabe der Netzwerknummer der Station, aus der gelesen wird.	1 bis 239 254 ●4	Benutzer
(s1)+5 Array_s1[6]	Nummer der Ziel-Station	Angabe der Stationsnummer der Ziel-Station.	1 bis 64	Benutzer
(s1)+6 Array_s1[7]	Dummy	Wird nicht verwendet.	—	—
(s1)+7 Array_s1[8]	Anzahl der Über-tragungsversuche	Angabe der Anzahl der Versuche, die unternom-men werden, um die vollständige Verarbeitung der SREAD-Anweisung innerhalb der in (s1)+8 (Array_s1[9]) gespeicherten Überwachungszeit durchzuführen.	1 bis15	Benutzer
	Anzahl der durch-geführten Übertra-gungsversuche	Speicherung der durchgeführten Übertragungs-versuche.		System
(s1)+8 Array_s1[9]	Zeiteinstellung des WDT zur Empfangs-zeitüberwachung	Angabe der Überwachungszeit für SREAD-Lese-Operationen in Sekunden. Erfolgt der Abschluss der Operation nicht innerhalb der angegebenen Zeit, wird die Übertragung mir der in (s1)+7 (Array_s1[8]) angegebenen Anzahl wiederholt.	1 bis 32767 0 = 10 s (fest eingestellt)	Benutzer
(s1)+9 Array_s1[10]	Empfangsdaten-länge	Angabe der Anzahl der auszulesenden Datenblöcke.	1 bis 480	Benutzer
(s1)+10 Array_s1[11]	Dummy	Wird nicht verwendet.	—	—
(s1)+11 Array_s1[12]	Flag der Uhr-Daten-speicherung (wird nur im Fehlerfall gesetzt)	Dieses Flag speichert den Status der in (s1)+0 (Array_s1[1]) angegebenen Speicheroption der Uhr-Daten: Keine Speicherung der Uhr-Daten = 0 Speicherung der Uhr-Daten = 1	—	System
(s1)+12 Array_s1[13]	Uhr-Daten (werden nur im Fehlerfall gesetzt)	Höchstwertiges Byte = Jahr (0 bis 99) Niedrigstwertiges Byte = Monat (1 bis 12)	—	System
(s1)+13 Array_s1[14]		Höchstwertiges Byte = Tag (1 bis 31) Niedrigstwertiges Byte = Stunde (0 bis 23)		
(s1)+14 Array_s1[15]		Höchstwertiges Byte = Minute (0 bis 59) Niedrigstwertiges Byte = Sekunde (0 bis 59)		
(s1)+15 Array_s1[16]		Höchstwertiges Byte = 00H Niedrigstwertiges Byte = Wochentag (0 bis 6) (Sonntag = 0, Samstag = 6)		
(s1)+16 Array_s1[17]	Nummer des Netz-werks, in dem der Fehler aufgetreten ist	Speicherung der Netzwerknummer der Station, in der der Fehler aufgetreten ist. Die Netzwerk-nummer liegt zwischen 1 und 239.	—	System

Operand	Bedeutung	Funktion	Werte-bereich	Festlegung durch
(s1)+17 Array_s1[18]	Nummer der Station, in der der Fehler aufgetreten ist	Speicherung der Nummer der Station, in der der Fehler aufgetreten ist. Die Speicherung wird nicht durchgeführt, wenn der Abschluss-Status der Anweisungsausführung "Kanal in Betrieb" lautet. Die Stationsnummer liegt zwischen 1 und 64.	—	System

●[3] Weitere Informationen über Fehlercodes enthält das Handbuch "MELSECNET/10 für QnA-Netzwerk-Systeme".

●[4] Die Netzwerknummer 254 ist ausgewählt, wenn für Jn der Wert 254 angegeben wird.

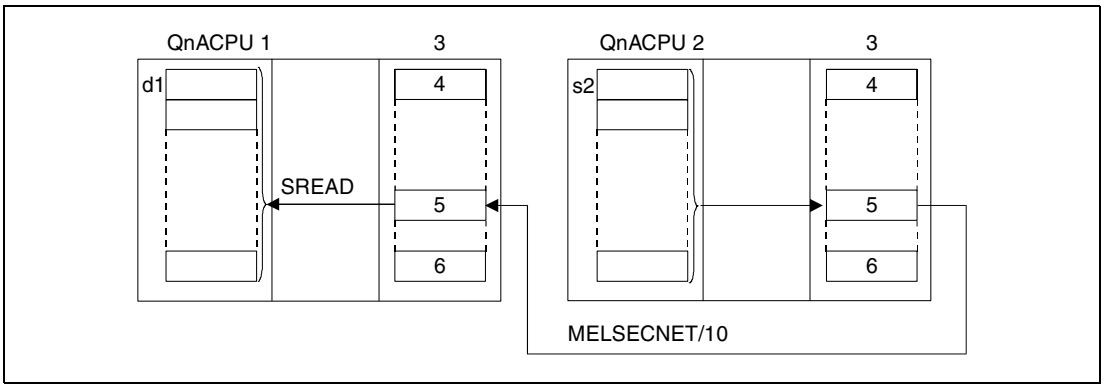
Funktions-weise

Lesen von Daten aus Wortoperanden anderer Stationen

SREAD Leseanweisung

Die SREAD-Anweisung liest die ab s2 gespeicherten Daten einer im MELSECNET/10 angeschlossenen Station. Die Stations- und Netzwerknummer sind in den Kontrolldaten angegeben. Die aus der Station gelesenen Daten werden ab d1 in der HOST-Station gespeichert.

Nach dem Abschluss der Lese-Operation werden in der HOST-Station der Operand d2 und in der Zielstation der Operand d3 gesetzt.



[1] HOST-Station
[2] Ziel-Station
[3] Netzwerkmodul
[4] Kanal 1
[5] Kanal n
[6] Kanal 8

Über eine Relais-Station und gesetzte Routing-Parameter kann auch auf Stationen in anderen Netzwerken zugegriffen werden.

Die Data-Link-Anweisungen können nicht von mehreren Stellen gleichzeitig mit gemeinsamem Zugriff auf die gleichen Kanäle ausgeführt werden. Bei gleichzeitiger Ausführung der Anweisungen von zwei oder mehr Stellen wird durch den Handshake-Betrieb der beiden aktiven Stationen eine Ausführung weiterer Data-Link-Anweisungen verhindert.

Der Ausführungs-Status und der Abschluss-Status (normal, nicht normal) der SREAD-Anweisung können mittels

- dem Kommunikations-Kanal-Flag (●[5]) des verwendeten Kanals,
- den Operanden, die nach Abschluss der Lese-Operation in der HOST-Station (d2) und in der Ziel-Station (d3) gesetzt werden, und
- der Status-Anzeige des Operationsabschlusses (Abschluss einer fehlerfreien oder fehlerhaften Übertragung) ((d2)+1)

wie folgt überprüft werden:

Kommunikations-Kanal-Flag

Dieses Flag wird während der Ausführung der SREAD-Anweisung gesetzt. Das Flag wird mit der Verarbeitung der END-Anweisung in dem Zyklus zurückgesetzt, in dem die Lese-Operation abgeschlossen wurde.

Operand der HOST-Station, der den Lese-Operationsabschluss anzeigt

Dieser Operand wird mit der Verarbeitung der END-Anweisung in dem Zyklus gesetzt, in dem die Lese-Operation abgeschlossen wurde. Bei der Verarbeitung der darauffolgenden END-Anweisung wird dieser Operand wieder zurückgesetzt.

Status-Anzeige des Lese-Operationsabschlusses

Dieser Operand wird in Abhängigkeit des Abschlussergebnisses der Lese-Operation gesetzt.

Beim Abschluss einer normalen (fehlerfreien) Übertragung bleibt der Operand unverändert zurückgesetzt.

Beim Abschluss einer fehlerhaften Übertragung wird dieser Operand mit der Verarbeitung der END-Anweisung in dem Zyklus gesetzt, in dem die Verarbeitung der SREAD-Anweisung abgeschlossen wurde. Bei der Verarbeitung der darauffolgenden END-Anweisung wird dieser Operand wieder zurückgesetzt.

Operand der Ziel-Station, der den Lese-Operationsabschluss anzeigt

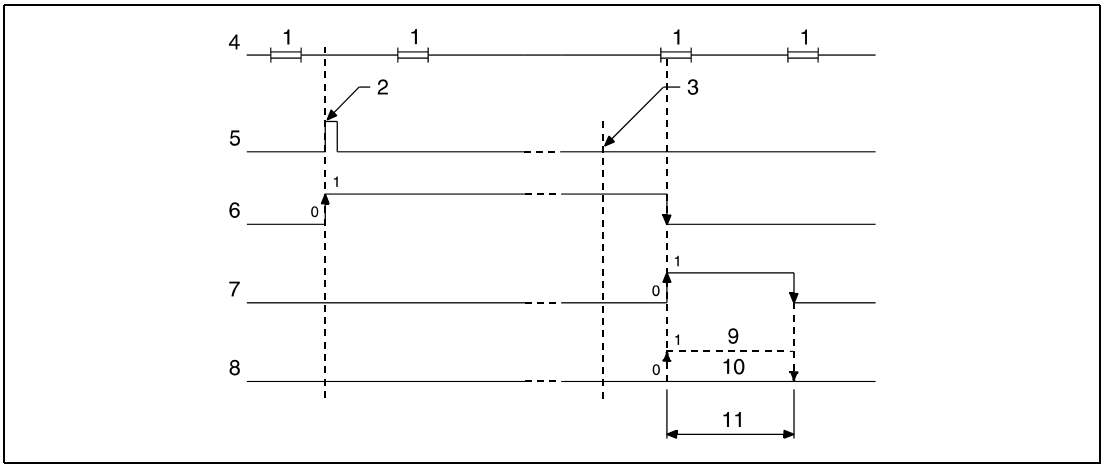
Dieser Operand wird mit der Verarbeitung der END-Anweisung in dem Zyklus gesetzt, in dem die Übertragung der Daten abgeschlossen wurde. Bei der Verarbeitung der darauffolgenden END-Anweisung wird dieser Operand wieder zurückgesetzt.

HINWEIS

●[5] *Die folgende Tabelle ordnet den Kanalnummern die entsprechenden Kommunikations-Kanal-Flags zu.*

Kanalnummer	1	2	3	4	5	6	7	8
Kommunikations-Kanal-Flag	SB30	SB32	SB34	SB36	SB38	SB40	SB42	SB44

Die folgende Abbildung zeigt die Operationen der HOST-Station während der Ausführung der SREAD-Anweisung.



1 END-Verarbeitung

2 Ausführung der SREAD-Anweisung

3 Abschluss der Operation

4 Programm der HOST-Station

5 SREAD-Anweisung

6 Kommunikations-Kanal-Flag

7 Operand der HOST-Station, der nach Abschluss der Operation gesetzt wird (d2)

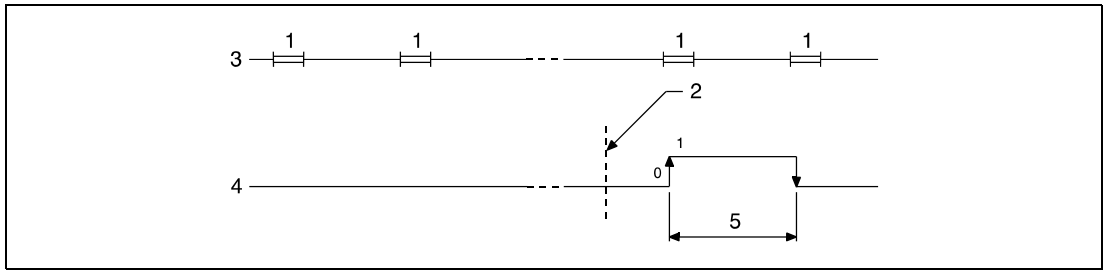
8 Status-Anzeige des Operationsabschlusses ((d2)+1)

9 Abschluss einer nicht normalen (fehlerhaften) Übertragung

10 Abschluss einer normalen (fehlerfreien) Übertragung

11 Ein Zyklus

Die folgende Abbildung zeigt die Operationen der Ziel-Station während der Ausführung der SREAD-Anweisung.



[1] END-Verarbeitung

[2] Abschluss der Operation

[3] Programm der Ziel-Station

[4] Operand der Ziel-Station, der nach Abschluss der Operation gesetzt wird (d3)

[5] Ein Zyklus

Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die Kontrolldateninhalte liegen nicht im gültigen gültigen Wertebereich (Fehlercode 4100).
- Das Netzwerk mit der mit Jn angegebenen Nummer ist nicht mit der Station verbunden (Fehlercode 4102).
- Das Modul an der mit Un angegebenen Ein-/Ausgangsadresse ist kein Netzwerkmodul (Fehlercode 2111).

8.6.3 WRITE

CPU

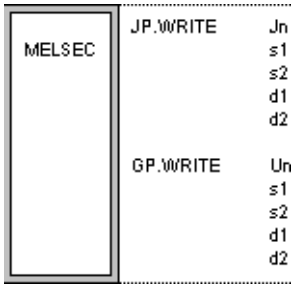
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	

Operanden MELSEC Q

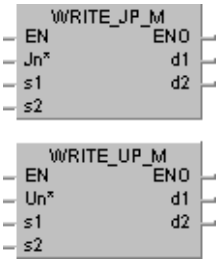
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	—	●	●	—	—	—	—	—	—	SM0	10
s2	—	●	●	—	—	—	—	—	—		
d1	●	●	●	—	—	—	—	—	—		
d2	●	●	●	—	—	—	—	—	—		

GX IEC Developer

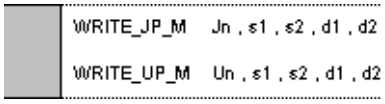
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



Variablen

Operand	Befehlswert	Datentyp	
		MELSEC	IEC
Jn	Netzwerknummer der HOST-Station. ●[1]	BIN-16-Bit	ANY16
Un	Kopf-Ein-/Ausgangsadresse des Netzwerkmoduls der HOST-Station. ●[2]		
s1	Erster Operand der HOST-Station, in dem die Kontrolldaten gespeichert sind.	Adresse	Array [1..18] of ANY16
s2	Erster Operand der Station, in dem die zu schreibenden Daten gespeichert sind.		ANY16
d1	Erster Operand der Ziel-Station, in den die Daten geschrieben werden.		
d2	Operand, der nach vollständiger Verarbeitung der Anweisung für einen Programmzyklus gesetzt wird.	Bit	BOOL

HINWEISE

- *[1] Die Netzwerknummer der HOST-Station muss zwischen 1 und 239 liegen. Das Netzwerk mit der Nummer 254 wird über Einstellungen so konfiguriert, dass andere Stationen auf die aktive Station zugreifen können.*
- *[2] Die Kopf-Ein-/Ausgangsadresse des Netzwerkmoduls der HOST-Station muss zwischen 0 und FE$_H$ liegen. Es ist zu beachten, dass der Compiler für Un eine hexadezimale Adressangabe erwartet. Eine Dezimalangabe wird automatisch in eine Hexadezimalzahl umgewandelt.*

Die WRITE-Anweisung kann nur ausgeführt werden, wenn sich in der Ziel-Station eine QnA-CPU befindet.

Die WRITE-Anweisung kann bei der Verwendung einer ACPU im MELSECNET/10 nicht angewendet werden.

Die Adressangabe "FF$_H$" (alle Stationen im Ziel-Netzwerk) kann mit der WRITE-Anweisung nur für eine Ziel-Sationsnummer in einem Netzwerk vorgenommen werden, in dem sich ausschließlich Stationen befinden, in denen QnA-CPUs eingesetzt sind. Die Adressangabe "FF$_H$" ist in Netzwerken mit gemischter Verwendung von Stationen mit QnA- und ACPUs nicht möglich.

Operandenübersicht der Kontrolldaten

Operand	Bedeutung	Funktion	Werte-bereich	Festlegung durch
(s1)+0 Array_s1[1]	Ausführungs-Modus	Bestätigung des Übertragungsabschlusses = Bit 0 auf 0 setzen Keine Bestätigung des Übertragungsabschlusses = Bit 0 auf 1 setzen	0000H 0001H 0080H 0081H	Benutzer
	Abschluss-Modus der Fehlerverar-beitung	Speichern der Uhr-Daten des Zeitpunktes, an dem die Fehlerverabeitung abgeschlossen ist: - Keine Speicherung der Uhr-Daten, Bit 7 (b7) = 0 - Speicherung der Uhr-Daten, Bit 7 (b7) = 1 (Uhr-Daten ab (s1)+10 (Array_s1[11]) aufwärts)		
(s1)+1 Array_s1[2]	Abschluss-Status der Anweisungs-verarbeitung	Statusanzeige nach Abschluss der Anweisungs-verarbeitung: - 0 = Keine Fehler (normaler Verarbeitungs-abschluss) - ungleich 0 = Fehlercode ●[3]	—	System
(s1)+2 Array_s1[3]	Von der HOST-Station verwendeter Kanal	Angabe des Kanals, den die HOST-Station verwenden wird.	1 bis 8	Benutzer
(s1)+3 Array_s1[4]	Dummy	Wird nicht verwendet.	0	—
(s1)+4 Array_s1[5]	Netzwerknummer der Ziel-Station	Angabe der Netzwerknummer der Ziel-Station.	1 bis 239 254 ●[4]	Benutzer
(s1)+5 Array_s1[6]	Nummer der Ziel-Station	Angabe der Stationsnummer der Ziel-Station.	Stationsnum-mer: 1 bis 64 Gruppen-adressierung: 81H bis 89H Alle Stationen im Zielnetzwerk: FFH	Benutzer
(s1)+6 Array_s1[7]	Dummy	Wird nicht verwendet.	—	—
(s1)+7 Array_s1[8]	Anzahl der Über-tragungsversuche	Angabe der Anzahl der Versuche, die unternom-men werden, um die vollständige Übertragung der Daten innerhalb der in (s1)+8 (Array_s1[9]) gespeicherten Überwachungszeit durchzuführen. Diese Option ist nur aktiv, wenn der Ausführungs-Modus ((s1)+0, Array_s1[1]) gesetzt (1) ist.	1 bis15	Benutzer
	Anzahl der durch-geführten Übertra-gungsversuche	Speicherung der durchgeführten Übertragungs-versuche.		System
(s1)+8 Array_s1[9]	Zeiteinstellung des WDT zur Empfangs-zeitüberwachung	Angabe der Überwachungszeit für die Operation in Sekunden. Erfolgt der Abschluss der Operation nicht innerhalb der angegebenen Zeit, wird die Übertragung mir der in (s1)+7 (Array_s1[8]) angegebenen Anzahl wiederholt.	1 bis 32767 0 = 10 s (fest eingestellt) Diese Option ist nur aktiv, wenn der Ausfühungs-modus gesetzt (1) ist	Benutzer
(s1)+9 Array_s1[10]	Sendedatenlänge	Angabe der Anzahl der zu schreibenden Datenblöcke.	1 bis 480	Benutzer
(s1)+10 Array_s1[11]	Dummy	Wird nicht verwendet.	—	—
(s1)+11 Array_s1[12]	Flag der Uhr-Daten-speicherung (wird nur im Fehlerfall gesetzt)	Dieses Flag speichert den Status der in (s1)+0 (Array_s1[1]) angegebenen Speicheroption der Uhr-Daten: Keine Speicherung der Uhr-Daten = 0 Speicherung der Uhr-Daten = 1	—	System

Operand	Bedeutung	Funktion	Werte-bereich	Festlegung durch
(s1)+12 Array_s1[13]	Uhr-Daten (werden nur im Fehlerfall gesetzt)	Höchstwertiges Byte = Jahr (0 bis 99) Niedrigstwertiges Byte = Monat (1 bis 12)	—	System
(s1)+13 Array_s1[14]		Höchstwertiges Byte = Tag (1 bis 31) Niedrigstwertiges Byte = Stunde (0 bis 23)		
(s1)+14 Array_s1[15]		Höchstwertiges Byte = Minute (0 bis 59) Niedrigstwertiges Byte = Sekunde (0 bis 59)		
(s1)+15 Array_s1[16]		Höchstwertiges Byte = 00H Niedrigstwertiges Byte = Wochentag (0 bis 6) (Sonntag = 0, Samstag = 6)		
(s1)+16 Array_s1[17]	Nummer des Netzwerks, in dem der Fehler aufgetreten ist	Speicherung der Netzwerknummer der Station, in der der Fehler aufgetreten ist. Die Netzwerknummer liegt zwischen 1 und 239.	—	System
(s1)+17 Array_s1[18]	Nummer der Station, in der der Fehler aufgetreten ist	Speicherung der Nummer der Station, in der der Fehler aufgetreten ist. Die Speicherung wird nicht durchgeführt, wenn der Abschluss-Status der Anweisungsausführung "Kanal in Betrieb" lautet. Die Stationsnummer liegt zwischen 1 und 64.	—	System

●[3] Weitere Informationen über Fehlercodes enthält das Handbuch "MELSECNET/10 für QnA-Netzwerk-Systeme".

●[4] Die Netzwerknummer 254 ist ausgewählt, wenn für Jn der Wert 254 angegeben wird.

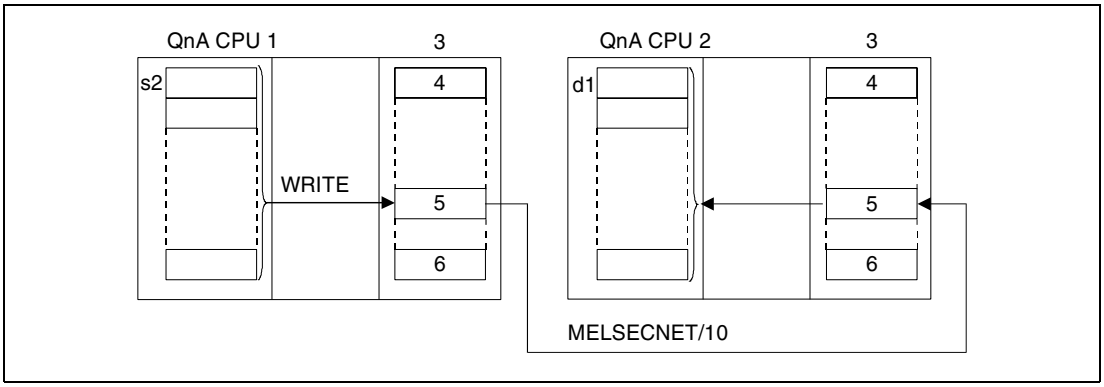
Funktionsweise

Schreiben von Daten in Wortoperanden anderer Stationen

WRITE Schreibanweisung

Die WRITE-Anweisung schreibt die ab s2 gespeicherten Daten der HOST-Station in eine im MELSECNET/10 angeschlossene Station. Die Stations- und Netzwerknummer sind in den Kontrolldaten angegeben. Die Daten werden ab d1 in der Ziel-Station gespeichert.

Nach dem Abschluss der Schreib-Operation wird der Operand d2 gesetzt.



[1] HOST-Station
[2] Ziel-Station
[3] Netzwerkmodul
[4] Kanal 1
[5] Kanal n
[6] Kanal 8

Über eine Relais-Station und gesetzte Routing-Parameter kann auch auf Stationen in anderen Netzwerken zugegriffen werden.

Die Data-Link-Anweisungen können nicht von mehreren Stellen gleichzeitig mit gemeinsamem Zugriff auf die gleichen Kanäle ausgeführt werden. Bei gleichzeitiger Ausführung der Anweisungen von zwei oder mehr Stellen wird durch den Handshake-Betrieb der beiden aktiven Stationen eine Ausführung weiterer Data-Link-Anweisungen verhindert.

Der Ausführungs-Status und der Abschluss-Status (normal, nicht normal) der WRITE-Anweisung können mittels

- dem Kommunikations-Kanal-Flag (●[5]) des verwendeten Kanals,
- dem Operanden, der nach Abschluss der Schreib-Operation gesetzt wird (d2) und
- der Status-Anzeige des Operationsabschlusses (Abschluss einer fehlerfreien oder fehlerhaften Übertragung) ((d2)+1)

wie folgt überprüft werden:

Kommunikations-Kanal-Flag

Dieses Flag wird während der Ausführung der WRITE-Anweisung gesetzt. Das Flag wird mit der Verarbeitung der END-Anweisung in dem Zyklus zurückgesetzt, in dem die Schreib-Operation abgeschlossen wurde.

Operand der HOST-Station, der den Schreib-Operationsabschluss anzeigt

Dieser Operand wird mit der Verarbeitung der END-Anweisung in dem Zyklus gesetzt, in dem die Schreib-Operation abgeschlossen wurde. Bei der Verarbeitung der darauffolgenden END-Anweisung wird dieser Operand wieder zurückgesetzt.

Status-Anzeige des Schreib-Operationsabschlusses

Dieser Operand wird in Abhängigkeit des Abschlussergebnisses der Schreib-Operation gesetzt.

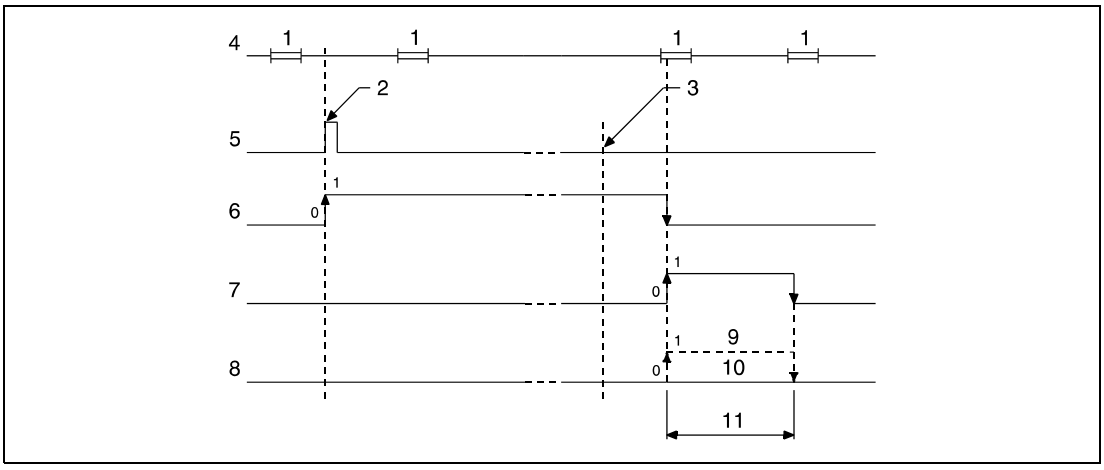
Beim Abschluss einer normalen (fehlerfreien) Übertragung bleibt der Operand unverändert zurückgesetzt.

Beim Abschluss einer nicht normalen (fehlerhaften) Übertragung wird dieser Operand mit der Verarbeitung der END-Anweisung in dem Zyklus gesetzt, in dem die Verarbeitung der WRITE-Anweisung abgeschlossen wurde. Bei der Verarbeitung der darauffolgenden END-Anweisung wird dieser Operand wieder zurückgesetzt.

HINWEIS ●[5] *Die folgende Tabelle ordnet den Kanalnummern die entsprechenden Kommunikations-Kanal-Flags zu.*

Kanalnummer	1	2	3	4	5	6	7	8
Kommunikations-Kanal-Flag	SB30	SB32	SB34	SB36	SB38	SB3A	SB3B	SB3D

Die folgende Abbildung zeigt die Operationen der HOST-Station während der Ausführung der WRITE-Anweisung.



[1] END-Verarbeitung
[2] Ausführung der WRITE-Anweisung
[3] Abschluss der Operation
[4] Programm der HOST-Station
[5] WRITE-Anweisung
[6] Kommunikations-Kanal-Flag
[7] Operand der HOST-Station, der nach Abschluss der Operation gesetzt wird (d2)
[8] Status-Anzeige des Operationsabschlusses ((d2)+1)
[9] Abschluss einer nicht normalen (fehlerhaften) Übertragung
[10] Abschluss einer normalen (fehlerfreien) Übertragung
[11] Ein Zyklus

Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die Kontrolldateninhalte liegen nicht im gültigen gültigen Wertebereich (Fehlercode 4100).
- Das Netzwerk mit der mit Jn angegebenen Nummer ist nicht mit der Station verbunden (Fehlercode 4102).
- Das Modul an der mit Un angegebenen Ein-/Ausgangsadresse ist kein Netzwerkmodul (Fehlercode 2111).

8.6.4 SWRITE

CPU

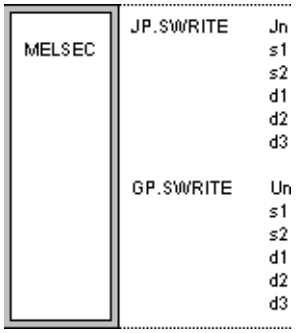
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	

Operanden MELSEC Q

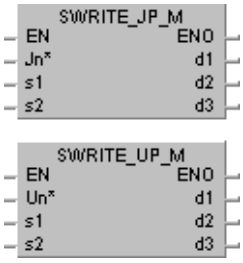
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	—	●	●	—	—	—	—	—	—	SM0	11
s2	—	●	●	—	—	—	—	—	—		
d1	●	●	●	—	—	—	—	—	—		
d2	●	●	●	—	—	—	—	—	—		
d3	●	●	●	—	—	—	—	—	—		

GX IEC Developer

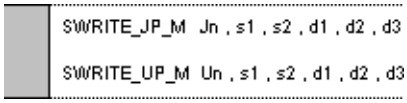
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



Variablen

Operand	Befehlswert	Datentyp	
		MELSEC	IEC
Jn	Netzwerknummer der HOST-Station. ●[1]	BIN-16-Bit	ANY16
Un	Kopf-Ein-/Ausgangsadresse des Netzwerkmoduls der HOST-Station. ●[2]		
s1	Erster Operand der HOST-Station, in dem die Kontrolldaten gespeichert sind.	Adresse	Array [1..18] of ANY16
s2	Erster Operand der Station, in dem die zu schreibenden Daten gespeichert sind.		ANY16
d1	Erster Operand der Ziel-Station, in den die Daten geschrieben werden.		
d2	Operand der HOST-Station, der nach vollständiger Verarbeitung der Anweisung für einen Programmzyklus gesetzt wird.	Bit	BOOL
d3	Operand der Ziel-Station, der nach vollständiger Verarbeitung der Anweisung für einen Programmzyklus gesetzt wird.		

HINWEISE

- *[1] Die Netzwerknummer der HOST-Station muss zwischen 1 und 239 liegen. Das Netzwerk mit der Nummer 254 wird über Einstellungen so konfiguriert, dass andere Stationen auf die aktive Station zugreifen können.*
- *[2] Die Kopf-Ein-/Ausgangsadresse des Netzwerkmoduls der HOST-Station muss zwischen 0 und FE$_H$ liegen. Es ist zu beachten, dass der Compiler für Un eine hexadezimale Adressangabe erwartet. Eine Dezimalangabe wird automatisch in eine Hexadezimalzahl umgewandelt.*

Die SWRITE-Anweisung kann nur ausgeführt werden, wenn sich in der Ziel-Station eine QnA-CPU befindet.

Die SWRITE-Anweisung kann bei der Verwendung einer ACPU im MELSECNET/10 nicht angewendet werden.

Die Adressangabe "FF$_H$" (alle Stationen im Ziel-Netzwerk) kann mit der SWRITE-Anweisung nur für eine Ziel-Sationsnummer in einem Netzwerk vorgenommen werden, in dem sich ausschließlich Stationen befinden, in denen QnA-CPUs eingesetzt sind. Die Adressangabe "FF$_H$" ist in Netzwerken mit gemischter Verwendung von Stationen mit QnA- und ACPUs nicht möglich.

Operandenübersicht der Kontrolldaten

Operand	Bedeutung	Funktion	Werte-bereich	Festlegung durch
(s1)+0 Array_s1[1]	Ausführungs-Modus	Bestätigung des Übertragungsabschlusses = Bit 0 auf 0 setzen Keine Bestätigung des Übertragungsabschlusses = Bit 0 auf 1 setzen	0000H 0001H 0080H 0081H	Benutzer
	Abschluss-Modus der Fehlerverar-beitung	Speichern der Uhr-Daten des Zeitpunktes, an dem die Fehlerverabeitung abgeschlossen ist: - Keine Speicherung der Uhr-Daten, Bit 7 (b7) = 0 - Speicherung der Uhr-Daten, Bit 7 (b7) = 1 (Uhr-Daten ab (s1)+10 (Array_s1[11]) aufwärts)		
(s1)+1 Array_s1[2]	Abschluss-Status der Anweisungs-verarbeitung	Statusanzeige nach Abschluss der Anweisungs-verarbeitung: - 0 = Keine Fehler (normaler Verarbeitungs-abschluss) - ungleich 0 = Fehlercode ●[3]	—	System
(s1)+2 Array_s1[3]	Von der HOST-Station verwendeter Kanal	Angabe des Kanals, den die HOST-Station verwenden wird.	1 bis 8	Benutzer
(s1)+3 Array_s1[4]	Dummy	Wird nicht verwendet.	0	—
(s1)+4 Array_s1[5]	Netzwerknummer der Ziel-Station	Angabe der Netzwerknummer der Ziel-Station.	1 bis 239 254 ●[4]	Benutzer
(s1)+5 Array_s1[6]	Nummer der Ziel-Station	Angabe der Stationsnummer der Ziel-Station.	Stationsnum-mer: 1 bis 64 Gruppen-adressierung: 81H bis 89H Alle Stationen im Zielnetzwerk: FFH	Benutzer
(s1)+6 Array_s1[7]	Dummy	Wird nicht verwendet.	—	—
(s1)+7 Array_s1[8]	Anzahl der Über-tragungsversuche	Angabe der Anzahl der Versuche, die unternom-men werden, um die vollständige Übertragung der Daten innerhalb der in (s1)+8 (Array_s1[9]) gespeicherten Überwachungszeit durchzuführen. Diese Option ist nur aktiv, wenn der Ausführungs-Modus ((s1)+0, Array_s1[1]) gesetzt (1) ist.	1 bis15	Benutzer
	Anzahl der durch-geführten Übertra-gungsversuche	Speicherung der durchgeführten Übertragungs-versuche.		System
(s1)+8 Array_s1[9]	Zeiteinstellung des WDT zur Empfangs-zeitüberwachung	Angabe der Überwachungszeit für die Operation in Sekunden. Erfolgt der Abschluss der Operation nicht innerhalb der angegebenen Zeit, wird die Übertragung mir der in (s1)+7 (Array_s1[8]) angegebenen Anzahl wiederholt.	1 bis 32767 0 = 10 s (fest eingestellt) Diese Option ist nur aktiv, wenn der Ausfühungs-modus gesetzt (1) ist	Benutzer
(s1)+9 Array_s1[10]	Empfangsdaten-länge	Angabe der Anzahl der zu schreibenden Datenblöcke.	1 bis 480	Benutzer
(s1)+10 Array_s1[11]	Dummy	Wird nicht verwendet.	—	—
(s1)+11 Array_s1[12]	Flag der Uhr-Daten-speicherung (wird nur im Fehlerfall gesetzt)	Dieses Flag speichert den Status der in (s1)+0 (Array_s1[1]) angegebenen Speicheroption der Uhr-Daten: Keine Speicherung der Uhr-Daten = 0 Speicherung der Uhr-Daten = 1	—	System

Operand	Bedeutung	Funktion	Werte-bereich	Festlegung durch
(s1)+12 Array_s1[13]	Uhr-Daten (werden nur im Fehlerfall gesetzt)	Höchstwertiges Byte = Jahr (0 bis 99) Niedrigstwertiges Byte = Monat (1 bis 12)	—	System
(s1)+13 Array_s1[14]		Höchstwertiges Byte = Tag (1 bis 31) Niedrigstwertiges Byte = Stunde (0 bis 23)		
(s1)+14 Array_s1[15]		Höchstwertiges Byte = Minute (0 bis 59) Niedrigstwertiges Byte = Sekunde (0 bis 59)		
(s1)+15 Array_s1[16]		Höchstwertiges Byte = 00H Niedrigstwertigesniedrigstwertiges Byte = Wochentag (0 bis 6) (Sonntag = 0, Samstag = 6)		
(s1)+16 Array_s1[17]	Nummer des Netzwerks, in dem der Fehler aufgetreten ist	Speicherung der Netzwerknummer der Station, in der der Fehler aufgetreten ist. Die Netzwerknummer liegt zwischen 1 und 239.	—	System
(s1)+17 Array_s1[18]	Nummer der Station, in der der Fehler aufgetreten ist	Speicherung der Nummer der Station, in der der Fehler aufgetreten ist. Die Speicherung wird nicht durchgeführt, wenn der Abschluss-Status der Anweisungsausführung "Kanal in Betrieb" lautet. Die Stationsnummer liegt zwischen 1 und 64.	—	System

●[3] Weitere Informationen über Fehlercodes enthält das Handbuch "MELSECNET/10 für QnA-Netzwerk-Systeme".

●[4] Die Netzwerknummer 254 ist ausgewählt, wenn für Jn der Wert 254 angegeben wird.

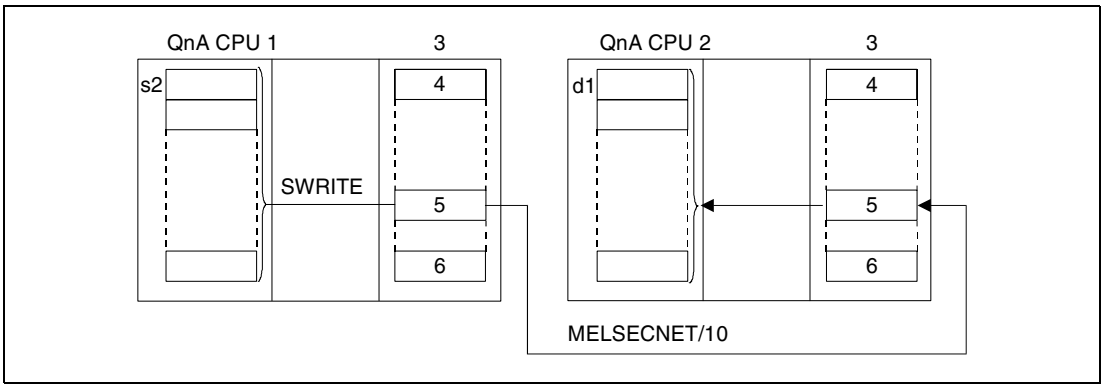
Funktions-weise

Schreiben von Daten in Wortoperanden anderer Stationen

SWRITE Schreibanweisung

Die SWRITE-Anweisung schreibt die ab s2 gespeicherten Daten der HOST-Station in eine im MELSECNET/10 angeschlossene Station. Die Stations- und Netzwerknummer sind in den Kontrolldaten angegeben. Die Daten werden ab d1 in der Ziel-Station gespeichert.

Nach dem Abschluss der Schreib-Operation werden in der HOST-Station der Operand d2 und in der Zielstation der Operand d3 gesetzt.



[1] HOST-Station
[2] Ziel-Station
[3] Netzwerkmodul
[4] Kanal 1
[5] Kanal n
[6] Kanal 8

Über eine Relais-Station und gesetzte Routing-Parameter kann auch auf Stationen in anderen Netzwerken zugegriffen werden.

Die Data-Link-Anweisungen können nicht von mehreren Stellen gleichzeitig mit gemeinsamem Zugriff auf die gleichen Kanäle ausgeführt werden. Bei gleichzeitiger Ausführung der Anweisungen von zwei oder mehr Stellen wird durch den Handshake-Betrieb der beiden aktiven Stationen eine Ausführung weiterer Data-Link-Anweisungen verhindert.

Der Ausführungs-Status und der Abschluss-Status (normal, nicht normal) der SWRITE-Anweisung können mittels

- dem Kommunikations-Kanal-Flag (●[5]) des verwendeten Kanals,
- den Operanden, die nach Abschluss der Schreib-Operation in der HOST-Station (d2) und in der Ziel-Station (d3) gesetzt werden und
- der Status-Anzeige des Operationsabschlusses (Abschluss einer fehlerfreien oder fehlerhaften Übertragung) ((d2)+1)

wie folgt überprüft werden:

Kommunikations-Kanal-Flag

Dieses Flag wird während der Ausführung der SWRITE-Anweisung gesetzt. Das Flag wird mit der Verarbeitung der END-Anweisung in dem Zyklus zurückgesetzt, in dem die Schreib-Operation abgeschlossen wurde.

Operand der HOST-Station, der den Schreib-Operationsabschluss anzeigt

Dieser Operand wird mit der Verarbeitung der END-Anweisung in dem Zyklus gesetzt, in dem die Schreib-Operation abgeschlossen wurde. Bei der Verarbeitung der darauffolgenden END-Anweisung wird dieser Operand wieder zurückgesetzt.

Status-Anzeige des Schreib-Operationsabschlusses

Dieser Operand wird in Abhängigkeit des Abschlussergebnisses der Schreib-Operation gesetzt.

Bei dem Abschluss einer normalen (fehlerfreien) Übertragung bleibt der Operand unverändert zurückgesetzt.

Bei dem Abschluss einer fehlerhaften Übertragung wird dieser Operand mit der Verarbeitung der END-Anweisung in dem Zyklus gesetzt, in dem die Verarbeitung der SWRITE-Anweisung abgeschlossen wurde. Bei der Verarbeitung der darauffolgenden END-Anweisung wird dieser Operand wieder zurückgesetzt.

Operand der Ziel-Station, der den Schreib-Operationsabschluss anzeigt

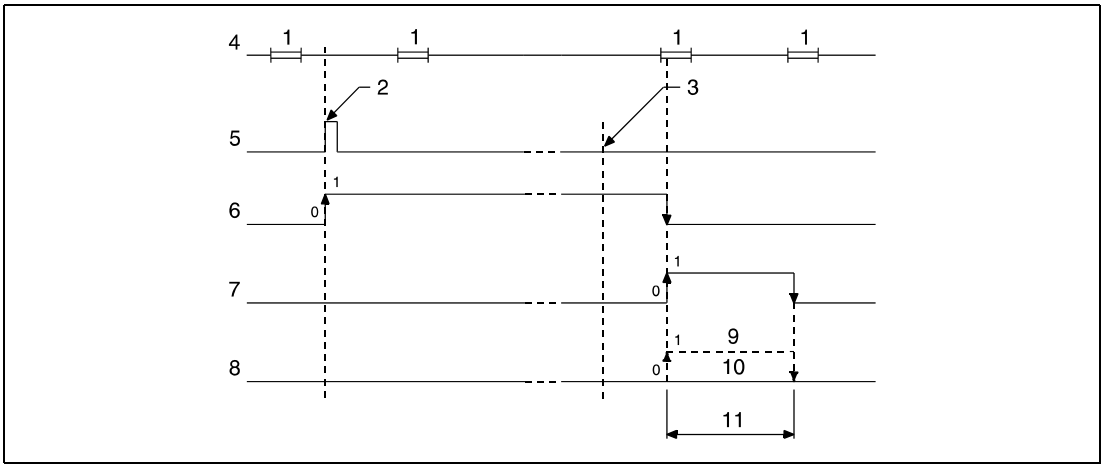
Dieser Operand wird mit der Verarbeitung der END-Anweisung in dem Zyklus gesetzt, in dem die Übertragung der in der SWRITE-Anweisung angegebenen Daten abgeschlossen wurde. Bei der Verarbeitung der darauffolgenden END-Anweisung wird dieser Operand wieder zurückgesetzt.

HINWEIS

●[5] *Die folgende Tabelle ordnet den Kanalnummern die entsprechenden Kommunikations-Kanal-Flags zu.*

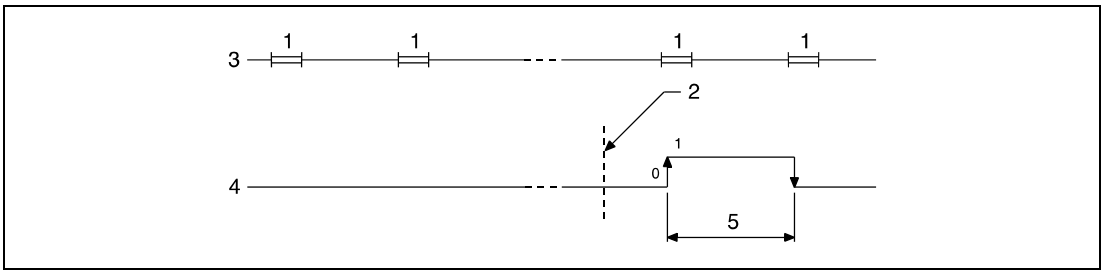
Kanalnummer	1	2	3	4	5	6	7	8
Kommunikations-Kanal-Flag	SB30	SB32	SB34	SB36	SB38	SB40	SB42	SB44

Die folgende Abbildung zeigt die Operationen der HOST-Station während der Ausführung der SWRITE-Anweisung.



1 END-Verarbeitung
2 Ausführung der SWRITE-Anweisung
3 Abschluss der Operation
4 Programm der HOST-Station
5 SWRITE-Anweisung
6 Kommunikations-Kanal-Flag
7 Operand der HOST-Station, der nach Abschluss der Operation gesetzt wird (d2)
8 Status-Anzeige des Operationsabschlusses ((d2)+1)
9 Abschluss einer nicht normalen (fehlerhaften) Übertragung
10 Abschluss einer normalen (fehlerfreien) Übertragung
11 Ein Zyklus

Die folgende Abbildung zeigt die Operationen der Ziel-Station während der Ausführung der SWRITE-Anweisung.



[1] END-Verarbeitung

[2] Abschluss der Operation

[3] Programm der Ziel-Station

[4] Operand der Ziel-Station, der nach Abschluss der Operation gesetzt wird (d3)

[5] Ein Zyklus

Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die Kontrolldateninhalte liegen nicht im gültigen Wertebereich (Fehlercode 4100).
- Das Netzwerk mit der mit Jn angegebenen Nummer ist nicht mit der Station verbunden (Fehlercode 4102).
- Das Modul an der mit Un angegebenen Ein-/Ausgangsadresse ist kein Netzwerkmodul (Fehlercode 2111).

8.6.5 SEND

CPU

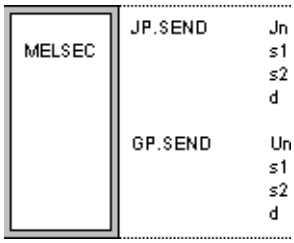
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	

Operanden MELSEC Q

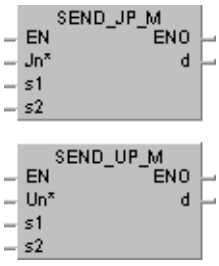
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	—	●	●	—	—	—	—	—	—	SM0	8
s2	—	●	●	—	—	—	—	—	—		
d	●	●	●	—	—	—	—	—	—		

GX IEC Developer

MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste

```
SEND_JP_M   Jn , s1 , s2 , d
SEND_UP_M   Un , s1 , s2 , d
```

Variablen

Operand	Befehlswert	Datentyp	
		MELSEC	IEC
Jn	Netzwerknummer der HOST-Station. ●[1]	BIN-16-Bit	ANY16
Un	Kopf-Ein-/Ausgangsadresse des Netzwerkmoduls der HOST-Station. ●[2]		
s1	Erste Adresse des Operanden, in dem die Kontrolldaten gespeichert sind.	Adresse	Array [1..18] of ANY16
s2	Erste Adresse des Operanden, in dem die zu sendenden Daten gespeichert sind.		ANY16
d	Operand, der nach vollständiger Verarbeitung der Anweisung für einen Programmzyklus gesetzt wird.	Bit	BOOL

HINWEISE

●[1] Die Netzwerknummer der HOST-Station muss zwischen 1 und 239 liegen. Das Netzwerk mit der Nummer 254 wird über Einstellungen so konfiguriert, dass andere Stationen auf die aktive Station zugreifen können.

●[2] Die Kopf-Ein-/Ausgangsadresse des Netzwerkmoduls der HOST-Station muss zwischen 0 und FE$_H$ liegen. Es ist zu beachten, dass der Compiler für Un eine hexadezimale Adressangabe erwartet. Eine Dezimalangabe wird automatisch in eine Hexadezimalzahl umgewandelt.

Operandenübersicht der Kontrolldaten

Operand	Bedeutung	Funktion	Werte-bereich	Festlegung durch
(s1)+0 Array_s1[1]	Ausführungs-Modus	Bestätigung des Übertragungsabschlusses = Bit 0 auf 0 setzen Keine Bestätigung des Übertragungsabschlusses = Bit 0 auf 1 setzen	0000H 0001H 0080H 0081H	Benutzer
	Abschluss-Modus der Fehlerverar-beitung	Speichern der Uhr-Daten des Zeitpunktes, an dem die Fehlerverabeitung abgeschlossen ist: - Keine Speicherung der Uhr-Daten, Bit 7 (b7) = 0 - Speicherung der Uhr-Daten, Bit 7 (b7) = 1 (Uhr-Daten ab (s1)+11 (Array_s1[12]) aufwärts)		
(s1)+1 Array_s1[2]	Abschluss-Status der Anweisungs-verarbeitung	Statusanzeige nach Abschluss der Anweisungs-verarbeitung: - 0 = Keine Fehler (normaler Verarbeitungs-abschluss) - ungleich 0 = Fehlercode ●[3]	—	System
(s1)+2 Array_s1[3]	Von der HOST-Station verwendeter Kanal	Angabe des Kanals, den die HOST-Station verwenden wird.	1 bis 8	Benutzer
(s1)+3 Array_s1[4]	Von der Ziel-Station verwendeter Kanal	Angabe der Netzwerknummer der Ziel-Station.	1 bis 8	Benutzer
(s1)+4 Array_s1[5]	Netzwerknummer der Ziel-Station	Angabe der Netzwerknummer der Ziel-Station.	1 bis 239 254 ●[4]	Benutzer
(s1)+5 Array_s1[6]	Nummer der Ziel-Station	Angabe der Stationsnummer der Ziel-Station. (Der Bereich von "81H" bis "89H" kann nur bei zurückgesetztem Ausführungs-Modus adressiert werden).	Stationsnum-mer: 1 bis 64 Gruppen-adressierung: 81H bis 89H Alle Stationen im Zielnetzwerk: FFH	Benutzer
(s1)+6 Array_s1[7]	Dummy	Wird nicht verwendet.	—	—
(s1)+7 Array_s1[8]	Anzahl der Über-tragungsversuche	Angabe der Anzahl der Versuche, die unternom-men werden, um die vollständige Übertragung der Daten innerhalb der in (s1)+8 (Array_s1[9]) gespeicherten Überwachungszeit durchzuführen. Diese Option ist nur aktiv, wenn der Ausführungs-Modus ((s1)+0, Array_s1[1]) gesetzt (1) ist.	1 bis15	Benutzer
	Anzahl der durch-geführten Übertra-gungsversuche	Speicherung der durchgeführten Übertragungs-versuche.		System
(s1)+8 Array_s1[9]	Zeiteinstellung des WDT zur Empfangs-zeitüberwachung	Angabe der Überwachungszeit für die Operation in Sekunden. Erfolgt der Abschluss der Operation nicht innerhalb der angegebenen Zeit, wird die Übertragung mir der in (s1)+7 (Array_s1[8]) angegebenen Anzahl wiederholt.	1 bis 32767 0 = 10 s (fest eingestellt) Diese Option ist nur aktiv, wenn der Ausfühungs-modus gesetzt (1) ist	Benutzer
(s1)+9 Array_s1[10]	Empfangsdaten-länge	Angabe der Anzahl der zu sendenden Datenblöcke.	1 bis 480	Benutzer
(s1)+10 Array_s1[11]	Dummy	Wird nicht verwendet.	—	—
(s1)+11 Array_s1[12]	Flag der Uhr-Daten-speicherung (wird nur im Fehlerfall gesetzt)	Dieses Flag speichert den Status der in (s1)+0 (Array_s1[1]) angegebenen Speicheroption der Uhr-Daten: Keine Speicherung der Uhr-Daten = 0 Speicherung der Uhr-Daten = 1	—	System

Operand	Bedeutung	Funktion	Werte-bereich	Festlegung durch
(s1)+12 Array_s1[13]	Uhr-Daten (werden nur im Fehlerfall gesetzt)	Höchstwertiges Byte = Jahr (0 bis 99) Niedrigstwertiges Byte = Monat (1 bis 12)	—	System
(s1)+13 Array_s1[14]		Höchstwertiges Byte = Tag (1 bis 31) Niedrigstwertiges Byte = Stunde (0 bis 23)		
(s1)+14 Array_s1[15]		Höchstwertiges Byte = Minute (0 bis 59) Niedrigstwertiges Byte = Sekunde (0 bis 59)		
(s1)+15 Array_s1[16]		Höchstwertiges Byte = 00H Niedrigstwertiges Byte = Wochentag (0 bis 6) (Sonntag = 0, Samstag = 6)		
(s1)+16 Array_s1[17]	Nummer des Netzwerks, in dem der Fehler aufgetreten ist	Speicherung der Netzwerknummer der Station, in der der Fehler aufgetreten ist. Die Netzwerknummer liegt zwischen 1 und 239.	—	System
(s1)+17 Array_s1[18]	Nummer der Station, in der der Fehler aufgetreten ist	Speicherung der Nummer der Station, in der der Fehler aufgetreten ist. Die Speicherung wird nicht durchgeführt, wenn der Abschluss-Status der Anweisungsausführung "Kanal in Betrieb" lautet. Die Stationsnummer liegt zwischen 1 und 64.	—	System

●[3] Weitere Informationen über Fehlercodes enthält das Handbuch "MELSECNET/10 für QnA-Netzwerk-Systeme".

●[4] Die Netzwerknummer 254 ist ausgewählt, wenn für Jn der Wert 254 angegeben wird.

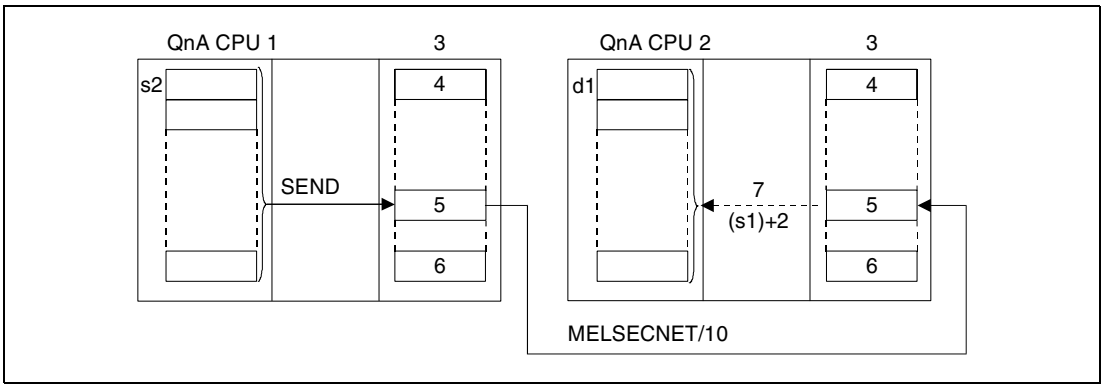
Funktions-weise

Senden von Daten an andere Stationen

SEND Sendeanweisung

Die SEND-Anweisung sendet die ab s2 gespeicherten Daten der HOST-Station an eine im MELSECNET/10 angeschlossene Station. Der Übertragungskanal ist in (s1)+2 angegeben. Die Stations- und Netzwerknummer sind in den Kontrolldaten angegeben.

Nach dem Abschluss der Schreib-Operation in der Ziel-Station wird der in d angegebene Operand gesetzt.



[1] HOST-Station

[2] Ziel-Station

[3] Netzwerkmodul

[4] Kanal 1

[5] Kanal n

[6] Kanal 8

[7] Die Lese-Operation wird mittels der RECV-Anweisung durchgeführt

Über eine Relais-Station und gesetzte Routing-Parameter kann auch auf Stationen in anderen Netzwerken zugegriffen werden.

Die Data-Link-Anweisungen können nicht von mehreren Stellen gleichzeitig mit gemeinsamem Zugriff auf die gleichen Kanäle ausgeführt werden. Bei gleichzeitiger Ausführung der Anweisungen von zwei oder mehr Stellen wird durch den Handshake-Betrieb der beiden aktiven Stationen eine Ausführung weiterer Data-Link-Anweisungen verhindert.

Der Ausführungs-Status und der Abschluss-Status (normal, nicht normal) der SEND-Anweisung können mittels

- dem Kommunikations-Kanal-Flag (●[5]) des verwendeten Kanals,
- dem Operanden, der nach Abschluss der Operation gesetzt wird (d) und
- der Status-Anzeige des Operationsabschlusses (Abschluss einer fehlerfreien oder fehlerhaften Übertragung) (d+1)

wie folgt überprüft werden:

Kommunikations-Kanal-Flag

Dieses Flag wird während der Ausführung der SEND-Anweisung gesetzt. Das Flag wird mit der Verarbeitung der END-Anweisung in dem Zyklus zurückgesetzt, in dem die Operation abgeschlossen wurde.

Operand der HOST-Station, der den Operationsabschluss anzeigt

Dieser Operand wird mit der Verarbeitung der END-Anweisung in dem Zyklus gesetzt, in dem die Operation abgeschlossen wurde. Bei der Verarbeitung der darauffolgenden END-Anweisung wird dieser Operand wieder zurückgesetzt.

Status-Anzeige des Operationsabschlusses

Dieser Operand wird in Abhängigkeit des Abschlussergebnisses der Operation gesetzt.

Beim Abschluss einer normalen (fehlerfreien) Übertragung bleibt der Operand unverändert zurückgesetzt.

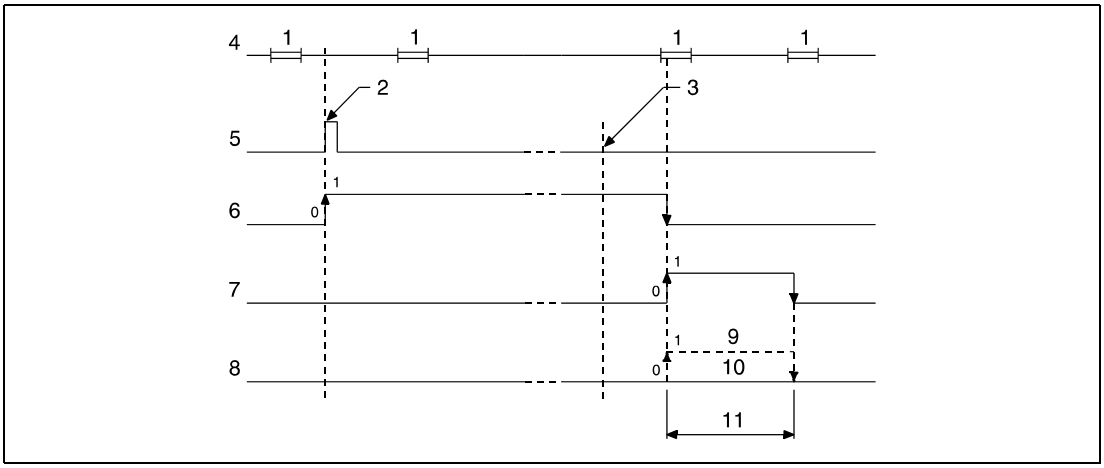
Beim Abschluss einer nicht normalen (fehlerhaften) Übertragung wird dieser Operand mit der Verarbeitung der END-Anweisung in dem Zyklus gesetzt, in dem die Verarbeitung der SEND-Anweisung abgeschlossen wurde. Bei der Verarbeitung der darauffolgenden END-Anweisung wird dieser Operand wieder zurückgesetzt.

HINWEIS

●[5] *Die folgende Tabelle ordnet den Kanalnummern die entsprechenden Kommunikations-Kanal-Flags zu.*

Kanalnummer	1	2	3	4	5	6	7	8
Kommunikations-Kanal-Flag	SB30	SB32	SB34	SB36	SB38	SB3A	SB3C	SB3E

Die folgende Abbildung zeigt die Operationen der HOST-Station während der Ausführung der SEND-Anweisung.



1 END-Verarbeitung
2 Ausführung der SEND-Anweisung
3 Abschluss der Operation
4 Programm der HOST-Station
5 SEND-Anweisung
6 Kommunikations-Kanal-Flag
7 Operand der HOST-Station, der nach Abschluss der Operation gesetzt wird (d)
8 Status-Anzeige des Operationsabschlusses (d+1)
9 Abschluss einer nicht normalen (fehlerhaften) Übertragung
10 Abschluss einer normalen (fehlerfreien) Übertragung
11 Ein Zyklus

Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die Kontrolldateninhalte liegen nicht im gültigen Wertebereich (Fehlercode 4100).
- Das Netzwerk mit der mit Jn angegebenen Nummer ist nicht mit der Station verbunden (Fehlercode 4102).
- Das Modul an der mit Un angegebenen Ein-/Ausgangsadresse ist kein Netzwerkmodul (Fehlercode 2111).

Beispiel JP.SEND

Das folgende Programm sendet mit positiver Flanke von M10 Daten der HOST-Station zu einer Ziel-Station. Die Ausführung der SEND-Anweisung ist über den Öffner des Flags SB34 verriegelt. Nähere Angaben zur HOST- und Ziel-Station und den verwendeten MOV-Anweisungen enthält die folgende Tabelle.

Operand/Anweisung	Bezeichnung/Funktion
HOST-Station	–
HOST-Netzwerk	7
HOST-Kanal	3
Kommunikations-Kanal-Flag	SB34
Ziel-Station	15
Ziel-Netzwerk	5
Ziel-Kanal	5
1. MOV-Anweisung	Setzen der Eingangsbedingung und der Uhr-Daten
2. MOV-Anweisung	Setzen des Kanals der HOST-Station
3. MOV-Anweisung	Setzen des Kanals der Ziel-Station
4. MOV-Anweisung	Setzen der Netzwerknummer der Ziel-Station
5. MOV-Anweisung	Setzen der Nummer der Ziel-Station
6. MOV-Anweisung	–
7. MOV-Anweisung	Setzen der Anzahl der Übertragungsversuche
8. MOV-Anweisung	Setzen der WDT-Zeiteinstellung (20 s)
9. MOV-Anweisung	Setzen der Anzahl der zu sendenden Blöcke (4)
10. MOV-Anweisung	Setzen der zu sendenden Daten
11. MOV-Anweisung	
12. MOV-Anweisung	
13. MOV-Anweisung	

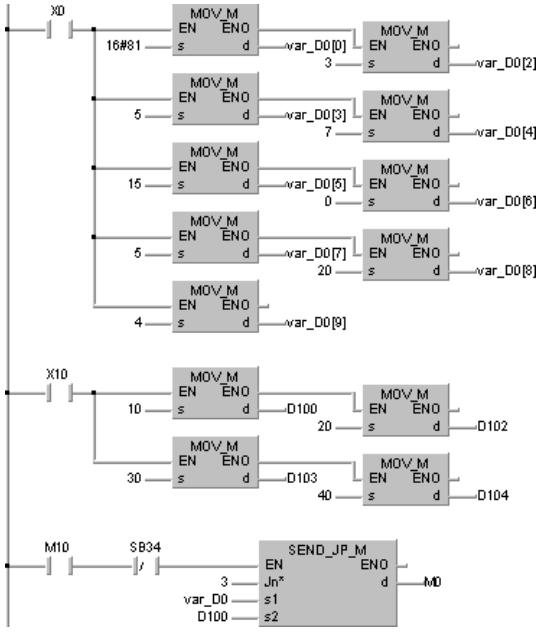
MELSEC-Anweisungsliste

```
MELSEC  LD      X0
        MOV     H81
                D0
        MOV     K3
                D2
        MOV     K5
                D3
        MOV     K7
                D4
        MOV     K15
                D5
        MOV     K0
                D6
        MOV     K5
                D7
        MOV     K20
                D8
        MOV     K4
                D9
        LD      X10
        MOV     K10
                D100
        MOV     K20
                D102
        MOV     K30
                D103
        MOV     K40
                D104
        LD      M10
        ANI     SB34
        JP.SEND J3
                D0
                D100
                M0
```

IEC-Anweisungsliste

```
LD          X0
MOV_M       H81 , var_D0[0]
MOV_M       K3 , var_D0[2]
MOV_M       K5 , var_D0[3]
MOV_M       K7 , var_D0[4]
MOV_M       K15 , var_D0[5]
MOV_M       K0 , var_D0[6]
MOV_M       K5 , var_D0[7]
MOV_M       K20 , var_D0[8]
MOV_M       K4 , var_D0[9]
LD          X10
MOV_M       K10 , D100
MOV_M       K20 , D102
MOV_M       K30 , D103
MOV_M       K40 , D104
LD          M10
ANDN        SB34
SEND_JP_M   K3 , var_D0 , D100 , M0
```

Kontaktplan



HINWEIS *Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- und Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

8.6.6 RECV

CPU

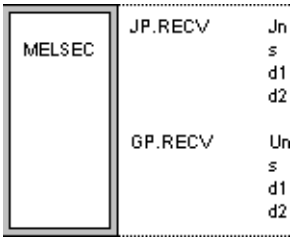
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	

Operanden MELSEC Q

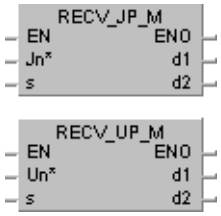
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	—	—	—	—	—	SM0	8
d1	—	●	●	—	—	—	—	—	—		
d2	●	●	●	—	—	—	—	—	—		

GX IEC Developer

MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste

```
RECV_JP_M   Jn , s , d1 , d2
RECV_UP_M   Un , s , d1 , d2
```

Variablen

Operand	Befehlswert	Datentyp	
		MELSEC	IEC
Jn	Netzwerknummer der HOST-Station. ●[1]	BIN-16-Bit	ANY16
Un	Kopf-Ein-/Ausgangsadresse des Netzwerkmoduls der HOST-Station. ●[2]		
s	Erste Adresse des Operanden, in dem die Kontrolldaten gespeichert sind.	Adresse	Array [1..16] of ANY16
d1	Erste Adresse des Operanden, in dem die gesendeten Daten gespeichert werden.		ANY16
d2	Operand, der nach vollständiger Verarbeitung der Anweisung für einen Programmzyklus gesetzt wird.	Bit	BOOL

HINWEISE

●[1] *Die Netzwerknummer der HOST-Station muss zwischen 1 und 239 liegen. Das Netzwerk mit der Nummer 254 wird über Einstellungen so konfiguriert, dass andere Stationen auf die aktive Station zugreifen können.*

●[2] *Die Kopf-Ein-/Ausgangsadresse des Netzwerkmoduls der HOST-Station muss zwischen 0 und FE$_H$ liegen. Es ist zu beachten, dass der Compiler für Un eine hexadezimale Adressangabe erwartet. Eine Dezimalangabe wird automatisch in eine Hexadezimalzahl umgewandelt.*

Operandenübersicht der Kontrolldaten

Operand	Bedeutung	Funktion	Werte-bereich	Festlegung durch
(s1)+0 Array_s1[1]	Ausführungs-Modus	Warten auf Daten (in diesem Modus wird wiederkehrend mit einer fest eingestellten Zeit auf Daten gewartet): (Bit 0 (b0) = 0, fest eingestellt)	0000H 0080H	Benutzer
	Abschluss-Modus der Fehlerverarbeitung	Speichern der Uhr-Daten des Zeitpunktes, an dem die Fehlerverabeitung abgeschlossen ist: - Keine Speicherung der Uhr-Daten, Bit 7 (b7) = 0 - Speicherung der Uhr-Daten, Bit 7 (b7) = 1 (Uhr-Daten ab (s1)+11 (Array_s1[12]) aufwärts)		
(s1)+1 Array_s1[2]	Abschluss-Status der Anweisungs-verarbeitung	Statusanzeige nach Abschluss der Anweisungs-verarbeitung: - 0 = Keine Fehler (normaler Verarbeitungs-abschluss) - ungleich 0 = Fehlercode ●3	—	System
(s1)+2 Array_s1[3]	Von der HOST-Station verwendeter Kanal	Angabe des Kanals, in welchem die zu empfangenden Daten gespeichert sind	1 bis 8	Benutzer
(s1)+3 Array_s1[4]	Von den Ziel-Stationen verwendete Känale	Speicherung des von der sendenden Station verwendeten Kanals.	1 bis 8	System
(s1)+4 Array_s1[5]	Netzwerk-Nummer der Ziel-Station	Speicherung der Netzwerknummer der sendenden Station.	1 bis 239	System
(s1)+5 Array_s1[6]	Nummer der Ziel-Station	Speicherung der Stationsnummer der sendenden Station.	1 bis 64	System
(s1)+6 Array_s1[7]	Dummy	Wird nicht verwendet.	—	—
(s1)+7 Array_s1[8]	Dummy	Wird nicht verwendet.	—	—
(s1)+8 Array_s1[9]	Zeiteinstellung des WDT zur Empfangs-zeitüberwachung	Angabe der Überwachungszeit für die Operation in Sekunden.	1 bis 32767 0 = 10 s (fest eingestellt) Diese Option ist nur aktiv, wenn der Ausfühungs-modus gesetzt (1) ist	Benutzer
(s1)+9 Array_s1[10]	Empfangsdaten-länge	Speicherung der Anzahl der empfangenen Datenblöcke.	1 bis 480	System
(s1)+10 Array_s1[11]	Dummy	Wird nicht verwendet.	—	—
(s1)+11 Array_s1[12]	Flag der Uhr-Daten-speicherung (wird nur im Fehlerfall gesetzt)	Dieses Flag speichert den Status der in (s1)+0 (Array_s1[1]) angegebenen Speicheroption der Uhr-Daten: Keine Speicherung der Uhr-Daten = 0 Speicherung der Uhr-Daten = 1	—	System
(s1)+12 Array_s1[13]	Uhr-Daten (werden nur im Fehlerfall gesetzt)	Höchstwertiges Byte = Jahr (0 bis 99) Niedrigstwertiges Byte = Monat (1 bis 12)	—	System
(s1)+13 Array_s1[14]		Höchstwertiges Byte = Tag (1 bis 31) Niedrigstwertiges Byte = Stunde (0 bis 23)		
(s1)+14 Array_s1[15]		Höchstwertiges Byte = Minute (0 bis 59) Niedrigstwertiges Byte = Sekunde (0 bis 59)		
(s1)+15 Array_s1[16]		Höchstwertiges Byte = 00H Niedrigstwertiges Byte = Wochentag (0 bis 6) (Sonntag = 0, Samstag = 6)		

●3 Weitere Informationen über Fehlercodes enthält das Handbuch "MELSECNET/10 für QnA-Netzwerk-Systeme".

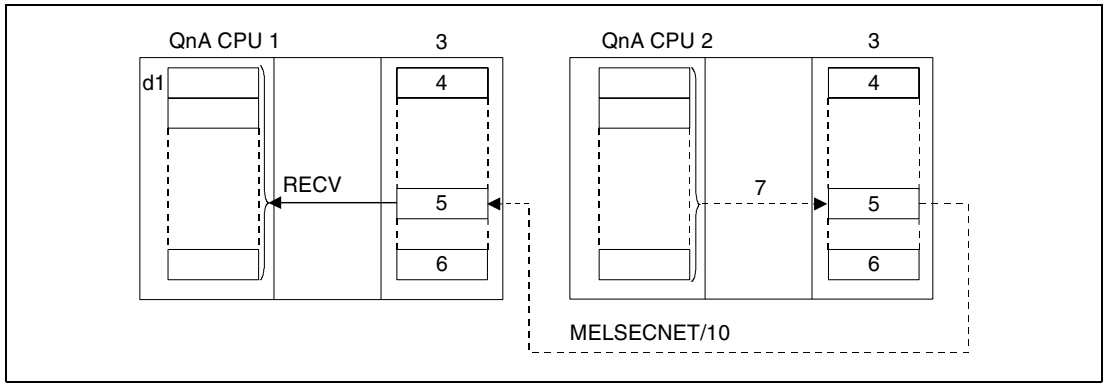
Funktions-weise

Empfangen von gesendeten Daten anderer Stationen

RECV Empfangsanweisung

Die RECV-Anweisung empfängt die mittels SEND-Anweisung gesendeten Daten einer im MELSECNET/10 angeschlossenen Station. Die Stations- und Netzwerknummer sind in den Kontrolldaten angegeben. Die Daten werden ab d1 gespeichert.

Nach dem Abschluss der Operation wird der in d2 angegebene Operand gesetzt.



[1] HOST-Station

[2] Ziel-Station

[3] Netzwerkmodul

[4] Kanal 1

[5] Kanal n

[6] Kanal 8

[7] Die Schreib-Operation wird mittels der SEND-Anweisung durchgeführt

Über eine Relais-Station und gesetzte Routing-Parameter kann auch auf Stationen in anderen Netzwerken zugegriffen werden.

Die Data-Link-Anweisungen können nicht von mehreren Stellen gleichzeitig mit gemeinsamem Zugriff auf die gleichen Kanäle ausgeführt werden. Bei gleichzeitiger Ausführung der Anweisungen von zwei oder mehr Stellen wird durch den Handshake-Betrieb der beiden aktiven Stationen eine Ausführung weiterer Data-Link-Anweisungen verhindert.

Der Ausführungs-Status und der Abschluss-Status (normal, nicht normal) der RECV-Anweisung können mittels

- dem Kommunikations-Kanal-Flag (●[4]) des verwendeten Kanals,
- dem Operanden, der nach Abschluss der Operation gesetzt wird (d2) und
- der Status-Anzeige des Operationsabschlusses (Abschluss einer fehlerfreien oder fehlerhaften Übertragung) ((d2)+1)

wie folgt überprüft werden:

Kommunikations-Kanal-Flag

Dieses Flag wird während der Ausführung der RECV-Anweisung gesetzt. Das Flag wird mit der Verarbeitung der END-Anweisung in dem Zyklus zurückgesetzt, in dem die Lese-Operation abgeschlossen wurde.

Operand der HOST-Station, der den Operationsabschluss anzeigt

Dieser Operand wird mit der Verarbeitung der END-Anweisung in dem Zyklus gesetzt, in dem die Operation abgeschlossen wurde. Bei der Verarbeitung der darauffolgenden END-Anweisung wird dieser Operand wieder zurückgesetzt.

Status-Anzeige des Operationsabschlusses

Dieser Operand wird in Abhängigkeit des Abschlussergebnisses der Operation gesetzt.

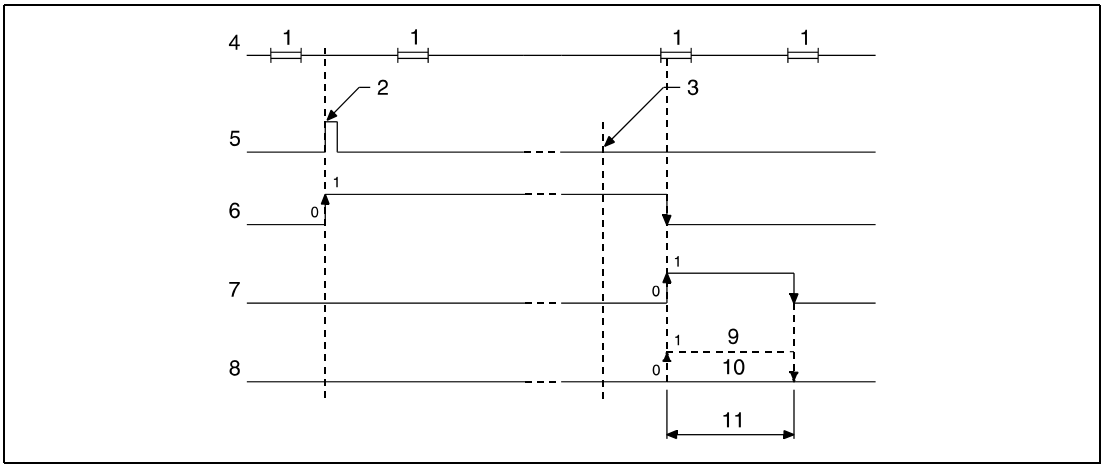
Bei dem Abschluss einer normalen (fehlerfreien) Übertragung bleibt der Operand unverändert zurückgesetzt.

Bei dem Abschluss einer nicht normalen (fehlerhaften) Übertragung wird dieser Operand mit der Verarbeitung der END-Anweisung in dem Zyklus gesetzt, in dem die Verarbeitung der RECV-Anweisung abgeschlossen wurde. Bei der Verarbeitung der darauffolgenden END-Anweisung wird dieser Operand wieder zurückgesetzt.

HINWEIS ●[4] *Die folgende Tabelle ordnet den Kanalnummern die entsprechenden Kommunikations-Kanal-Flags zu.*

Kanalnummer	1	2	3	4	5	6	7	8
Kommunikations-Kanal-Flag	SB30	SB32	SB34	SB36	SB38	SB3A	SB3C	SB3E

Die folgende Abbildung zeigt die Operationen der HOST-Station während der Ausführung der RECV-Anweisung.



[1] END-Verarbeitung
[2] Ausführung der RECV-Anweisung
[3] Abschluss der Operation
[4] Programm der HOST-Station
[5] RECV-Anweisung
[6] Kommunikations-Kanal-Flag
[7] Operand der HOST-Station, der nach Abschluss der Operation gesetzt wird (d2)
[8] Status-Anzeige des Operationsabschlusses ((d2)+1)
[9] Abschluss einer nicht normalen (fehlerhaften) Übertragung
[10] Abschluss einer normalen (fehlerfreien) Übertragung
[11] Ein Zyklus

Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die Kontrolldateninhalte liegen nicht im gültigen Wertebereich (Fehlercode 4100).
- Das Netzwerk mit der mit Jn angegebenen Nummer ist nicht mit der Station verbunden (Fehlercode 4102).
- Das Modul an der mit Un angegebenen Ein-/Ausgangsadresse ist kein Netzwerkmodul (Fehlercode 2111).

Beispiel

JP.RECV

Das folgende Programm liest mit positiver Flanke von X10 die mittels der SEND-Anweisung gesendeten Daten einer Station. Die Ausführung der RECV-Anweisung ist über den Öffner des Flags SB34 verriegelt. Nähere Angaben zur HOST- und Sende-Station und den verwendeten MOV-Anweisungen enthält die folgende Tabelle.

Operand/Anweisung	Bezeichnung/Funktion
HOST-Station	–
HOST-Netzwerk	–
HOST-Kanal	3
Kommunikations-Kanal-Flag	SB34
Sende-Station	–
Netzwerk der Sende-Station	3
Kanal der Sende-Station	3
1. MOV-Anweisung	Setzen der Uhr-Daten
2. MOV-Anweisung	Setzen des Kanals der HOST-Station
3. MOV-Anweisung	Setzen der WDT-Zeiteinstellung (20 s)

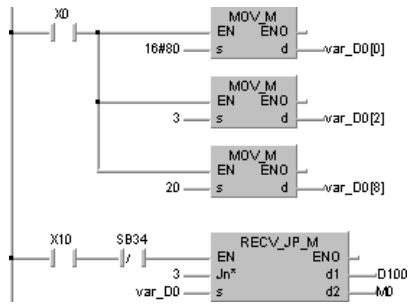
MELSEC-Anweisungsliste

```
MELSEC  LD      X0
        MOV     H80
                D0
        MOV     K3
                D2
        MOV     K20
                D8
        LD      X10
        ANI     SB34
        JP.RECV J3
                D0
                D100
                M0
```

Kontaktplan

```
        LD      X0
        MOV_M   16#80 , var_D0[0]
        MOV_M   3 , var_D0[2]
        MOV_M   20 , var_D0[8]
        LD      X10
        ANDN    SB34
        RECV_JP_M  3 , var_D0 , D100 , M0
```

IEC-Anweisungsliste



HINWEIS

Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- und Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.

8.6.7 REQ

CPU

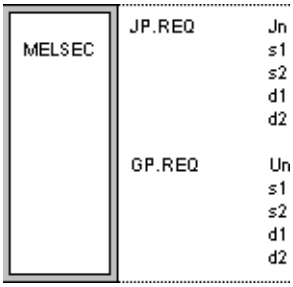
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	

Operanden MELSEC Q

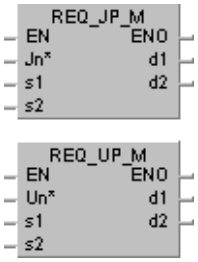
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	—	●	●	—	—	—	—	—	—	SM0	8
s2	—	●	●	—	—	—	—	—	—		
d1	—	●	●	—	—	—	—	—	—		
d2	●	●	●	—	—	—	—	—	—		

GX IEC Developer

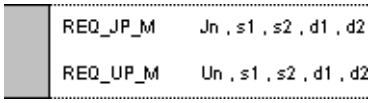
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



Variablen

Operand	Befehlswert	Datentyp	
		MELSEC	IEC
Jn	Netzwerknummer der HOST-Station. ●[1]	BIN-16-Bit	ANY16
Un	Kopf-Ein-/Ausgangsadresse des Netzwerkmoduls der HOST-Station. ●[2]		
s1	Erste Adresse des Operanden, in dem die Kontrolldaten gespeichert sind.	Adresse	Array [1..18] of ANY16
s2	Erste Adresse des Operanden, in dem die angeforderten Daten gespeichert sind.		Array [1..7] of ANY16
d1	Erste Adresse des Operanden, in dem die Antwort-Daten gespeichert werden.		Array [1..4] of ANY16
d2	Operand, der nach vollständiger Verarbeitung der Anweisung für einen Programmzyklus gesetzt wird.	Bit	BOOL

HINWEISE

- *[1] Die Netzwerknummer der HOST-Station muss zwischen 1 und 239 liegen. Das Netzwerk mit der Nummer 254 wird über Einstellungen so konfiguriert, dass andere Stationen auf die aktive Station zugreifen können.*
- *[2] Die Kopf-Ein-/Ausgangsadresse des Netzwerkmoduls der HOST-Station muss zwischen 0 und FE$_H$ liegen. Es ist zu beachten, dass der Compiler für Un eine hexadezimale Adressangabe erwartet. Eine Dezimalangabe wird automatisch in eine Hexadezimalzahl umgewandelt.*

Die REQ-Anweisung kann nur ausgeführt werden, wenn sich in der Ziel-Station eine QnA-CPU befindet.

Die REQ-Anweisung kann bei der Verwendung einer ACPU im MELSECNET/10 nicht angewendet werden.

Es können nur Stationsnummern für QnA-CPUs als Nummer der Ziel-Station angegeben werden.

Operandenübersicht der Kontrolldaten

Operand	Bedeutung	Funktion	Werte-bereich	Festlegung durch
(s1)+0 Array_s1[1]	Ausführungs-Modus	Die Eingangsbestätigung ist gesetzt: (Bit 0 (b0) = 1, fest eingestellt)	0001H 0081H	Benutzer
	Abschluss-Modus der Fehlerverar-beitung	Speichern der Uhr-Daten des Zeitpunktes, an dem die Fehlerverabeitung abgeschlossen ist: - Keine Speicherung der Uhr-Daten, Bit 7 (b7) = 0 - Speicherung der Uhr-Daten, Bit 7 (b7) = 1 (Uhr-Daten ab (s1)+11 (Array_s1[12]) aufwärts)		
(s1)+1 Array_s1[2]	Abschluss-Status der Anweisungs-verarbeitung	Statusanzeige nach Abschluss der Anweisungs-verarbeitung: - 0 = Keine Fehler (normaler Verarbeitungs-abschluss) - ungleich 0 = Fehlercode ●[1]	—	System
(s1)+2 Array_s1[3]	Von der HOST-Station verwendeter Kanal	Angabe des Kanals, den die HOST-Station verwenden wird.	1 bis 8	Benutzer
(s1)+3 Array_s1[4]	Dummy	Wird nicht verwendet.	—	—
(s1)+4 Array_s1[5]	Netzwerknummer der Ziel-Station	Angabe der Netzwerknummer der Station, aus der gelesen wird.	1 bis 239 254 ●[2]	Benutzer
(s1)+5 Array_s1[6]	Nummer der Ziel-Station	Angabe der Stationsnummer der Ziel-Station.	1 bis 64	Benutzer
(s1)+6 Array_s1[7]	Dummy	Wird nicht verwendet.	—	—
(s1)+7 Array_s1[8]	Anzahl der Über-tragungsversuche	Angabe der Anzahl der Versuche, die unternom-men werden, um die vollständige Verarbeitung der REQ-Anweisung innerhalb der in (s1)+8 (Array_s1[9]) gespeicherten Überwachungszeit durchzuführen.	1 bis15	Benutzer
	Anzahl der durch-geführten Übertra-gungsversuche	Speicherung der durchgeführten Übertragungs-versuche.	—	System
(s1)+8 Array_s1[9]	Zeiteinstellung des WDT zur Empfangs-zeitüberwachung	Angabe der Überwachungszeit für die Operation in Sekunden. Erfolgt der Abschluss der Operation nicht innerhalb der angegebenen Zeit, wird die Übertragung mir der in (s1)+7 (Array_s1[8]) angegebenen Anzahl wiederholt.	1 bis 32767 0 = 10 Sekunden (fest eingestellt)	Benutzer
(s1)+9 Array_s1[10]	Länge der angeforderten Daten	Angabe der Länge der angeforderten Daten. Wenn Uhr-Daten gelesen werden = 2 Wenn Uhr-Daten geschrieben werden = 7 RUN-/STOP eines ausgelagerten Moduls = 4	2, 7, 4	Benutzer
(s1)+10 Array_s1[11]	Länge der Antwort-Daten	Speicherung der Länge der Antwort-Daten. Wenn Uhr-Daten gelesen werden = 2	0, 4	Benutzer
(s1)+11 Array_s1[12]	Flag der Uhr-Daten-speicherung (wird nur im Fehlerfall gesetzt)	Dieses Flag speichert den Status der in (s1)+0 (Array_s1[1]) angegebenen Speicheroption der Uhr-Daten: Keine Speicherung der Uhr-Daten = 0 Speicherung der Uhr-Daten = 1	—	System
(s1)+12 Array_s1[13]	Uhr-Daten (werden nur im Fehlerfall gesetzt)	Höchstwertiges Byte = Jahr (0 bis 99) Niedrigstwertiges Byte = Monat (1 bis 12)	—	System
(s1)+13 Array_s1[14]		Höchstwertiges Byte = Tag (1 bis 31) Niedrigstwertiges Byte = Stunde (0 bis 23)		
(s1)+14 Array_s1[15]		Höchstwertiges Byte = Minute (0 bis 59) Niedrigstwertiges Byte = Sekunde (0 bis 59)		
(s1)+15 Array_s1[16]		Höchstwertiges Byte = 00H Niedrigstwertiges Byte = Wochentag (0 bis 6) (Sonntag = 0, Samstag = 6)		

Operand	Bedeutung	Funktion	Werte-bereich	Festlegung durch
(s1)+16 Array_s1[17]	Nummer des Netzwerks, in dem der Fehler aufgetreten ist	Speicherung der Nummer des Netzwerks der Station, in der der Fehler aufgetreten ist. Die Speicherung wird nicht durchgeführt, wenn der Abschluss-Status der Anweisungsausführung "Kanal in Betrieb (F7C1H)" lautet. Die Netzwerknummer liegt zwischen 1 und 239.	—	System
(s1)+17 Array_s1[18]	Nummer der Station, in der der Fehler aufgetreten ist	Speicherung der Nummer der Station, in der der Fehler aufgetreten ist. Die Speicherung wird nicht durchgeführt, wenn der Abschluss-Status der Anweisungsausführung "Kanal in Betrieb (F7C1H)" lautet. Die Stationsnummer liegt zwischen 1 und 64.	—	System

●[1] Weitere Informationen über Fehlercodes enthält das Handbuch "MELSECNET/10 für QnA-Netzwerk-Systeme".

●[2] Die Netzwerknummer 254 ist ausgewählt, wenn für Jn der Wert 254 angegeben wird.

Anforderungs-/Antwort-Daten während der Schreib-/Lese-Operation von Uhr-Daten

Anforderungs-Daten

Operand	Bedeutung	Funktion	Lesen von Uhr-Daten	Schreiben von Uhr-Daten
(s2)+0 Array-s2[1]	Anforderungs-Modus	0001H = Lesen von Uhr-Daten 0011H = Schreiben von Uhr-Daten	●	●
(s2)+1 Array_s2[2]	Anforderungs-Modus der Unterroutine	0002H = Lesen von Uhr-Daten 0001H = Schreiben von Uhr-Daten	●	●
(s2)+2 Array_s2[3]	Aktualisierungs-Schema	Angabe, welche Uhr-Daten in (s2)+3 (Array_s2[4]) bis (s2)+6 (Array_s2[7]) aktualisiert werden. Ist der Operand gesetzt (1), wird dieses Uhr-Datum aktualisiert. b15 - - - - - - - - - - b7 b6 b5 b4 b3 b2 b1 b0 0 | | 0 | W | Sec | Min | H | D | M | Y		●
(s2)+3 Array_s2[4]	Zu aktualisierende Monats- und Jahresangabe	Im BCD-Code gespeicherte Monats- und Jahresangabe (letzten 2 Stellen). b15 - - - - - - - - - - b8 b7 - - - - - - - - - - - b0 M (01H - 12H) | Y (00H - 99H)		●
(s2)+4 Array_s2[5]	Zu aktualisierende Stunden- und Tagesangabe	Im BCD-Code gespeicherte Stunden- und Tagesangabe. b15 - - - - - - - - - - b8 b7 - - - - - - - - - - - b0 H (00H - 23H) | D (01H - 31H)		●
(s2)+5 Array_s2[6]	Zu aktualisierende Minuten- und Sekundenangabe	Im BCD-Code gespeicherte Sekunden- und Minutenangabe. b15 - - - - - - - - - - b8 b7 - - - - - - - - - - - b0 Sec (00H - 59H) | Min (00H - 59H)		●
(s2)+6 Array_s2[7]	Zu aktualisierende Angabe des Wochentags	Im BCD-Code gespeicherte Angabe des Wochentags (00H = Sontag, 06H = Samstag). b15 - - - - - - - - - - b8 b7 - - - - - - - - - - - b0 00H | W (00H - 06H)		●

M = Monat
Y = Jahr
H = Stunde
D = Tag
Sec = Sekunde
Min = Minute
W = Wochentag

Antwort-Daten

Operand	Bedeutung	Funktion	Lesen von Uhr-Daten	Schreiben von Uhr-Daten
(d1)+0 Array-d1[1]	Gelesene Monats- und Jahresangabe	Im BCD-Code gespeicherte Monats- und Jahresangabe (letzten 2 Stellen). b15 ---------- b8 b7 ---------- b0 M (01H - 12H) | Y (00H - 99H)	●	
(d1)+1 Array_d1[2]	Gelesene Stunden- und Tagesangabe	Im BCD-Code gespeicherte Stunden- und Tagesangabe. b15 ---------- b8 b7 ---------- b0 H (00H - 23H) | D (01H - 31H)	●	
(d1)+2 Array_d1[3]	Gelesene Minuten- und Sekundenangabe	Im BCD-Code gespeicherte Sekunden- und Minutenangabe. b15 ---------- b8 b7 ---------- b0 Sec (00H - 59H) | Min (00H - 59H)	●	
(d1)+3 Array_d1[4]	Gelesene Angabe des Wochentags	Im BCD-Code gespeicherte Angabe des Wochentags (00H = Sontag, 06H = Samstag). b15 ---------- b8 b7 ---------- b0 00H | W (00H - 06H)	●	

M = Monat
Y = Jahr
H = Stunde
D = Tag
Sec = Sekunde
Min = Minute
W = Wochentag

HINWEIS *Schreib-/Lese-Operationen sind nicht möglich, wenn der Schreibschutz an der CPU der Ziel-Station aktiviert ist (System-Schalter 1, SW5 (QnA, Q4AR) bzw. SW1 (QnAS) ist eingeschaltet).*

Anforderungs-Daten während der RUN-/STOP-Operation an einer anderen Station

Operand	Bedeutung	Funktion	RUN-Operation	STOP-Operation
(s2)+0 Array-s2[1]	Anforderungs-Modus	0010H	●	●
(s2)+1 Array_s2[2]	Anforderungs-Modus der Unterroutine	0001H = RUN-Operation an einer anderen Station 0002H = STOP-Operation an einer anderen Station	●	●
(s2)+2 Array_s2[3]	Modus	Angabe, ob eine RUN-Operation zwingend an einer anderen Station ausgeführt werden soll. 0001H = Keine zwingende Ausführung der RUN-Operation 0003H = Zwingende Ausführung der RUN-Operation (wird während der STOP-Operation an der anderen Station gesetzt) Wenn die Station, die an einer anderen Station eine STOP-Operation ausführt, keine RUN-Operation ausführen kann, ist die Ausführung der RUN-Operation von einer anderen Station möglich.	●	●
(s2)+3 Array_s2[4]	Lösch-Modus	Angabe des Speicher-Status der CPU während der Ausführung der RUN-Operation an einer anderen Station. 0000H = Kein Löschen (wird während der STOP-Operation an der anderen Station gesetzt) 0001H = Löschen (ausschließlich der remanenten Bereiche) 0002H = Löschen (einschließlich der remanenten Bereiche)	●	●

HINWEISE *Die RUN-/STOP-Funktion an einer anderen Station kann nur dann ausgeführt werden, wenn der RUN-/STOP-Schalter der QnA-CPU der Ziel-Station auf RUN geschaltet ist.*

Schreib-/Lese-Operationen sind nicht möglich, wenn der Schreibschutz an der QnA-CPU der Ziel-Station aktiviert ist (System-Schalter 1, SW5 (QnA, Q4AR) bzw. SW1 (QnAS) ist eingeschaltet).

Wenn die Ziel-Station bereits durch eine andere Station in den STOP-/PAUSE-Modus umgeschaltet wurde, kann die RUN-Operation nur dann zwingend ausgeführt werden, wenn der Modus in (s2)+2 auf "Keine zwingende Ausführung der RUN-Operation (0001H)" gesetzt ist.

Wird die QnA-CPU der Ziel-Station, die die RUN-/STOP-Operation ausführt, zurückgesetzt, gehen die Informationen der RUN-/STOP-Operation in der Ziel-Station verloren.

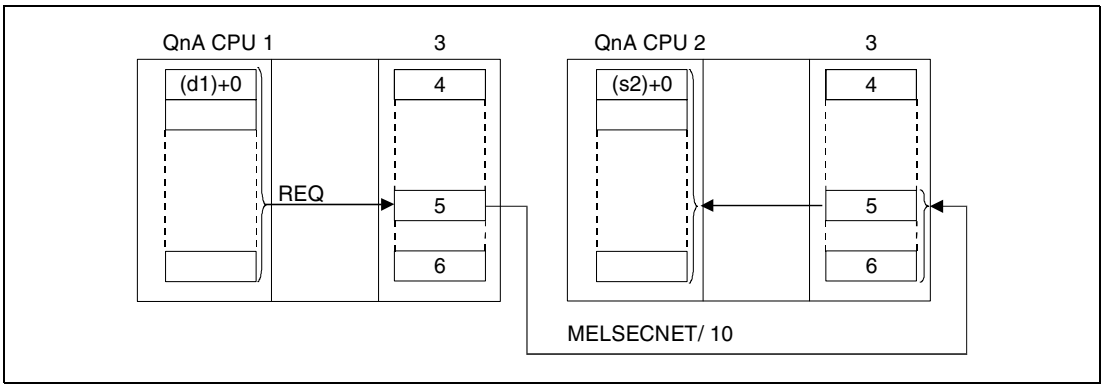
Funktions-weise

Datenanforderung an andere Stationen

REQ Anforderungsanweisung

Die REQ-Anweisung überträgt die ab (d1)+0 (Array _d1[1]) gespeicherten, angeforderten Daten einer im MELSECNET/10 angeschlossenen Station. Die Stations- und Netzwerknummer sind in den Kontrolldaten angegeben. Die Daten werden ab (s2)+0 (Array_s2[1]) gespeichert.

Nach dem Abschluss der Übertragungs-Operation der angeforderten Daten in der Ziel-Station wird der in d2 angegebene Operand gesetzt.



[1] HOST-Station

[2] Ziel-Station

[3] Netzwerkmodul

[4] Kanal 1

[5] Kanal n

[6] Kanal 8

Über eine Relais-Station und gesetzte Routing-Parameter kann auch auf Stationen in anderen Netzwerken zugegriffen werden.

Die Data-Link-Anweisungen können nicht von mehreren Stellen gleichzeitig mit gemeinsamem Zugriff auf die gleichen Kanäle ausgeführt werden. Bei gleichzeitiger Ausführung der Anweisungen von zwei oder mehr Stellen wird durch den Handshake-Betrieb der beiden aktiven Stationen eine Ausführung weiterer Data-Link-Anweisungen verhindert.

Der Ausführungs-Status und der Abschluss-Status (normal, nicht normal) der REQ-Anweisung können mittels

- dem Kommunikations-Kanal-Flag (●[3]) des verwendeten Kanals,
- dem Operanden, der nach Abschluss der Operation gesetzt wird (d2) und
- der Status-Anzeige des Operationsabschlusses (Abschluss einer fehlerfreien oder fehlerhaften Übertragung) ((d2)+1)

wie folgt überprüft werden:

Kommunikations-Kanal-Flag

Dieses Flag wird während der Ausführung der REQ-Anweisung gesetzt. Das Flag wird mit der Verarbeitung der END-Anweisung in dem Zyklus zurückgesetzt, in dem die Übertragungs-Operation abgeschlossen wurde.

Operand der HOST-Station, der den Operationsabschluss anzeigt

Dieser Operand wird mit der Verarbeitung der END-Anweisung in dem Zyklus gesetzt, in dem die Operation abgeschlossen wurde. Bei der Verarbeitung der darauffolgenden END-Anweisung wird dieser Operand wieder zurückgesetzt.

Status-Anzeige des Operationsabschlusses

Dieser Operand wird in Abhängigkeit des Abschlussergebnisses der Operation gesetzt.

Bei dem Abschluss einer normalen (fehlerfreien) Übertragung bleibt der Operand unverändert zurückgesetzt.

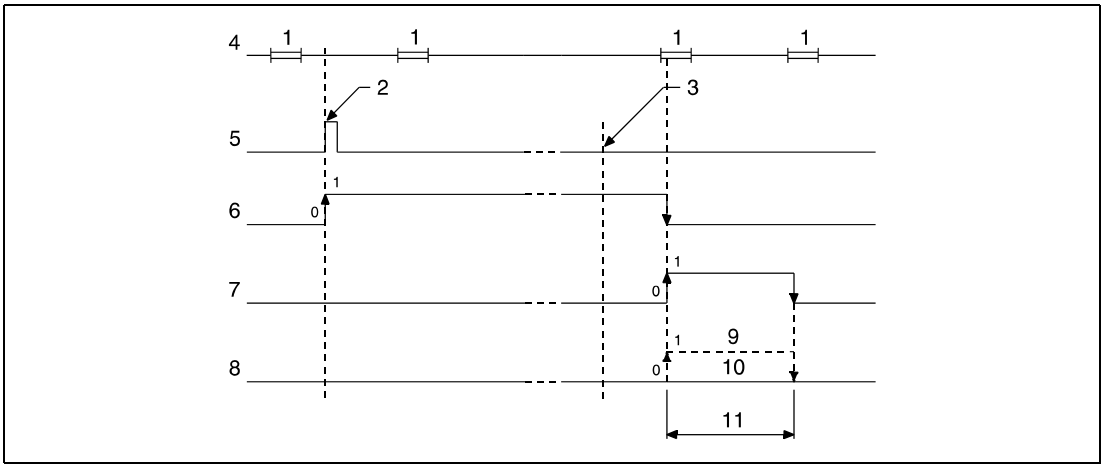
Bei dem Abschluss einer nicht normalen (fehlerhaften) Übertragung wird dieser Operand mit der Verarbeitung der END-Anweisung in dem Zyklus gesetzt, in dem die Verarbeitung der REQ-Anweisung abgeschlossen wurde. Bei der Verarbeitung der darauffolgenden END-Anweisung wird dieser Operand wieder zurückgesetzt.

HINWEIS

●[3] *Die folgende Tabelle ordnet den Kanalnummern die entsprechenden Kommunikations-Kanal-Flags zu.*

Kanalnummer	1	2	3	4	5	6	7	8
Kommunikations-Kanal-Flag	SB30	SB32	SB34	SB36	SB38	SB3A	SB3C	SB3E

Die folgende Abbildung zeigt die Operationen der HOST-Station während der Ausführung der REQ-Anweisung.



1 END-Verarbeitung
2 Ausführung der REQ-Anweisung
3 Abschluss der Operation
4 Programm der HOST-Station
5 REQ-Anweisung
6 Kommunikations-Kanal-Flag
7 Operand der HOST-Station, der nach Abschluss der Operation gesetzt wird (d2)
8 Status-Anzeige des Operationsabschlusses ((d2)+1)
9 Abschluss einer nicht normalen (fehlerhaften) Übertragung
10 Abschluss einer normalen (fehlerfreien) Übertragung
11 Ein Zyklus

Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die Kontrolldateninhalte liegen nicht im gültigen Wertebereich (Fehlercode 4100).
- Das Netzwerk mit der mit Jn angegebenen Nummer ist nicht mit der Station verbunden (Fehlercode 4102).
- Das Modul an der mit Un angegebenen Ein-/Ausgangsadresse ist kein Netzwerkmodul (Fehlercode 2111).

Beispiel JP.REQ

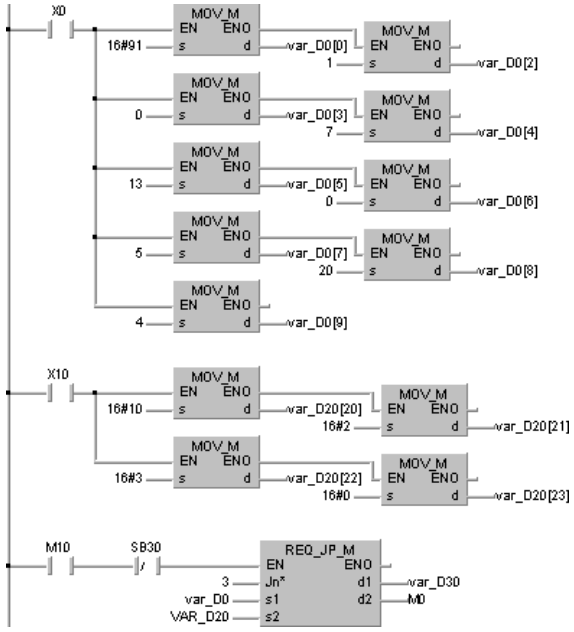
Das folgende Programm führt mit positiver Flanke von X10 eine STOP-Operation an einer Ziel-Station durch. Die Ausführung der REQ-Anweisung ist über den Öffner des Flags SB30 verriegelt. Nähere Angaben zur HOST- und Ziel-Station und den verwendeten MOV-Anweisungen enthält die folgende Tabelle.

Operand/Anweisung	Bezeichnung/Funktion
HOST-Station	–
HOST-Netzwerk	–
HOST-Kanal	1
Kommunikations-Kanal-Flag	SB30
Ziel-Station	13
Ziel-Netzwerk	7
Ziel-Kanal	–
1. MOV-Anweisung	Setzen der Uhr-Daten
2. MOV-Anweisung	Setzen des Kanals der HOST-Station
3. MOV-Anweisung	–
4. MOV-Anweisung	Setzen der Netzwerknummer der Ziel-Station
5. MOV-Anweisung	Setzen der Nummer der Ziel-Station
6. MOV-Anweisung	–
7. MOV-Anweisung	Setzen der Anzahl der Übertragungsversuche
8. MOV-Anweisung	Setzen der WDT-Zeiteinstellung (20 s)
9. MOV-Anweisung	Setzen der Anzahl der zu übertragenden Blöcke (4)
10. MOV-Anweisung	Setzen des Anforderungsmodus
11. MOV-Anweisung	Setzen des Anforderungsmodus der Unterroutine
12. MOV-Anweisung	Setzen des Modus
13. MOV-Anweisung	Setzen des Löschmodus

MELSEC-Anweisungsliste

```
MELSEC  LD      X0
        MOV     H91
                D0
        MOV     K1
                D2
        MOV     K0
                D3
        MOV     K7
                D4
        MOV     K13
                D5
        MOV     K0
                D6
        MOV     K5
                D7
        MOV     K20
                D8
        MOV     K4
                D9
        LD      X10
        MOV     H10
                D20
        MOV     H2
                D21
        MOV     H3
                D22
        MOV     H0
                D23
        LD      M10
        ANI     SB30
        JP.REQ  J3
                D0
                D20
                D30
                M0
```

Kontaktplan



IEC-Anweisungsliste

```
LD          X0
MOV_M       16#91 , var_D0[0]
MOV_M       1 , var_D0[2]
MOV_M       0 , var_D0[3]
MOV_M       7 , var_D0[4]
MOV_M       13 , var_D0[5]
MOV_M       0 , var_D0[6]
MOV_M       5 , var_D0[7]
MOV_M       20 , var_D0[8]
MOV_M       4 , var_D0[9]
LD          X10
MOV_M       16#10 , var_D20[20]
MOV_M       16#2 , var_D20[21]
MOV_M       16#3 , var_D20[22]
MOV_M       16#0 , var_D20[23]
LD          M10
ANDN        SB30
REQ_JP_M    3 , var_D0 , VAR_D20 - var_D30 , M0
```

HINWEIS *Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- und Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

8.6.8 ZNFR

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	

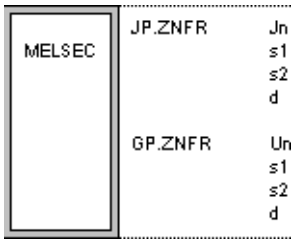
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	—	●	●	—	—	—	—	—	—	—	8
s2	—	●[1]	—	—	—	—	—	—	—		
d	●	—	—	—	—	—	—	—	—		

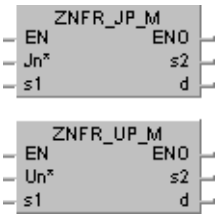
[1] Nur Link-Register

GX IEC Developer

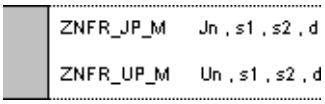
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



Variablen

Operand	Befehlswert	Datentyp	
		MELSEC	IEC
Jn	Netzwerknummer der HOST-Station. ●[1]	BIN-16-Bit	ANY16
Un	Kopf-Ein-/Ausgangsadresse des Netzwerkmoduls der HOST-Station. ●[2]		
s1	Erste Adresse des Operanden, in dem die Kontrolldaten gespeichert sind.	Adresse	Array [1..15] of ANY16
s2	Erste Adresse des Link-Registers (W) in der HOST-Station, in dem die gelesenen Daten gespeichert werden.		ANY16
d	Operand, der nach vollständiger Verarbeitung der Anweisung für einen Programmzyklus gesetzt wird.	Bit	BOOL

HINWEISE

- ●[1] *Die Netzwerknummer der HOST-Station muss zwischen 1 und 239 liegen. Das Netzwerk mit der Nummer 254 wird über Einstellungen so konfiguriert, dass andere Stationen auf die aktive Station zugreifen können.*
- ●[2] *Die Kopf-Ein-/Ausgangsadresse des Netzwerkmoduls der HOST-Station muss zwischen 0 und FE$_H$ liegen. Es ist zu beachten, dass der Compiler für Un eine hexadezimale Adressangabe erwartet. Eine Dezimalangabe wird automatisch in eine Hexadezimalzahl umgewandelt.*

Operandenübersicht der Kontrolldaten

Operand	Bedeutung	Funktion	Werte-bereich	Festlegung durch
(s1)+0 Array_s1[1]	Ausführungs-Modus	Die Eingangsbestätigung ist gesetzt: (Bit 0 (b0) = 1, fest eingestellt)	0001H 0081H	Benutzer
	Abschluss-Modus der Fehlerverar-beitung	Speichern der Uhr-Daten des Zeitpunktes, an dem die Fehlerverabeitung abgeschlossen ist: - Keine Speicherung der Uhr-Daten, Bit 7 (b7) = 0 - Speicherung der Uhr-Daten, Bit 7 (b7) = 1 (Uhr-Daten ab (s1)+11 (Array_s1[12]) aufwärts)		
(s1)+1 Array_s1[2]	Abschluss-Status der Anweisungs-verarbeitung	Statusanzeige nach Abschluss der Anweisungs-verarbeitung: - 0 = Keine Fehler (normaler Verarbeitungs-abschluss) - ungleich 0 = Fehlercode ●[3]	—	System
(s1)+2 Array_s1[3]	Dummy	Wird nicht verwendet.	—	—
(s1)+3 Array_s1[4]	Adresse des Pufferspeichers	Angabe der ersten Adresse des Pufferspeichers.	●[4]	Benutzer
(s1)+4 Array_s1[5]	Dummy	Wird nicht verwendet.	—	—
(s1)+5 Array_s1[6]	Nummer der Ziel-Station	Angabe der Nummer der ausgelagerten Ein-/Ausgabe-Station, welches die Daten lesen wird.	1 bis 64	Benutzer
(s1)+6 Array_s1[7]	Position des Sondermoduls in der Modul-Reihe	Angabe der Position, an welcher sich das Sonder-modul in einer Reihe von Sondermodulen in der Ziel-Station befindet.	—	Benutzer
(s1)+7 Array_s1[8]	Dummy	Wird nicht verwendet.	—	—
(s1)+8 Array_s1[9]	Dummy	Wird nicht verwendet.	—	—
(s1)+9 Array_s1[10]	Länge der zu lesenden Daten	Angabe der Anzahl der zu lesenden Daten.	1 bis 256	Benutzer
(s1)+10 Array_s1[11]	Flag der Uhr-Daten-speicherung (wird nur im Fehlerfall gesetzt)	Dieses Flag speichert den Status der in (s1)+0 (Array_s1[1]) angegebenen Speicheroption der Uhr-Daten: Keine Speicherung der Uhr-Daten = 0 Speicherung der Uhr-Daten = 1	—	System
(s1)+11 Array_s1[12]	Uhr-Daten (werden nur im Fehlerfall gesetzt)	Höchstwertiges Byte = Jahr (0 bis 99) Niedrigstwertiges Byte = Monat (1 bis 12)	—	System
(s1)+12 Array_s1[13]		Höchstwertiges Byte = Tag (1 bis 31) Niedrigstwertiges Byte = Stunde (0 bis 23)		
(s1)+13 Array_s1[14]		Höchstwertiges Byte = Minute (0 bis 59) Niedrigstwertiges Byte = Sekunde (0 bis 59)		
(s1)+14 Array_s1[15]		Höchstwertiges Byte = 00H Niedrigstwertiges Byte = Wochentag (0 bis 6) (Sonntag = 0, Samstag = 6)		

●[3] Weitere Informationen über Fehlercodes enthält das Handbuch "MELSECNET/10 für QnA-Netzwerk-Systeme".

●[4] Weitere Informationen enthält das Handbuch des Sondermoduls, mit welchem die Lese-Operation ausgeführt wird.

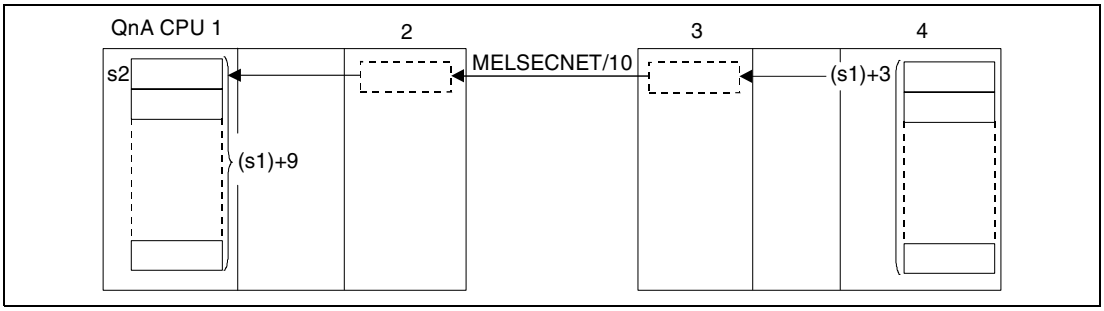
Funktionsweise

Lesen von Daten aus Sondermodulen in ausgelagerten Ein-/Ausgabe-Stationen

ZNFR Lese-Anweisung

Die ZNFR-Anweisung liest die in dem Pufferspeicher eines Sondermoduls gespeicherten Daten einer im MELSECNET/10 angeschlossenen, ausgelagerten Ein-/Ausgabe-Station. Die Ein-/Ausgabe-Station ist in den Kontrolldaten angegeben. Die aus dem Modul gelesenen Daten werden ab s2 in der HOST-Station gespeichert.

Nach dem Abschluss der Lese-Operation in der ausgelagerten Ein-/Ausgabe-Station wird der in d angegebene Operand gesetzt.



[1] HOST-Station/Master-Station

[2] Netzwerkmodul (HOST-Station/Master-Station)

[3] Ausgelagerte Ein-/Ausgabe-Station (Ziel-Station)

[4] Sondermodul (Ziel-Station/ausgelagerte Ein-/Ausgabe-Station)

Die Lese-Operation von einer ausgelagerten Ein-/Ausgabe-Station kann nur über ein Netzwerkmodul ausgeführt werden, das im gleichen Netzwerk angeschlossen ist wie die im MELSECNET/10 angeschlossene ausgelagerte Ein-/Ausgabe-Station.

Die ZNFR-Anweisung kann nicht von mehreren Stellen gleichzeitig von dem gleichen Sondermodul ausgeführt werden. Bei gleichzeitiger Ausführung der Anweisungen von zwei oder mehr Stellen wird durch den Handshake-Betrieb der beiden aktiven Stationen eine Ausführung weiterer ZNFR-Anweisungen verhindert.

Das während der Ausführung der ZNFR-Anweisung gesendete Interlock-Signal enthält

- Lese-/Schreib-Anforderungssignale,
- Abschlusssignale der Lese-/Schreib-Operationen,
- den Operanden, der nach Abschluss der Lese-Operation gesetzt wird (d)
- und die Status-Anzeige des Operationsabschlusses (Abschluss einer fehlerfreien oder fehlerhaften Übertragung) (d+1). Die Signale und Operanden sind im folgenden beschrieben:

Lese-/Schreib-Anforderungssignale

Dieses Signal wird mit der Ausführung der erweiterten Data-Link-Anweisungen der QnA-Serie gesetzt. Das Signal wird mit der Verarbeitung der END-Anweisung in dem Zyklus, in dem die Lese-/Schreib-Operationen abgeschlossen wurden, zurückgesetzt.

Abschlusssignale der Lese-/Schreib-Operationen

Dieses Signal wird mit der Ausführung der erweiterten Data-Link-Anweisungen der QnA-Serie zurückgesetzt. Das Signal wird mit der Verarbeitung der END-Anweisung in dem Zyklus gesetzt, in dem die Lese-/Schreib-Operationen abgeschlossen wurden.

Operand der HOST-Station, der den Operationsabschluss anzeigt

Dieser Operand wird mit der Verarbeitung der END-Anweisung in dem Zyklus gesetzt, in dem die Verarbeitung der ZNFR-Anweisung abgeschlossen wurde. Bei der Verarbeitung der darauffolgenden END-Anweisung wird dieser Operand wieder zurückgesetzt.

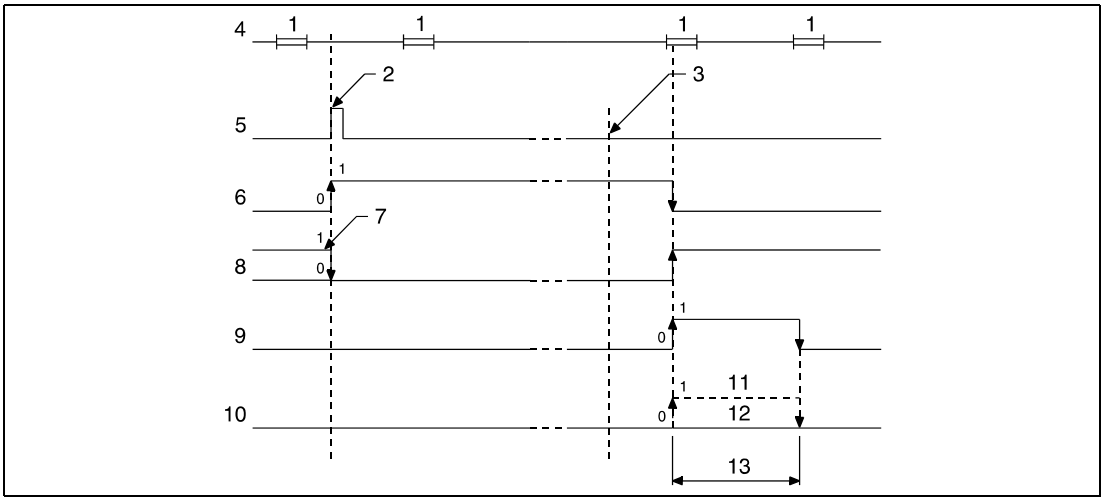
Status-Anzeige des Operationsabschlusses

Dieser Operand wird in Abhängigkeit des Abschlussergebnisses der Verarbeitung der ZNFR-Anweisung gesetzt.

Beim Abschluss einer normalen (fehlerfreien) Übertragung bleibt der Operand unverändert zurückgesetzt.

Beim Abschluss einer nicht normalen (fehlerhaften) Übertragung wird dieser Operand mit der Verarbeitung der END-Anweisung in dem Zyklus gesetzt, in dem die Verarbeitung der ZNFR-Anweisung abgeschlossen wurde. Bei der Verarbeitung der darauffolgenden END-Anweisung wird dieser Operand wieder zurückgesetzt.

Die folgende Abbildung zeigt die Operationen der HOST-Station während der Ausführung der ZNFR-Anweisung.



1 END-Verarbeitung
2 Ausführung der ZNFR-Anweisung
3 Abschluss der Operation
4 Programm der HOST-Station
5 ZNFR-Anweisung
6 Lese-/Schreib-Anforderungssignal
7 Nach Ausführung der erweiterten Data-Link-Anweisungen
8 Abschluss der Lese-/Schreib-Operation
9 Operand der HOST-Station, der nach Abschluss der Operation gesetzt wird (d)
10 Status-Anzeige des Operationsabschlusses (d+1)
11 Abschluss einer nicht normalen (fehlerhaften) Übertragung
12 Abschluss einer normalen (fehlerfreien) Übertragung
13 Ein Zyklus

Die Link-Register in s2 werden durch die Netzwerk-Parameter "M ← R (zur Master-Station von der ausgelagerten Ein-/Ausgabe-Station)" gesetzt und liegen innerhalb des durch die Link-Refresh-Parameter festgelegten Bereichs.

Für die Ausführung der ZNFR-Anweisung werden Link-Merker und Link-Register benötigt, die von dem Betriebssystem verwendet werden können. Die Anzahl der Link-Merker und Link-Register, die von dem Betriebssystem für das entsprechende Sondermodul verwendet werden, lautet wie folgt:

Für M → R (von der Master-Station zur ausgelagerten Ein-/Ausgabe-Station):
Link-Merker = 4, Link-Register = 4

Für M ← R (zur Master-Station von der ausgelagerten Ein-/Ausgabe-Station):
Link-Merker = 4, Link-Register = 4

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die Kontrolldateninhalte liegen nicht im gültigen Wertebereich (Fehlercode 4100).
- Das Netzwerk mit der mit Jn angegebenen Nummer ist nicht mit der Station verbunden (Fehlercode 4102).
- Das Modul an der mit Un angegebenen Ein-/Ausgangsadresse ist kein Netzwerkmodul (Fehlercode 2111).

Beispiel JP.ZNFR

Das folgende Programm liest mit positiver Flanke von X10 die Adressen 6 bis 9 des Pufferspeichers eines Sondermoduls einer Ein-/Ausgabe-Station aus. Die gelesenen Daten werden in den Link-Registern W108 bis W10B gespeichert. Nähere Angaben zur Ein-/Ausgabe-Station und den verwendeten MOV-Anweisungen enthält die folgende Tabelle.

Operand/Anweisung	Bezeichnung/Funktion
Ein-/Ausgabe-Station	1R1
Ein-/Ausgabe-Stationsnetzwerk	3
Sondermodul	1
1. MOV-Anweisung	Setzen der Uhr-Daten
2. MOV-Anweisung	Setzen der ersten Adresse des Pufferspeichers (6)
3. MOV-Anweisung	Setzen der Nummer der Ein-/Ausgabe-Station
4. MOV-Anweisung	Setzen der Position des Sondermoduls in der Modul-Reihe
5. MOV-Anweisung	Setzen der Länge der zu lesenden Daten

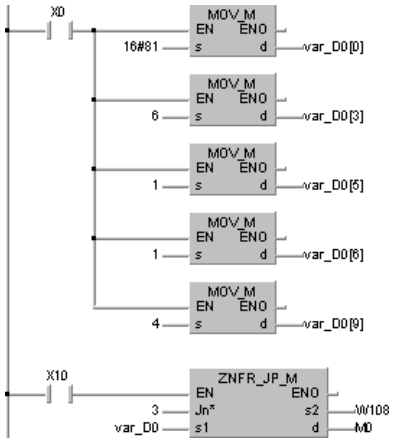
MELSEC-Anweisungsliste

```
MELSEC  LD      X0
        MOV     H81
                D0
        MOV     K6
                D3
        MOV     K1
                D5
        MOV     K1
                D6
        MOV     K4
                D9
        LD      X10
        JP.ZNFR J3
                D0
                W108
                M0
```

IEC-Anweisungsliste

```
LD          X0
MOV_M       16#81 , var_D0[0]
MOV_M       6 , var_D0[3]
MOV_M       1 , var_D0[5]
MOV_M       1 , var_D0[6]
MOV_M       4 , var_D0[9]
LD          X10
ZNFR_JP_M   3 , var_D0 , W108 , M0
```

Kontaktplan



HINWEIS *Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- und Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

8.6.9 ZNTO

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	

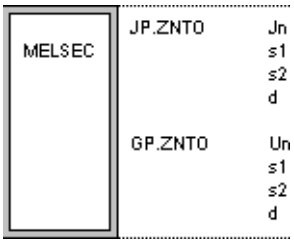
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	—	●	●	—	—	—	—	—	—	—	8
s2	—	●[1]	—	—	—	—	—	—	—		
d	●	—	—	—	—	—	—	—	—		

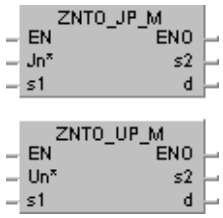
[1] Nur Link-Register

GX IEC Developer

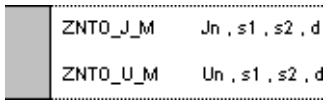
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



Variablen

Operand	Befehlswert	Datentyp	
		MELSEC	IEC
Jn	Netzwerknummer der HOST-Station. ●[1]	BIN-16-Bit	ANY16
Un	Kopf-Ein-/Ausgangsadresse des Netzwerkmoduls der HOST-Station. ●[2]		
s1	Erste Adresse des Operanden, in dem die Kontrolldaten gespeichert sind.	Adresse	Array [1..16] of ANY16
s2	Erste Adresse des Link-Registers (W) in der HOST-Station, in dem die zu schreibenden Daten gespeichert sind.		ANY16
d	Operand, der nach vollständiger Verarbeitung der Anweisung für einen Programmzyklus gesetzt wird.	Bit	BOOL

HINWEISE

●[1] *Die Netzwerknummer der HOST-Station muss zwischen 1 und 239 liegen. Das Netzwerk mit der Nummer 254 wird über Einstellungen so konfiguriert, dass andere Stationen auf die aktive Station zugreifen können.*

●[2] *Die Kopf-Ein-/Ausgangsadresse des Netzwerkmoduls der HOST-Station muss zwischen 0 und FE$_H$ liegen. Es ist zu beachten, dass der Compiler für Un eine hexadezimale Adressangabe erwartet. Eine Dezimalangabe wird automatisch in eine Hexadezimalzahl umgewandelt.*

Operandenübersicht der Kontrolldaten

Operand	Bedeutung	Funktion	Werte-bereich	Festlegung durch
(s1)+0 Array_s1[1]	Ausführungs-Modus	Die Eingangsbestätigung ist gesetzt: (Bit 0 (b0) = 1, fest eingestellt)	0001H 0081H	Benutzer
	Abschluss-Modus der Fehlerverar-beitung	Speichern der Uhr-Daten des Zeitpunktes, an dem die Fehlerverabeitung abgeschlossen ist: - Keine Speicherung der Uhr-Daten, Bit 7 (b7) = 0 - Speicherung der Uhr-Daten, Bit 7 (b7) = 1 (Uhr-Daten ab (s1)+11 (Array_s1[12]) aufwärts)		
(s1)+1 Array_s1[2]	Abschluss-Status der Anweisungs-verarbeitung	Statusanzeige nach Abschluss der Anweisungs-verarbeitung: - 0 = Keine Fehler (normaler Verarbeitung-abschluss) - ungleich 0 = Fehlercode ●[3]	—	System
(s1)+2 Array_s1[3]	Dummy	Wird nicht verwendet.	—	—
(s1)+3 Array_s1[4]	Adresse des Pufferspeichers	Angabe der ersten Adresse des Pufferspeichers.	●[4]	Benutzer
(s1)+4 Array_s1[5]	Dummy	Wird nicht verwendet.	—	—
(s1)+5 Array_s1[6]	Nummer der Ziel-Station	Angabe der Nummer der Ziel-Station.	1 bis 64	Benutzer
(s1)+6 Array_s1[7]	Nummer des Sondermoduls	Angabe der fortlaufenden Nummer des entsprechenden Sondermoduls in der Reihe der Sondermodule in der Ziel-Station.	—	Benutzer
(s1)+7 Array_s1[8]	Dummy	Wird nicht verwendet.	—	—
(s1)+8 Array_s1[9]	Dummy	Wird nicht verwendet.	—	—
(s1)+9 Array_s1[10]	Länge der zu schreibenden Daten	Angabe der Anzahl der zu schreibenden Daten.	1 bis 256	Benutzer
(s1)+10 Array_s1[11]	Dummy	Wird nicht verwendet.	—	—
(s1)+11 Array_s1[12]	Flag der Uhr-Daten-speicherung (wird nur im Fehlerfall gesetzt)	Dieses Flag speichert den Status der in (s1)+0 (Array_s1[1]) angegebenen Speicheroption der Uhr-Daten: Keine Speicherung der Uhr-Daten = 0 Speicherung der Uhr-Daten = 1	—	System
(s1)+12 Array_s1[13]	Uhr-Daten (werden nur im Fehlerfall gesetzt)	Höchstwertiges Byte = Jahr (0 bis 99) Niedrigstwertiges Byte = Monat (1 bis 12)	—	System
(s1)+13 Array_s1[14]		Höchstwertiges Byte = Tag (1 bis 31) Niedrigstwertiges Byte = Stunde (0 bis 23)		
(s1)+14 Array_s1[15]		Höchstwertiges Byte = Minute (0 bis 59) Niedrigstwertiges Byte = Sekunde (0 bis 59)		
(s1)+15 Array_s1[16]		Höchstwertiges Byte = 00H Niedrigstwertiges Byte = Wochentag (0 bis 6) (Sonntag = 0, Samstag = 6)		

●[3] Weitere Informationen über Fehlercodes enthält das Handbuch "MELSECNET/10 für QnA-Netzwerk-Systeme".

●[4] Weitere Informationen enthält das Handbuch des Sondermoduls, mit welchem die Schreib-Operation ausgeführt wird.

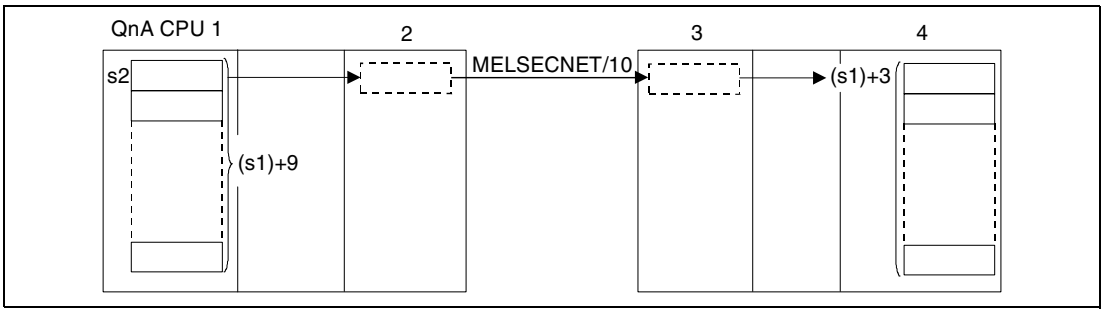
Funktions-weise

Schreiben von Daten in Sondermodule in ausgelagerten Ein-/Ausgabe-Stationen

ZNTO Schreib-Anweisung

Die ZNTO-Anweisung schreibt die ab s2 gespeicherten Daten der HOST-Station in den Pufferspeicher eines Sondermoduls einer im MELSECNET/10 angeschlossenen ausgelagerten Ein-/Ausgabe-Station. Die Ein-/Ausgabe-Station ist in den Kontrolldaten angegeben.

Nach dem Abschluss der Schreib-Operation in der ausgelagerten Ein-/Ausgabe-Station wird der in d angegebene Operand gesetzt.



[1] HOST-Station/Master-Station

[2] Netzwerkmodul (HOST-Station/Master-Station)

[3] Ausgelagerte Ein-/Ausgabe-Station (Ziel-Station)

[4] Sondermodul (Ziel-Station/ausgelagerte Ein-/Ausgabe-Station)

Die Schreib-Operationen in eine ausgelagerte Ein-/Ausgabe-Station können nur ausgeführt werden, wenn diese Schreib-Operationen von einer ausgelagerten Master-Station im MELSECNET/10 in eine ausgelagerte Ein-/Ausgabe-Station im gleichen Netzwerk durchgeführt wird.

Die ZNTO-Anweisung kann nicht von mehreren Stellen gleichzeitig von dem gleichen Sondermodul ausgeführt werden. Bei gleichzeitiger Ausführung der Anweisungen von zwei oder mehr Stellen wird durch den Handshake-Betrieb der beiden aktiven Stationen eine Ausführung weiterer ZNTO-Anweisungen verhindert.

Das während der Ausführung der ZNTO-Anweisung gesendete Interlock-Signal enthält

- Lese-/Schreib-Anforderungssignale,
- Abschlusssignale der Lese-/Schreib-Operationen,
- den Operanden, der nach Abschluss der Schreib-Operation gesetzt wird (d)
- und die Status-Anzeige des Operationsabschlusses (Abschluss einer fehlerfreien oder fehlerhaften Übertragung) (d+1). Die Signale und Operanden sind im folgenden beschrieben:

Lese-/Schreib-Anforderungssignale

Dieses Signal wird mit der Ausführung der erweiterten Data-Link-Anweisungen der QnA-Serie gesetzt. Das Signal wird mit der Verarbeitung der END-Anweisung in dem Zyklus, in dem die Lese-/Schreib-Operationen abgeschlossen wurden, zurückgesetzt.

Abschlusssignale der Lese-/Schreib-Operationen

Dieses Signal wird mit der Ausführung der erweiterten Data-Link-Anweisungen der QnA-Serie zurückgesetzt. Das Signal wird mit der Verarbeitung der END-Anweisung in dem Zyklus gesetzt, in dem die Lese-/Schreib-Operationen abgeschlossen wurden.

Operand der HOST-Station, der den Operationsabschluss anzeigt

Dieser Operand wird mit der Verarbeitung der END-Anweisung in dem Zyklus gesetzt, in dem die Verarbeitung der ZNTO-Anweisung abgeschlossen wurde. Bei der Verarbeitung der darauffolgenden END-Anweisung wird dieser Operand wieder zurückgesetzt.

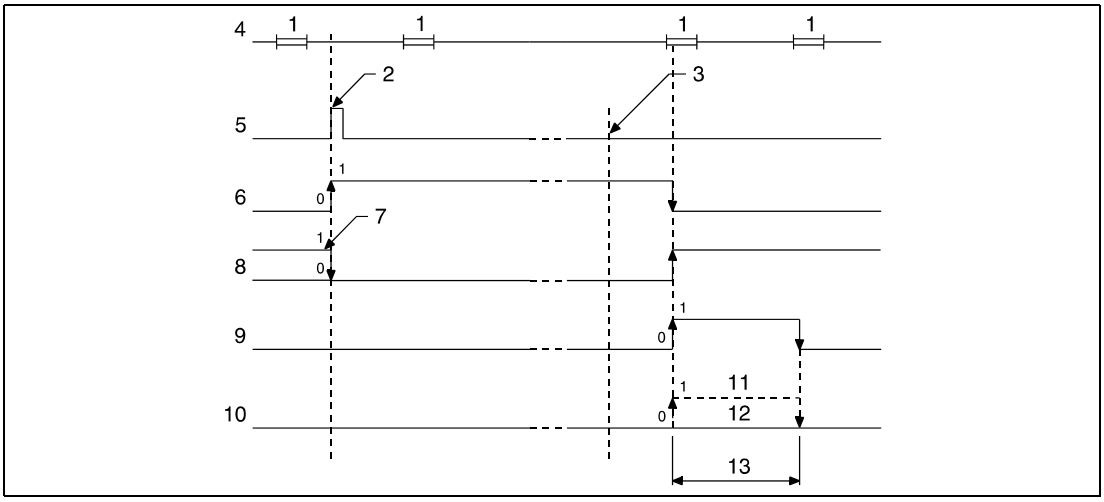
Status-Anzeige des Operationsabschlusses

Dieser Operand wird in Abhängigkeit des Abschlussergebnisses der Verarbeitung der ZNTO-Anweisung gesetzt.

Beim Abschluss einer normalen (fehlerfreien) Übertragung bleibt der Operand unverändert zurückgesetzt.

Beim Abschluss einer nicht normalen (fehlerhaften) Übertragung wird dieser Operand mit der Verarbeitung der END-Anweisung in dem Zyklus gesetzt, in dem die Verarbeitung der ZNTO-Anweisung abgeschlossen wurde. Bei der Verarbeitung der darauffolgenden END-Anweisung wird dieser Operand wieder zurückgesetzt.

Die folgende Abbildung zeigt die Operationen der HOST-Station während der Ausführung der ZNTO-Anweisung.



[1] END-Verarbeitung
[2] Ausführung der ZNTO-Anweisung
[3] Abschluss der Operation
[4] Programm der HOST-Station
[5] ZNTO-Anweisung
[6] Lese-/Schreib-Anforderungssignal
[7] Nach Ausführung der erweiterten Data-Link-Anweisungen
[8] Abschluss der Lese-/Schreib-Operation
[9] Operand der HOST-Station, der nach Abschluss der Operation gesetzt wird (d)
[10] Status-Anzeige des Operationsabschlusses (d+1)
[11] Abschluss einer nicht normalen (fehlerhaften) Übertragung
[12] Abschluss einer normalen (fehlerfreien) Übertragung
[13] Ein Zyklus

Die Link-Register in s2 werden durch die Netzwerk-Parameter "M ← R (zur Master-Station von der ausgelagerten Ein-/Ausgabe-Station)" gesetzt, und liegen innerhalb des durch die Link-Refresh-Parameter festgelegten Bereichs.

Für die Ausführung der ZNTO-Anweisung werden Link-Merker und Link-Register benötigt, die von dem Betriebssystem verwendet werden können. Die Anzahl der Link-Merker und Link-Register, die von dem Betriebssystem für das entsprechende Sondermodul verwendet werden, lautet wie folgt:

Für M $\rightarrow$ R (von der Master-Station zur ausgelagerten Ein-/Ausgabe-Station):
Link-Merker = 4, Link-Register = 4

Für M $\leftarrow$ R (zur Master-Station von der ausgelagerten Ein-/Ausgabe-Station):
Link-Merker = 4, Link-Register = 4

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die Kontrolldateninhalte liegen nicht im gültigen Wertebereich (Fehlercode 4100).
- Das Netzwerk mit der mit Jn angegebenen Nummer ist nicht mit der Station verbunden (Fehlercode 4102).
- Das Modul an der mit Un angegebenen Ein-/Ausgangsadresse ist kein Netzwerkmodul (Fehlercode 2111).

Beispiel JP.ZNTO

Das folgende Programm schreibt mit positiver Flanke von X10 Daten in die Adressen 10 bis 12 des Pufferspeichers eines Sondermoduls einer Ein-/Ausgabe-Station. Die zu schreibenden Daten sind in der Host-Station in den Link-Registern W18 bis W1A gespeichert. Nähere Angaben zur Ein-/Ausgabe-Station und den verwendeten MOV-Anweisungen enthält die folgende Tabelle.

Operand/Anweisung	Bezeichnung/Funktion
Ein-/Ausgabe-Station	1R2
Ein-/Ausgabe-Stationsnetzwerk	3
Sondermodul	2
1. MOV-Anweisung	Setzen der Uhr-Daten
2. MOV-Anweisung	Setzen der ersten Adresse des Pufferspeichers (10)
3. MOV-Anweisung	Setzen der Nummer der Ein-/Ausgabe-Station
4. MOV-Anweisung	Setzen der Position des Sondermoduls in der Modul-Reihe
5. MOV-Anweisung	Setzen der Länge der zu schreibenden Daten
6. MOV-Anweisung	Schreiben der Daten in die Link-Register W18 bis W1A
7. MOV-Anweisung	
8. MOV-Anweisung	

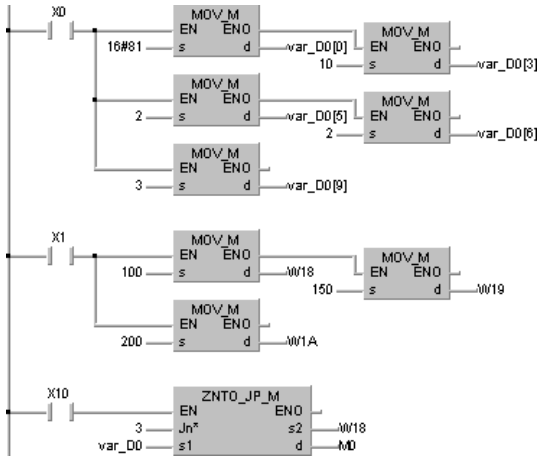
MELSEC-Anweisungsliste

```
MELSEC   LD     X0
         MOV    H81
                D0
         MOV    K10
                D3
         MOV    K2
                D5
         MOV    K2
                D6
         MOV    K3
                D9
         LD     X1
         MOV    K100
                W18
         MOV    K150
                W19
         MOV    K200
                W1A
         LD     X10
         JP.ZNTO J3
                D0
                W18
                M0
```

IEC-Anweisungsliste

```
LD          X0
MOV_M       16#81 , var_D0[0]
MOV_M       10 , var_D0[3]
MOV_M       2 , var_D0[5]
MOV_M       2 , var_D0[6]
MOV_M       3 , var_D0[9]
LD          X1
MOV_M       100 , W18
MOV_M       150 , W19
MOV_M       200 , W1A
LD          X10
ZNTO_JP_M   3 , var_D0 , W18 , M0
```

Kontaktplan



HINWEIS *Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- und Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.*

8.7 Zur A-Serie kompatible Data-Link-Anweisungen

Mit diesen Anweisungen ist die Datenkommunikation zwischen Stationen mit QnA-CPUs, zwischen Stationen mit QnA- und ACPUs sowie zwischen Stationen mit QnA- oder ACPUs und ausgelagerten Ein-/Ausgabe-Stationen im MELSECNET und MELSECNET/10 möglich. Die folgende Tabelle liefert eine Übersicht der Anweisungen.

Funktion	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor	Im MELSECNET/10 angesprochene Station			MELSECNET
			QnA-CPU	ACPU	Ausgelagerte Ein-/Ausgabe-Stationen	
Lesen von QnA-Daten aus Ziel-Stationen in Ziel-Netzwerken	J.ZNRD	ZNRD_J_M	●	●	—	—
	JP.ZNRD	ZNRD_JP_M				
Lesen von Daten aus lokalen Stationen (nur an Master-Stationen möglich)	J.ZNRD	ZNRD_J_M	—	—	—	●
	JP.ZNRD	ZNRD_JP_M				
Schreiben von QnA-Daten in Ziel-Stationen in Ziel-Netzwerken	J.ZNWR	ZNWR_J_M	●	●	—	—
	JP.ZNWR	ZNWR_JP_M				
Schreiben von Daten in lokalen Stationen (nur an Master-Stationen möglich)	J.ZNWR	ZNWR_J_M	—	—	—	●
	JP.ZNWR	ZNWR_JP_M				
Nur A-Serie: Lesen von Daten aus lokalen Stationen (nur an Master-Stationen möglich)	LRDP	LRDP_M	—	—	—	●
		LRDP_MD				
		LRDP_P_MD				
Nur A-Serie: Schreiben von Daten in lokale Stationen (nur an Master-Stationen möglich)	LWTP	LWTP_M	—	—	—	●
		LWTP_MD				
		LWTP_P_MD				
Lesen von Daten aus Sondermodulen in ausgelagerten Ein-/Ausgabe-Stationen.	RFRP/ G.RFRP	RFRP_U_M	—	—	—	●
		RFRP_UP_M				
Schreiben von Daten in Sondermodule in ausgelagerten Ein-/Ausgabe-Stationen.	RTOP G.RTOP	RTOP_U_M	—	—	—	●
		RTOP_UP_M				

8.7.1 ZNRD

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	

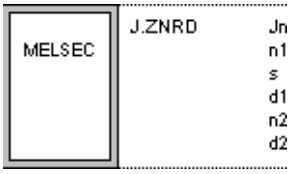
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
n1	●	●	●	—	—	—	—	●	—	SM0	32
s	—	●[1]	—	—	—	—	—	—	—		
d1	—	●	●	—	—	—	—	—	—		
n2	●	●	●	—	—	—	—	●	—		
d2	●	●	●	—	—	—	—	—	—		

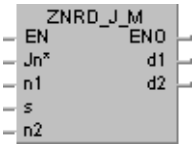
[1] Nur T, C, D, und W

GX IEC Developer

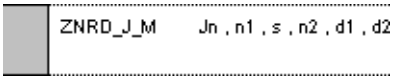
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



Variablen

Operand	Befehlswert	Datentyp
Jn	Netzwerknummer der HOST-Station. ●[1]	BIN-16-Bit
n1	Nummer der Ziel-Station.	
s	Erster Operand der Station, in dem die zu lesenden Daten gespeichert sind.	
d1	Erster Operand der HOST-Station, in dem die gelesenen Daten gespeichert werden.	
n2	Empfangsdatenlänge.	
d2	Operand, der nach vollständiger Verarbeitung der Anweisung für einen Programmzyklus gesetzt wird.	Bit

HINWEIS ●[1] *Wenn die Anweisung in einem MELSECNET(I/II/B) verwendet wird, ist J0 anzugeben.*

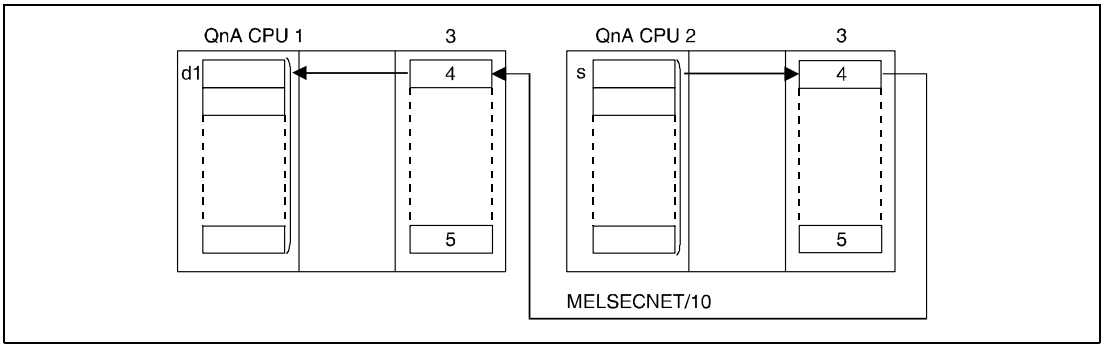
Funktions-weise

Lesen von Daten anderer Stationen

ZNRD Lese-Anweisung

Die ZNRD-Anweisung liest die mit n2 angegebene Anzahl der ab s gespeicherten Datenwörter der Ziel-Station im MELSECNET/10. Die Ziel-Stationsnummer ist in n1 angegeben. Die Netzwerknummer ist in Jn angegeben. Die gelesenen Daten werden ab d1 in der HOST-Station gespeichert.

Nach dem Abschluss der Lese-Operation in der Ziel-Station wird der in d2 angegebene Operand gesetzt.



[1] HOST-Station

[2] Ziel-Station

[3] Netzwerkmodul

[4] Kanal 1

[5] Kanal 8

Die Lese-Operation kann nur mit einer Ziel-Station ausgeführt werden, die im selben Netzwerk des MELSECNET/10-Netzwerksystems wie die HOST-Station angeschlossen ist.

Die Lese-Operation von einer lokalen Station kann nur mit einer Master-Station im Netzwerk-System MELSECNET durchgeführt werden.

Für Jn können Netzwerknummern zwischen 1 und 239 vergeben werden. Die Angabe der Netzwerknummer 0 (J0) ist mit der Angabe des MELSECNET als Netzwerk-System vergleichbar.

Im MELSECNET-System ist die Nummer des Ziel-Netzwerks (Jn) auf 0 (J0) fest eingestellt. Die Ziel-Netzwerknummern (Jn) 1 bis 239 werden im MELSECNET/10 verwendet.

Für n1 kann eine Stationsnummer zwischen 1 und 64 angegeben werden.

Bei Verwendung des Netzwerk-Systems MELSECNET/B kann eine Stationsnummer zwischen 1 und 31 angegeben werden.

Für n2 kann eine Empfangsdatenlänge (Anzahl der Datenworte) zwischen 1 und 230 angegeben werden.

Lese-Operationen von anderen Stationen mittels ZNRD-Anweisung können gleichermaßen von Stationen mit AnU CPUs und QnA CPUs ausgeführt werden.

Die Data-Link-Anweisungen können nicht von mehreren Stellen gleichzeitig mit gemeinsamem Zugriff auf die gleichen Kanäle ausgeführt werden. Bei gleichzeitiger Ausführung der Anweisungen von zwei oder mehr Stellen wird durch den Handshake-Betrieb der beiden aktiven Stationen eine Ausführung weiterer Data-Link-Anweisungen verhindert.

Die HOST- und die Ziel-Station benutzen für die Ausführung der ZNRD-Anweisung beide den Kanal 1 des Netzwerkmoduls. Bei der Verwendung mehrerer ZNRD-Anweisungen wird mehrmals auf Kanal 1 zugegriffen, wogegen der Kanal 1 des Netzwerkmoduls nur einmal von einer Anweisung gleichzeitig genutzt werden kann. Um eine gleichzeitige Ausführung mehrerer Anweisungen zu verhindern, ist mit dem Lese-/Schreibanforderungssignal und dem Operanden, der nach dem Operationsabschluss gesetzt wird, eine Verriegelung zu realisieren.

Die Ausführung und das Ausführungsende der ZNRD-Anweisung wird mittels Kommunikations-Kanal-Flag (SB30) und dem Operanden, der bei Operationsabschluss gesetzt wird (d2), wie folgt angezeigt:

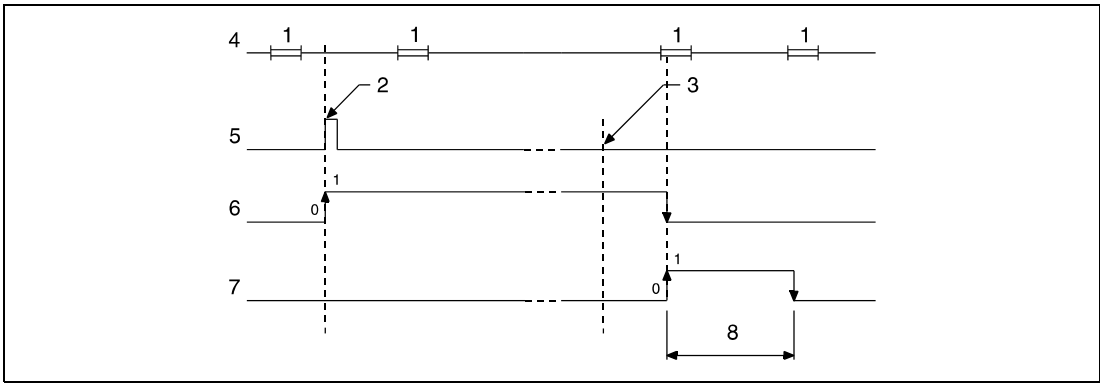
Kommunikations-Kanal-Flag

Dieses Flag wird während der Ausführung der ZNRD-Anweisung gesetzt. Das Flag wird mit der Verarbeitung der END-Anweisung in dem Zyklus zurückgesetzt, in dem die Lese-Operation abgeschlossen wurde.

Operand der HOST-Station, der den Lese-Operationsabschluss anzeigt

Dieser Operand wird mit der Verarbeitung der END-Anweisung in dem Zyklus gesetzt, in dem die Lese-Operation abgeschlossen wurde. Bei der Verarbeitung der darauffolgenden END-Anweisung wird dieser Operand wieder zurückgesetzt.

Die folgende Abbildung zeigt die Operationen der HOST-Station während der Ausführung der ZNRD-Anweisung.



1 END-Verarbeitung
2 Ausführung der ZNRD-Anweisung
3 Abschluss der Operation
4 Programm der HOST-Station
5 ZNRD-Anweisung
6 Kommunikations-Kanal-Flag (SB30)
7 Operand der HOST-Station, der nach Abschluss der Operation gesetzt wird (d2)
8 Ein Zyklus

Der Abschluss-Status der ZNRD-Anweisung (normal, nicht normal) wird mit dem Register für den Verarbeitungsabschluss der ZNRD-Anweisung (SW31) wie folgt angezeigt:

Bei normalem (fehlerfreiem) Operations-Abschluss ist der Inhalt des Registers SW31 gleich 0.

Bei nicht normalem (fehlerhaftem) Operations-Abschluss wird der entsrechende Fehlercode in dem Register SW31 gespeichert.

HINWEIS *Weitere Informationen über Fehlercodes enthält das Handbuch "MELSECNET/10 für QnA-Netzwerk-Systeme".*

Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die mit n2 angegebene Empfangsdatenlänge liegt außerhalb des für die Speicherung vorgesehenen Bereichs des Operanden s1 (Fehlercode 4101).
- Das Netzwerk mit der mit Jn angegebenen Nummer existiert nicht (Fehlercode 4102).
- Die Station mit der in n1 angegebenen Stationsnummer existiert nicht (Fehlercode 4102).
- Die mit n2 angegebene Empfangsdatenlänge liegt nicht zwischen 1 und 230 (Fehlercode 4100).

Beispiel

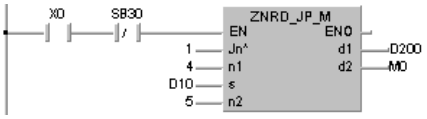
JP.ZNRD

Das folgende Programm liest mit positiver Flanke von X0 die Daten aus den Registern D10 bis D14 aus der Station Nummer 4 und speichert die gelesenen Daten in den Registern D200 bis D204 in der HOST-Station. Die HOST- und die Ziel-Station befinden sich in Netzwerk 1.

MELSEC-Anweisungsliste

```
MELSEC   LD       X0
         ANI      SB30
         JP.ZNRD  J1
                  K4
                  D10
                  D200
                  K5
                  M0
```

Kontaktplan



IEC-Anweisungsliste

```
         LD       X0
         ANDN     SB30
ZNRD_JP_M  1 , 4 , D10 , 5 , D200 , M0
```

8.7.2 ZNWR

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	

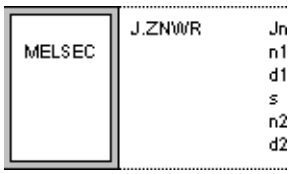
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
n1	●	●	●	—	—	—	—	●	—	SM0	32
d1	—	●[1]	—	—	—	—	—	—	—		
s	—	●	●	—	—	—	—	—	—		
n2	—	●	●	—	—	—	—	●	—		
d2	●	●	●	—	—	—	—	—	—		

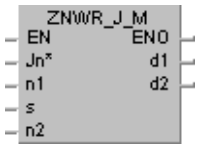
[1] Nur T, C, D, und W

GX IEC Developer

MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste

```
ZNWR_J_M    Jn , n1 , s , n2 , d1 , d2
```

Variablen

Operand	Befehlswert	Datentyp
Jn	Netzwerknummer der HOST-Station. ●[1]	BIN-16-Bit
n1	Nummer der Ziel-Station	
d1	Erster Operand der Ziel-Station, in den die Daten geschrieben werden.	
s	Erster Operand der HOST-Station, in dem die zu schreibenden Daten gespeichert sind.	
n2	Sendedatenlänge.	
d2	Operand, der nach vollständiger Verarbeitung der Anweisung für einen Programmzyklus gesetzt wird.	Bit

HINWEIS *●[1] Wenn die Anweisung in einem MELSECNET(I/II/B) verwendet wird, ist J0 anzugeben.*

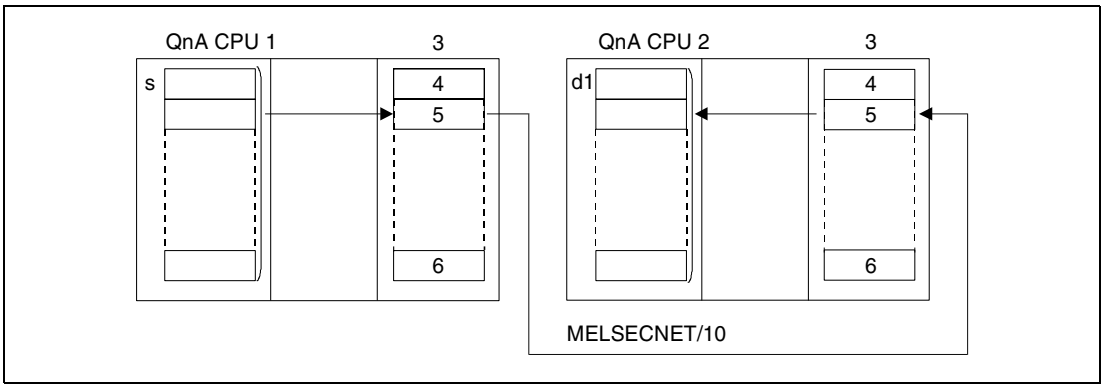
Funktions-weise

Schreiben von Daten in andere Stationen

ZNWR Schreib-Anweisung

Die ZNWR-Anweisung schreibt die mit n2 angegebene Anzahl der in s gespeicherten Datenwörter der HOST-Station in eine Zielstation im MELSECNET/10. Die Ziel-Stationsnummer ist in n1 angegeben. Die Netzwerknummer ist in Jn angegeben.

Nach dem Abschluss der Schreib-Operation in der Ziel-Station wird der in d2 angegebene Operand gesetzt.



[1] HOST-Station
[2] Ziel-Station
[3] Netzwerkmodul
[4] Kanal 1
[5] Kanal 2
[6] Kanal 8

Die Schreib-Operation kann nur mit einer Ziel-Station ausgeführt werden, die im selben Netzwerk des MELSECNET/10-Netzwerksystems wie die HOST-Station angeschlossen ist.

Die Schreib-Operation von einer lokalen Station kann nur mit einer Master-Station im Netzwerk-System MELSECNET durchgeführt werden.

Für Jn können Netzwerknummern zwischen 1 und 239 vergeben werden. Die Angabe der Netzwerknummer 0 (J0) ist mit der Angabe des MELSECNET als Netzwerk-System vergleichbar.

Im MELSECNET-System ist die Nummer des Ziel-Netzwerks (Jn) auf 0 (J0) fest eingestellt. Die Ziel-Netzwerknummern (Jn) 1 bis 239 werden im MELSECNET/10 verwendet.

Für n1 kann eine Stationsnummer zwischen 1 und 64 angegeben werden.

Bei Verwendung des Netzwerk-Systems MELSECNET/B kann eine Stationsnummer zwischen 1 und 31 angegeben werden.

Für n2 kann eine Sendedatenlänge (Anzahl der Datenworte) zwischen 1 und 230 angegeben werden.

Schreib-Operationen mit anderen Stationen mittels ZNWR-Anweisung können gleichermaßen mit Stationen mit AnU CPUs und QnA CPUs ausgeführt werden.

Die HOST- und die Ziel-Station benutzen für die Ausführung der ZNWR-Anweisung beide den Kanal 2 des Netzwerkmoduls. Bei der Verwendung mehrerer ZNWR-Anweisungen wird mehrmals auf Kanal 2 zugegriffen, wogegen der Kanal 2 des Netzwerkmoduls nur einmal von einer Anweisung gleichzeitig genutzt werden kann. Um eine gleichzeitige Ausführung mehrerer Anweisungen zu verhindern, ist mit dem Lese-/Schreibanforderungssignal und dem Operanden, der nach dem Operationsabschluss gesetzt wird, eine Verriegelung zu realisieren.

Die Ausführung und das Ausführungsende der ZNWR-Anweisung wird mittels Kommunikations-Kanal-Flag (SB32) und dem Operanden, der bei Operationsabschluss gesetzt wird (d2), wie folgt angezeigt:

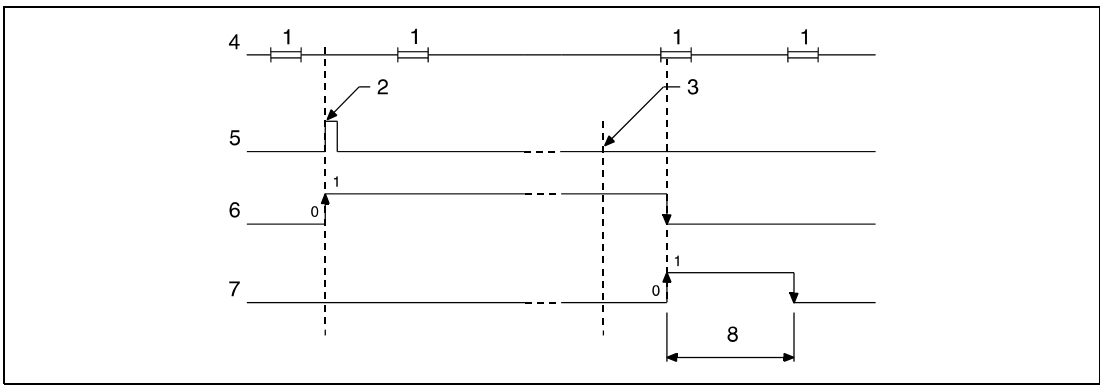
Kommunikations-Kanal-Flag

Dieses Flag wird während der Ausführung der ZNWR-Anweisung gesetzt. Das Flag wird mit der Verarbeitung der END-Anweisung in dem Zyklus zurückgesetzt, in dem die Schreib-Operation abgeschlossen wurde.

Operand der HOST-Station, der den Schreib-Operationsabschluss anzeigt

Dieser Operand wird mit der Verarbeitung der END-Anweisung in dem Zyklus gesetzt, in dem die Schreib-Operation abgeschlossen wurde. Bei der Verarbeitung der darauffolgenden END-Anweisung wird dieser Operand wieder zurückgesetzt.

Die folgende Abbildung zeigt die Operationen der HOST-Station während der Ausführung der ZNWR-Anweisung.



1 END-Verarbeitung
2 Ausführung der ZNWR-Anweisung
3 Abschluss der Operation
4 Programm der HOST-Station
5 ZNWR-Anweisung
6 Kommunikations-Kanal-Flag (SB32)
7 Operand der HOST-Station, der nach Abschluss der Operation gesetzt wird (d2)
8 Ein Zyklus

Der Abschluss-Status der ZNWR-Anweisung (normal, nicht normal) wird mit dem Register für den Verarbeitungsabschluss der ZNWR-Anweisung (SW33) wie folgt angezeigt:

Bei normalem (fehlerfreiem) Operationsabschluss ist der Inhalt des Registers SW33 gleich 0.

Bei nicht normalem (fehlerhaftem) Operationsabschluss wird der entsrechende Fehlercode in dem Register SW33 gespeichert.

HINWEIS *Weitere Informationen über Fehlercodes enthält das Handbuch "MELSECNET/10 für QnA-Netzwerk-Systeme".*

Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

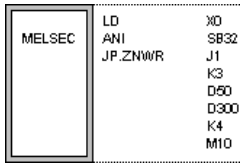
- Die mit n2 angegebene Sendedatenlänge liegt außerhalb des für die Speicherung vorgesehenen Bereichs des Operanden s1 (Fehlercode 4101).
- Das Netzwerk mit der mit Jn angegebenen Nummer existiert nicht (Fehlercode 4102).
- Die Station mit der in n1 angegebenen Stationsnummer existiert nicht (Fehlercode 4102).
- Die mit n2 angegebene Sendedatenlänge liegt nicht zwischen 1 und 230 (Fehlercode 4100).

Beispiel

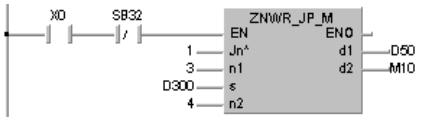
JP.ZNWR

Das folgende Programm schreibt mit positiver Flanke von X0 die Daten aus den Registern D300 bis D303 der HOST-Station in die Register D50 bis D53 der Station 3. Die HOST- und die Ziel-Station befinden sich in Netzwerk 1.

MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste

```
LD      X0
ANDN    SB32
ZNWR_JP_M  1 , 3 , D300 , 4 , D50 , M10
```

8.7.3 LRDP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●		

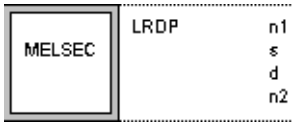
Operanden MELSEC A

	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
n1																	●	●								
s								●	●	●	●												11 [1]	●		●
d								●	●	●	●															
n2																	●	●								

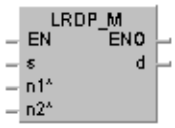
[1] Die Anzahl der Schritte bei der Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

GX IEC Developer

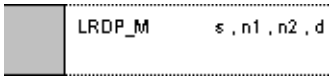
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



Variablen

Operand	Befehlswert	Datentyp
n1	Nummer der lokalen Station.	BIN-16-Bit
s	Startadresse des zu lesenden Datenbereichs der lokalen Station.	
d	Adressbereich der Master-Station, in dem die gelesenen Daten gespeichert werden.	
n2	Empfangsdatenlänge.	

Funktionsweise

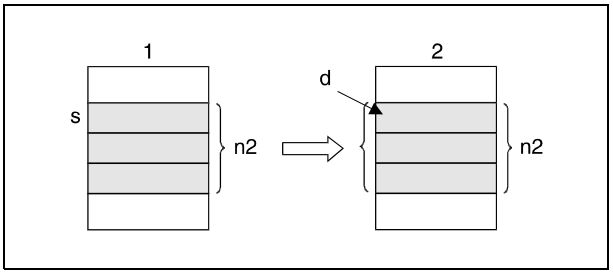
Lesen von Daten aus einer lokalen Station

LRDP Lese-Anweisung

Die LRDP-Anweisung liest Daten aus einer lokalen Station und speichert sie in einem vorgegebenen Adressbereich der Master-Station. Die Startadresse des zu lesenden Datenbereiches wird in s, die Anzahl der Daten (1 bis 32) in n2 und die Nummer der lokalen Station in n1 vorgegeben. Der Adressbereich der Master-Station, in dem die Daten gespeichert werden sollen, wird in d festgelegt.

Während der Ausführung der LRDP-Anweisung ist der Sondermerker M9200 in der Master-Station gesetzt. Nach Beendigung der Ausführung wird M9201 gesetzt. Beide Sondermerker bleiben auch nach der Verarbeitung der Anweisung gesetzt. Das Rücksetzen muss über das Ablaufprogramm erfolgen.

Es ist nicht möglich, zwei oder mehr LRDP-Anweisungen gleichzeitig auszuführen. Die LRDP-Anweisung kann auch nicht gleichzeitig mit einer LWTP-Anweisung auf eine lokale Station zugreifen.



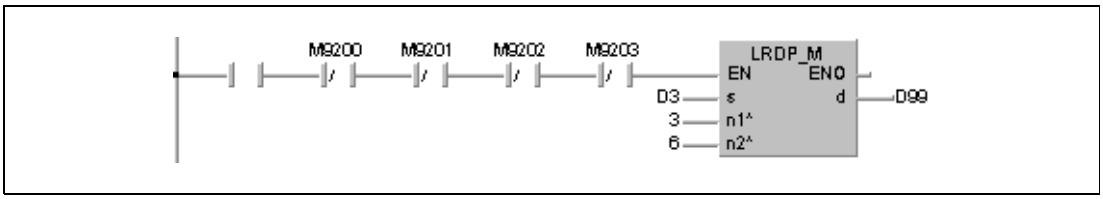
[1] Lokale Station (n1)
[2] Master-Station

HINWEIS *Die Sondermerker M9200, M9201, M9202 und M9203 sollten so als Verriegelung und Eingangsbedingung zur LRDP- bzw. LWTP-Anweisung programmiert werden, dass eine weitere Ausführung einer LRDP- und/oder LWTP-Anweisung ausgeschlossen ist.*

Das Ausführungsergebnis der LRDP-Anweisung wird in der Master-Station durch den Datenwert in D9200 angezeigt (siehe folgende Tabelle).

Datenwert LRDP D9200	Bedeutung
0	Ausführung fehlerfrei beendet.
2	Verarbeitungsfehler aufgrund fehlerhafter Adressierung: Die Adressen für s und d liegen außerhalb der für die Speicherung vorgesehenen Bereiche der Operanden. Der Wert von n1 liegt nicht zwischen 1 und 64. Der Wert von n2 liegt nicht zwischen 1 und 32.
3	Die lokale Station, auf die zugeriffen werden soll, befindet sich im Offline-Zustand und ist nicht ansprechbar.
4	An der angegebenen Stationsnummer befindet sich keine lokale Station (Fehlerverarbeitung).

Die folgende Abbildung zeigt die Verriegelung der LRDP-Anweisung.



Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- An der mit n1 angegebenen Nummer befindet sich keine lokale Station oder die angegebene Nummer liegt nicht zwischen 1 und 64.
- Die Anzahl der in n2 angegebenen Adressen in den in s und d angegebenen Operanden liegt außerhalb der für die Speicherung vorgesehenen Bereiche der Operanden.
- Die Anzahl der in n2 vorgegebenen Daten liegt nicht im Bereich von 1 bis 32.
- Die LRDP-Anweisung wird in einem Programm einer lokalen Station ausgeführt.

HINWEIS *Ist die CPU für den Data-Link-Betrieb nicht geeignet oder ist die Betriebsart Offline geschaltet, wird die LRDP-Anweisung nicht ausgeführt. Es erfolgt keine Fehlerkennung. M9200 wird jedoch gesetzt.*

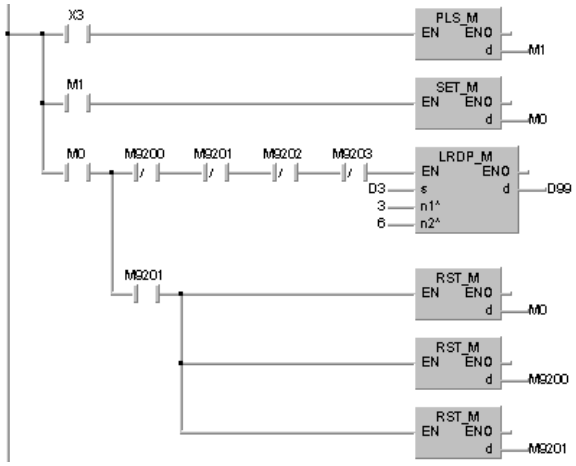
Beispiel LRDP

Das folgende Programm liest die Daten aus D3 bis D8 der lokalen Station 3 und speichert sie in D99 bis D104 der Master-Station. Nach dem Einschalten von X3 wird M0 gesetzt und die LRDP-Anweisung ausgeführt. Mit dem Beginn der Datenübertragung wird M9200 gesetzt. Ist die Übertragung beendet, wird M9201 gesetzt. Die Ausführung der LRDP-Anweisung erfolgt nicht, wenn bereits eine andere LRDP- oder LWTP-Anweisung ausgeführt wird. Nach Abschluss der Übertragung (M9201 ist gesetzt) werden im weiteren Verlauf des Programms M0, M9200 und M9201 zurückgesetzt.

MELSEC-Anweisungsliste

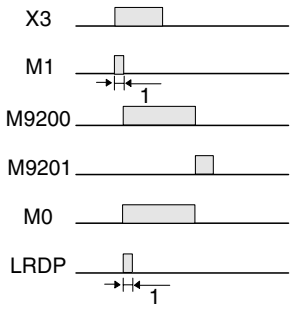
```
MELSEC  LD    X3
        PLS   M1
        LD    M1
        SET   M0
        LD    M0
        MPS
        ANI   M9200
        ANI   M9201
        ANI   M9202
        ANI   M9203
        LRDP  K3
              D3
              D99
              K6
        MPP
        AND   M9201
        RST   M0
        RST   M9200
        RST   M9201
        END
```

Kontaktplan



IEC-Anweisungsliste

```
        LD      X3
        PLS_M   M1
        LD      M1
        SET_M   M0
        LD      M0
        MPS
        ANDN    M9200
        ANDN    M9201
        ANDN    M9203
        ANDN    M9204
        LRDP_M  D3 , K3 , K6 , D99
        MPP
        AND     M9201
        RST_M   M0
        RST_M   M9200
        RST_M   M9201
        END_M
```



[1] Einmalige Ausführung

HINWEISE *Der zu M1 gehörende Kontakt muss als gepulstes Signal ausgeführt sein, da andernfalls eine vollständige Ausführung der LRDP-Anweisung nicht möglich ist.*

Der zu M0 gehörende Kontakt sollte über eine SET-Anweisung gesetzt werden. Wird anstelle der SET-Anweisung eine OUT- oder PLS-Anweisung programmiert, kann es bei der Ausführung der LRDP-Anweisung zu Fehlern in der Verarbeitung kommen.

Damit eine gleichzeitige Ausführung von zwei LRDP-Anweisungen ausgeschlossen werden kann, müssen die Sondermerker M9200 und M9201 als Verriegelung programmiert werden.

Soll im gleichen Programm mit einer LWTP-Anweisung auf eine lokale Station zugegriffen werden, müssen die Sondermerker M9202 und M9203 zusätzlich als Verriegelung programmiert werden.

8.7.4 LWTP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●		

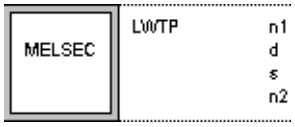
Operanden MELSEC A

	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
n1																	●	●								
d								●	●	●	●												11[1]	●		●
s								●	●	●	●															
n2																	●	●								

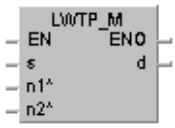
[1] Die Anzahl der Schritte bei der Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

GX IEC Developer

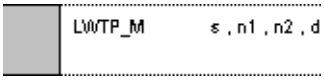
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



Variablen

Operand	Befehlswert	Datentyp
n1	Nummer der lokalen Station.	BIN-16-Bit
d	Adressbereich der lokalen Station, in den die Daten geschrieben werden.	
s	Startadresse des zu schreibenden Datenbereichs der Master-Station.	
n2	Sendedatenlänge.	

Funktionsweise

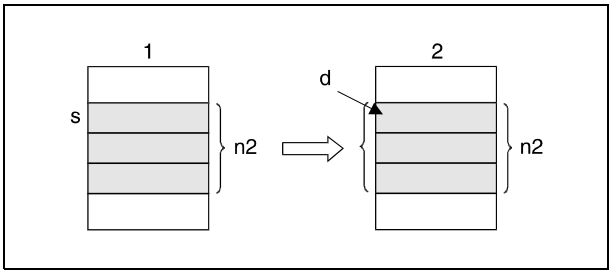
Schreiben von Daten in eine lokale Station

LWTP Schreib-Anweisung

Die LWTP-Anweisung schreibt Daten der Master-Station in einen vorgegebenen Adressbereich einer lokalen Station. Die Startadresse der zu übertragenden Daten wird in s, die Anzahl der Daten (1 bis 32) in n2 und die Nummer der lokalen Station in n1 vorgegeben. Der Adressbereich der lokalen Station, in den die Daten übertragen werden sollen, wird in d festgelegt.

Während der Ausführung der LWTP-Anweisung ist der Sondermerker M9202 in der Master-Station gesetzt. Nach Beendigung der Ausführung wird M9203 gesetzt. Beide Sondermerker bleiben auch nach der Verarbeitung der Anweisung gesetzt. Das Rücksetzen muss über das Ablaufprogramm erfolgen.

Es ist nicht möglich, zwei oder mehr LWTP-Anweisungen gleichzeitig auszuführen. Die LWTP-Anweisung kann auch nicht gleichzeitig mit einer LRDP-Anweisung auf eine lokale Station zugreifen.



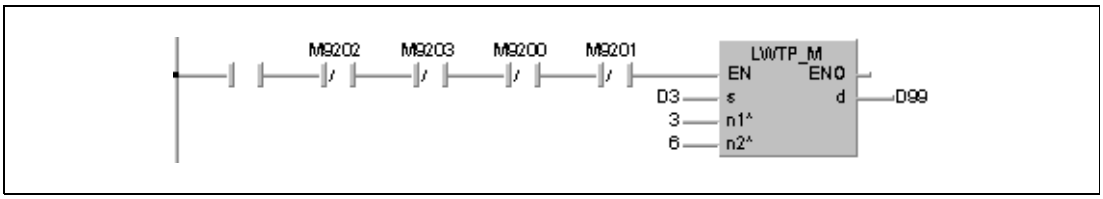
[1] Master-Station
[2] Lokale Station (n1)

HINWEIS *Die Sondermerker M9200, M9201, M9202 und M9203 sollten so als Verriegelung und Eingangsbedingung zur LRDP- bzw. LWTP-Anweisung programmiert werden, dass eine weitere Ausführung einer LRDP- und/oder LWTP-Anweisung ausgeschlossen ist.*

Das Ausführungsergebnis der LWTP-Anweisung wird in der Master-Station durch den Datenwert in D9001 angezeigt (siehe folgende Tabelle).

Datenwert LWTP D9201	Bedeutung
0	Ausführung fehlerfrei beendet.
2	Verarbeitungsfehler aufgrund fehlerhafter Adressierung: Die Adressen für s und d liegen außerhalb der für die Speicherung vorgesehenen Bereiche der Operanden. Der Wert von n1 liegt nicht zwischen 1 und 64. Der Wert von n2 liegt nicht zwischen 1 und 32.
3	Die lokale Station, auf die zugeriffen werden soll, befindet sich im Offline-Zustand und ist nicht ansprechbar.
4	An der angegebenen Stationsnummer befindet sich keine lokale Station (Fehlerverarbeitung).

Die folgende Abbildung zeigt die Verriegelung LWTP-Anweisung.



Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- An der mit n1 angegebenen Nummer befindet sich keine lokale Station oder die angegebene Nummer liegt nicht zwischen 1 und 64.
- Die Anzahl der in n2 angegebenen Adressen in den in s und d angegebenen Operanden liegt außerhalb der für die Speicherung vorgesehenen Bereiche der Operanden.
- Die Anzahl der in n2 vorgegebenen Daten liegt nicht im Bereich von 1 bis 32.
- Die LWTP-Anweisung wird in einem Programm einer lokalen Station ausgeführt.

HINWEIS *Ist die CPU für den Data-Link-Betrieb nicht geeignet oder ist die Betriebsart Offline geschaltet, wird die LWTP-Anweisung nicht ausgeführt. Es erfolgt keine Fehlerkennung. M9202 wird jedoch gesetzt.*

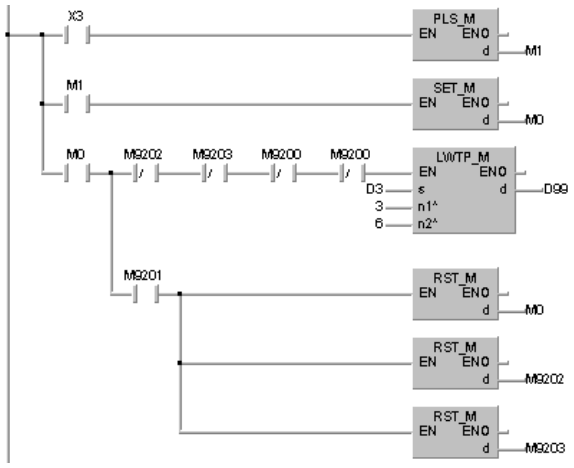
Beispiel LWTP

Das folgende Programm überträgt die Daten aus D99 bis D104 der Master-Station nach D3 bis D8 der lokalen Station 3. Nach dem Einschalten von X3 wird M0 gesetzt und die LWTP-Anweisung ausgeführt. Mit dem Beginn der Datenübertragung wird M9202 gesetzt. Ist die Übertragung beendet, wird M9203 gesetzt. Die Ausführung der LWTP-Anweisung erfolgt nicht, wenn bereits eine andere LWTP- oder LRDP-Anweisung ausgeführt wird. Nach Abschluss der Übertragung werden im weiteren Verlauf des Programms M0, M9202 und M9203 zurückgesetzt.

MELSEC-Anweisungsliste

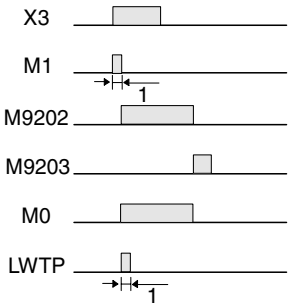
```
MELSEC  LD      X3
        PLS     M1
        LD      M1
        SET     M0
        LD      M0
        MPS
        ANI     M9202
        ANI     M9203
        ANI     M9200
        ANI     M9201
        LWTP    K3
                D3
                D99
                K6
        MPP
        AND     M9203
        RST     M0
        RST     M9202
        RST     M9203
        END
```

Kontaktplan



IEC-Anweisungsliste

```
        LD      X3
        PLS_M   M1
        LD      M1
        SET_M   M0
        LD      M0
        MPS
        ANDN    M9202
        ANDN    M9203
        ANDN    M9200
        ANDN    M9201
        LWTP_M  D3 , K3 , K6 , D99
        MPP
        AND     M9203
        RST_M   M0
        RST_M   M9202
        RST_M   M9203
        END_M
```



[1] Einmalige Ausführung

HINWEISE *Der zu M1 gehörende Kontakt muss als gepulstes Signal ausgeführt sein, da andernfalls eine vollständige Ausführung der LWTP-Anweisung nicht möglich ist.*

Der zu M0 gehörende Kontakt sollte über eine SET-Anweisung gesetzt werden. Wird anstelle der SET-Anweisung eine OUT- oder PLS-Anweisung programmiert, kann es bei der Ausführung der LWTP-Anweisung zu Fehlern in der Verarbeitung kommen.

Damit eine gleichzeitige Ausführung von zwei LWTP-Anweisungen ausgeschlossen werden kann, müssen die Sondermerker M9202 und M9203 als Verriegelung programmiert werden.

Soll im gleichen Programm mit einer LRDP-Anweisung auf eine lokale Station zugegriffen werden, müssen die Sondermerker M9200 und M9201 zusätzlich als Verriegelung programmiert werden.

8.7.5 RFRP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	

Operanden MELSEC A

	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)								Konstante		Pointer		Ebene				
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
n1																	●	●								
n2																	●	●								
d											●												11 [1] ●	●		●
n3																	●	●								

[1] Die Anzahl der Schritte bei der Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

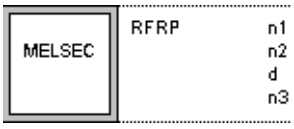
Operanden MELSEC Q

	Operanden								Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
n1	●	●	●	—	—	—	—	●	—	SM0	11
d1	—	● [1]	—	—	—	—	—	—	—		
n2	●	●	●	—	—	—	—	●	—		
d2	●	●	●	—	—	—	—	—	—		

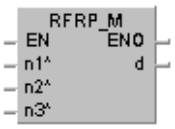
[1] Nur Link-Register

A-Serie GX IEC Developer

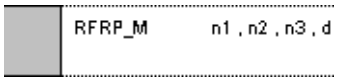
MELSEC-Anweisungsliste



Kontaktplan

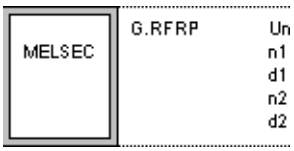


IEC-Anweisungsliste

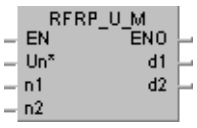


Q-Serie GX IEC Developer

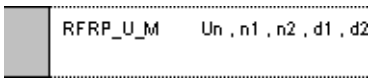
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



Variablen

Operand		Befehlswert	Datentyp
A-Serie	**Q-Serie**		
n1	Un	Kopf-Ein-/Ausgangsadresse des Sondermoduls in der ausgelagerten Ein-/Ausgabe-Station. ●[1]	BIN-16-Bit
n2	n1	Erste Adresse des Pufferspeichers des Sondermoduls, in dem die zu lesenden Daten gespeichert sind.	
d	d1	Erste Adresse des Link-Registers der HOST-Station, in dem die gelesenen Daten gespeichert werden.	Adresse
n3	n2	Empfangsdatenlänge.	BIN-16-Bit
	d2	Operand, der nach vollständiger Verarbeitung der Anweisung für einen Programmzyklus gesetzt wird.	Bit

HINWEIS

●[1] Die Kopf-Ein-/Ausgangsadresse des Netzwerkmoduls der HOST-Station muss zwischen 0 und FE$_H$ liegen. Es ist zu beachten, dass der Compiler für Un eine hexadezimale Adressangabe erwartet. Eine Dezimalangabe wird automatisch in eine Hexadezimalzahl umgewandelt.

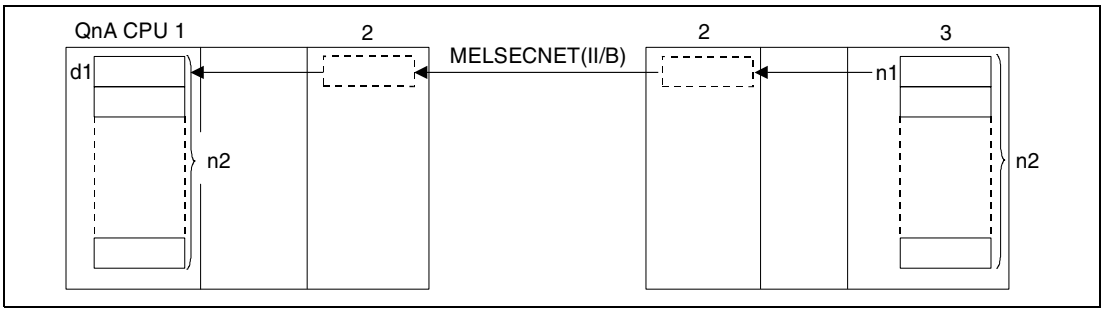
Funktions-weise

Lesen von Daten aus einer Remote-Station

RFRP Lese-Anweisung

Die RFRP-Anweisung liest Daten aus dem Pufferspeicher eines Sondermoduls einer Remote-Station im MELSECNET. Die Anzahl der zu lesenden Datenwörter ist in n2 (A-Serie = n3) angegeben. Der Adressbereich des Pufferspeichers ist ab n1 (A-Serie = n2) angegeben. Die Ein-/Ausgangsadresse des angeschlossenen Sondermoduls ist in Un (A-Serie = n1) angegeben. Die gelesenen Daten werden ab dem in d1 (A-Serie = d) angegebenen Link-Register der Master-Station gespeichert.

Nach dem Abschluss der Lese-Operation in der ausgelagerten Ein-/Ausgabe-Station wird der in d2 angegebene Operand gesetzt (nur QnA-Serie).



[1] HOST-Station (Master-Station)

[2] Data-Link-Modul

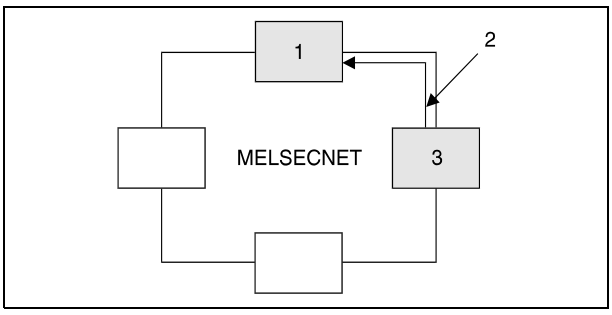
[3] Sondermodul (Ziel-Station/ausgelagerte Ein-/Ausgabe-Station)

HINWEIS *Auch wenn nur eine Lese-Operation ausgeführt wird, muss der Adressbereich der Link-Register in d1 (A-Serie = d) innerhalb der MELSECNET-Parametrierung der Remote- und Master-Station liegen.*

Bei den folgenden Beschreibungen der Ein-/Ausgangsadressen des Sondermoduls sind die Adressen der QnA-Serie beschrieben. Diese Angaben gelten gleichermaßen für die A-Serie. In diesem Fall ist für n der Wert von n1 einzusetzen (z. B. QnA-Serie = Y(n+E) $\Rightarrow$ A-Serie = Y(n1+E)).

Während der Ausführung der RFRP-Anweisung ist der Ausgang Y(n+E) gesetzt. X(n+1E) wird gesetzt, sobald die Ausführung der Anweisung abgeschlossen ist. Y(n+E) bleibt auch nach Ausführungsende gesetzt und muss daher über das Ablaufprogramm zurückgesetzt werden. Die Adressierung erfolgt automatisch und darf nicht verändert werden.

Lese-Operationen von ausgelagerten Ein-/Ausgabe-Stationen können durch eine im MELSECNET angeschlossene Master-Station ausgeführt werden.



[1] Station, die die RFRP-Anweisung ausführt (Master-Station)

[2] Lese-Operation der Daten aus dem Sondermodul

[3] Ausgelagerte Ein-/Ausgabe-Station

Kann die RFRP-Anweisung aufgrund eines Fehlers im angesprochenen Sondermodul der Remote-Station nicht ausgeführt werden, wird X(n+1D) gesetzt. In diesem Fall sollte das entsprechende Modul überprüft werden. X(n+1D) wird zurückgesetzt, sobald Y(n+D) gesetzt wird.

Die mit Un angegebenen Kopf-Ein-/Ausgangsadressen von Sondermodulen werden bei 4-stelliger Darstellung in den oberen 3 Stellen abgelegt. Beispielsweise muss für die Angabe der Adressen X/Y0200 der Wert 20 angegeben werden (nur Q-Serie).

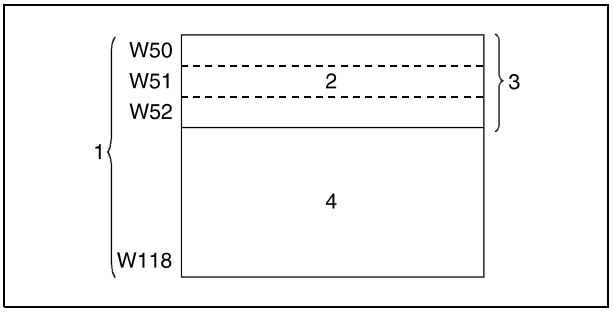
HINWEIS *Weitere Informationen über den in n1 (A-Serie = n2) anzugebenden möglichen Adressbereich von Pufferspeichern in Sondermodulen sind der Anleitung des entsprechenden Sondermoduls zu entnehmen.*

Die mit n2 (A-Serie = n3) angegebene Empfangsdatenlänge (Anzahl der Datenwörter) kann zwischen 1 und 16 liegen.

Der Adressbereich der Link-Register in d1 (A-Serie = d) muss innerhalb des Link-Parameterbereichs der Remote- und Master-Station liegen.

Der Bereich der Link-Register Wxxx zwischen Master- und Remote-Station muss genau differenziert werden. Die Anzahl der Link-Register-Adressen, die vom Betriebssystem genutzt werden, entspricht der Anzahl von Sondermodulen, die sich in den Remote-Stationen eines Netzwerks befinden. Der zur Datenspeicherung nutzbare Bereich ist der Parameterbereich abzüglich der vom Betriebssystem genutzten Link-Register-Adressen.

Das Beispiel in der folgenden Abbildung zeigt eine Aufteilung eines Link-Register-Bereiches. Der Bereich zwischen Master- und Remote-Station ist in den Parametern mit W050 bis W118 (A-Serie = W09F) festgelegt. In diesem Bereich befinden sich 2 Sondermodule, so dass die ersten beiden Link-Register W50 und W51 (2 Adressen) durch das Betriebssystem der CPU belegt sind. Der nutzbare Bereich zur Datenspeicherung liegt demnach zwischen W52 und W118 (A-Serie = W09F).



[1] Mit Link-Parametern festgelegter Bereich
[2] Wird vom System genutzt
[3] Anzahl der Register für die Anzahl der Sondermodule
[4] Für die Programmierung vorgesehener Bereich

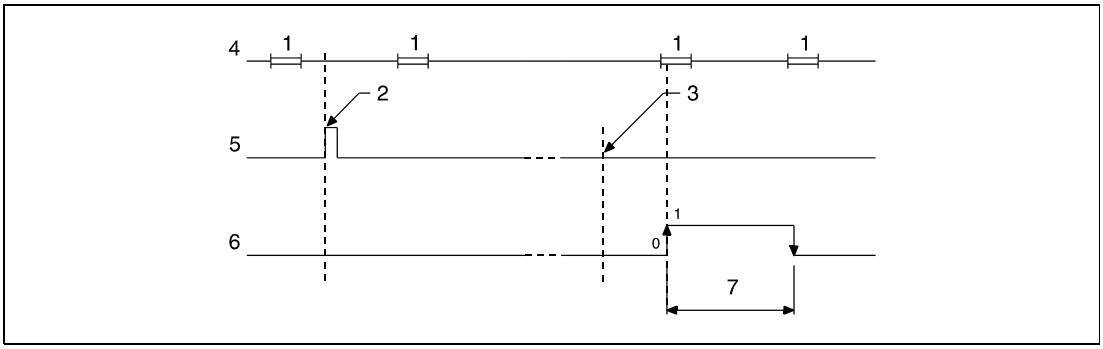
Die RFRP- und RTOP-Anweisungen können nicht von mehreren Stellen gleichzeitig von dem gleichen Sondermodul ausgeführt werden. Bei gleichzeitiger Ausführung der Anweisungen von zwei oder mehr Stellen wird durch den Handshake-Betrieb der beiden aktiven Stationen eine Ausführung weiterer RFRP- und RTOP-Anweisungen verhindert.

Die Ein- und Ausgänge X(n+1E) und Y(n+E) sollten so als Verriegelung der RFRP-Anweisung programmiert werden, dass die gleichzeitige Ausführung einer anderen RFRP- oder RTOP-Anweisung ausgeschlossen wird.

Der Operand der HOST-Station, der den Lese-Operationsabschluss anzeigt (d2), wird mit der Verarbeitung der END-Anweisung in dem Zyklus gesetzt, in dem die Lese-Operation abgeschlossen wurde. Bei der Verarbeitung der darauffolgenden END-Anweisung wird dieser Operand wieder zurückgesetzt (nur Q-Serie).

Die MELSEC A-Serie verfügt für die Datenübertragung im MELSECNET über eine Reihe von Sonderregistern, die verschiedene Kommunikationszustände registrieren. So wird beispielsweise der Zustand der Remote-E/A-Stationen über die Sonderregister D9228 bis D9231 registriert. Parameterzugriffe werden über die Sondermerker M9224 bis M9227 ausgewertet (nur A-Serie).

Die folgende Abbildung zeigt die Operationen der HOST-Station während der Ausführung der RFRP-Anweisung.



[1] END-Verarbeitung

[2] Ausführung der RFRP-Anweisung

[3] Abschluss der Operation

[4] Programm der HOST-Station

[5] RFRP-Anweisung

[6] Operand der HOST-Station, der nach Abschluss der Operation gesetzt wird (d2) (nur Q-Serie)

[7] Ein Zyklus

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die mit Un (A-Serie = n1) angegebene Ein-/Ausgangsadresse ist keine Adresse einer ausgelagerten Ein-/Ausgabe-Station (Q-Serie = Fehlercode 4102).
- Die mit n1 (A-Serie = n2) angegebene Ein-/Ausgangsadresse ist keine Kopf-Ein-/Ausgangsadresse eines Sondermoduls (Fehlercode 4102).
- Die in n2 (A-Serie = n3) angegebenen Anzahl von Adressen liegt außerhalb des Adressbereichs ab d1 (A-Serie = d, W0 bis W3FF)(Q-Serie = Fehlercode 4101).
- Das mit Un (A-Serie = n1) angegebene Netzwerk existiert nicht (Fehlercode 2413).
- Der für n2 (A-Serie = n3) angegebene Wert liegt nicht zwischen 1 und 16 (Fehlercode 4100).

Beispiel RFRP (A-Serie)

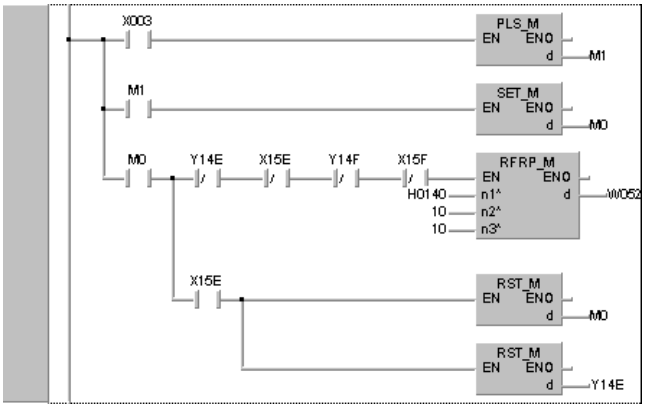
Das folgende Programm liest die Daten aus 10 aufeinanderfolgenden Adressen - beginnend mit Adresse 10 - aus einem Sondermodul (z.B. A68AD). Das Modul befindet sich in der zweiten Remote-Station. Der Adressbereich liegt zwischen 140 und 15F. Die gelesenen Daten werden in den Link-Registern W52 bis W61 der Master-Station gespeichert.

Nach dem Einschalten von X3 wird M0 gesetzt und die RFRP-Anweisung ausgeführt. Mit dem Beginn der Datenübertragung wird Y(n1+E) = Y14E gesetzt. Ist die Übertragung beendet, wird X(n1+1E) = X15E gesetzt. Die Ausführung der RFRP-Anweisung erfolgt nicht, wenn bereits eine andere RFRP- oder RTOP-Anweisung ausgeführt wird. Nach Abschluss der Übertragung werden im weiteren Verlauf des Programms M0 und Y14E zurückgesetzt.

MELSEC-Anweisungsliste

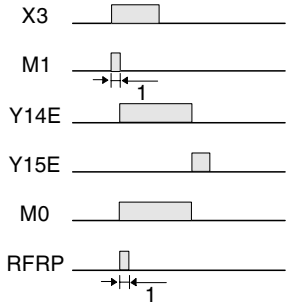
```
MELSEC  LD      X003
        PLS     M1
        LD      M1
        SET     M0
        LD      M0
        MPS
        ANI     Y14E
        ANI     X15E
        ANI     Y14F
        ANI     X15F
        RFRP    H0140
                K10
                W052
                K10
        MPP
        AND     X15E
        RST     M0
        RST     Y14E
        END
```

Kontaktplan



IEC-Anweisungsliste

```
LD          X3
PLS_M       M1
LD          M1
SET_M       M0
LD          M0
MPS
ANDN        Y14E
ANDN        X15E
ANDN        Y14F
ANDN        X15F
RFRP_M      H0140 , K10 , W052 , K10
MPP
AND         X15E
RST_M       M0
RST_M       Y14E
END_M
```



[1] Einmalige Ausführung

HINWEISE *Der zu M1 gehörende Kontakt muss als gepulstes Signal ausgeführt sein, da andernfalls eine vollständige Ausführung der RFRP-Anweisung nicht möglich ist.*

Der zu M0 gehörende Kontakt sollte über eine SET-Anweisung gesetzt werden. Wird anstelle der SET-Anweisung eine OUT- oder PLS-Anweisung programmiert, kann es bei der Ausführung der RFRP-Anweisung zu Fehlern in der Verarbeitung kommen.

Damit eine gleichzeitige Ausführung von zwei RFRP-Anweisungen ausgeschlossen werden kann, müssen der Ausgang Y14E und der Eingang X15E als Verriegelung programmiert werden.

Soll im gleichen Programm mit einer RTOP-Anweisung auf diese Station zugegriffen werden, müssen der Ausgang Y14F und der Eingang X15F zusätzlich als Verriegelung programmiert werden.

8.7.6 RTOP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	

Operanden MELSEC A

	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)								Konstante		Pointer		Ebene				
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
n1																	●	●								
n2																	●	●								
s											●												11 [1] ●	●		●
n3																	●	●								

[1] Die Anzahl der Schritte bei der Verwendung einer AnA, AnAS oder AnU CPU ist dem Abs. 3.10.2 „Bei einer AnA, AnAS und AnU CPU“ dieser Programmieranleitung zu entnehmen.

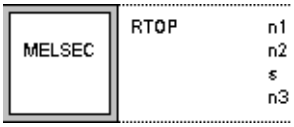
Operanden MELSEC Q

	Operanden								Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
n1	●	●	●	—	—	—	—	●	—	SM0	11
s	—	● [1]	—	—	—	—	—	—	—		
n2	●	●	●	—	—	—	—	—	—		
d	●	●	●	—	—	—	—	●	—		

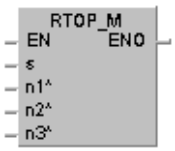
[1] Nur Link-Register

A-Serie GX IEC Developer

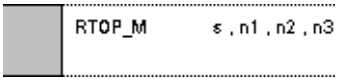
MELSEC-Anweisungsliste



Kontaktplan

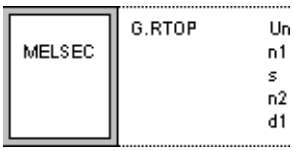


IEC-Anweisungsliste

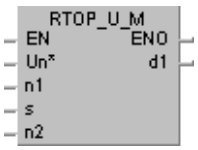


Q-Serie GX IEC Developer

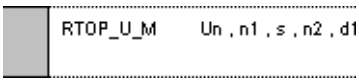
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



Variablen

Operand		Befehlswert	Datentyp
A-Serie	**Q-Serie**		
n1	Un	Kopf-Ein-/Ausgangsadresse des Sondermoduls in der ausgelagerten Ein-/ Ausgabe-Station. ●[1]	BIN-16-Bit
n2	n1	Erste Adresse des Pufferspeichers des Sondermoduls, in den die Daten geschrieben werden.	
s	s	Erste Adresse des Link-Registers der HOST-Station, in dem die zu schreibenden Daten gespeichert sind.	Adresse
n3	n2	Sendedatenlänge.	BIN-16-Bit
	d	Operand, der nach vollständiger Verarbeitung der Anweisung für einen Programmzyklus gesetzt wird.	Bit

HINWEIS

●[1] Die Kopf-Ein-/Ausgangsadresse des Netzwerkmoduls der HOST-Station muss zwischen 0 und FE$_H$ liegen. Es ist zu beachten, dass der Compiler für Un eine hexadezimale Adressangabe erwartet. Eine Dezimalangabe wird automatisch in eine Hexadezimalzahl umgewandelt.

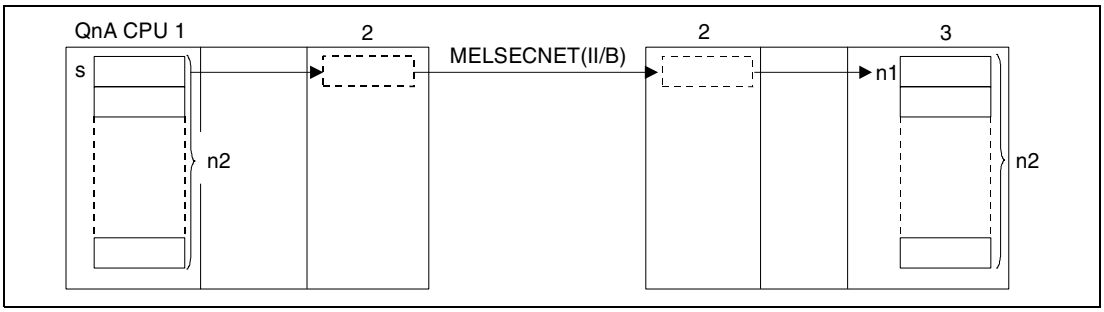
Funktionsweise

Schreiben von Daten in eine Remote-Station

RTOP Schreib-Anweisung

Die RTOP-Anweisung schreibt Daten in den Pufferspeicher eines Sondermoduls einer Remote-Station im MELSECNET. Die Anzahl der zu schreibenden Datenwörter ist in n2 (A-Serie = n3) angegeben. Der Adressbereich des Pufferspeichers ist ab n1 (A-Serie = n2) angegeben. Die Ein-/Ausgangsadresse des angeschlossenen Sondermoduls ist in Un (A-Serie = n1) angegeben. Die zu schreibenden Daten sind ab dem in s angegebenen Link-Register der Master-Station gespeichert.

Nach dem Abschluss der Schreib-Operation in der ausgelagerten Ein-/Ausgabe-Station wird der in d angegebene Operand gesetzt (nur Q-Serie).



[1] HOST-Station (Master-Station)

[2] Data-Link-Modul

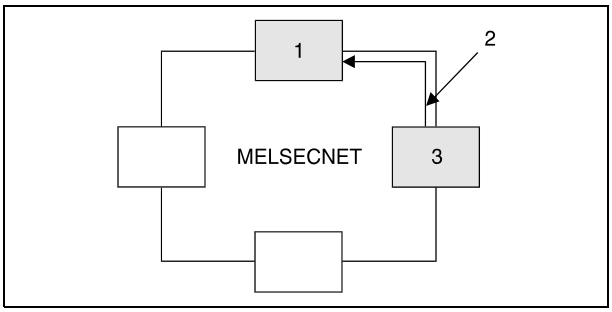
[3] Sondermodul (Ziel-Station/ausgelagerte Ein-/Ausgabe-Station))

HINWEIS *Auch wenn nur eine Schreib-Operation ausgeführt wird, muss der Adressbereich der Link-Register in s innerhalb der MELSECNET-Parametrierung der Remote- und Master-Station liegen.*

Bei den folgenden Beschreibungen der Ein-/Ausgangsadressen des Sondermoduls sind die Adressen der QnA-Serie beschrieben. Diese Angaben gelten gleichermaßen für die A-Serie. In diesem Fall ist für n der Wert von n1 einzusetzen (z. B. QnA-Serie = Y(n+F) $\Rightarrow$ A-Serie = Y(n1+F)).

Während der Ausführung der RTOP-Anweisung ist der Ausgang Y(n+F) gesetzt. X(n+1F) wird gesetzt, sobald die Ausführung der Anweisung abgeschlossen ist. Y(n+F) bleibt auch nach Ausführungsende gesetzt und muss daher über das Ablaufprogramm zurückgesetzt werden. Die Adressierung erfolgt automatisch und darf nicht verändert werden.

Schreib-Operationen mit ausgelagerten Ein-/Ausgabe-Stationen können durch eine im MELSECNET angeschlossene Master-Station ausgeführt werden.



[1] Station, die die RTOP-Anweisung ausführt (Master-Station)

[2] Schreib-Operation der Daten in das Sondermodul

[3] Ausgelagerte Ein-/Ausgabe-Station

Kann die RTOP-Anweisung aufgrund eines Fehlers im angesprochenen Sondermodul der Remote-Station nicht ausgeführt werden, wird X(n+1D) gesetzt. In diesem Fall sollte das entsprechende Modul überprüft werden. X(n+1D) wird zurückgesetzt, sobald Y(n+D) gesetzt wird.

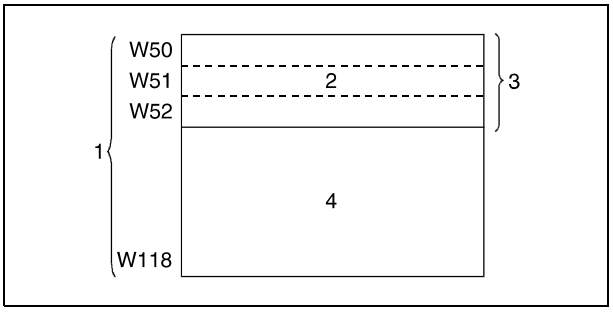
Die mit Un angegebenen Kopf-Ein-/Ausgangsadressen von Sondermodulen werden bei 4-stelliger Darstellung in den oberen 3 Stellen abgelegt. Beispielsweise muss für die Angabe der Adressen X/Y0200 der Wert 20 angegeben werden (nur Q-Serie).

HINWEIS *Weitere Informationen über den in n1 (A-Serie = n2) anzugebenden, möglichen Adressbereich von Pufferspeichern in Sondermodulen sind der Anleitung des entsprechenden Sondermoduls zu entnehmen.*

Die mit n2 (A-Serie = n3) angegebene Sendedatenlänge (Anzahl der Datenwörter) kann zwischen 1 und 16 liegen.

Der Bereich der Link-Register Wxxx zwischen Master- und Remote-Station muss genau differenziert werden. Die Anzahl der Link-Register-Adressen, die vom Betriebssystem genutzt werden, entspricht der Anzahl von Sondermodulen, die sich in den Remote-Stationen eines Netzwerks befinden. Der zur Datenspeicherung nutzbare Bereich ist der Parameterbereich abzüglich der vom Betriebssystem genutzten Link-Register-Adressen.

Das Beispiel in der folgenden Abbildung zeigt eine Aufteilung eines Link-Register-Bereiches. Der Bereich zwischen Master- und Remote-Station ist in den Parametern mit W050 bis W118 (A-Serie = W09F) festgelegt. In diesem Bereich befinden sich 2 Sondermodule, so dass die ersten beiden Link-Register W50 und W51 (2 Adressen) durch das Betriebssystem der CPU belegt sind. Der nutzbare Bereich zur Datenspeicherung liegt demnach zwischen W52 und W118 (A-Serie = W09F).



[1] Mit Link-Parametern festgelegter Bereich
[2] Wird vom System genutzt
[3] Anzahl der Register für die Anzahl der Sondermodule
[4] Für die Programmierung vorgesehener Bereich

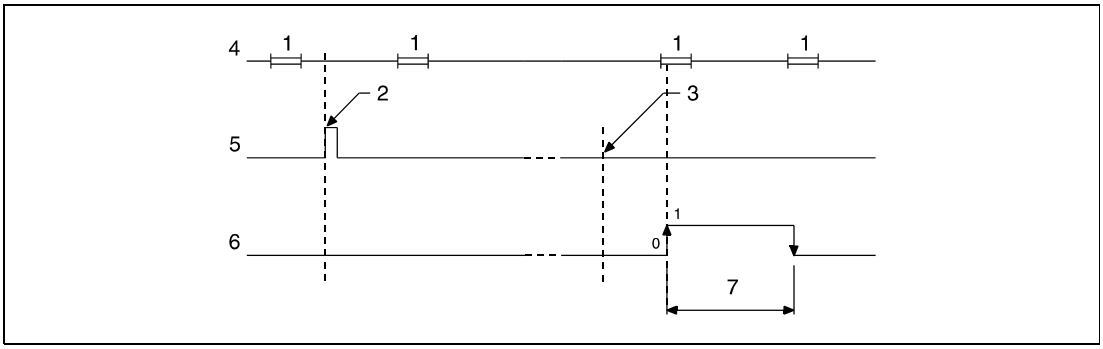
Die RFRP- und RTOP-Anweisungen können nicht von mehreren Stellen gleichzeitig von dem gleichen Sondermodul ausgeführt werden. Bei gleichzeitiger Ausführung der Anweisungen von zwei oder mehr Stellen wird durch den Handshake-Betrieb der beiden aktiven Stationen eine Ausführung weiterer RFRP- und RTOP-Anweisungen verhindert.

Die Ein- und Ausgänge X(n+1F) und Y(n+F) sollten so als Verriegelung der RTOP-Anweisung programmiert werden, dass die gleichzeitige Ausführung einer anderen RTOP- oder RFRP-Anweisung ausgeschlossen wird.

Der Operand der HOST-Station, der den Schreib-Operationsabschluss anzeigt (d), wird mit der Verarbeitung der END-Anweisung in dem Zyklus gesetzt, in dem die Lese-Operation abgeschlossen wurde. Bei der Verarbeitung der darauffolgenden END-Anweisung wird dieser Operand wieder zurückgesetzt (nur Q-Serie).

Die MELSEC A-Serie verfügt für die Datenübertragung im MELSECNET über eine Reihe von Sonderregistern, die verschiedene Kommunikationszustände registrieren. So wird beispielsweise der Zustand der Remote-E/A-Stationen über die Sonderregister D9228 bis D9231 registriert. Parameterzugriffe werden über die Sondermerker M9224 bis M9227 ausgewertet (nur A-Serie).

Die folgende Abbildung zeigt die Operationen der HOST-Station während der Ausführung der RTOP-Anweisung.



[1] END-Verarbeitung

[2] Ausführung der RTOP-Anweisung

[3] Abschluss der Operation

[4] Programm der HOST-Station

[5] RTOP-Anweisung

[6] Operand der HOST-Station, der nach Abschluss der Operation gesetzt wird (d)(nur Q-Serie)

[7] Ein Zyklus

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die mit Un (A-Serie = n1) angegebene Ein-/Ausgangsadresse ist keine Adresse einer ausgelagerten Ein-/Ausgabe-Station (Q-Serie = Fehlercode 4102).
- Die mit n1 (A-Serie = n2) angegebene Ein-/Ausgangsadresse ist keine Kopf-Ein-/Ausgangsadresse eines Sondermoduls (Q-Serie = Fehlercode 4102).
- Die in n2 (A-Serie = n3) angegebenen Anzahl von Adressen liegt außerhalb des Adressbereichs ab s (A-Serie = W0 bis W3FF)(Q-Serie = Fehlercode 4101).
- Das mit Un (A-Serie = n1) angegebene Netzwerk existiert nicht (Fehlercode 2413).
- Der für n2 (A-Serie = n3) angegebene Wert liegt nicht zwischen 1 und 16 (Fehlercode 4100).

Beispiel RTOP (A-Serie)

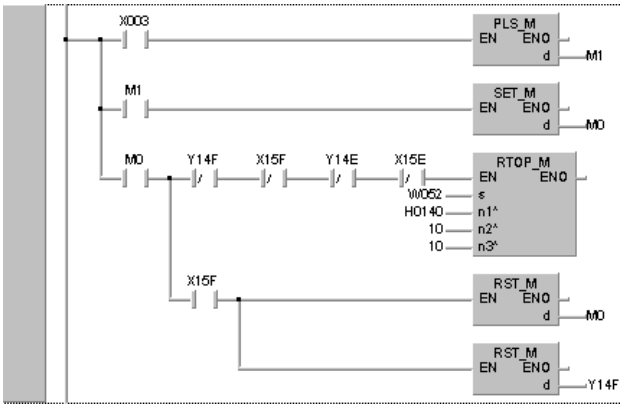
Das folgende Programm schreibt die Daten aus den Link-Registern W52 bis W61 der Master-Station in 10 aufeinanderfolgende Adressen eines Sondermoduls (z.B. A68AD) der zweiten Remote-Station. Der Adressbereich liegt zwischen 140 und 15F. Der Adressbereich, in den die Daten übertragen werden sollen, beginnt mit Adresse 10.

Nach dem Einschalten von X3 wird M0 gesetzt und die RTOP-Anweisung ausgeführt. Mit dem Beginn der Datenübertragung wird Y(n1+F) = Y14F gesetzt. Ist die Übertragung beendet, wird X(n1+1F) = X15F gesetzt. Die Ausführung der RTOP-Anweisung erfolgt nicht, wenn bereits eine andere RTOP- oder RFRP-Anweisung ausgeführt wird. Nach Abschluss der Übertragung werden im weiteren Verlauf des Programms M0 und Y14F zurückgesetzt.

MELSEC-Anweisungsliste

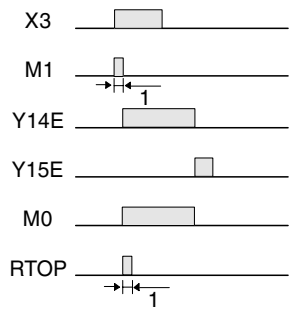
```
MELSEC   LD     X003
         PLS    M1
         LD     M1
         SET    M0
         LD     M0
         MPS
         ANI    Y14F
         ANI    X15F
         ANI    Y14E
         ANI    X15E
         RTOP   H0140
                K10
                W052
                K10
         MPP
         AND    X15F
         RST    M0
         RST    Y14F
         END
```

Kontaktplan



IEC-Anweisungsliste

```
LD        X3
PLS_M     M1
LD        M1
SET_M     M0
LD        M0
MPS
ANDN      Y14F
ANDN      X15F
ANDN      Y14E
ANDN      X15E
RTOP_M    H0140 , K10 , W052 , K10
MPP
AND       X15F
RST_M     M0
RST_M     Y14F
END_M
```



[1] Einmalige Ausführung

HINWEISE *Der zu M1 gehörende Kontakt muss als gepulstes Signal ausgeführt sein, da andernfalls eine vollständige Ausführung der RTOP-Anweisung nicht möglich ist.*

Der zu M0 gehörende Kontakt sollte über eine SET-Anweisung gesetzt werden. Wird anstelle der SET-Anweisung eine OUT- oder PLS-Anweisung programmiert, kann es bei der Ausführung der RTOP-Anweisung zu Fehlern in der Verarbeitung kommen.

Damit die gleichzeitige Ausführung von zwei RTOP-Anweisungen ausgeschlossen werden kann, müssen Ausgang Y14F und der Eingang X15F als Verriegelung programmiert werden.

Soll im gleichen Programm mit einer RFRP-Anweisung auf diese Station zugegriffen werden, müssen der Ausgang Y14E und der Eingang X15E zusätzlich als Verriegelung programmiert werden.

8.8 Lesen und Schreiben von Routing-Informationen

Mit diesen Anweisungen ist das Lesen und Schreiben von Routing-Informationen möglich. Die Routing-Parameter umfassen die Netzwerks- und Stationsnummer der Relais-Station und die Stationsnummer der Routing-Station.

Funktion	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor
Lesen von Routing-Informationen	Z.RTREAD	RTREAD_M
	ZP.RTREAD	RTREADP_M
Schreiben von Routing-Informationen	Z.RTWRITE	RTWRITE_M
	ZP.RTWRITE	RTWRITEP_M

8.8.1 RTREAD

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	Q
				●	●[1]

[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

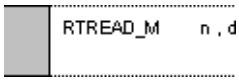
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
n	—	●	●	—	—	—	—	●	—	SM0	7
d	—	●	●	—	—	—	—	—	—		

GX IEC Developer

MELSEC-Anweisungsliste

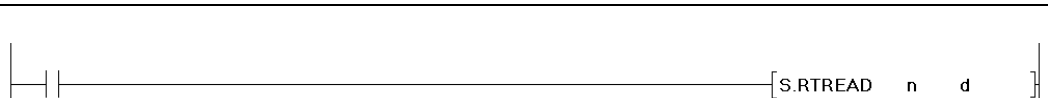
Kontaktplan

IEC-Anweisungsliste



GX Developer (QnA-CPU)

GX Developer (System Q)



Variablen

Operand	Befehlswert	Datentyp	
		MELSEC	IEC
n	Ziel-Netzwerk der Übertragung (1 bis 239).	BIN-16-Bit	ANY16
d	Erste Adresse des Operanden, in dem die gelesenen Routing-Informationen gespeichert werden.	Adresse	Array [1..3] of ANY16

Funktions-weise

Lesen von Routing-Informationen

RTREAD Lese-Anweisung

Die RTREAD-Anweisung liest die Routing-Informationen des in n angegebenen Ziel-Netzwerks der Datenübertragung aus. Die Routing-Informationen sind in den Routing-Parametern gespeichert. Die ausgelesenen Routing-Informationen werden ab d+0 (Array_d[1]) gespeichert.

Wenn für die Übertragung keine Daten angegeben sind, werden die ab d (Array_d[1] bis Array_d[3]) angegebenen Operanden mit dem Wert 0 beschrieben.

Die ab d+0 (Array_d[1]) angegebenen Inhalte sind in der folgenden Abbildung dargestellt.

d+0	1	(1 - 239)
d+1	2	(1 - 64)
d+2	3	(1 - 64)

[1] Netzwerknummer der Relais-Station
[2] Stationsnummer der Relais-Station
[3] Stationsnummer der Routing-Station

Fehler-quellen

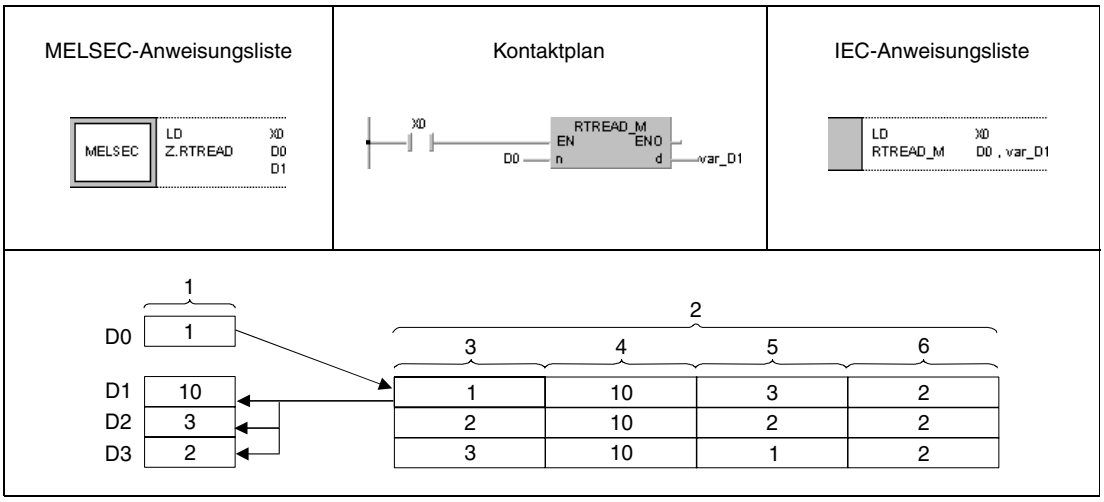
In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Der für n angegebene Datenwert liegt nicht zwischen 1 und 239 (Fehlercode 4100).

Beispiel

Z.RTREAD

Das folgende Programm liest für die Einschaltdauer von X0 die Routing-Informationen des in D0 angegebenen Netzwerks (1) und speichert die Daten in D1 bis D3 (var_D1[1] bis var_D1[3]).



[1] Operation
[2] Inhalte der Routing-Parameter-Einstellungen
[3] Netzwerknummer des Ziel-Netzwerks der Übertragung
[4] Netzwerknummer der Relais-Station
[5] Stationsnummer der Relais-Station
[6] Stationsnummer der Routing-Station

HINWEIS

Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- und Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.

8.8.2 RTWRITE

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●[1]

[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

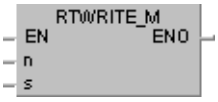
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
n	—	●	●	—	—	—	—	●	—	SM0	8
s	—	●	●	—	—	—	—	—	—		

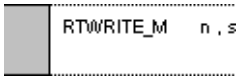
GX IEC Developer

MELSEC-Anweisungsliste

Kontaktplan

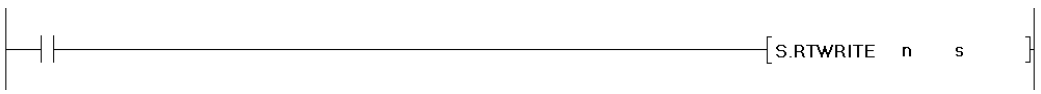


IEC-Anweisungsliste



GX Developer (QnA-CPU)

GX Developer (System Q)



Variablen

Operand	Befehlswert	Datentyp	
		MELSEC	IEC
n	Ziel-Netzwerk der Übertragung (1 bis 239).	BIN-16-Bit	ANY16
s	Erste Adresse des Operanden, in dem die zu schreibenden Routing-Informationen gespeichert werden.	Adresse	Array [1..3] of ANY16

Funktions-weise

Schreiben von Routing-Informationen

RTWRITE Schreib-Anweisung

Die RTWRITE-Anweisung schreibt die Routing-Informationen für das in n angegebenen Ziel-Netzwerk der Datenübertragung. Die Routing-Informationen sind in den Routing-Parametern ab s+0 (Array_s[1]) gespeichert.

Wenn die Daten für das Ziel-Netzwerk in den Routing-Parametern gesetzt sind, werden diese zur Aktualisierung der ab s+0 (Array_s[1]) gespeicherten Daten verwendet.

Wenn die ab s (Array_s[1]) gespeicherten Daten den Wert 0 besitzen, werden die Routing-Parameter des mit n angegebenen Ziel-Netzwerks gelöscht.

Die ab s+0 (Array_s[1]) angegebenen Inhalte sind in der folgenden Abbildung dargestellt.

s+0	1	(1 - 239)
s+1	2	(1 - 64)
s+2	3	(1 - 64)

[1] Netzwerknummer der Relais-Station
[2] Stationsnummer der Relais-Station
[3] Stationsnummer der Routing-Station

Fehler-quellen

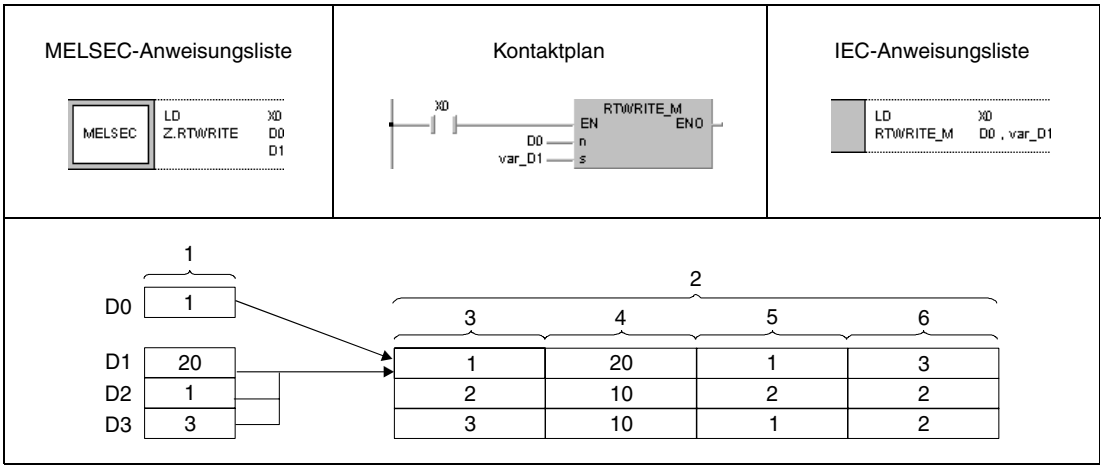
In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Der für n angegebene Datenwert liegt nicht zwischen 1 und 239 (Fehlercode 4100).
- Die in s angegebenen Daten liegen nicht in den zulässigen Bereichen (Fehlercode 4100).

Beispiel

Z.RTWRITE

Das folgende Programm schreibt für die Einschaltdauer von X0 die in D1 bis D3 (var_D1[1] bis var_D1[3]) gespeicherten Routing-Informationen als Routing-Parameter für das in D0 angegebene Netzwerk (1).



[1] Operation
[2] Inhalte der Routing-Parameter-Einstellungen
[3] Netzwerknummer des Ziel-Netzwerks der Übertragung
[4] Netzwerknummer der Relais-Station
[5] Stationsnummer der Relais-Station
[6] Stationsnummer der Routing-Station

HINWEIS

Dieses Programmbeispiel ist ohne Variablendefinition im Header der Programmorganisationseinheit (POE) nicht lauffähig. Compiler- und Checker-Fehlermeldungen können die Folge sein. Weitere Informationen sind dem Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung zu entnehmen.

9 Anweisungen für CPUs des System Q

Die in diesem Kapitel beschriebenen Anweisungen stehen nur bei einer CPU des System Q zur Verfügung.

Funktion	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor
Modul-Informationen auslesen	UNIRD	UNIRD_M
	UNIRDP	UNIRDP_M
Fehlererkennung und Fehlerbeseitigung	TRACE	TRACE_M
	TRACER	TRACER_M
Transfer von Daten in und aus Dateien	FWRITE	FWRITE_M
	FREAD	FREAD_M
Programmanweisungen	PLOADP	PLOADP_M
	PUNLOADP	PUNLOADP_M
	PSWAPP	PSWAPP_M
Transferanweisungen	RBMOV	RBMOV_M
	RBMOVP	RBMOVP_M

Bei Q02-, Q02H-, Q06H-, Q12H-, Q12PH-, Q25H und Q25PHCPUs ab der Softwareversion B können die folgenden Anweisungen für den Multi-CPU-Betrieb genutzt werden:

Funktion	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor
Daten in den gemeinsamen Speicherbereich für Multi-CPU-Betrieb eintragen	S.TO	TO_S_M
	S.TOP	TO_SP_M
Daten aus dem gemeinsamen Speicherbereich einer anderen CPU im Multi-CPU-System lesen	FROM	FROM_M
	FROMP	FROMP_M

9.1 Modul-Informationen lesen

9.1.1 UNIRD, UNIRDP

CPU

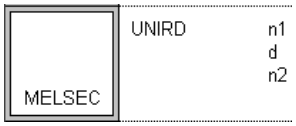
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					●

Operanden MELSEC Q

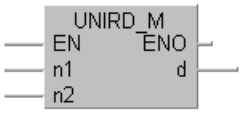
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□N□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
n1	●	●	●	—	—	—	—	●	—	SM0	4
d	—	●	●	—	—	—	—	—	—		
n2	●	●	●	—	—	—	—	●	—		

GX IEC Developer

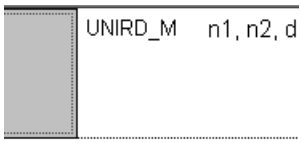
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

Operand	Befehlswert	Datentyp
n1	Wert, der entsteht, wenn die Startadresse des Moduls, von dem Informationen gelesen werden sollen, durch 16 geteilt wird (0 bis FF_H).	BIN-16-Bit
d	Erster Operand des Zielbereiches, in dem die Daten gespeichert werden.	Adresse
n2	Anzahl der zu lesenden Modulinformationen (0 bis 256).	BIN-16-Bit

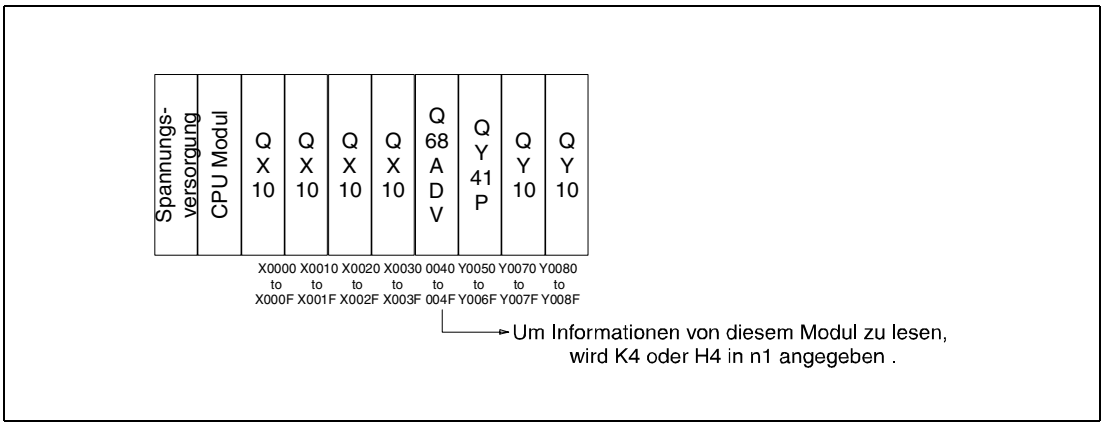
Funktions-weise

Modul-Informationen auslesen

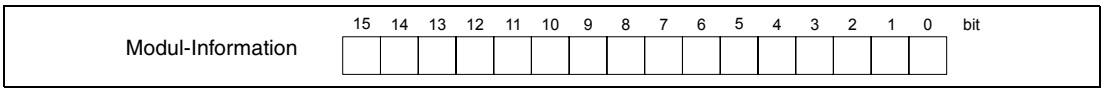
UNIRD Leseanweisung

Die Modulinformationen werden ab der E/A-Adresse, die in n1 angegeben ist, gelesen und ab der Adresse abgelegt, die in d angegeben wird. Die Länge der Daten wird in n2 angegeben. Der Wert für n1 ergibt sich, wenn die Startadresse des Moduls durch 16 geteilt wird.
Mit der UNIRD-Anweisung kann, unabhängig von der Zuordnung der Module in der E/A-Parametrierung, der Zustand der tatsächlich installierten Module erfasst werden.

HINWEIS *Der Wert für n1 besteht aus den drei höherwertigen Bit-Blöcken der Anfangsadresse des Steckplatzes, von dem die Modul-Information gelesen werden soll, wenn die Adresse in vier 4er-Blöcke aufgeteilt wird.*



In der folgenden Tabelle ist die Bedeutung der einzelnen Bits der Modul-Information dargestellt.



Bit	Bedeutung	Beschreibung
0 1 2	Anzahl der Ein- oder Ausgänge	000: 16 001: 32 010: 48 011: 64 100: 128 101: 256 110: 512 111: 1024
3 4 5	Typ des Moduls	000: Eingangsmodul 001: Ausgangsmodul 010: Ein-/Ausgangsmodul 011: Sondermodul
6	Zustand der externen Versorgungsspannung (In Vorbereitung)	EIN: Externe Spannungsversorgung ist vorhanden AUS: Externe Spannungsversorgung fehlt
7	Sicherungsüberwachung	EIN: Defekte Sicherung vorhanden AUS: Keine defekte Sicherung vorhanden
8	Nicht belegt	
9	Leichter oder mittelschwerer Fehler vorhanden	EIN: Leichter oder mittelschwerer Fehler vorhanden AUS: Kein leichter oder mittelschwerer Fehler vorhanden
10 11	Fehlerstatus des Moduls	00: Kein Fehler des Moduls 01: Leichter Fehler 10: Mittelschwerer Fehler 11: Schwerer Fehler
12	Standby-Zustand des Moduls	EIN: Normalbetrieb AUS: Ein Fehler ist aufgetreten
13	Nicht belegt	
14	A-/Q-Modul	EIN: Das Modul gehört zur A-Serie AUS: Das Modul gehört zur Q-Serie
15	Installationszustand der Module	EIN: Module sind installiert AUS: Kein Modul installiert

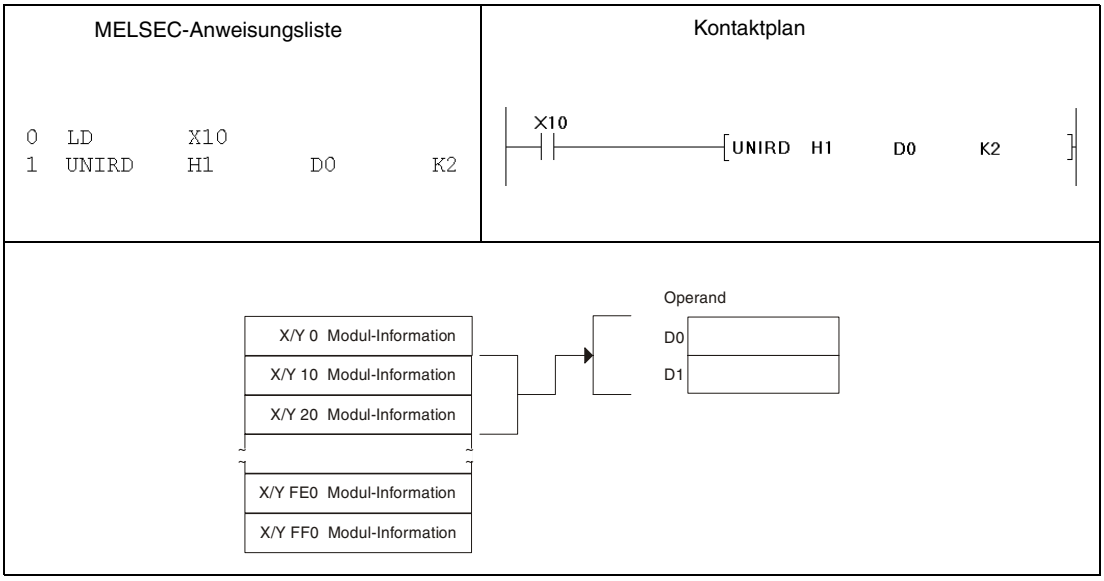
Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

Der Wert von n1 liegt ausserhalb des Bereiches 0 bis FF_H (Fehlercode 4100).
Der Wert von n2 liegt ausserhalb des Bereiches 0 bis FF_H (Fehlercode 4100).
Die Summe der Werte für n1 und n2 ist größer als 256 (Fehlercode 4100).

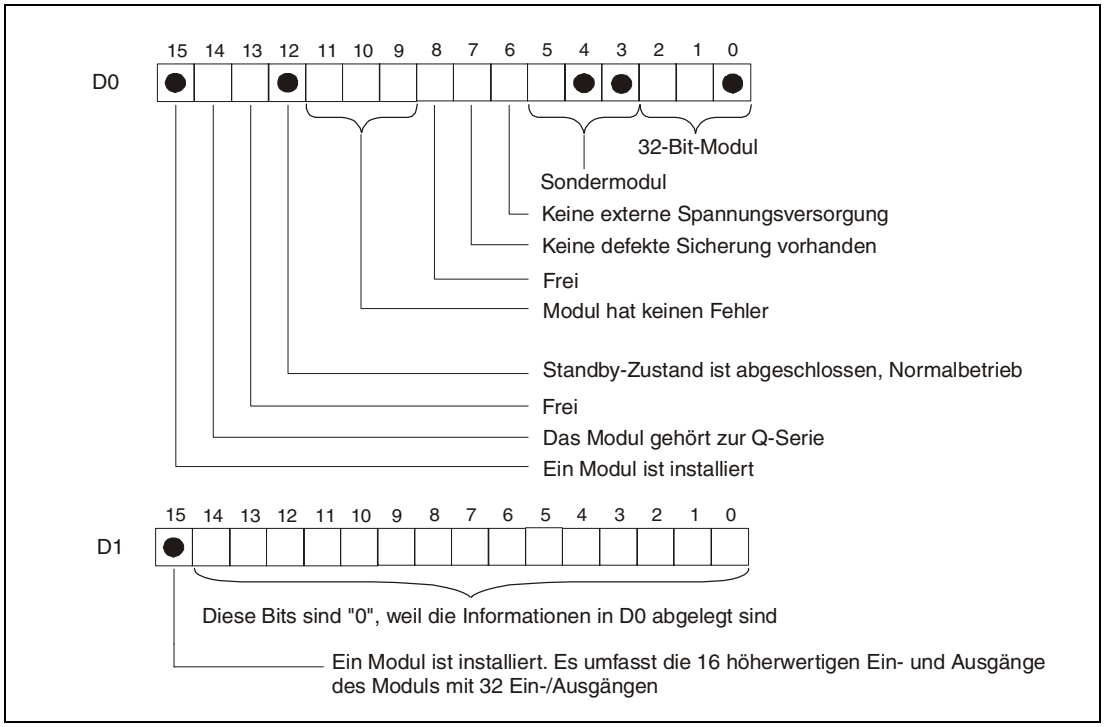
Beispiel UNIRD (GX Developer)

Im folgenden Programm werden die Informationen der Module, die die E/A-Adressen 10_H bis 20_H belegen, in D0 und D1 gespeichert, wenn X10 eingeschaltet wird.

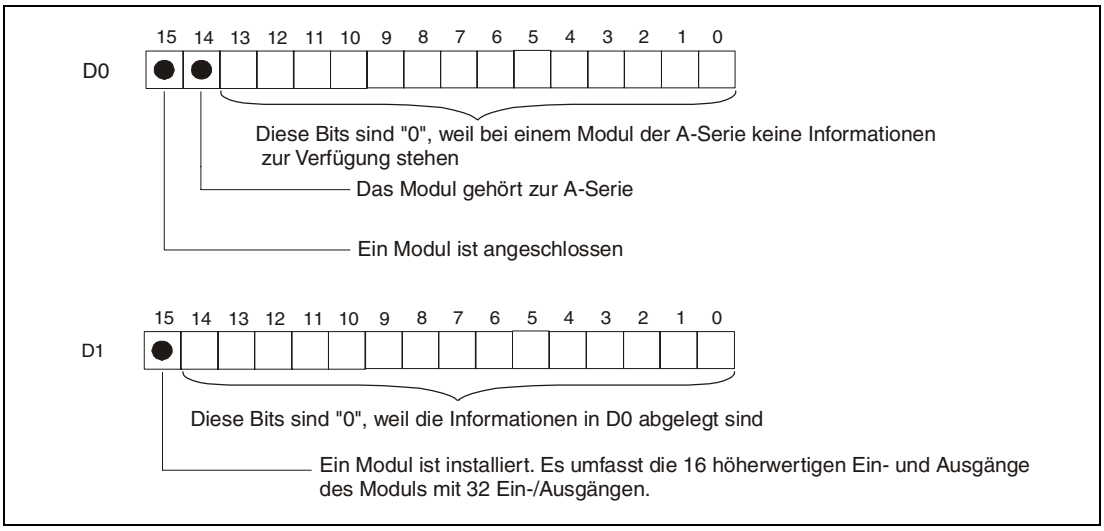


Die Modulinformationen können (im vorherigen Beispiel) D0 und D1 entnommen werden. Nachfolgend sind beispielhaft einige mögliche Modul-Informationen dargestellt.

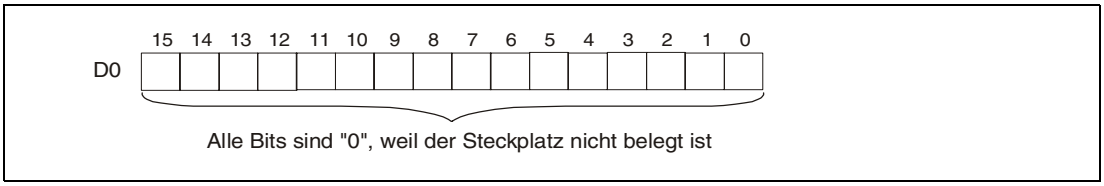
- Bei einem 32-Bit-Sondermodul des System Q. Bei einem 48- oder 64-Bit-Modul ist der Inhalt von D2 bzw. D2 und D3 identisch mit dem Inhalt von D1.



- Bei einem 32-Bit-Sondermodul der A-Serie. Bei einem 48- oder 64-Bit-Modul ist der Inhalt von D2 bzw. D2 und D3 identisch mit dem Inhalt von D1.



- Bei einem freien Steckplatz



9.2 Fehlersuche und Fehlerbeseitigung

9.2.1 TRACE, TRACER

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					●[1]

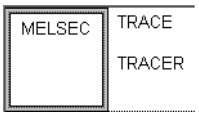
[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

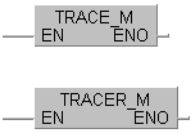
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sonder-module U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
—	—	—	—	—	—	—	—	—	—	—	1

GX IEC Developer

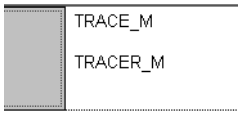
MELSEC-Anweisungsliste



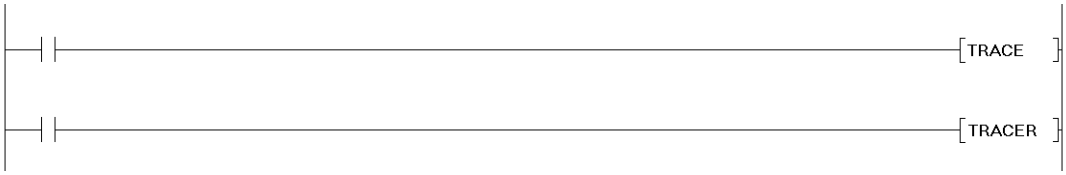
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

Operand	Befehlswert	Datentyp
—	—	—

Funktions-weise

Überwachung setzen und rücksetzen

TRACE Überwachung setzen

Die Funktion des Trace speichert die Daten der durch ein Programmiergerät ausgewählten Operanden in ein Trace-File auf der Speicherkarte, wenn SM800, SM801 und SM802 gesetzt sind. Wenn die TRACE-Anweisung ausgeführt wird, wird SM803 gesetzt. Wenn das Erfassen der Daten entsprechend den Vorgaben abgeschlossen ist, werden die Daten gesichert und das Trace gestoppt.

Wenn SM801 während der Ausführung des Trace rückgesetzt wird, wird die Erfassung der Daten angehalten.

Nachdem die TRACE-Anweisung ausgeführt wurde, wird SM805 gesetzt.

Während der Ausführung des Trace werden weitere TRACE-Anweisungen ignoriert. Erst nachdem die TRACER-Anweisung beendet wurde, ist eine erneute Ausführung der TRACE-Anweisung möglich.

TRACER Überwachung rücksetzen

Mit der TRACER-Anweisung werden die TRACE-Anweisung und die Merker SM803 bis SM805 rückgesetzt. Nachdem die TRACER-Anweisung beendet wurde, ist eine erneute Ausführung der TRACE-Anweisung freigegeben.

HINWEISE

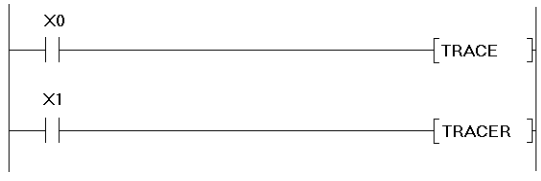
Weitere Informationen zum Thema Trace können Sie in der Bedienungsanleitung der CPU-Module des System Q und den Benutzerhandbüchern des GX Developers und GX IEC Developers nachlesen.

Beispiel

TRACE, TRACER (GX Developer)

Im folgenden Programm wird die TRACE-Anweisung ausgeführt, wenn X0 eingeschaltet wird. Wenn X1 eingeschaltet wird, wird die TRACE-Anweisung durch die TRACER-Anweisung rückgesetzt.

```
0  LD       X0
1  TRACE
2  LD       X1
3  TRACER
```



9.3 Datentransfer in und aus Dateien

9.3.1 FWRITE

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					●[1]

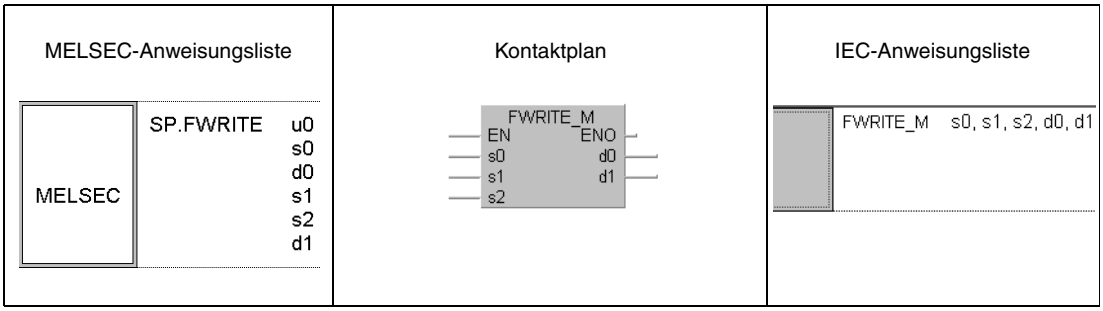
[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

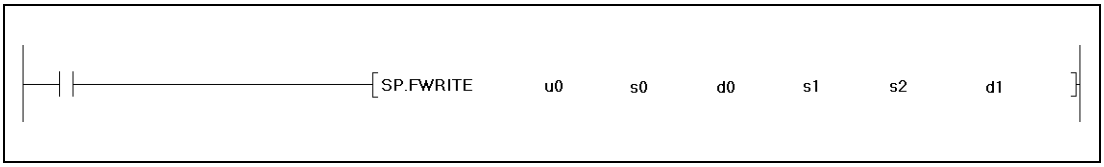
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s0	●	●	●	—	—	—	—	●	—	SM0	11
d0	—	●	●	—	—	—	—	—	—		
s1	—	●	●	—	—	—	—	—	—		
s2	—	●	●	—	—	—	—	—	●		
d1	●*	●*	●*	—	—	—	—	—	—		

* Lokale Operanden und Operanden, die für Programme reserviert sind, sind nicht zulässig.

GX IEC Developer



GX Developer



Variablen

<table>
<tr><th>Operand</th><th colspan="4">Bedeutung</th><th>Werte-bereich</th><th>Festlegung durch</th><th>Datentyp</th></tr>
<tr><td>u0</td><td colspan="4">Dummy (Wird nicht verwendet)</td><td>—</td><td>—</td><td rowspan="2">BIN-16-Bit</td></tr>
<tr><td>s0</td><td colspan="4">Angabe des Laufwerkes.
Es kann nur ein Laufwerk mit einer ATA-Speicherkarte angegeben werden (Laufwerk 2). Eine ROM-Speicherkarte oder eine normale RAM- oder ROM-Speicherkarte kann nicht benutzt werden.</td><td>2</td><td>Benutzer</td></tr>
<tr><td rowspan="10">d0</td><td colspan="4">Erster Operand des Bereiches, in dem die Kontrolldaten gespeichert sind.
Die folgenden Kontrolldaten sind notwendig.</td><td></td><td></td><td rowspan="10">BIN-16-Bit</td></tr>
<tr><td>Operand</td><td>Bedeutung</td><td colspan="2">Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(d0)</td><td>Ausführungs-modus</td><td colspan="2">Legt die Art der Datenübertragung fest:
0000_H: Binäre Daten übertragen
0100_H: Daten vor Übertragung in CSV-Format wandeln</td><td>0000_H
0100_H</td><td>Benutzer</td></tr>
<tr><td>(d0)+1</td><td>Reserviert</td><td colspan="2">Wird vom System benutzt</td><td>—</td><td>System</td></tr>
<tr><td>(d0)+2</td><td>Anzahl der übertragenen Daten</td><td colspan="2">Enthält die Anzahl der tatsächlich übertragenen Daten. Die Einheit des Wertes wird durch die Festlegung, ob wort- oder byteweise übertragen wird, festgelegt.</td><td>—</td><td>System</td></tr>
<tr><td>(d0)+3</td><td>Nicht benutzt</td><td colspan="2">—</td><td>—</td><td>—</td></tr>
<tr><td>(d0)+4
(d0)+5</td><td>Anfangsadresse in Zieldatei</td><td colspan="2">Auswahl, ab welcher Adresse Daten in die Zieldatei eingetragen werden sollen.
Binäre Daten (d0 = 0000_H):
00000000_H: Anfang der Datei
00000001_H bis $FFFFFFFE_H$: Ab der angegebenen Adresse.
Die Einheit des Wertes wird durch die Festlegung, ob wort- oder byteweise übertragen wird, festgelegt.
$FFFFFFFF_H$: Die Daten werden an das Ende der Datei angehangen

Wenn Daten nach der Wandlung aus dem CSV-Format übertragen werden (d0 = 0100):
Bei CPUs bis zur Seriennummer 01111 muss als Zieladresse der Anfang der Datei gewählt werden ($00000000_{H)}$.
Bei CPUs ab der Seriennr. 01112:
00000000_H bis $FFFFFFFE_H$:
Anfang der Datei
$FFFFFFFF_H$: Die Daten werden an das Ende der Datei angehangen.</td><td>00000000_H
bis
$FFFFFFFF_H$</td><td>Benutzer</td></tr>
<tr><td>(d0)+6</td><td>Anzahl der Spalten</td><td colspan="2">Festlegung, in wieviele Spalten Daten im CSV-Format in die Zieldatei eingetragen werden.
0: : Keine Spalten angewählt.
> 0 : Daten werden in eine entsprechende Anzahl Spalten eingetragen</td><td>0 bis 65535</td><td>Benutzer</td></tr>
<tr><td>(d0)+7</td><td>Festlegung, ob Wort- oder Byte-Daten übertragen werden</td><td colspan="2">0: Wort-Daten
1: Byte-Daten</td><td>0, 1</td><td>Benutzer</td></tr>
</table>

Variablen

<table>
<tr><th>Operand</th><th colspan="3">Bedeutung</th><th>Werte-bereich</th><th>Festlegung durch</th><th>Datentyp</th></tr>
<tr><td rowspan="3">s1</td><td colspan="5">Erster Operand des Bereiches, in dem der Dateiname gespeichert ist.</td><td rowspan="3">BIN-16-Bit</td></tr>
<tr><td>Operand</td><td>Bedeutung</td><td>Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(s1) bis (s1)+n</td><td>Dateibezeichnung</td><td>Die Dateibezeichnung besteht aus dem Dateinamen (maximal 8 Zeichen), einem Punkt und der 3-stelligen Erweiterung (ABD.BIN). Die Erweiterung kann entfallen, wenn auch der Punkt weggelassen wird.
Wenn mehr als 8 Zeichen angegeben werden, wird die Erweiterung, auch wenn sie vorhanden ist, nicht berücksichtig. Die Erweiterung BIN oder CSV wird automatisch angefügt.</td><td>Zeichen-folge</td><td>Benutzer</td></tr>
<tr><td rowspan="4">s2</td><td colspan="5">Erster Operand des Bereiches, in dem die Daten gespeichert ist.</td><td rowspan="4">BIN-16-Bit</td></tr>
<tr><td>Operand</td><td>Bedeutung</td><td>Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(s2)</td><td>Länge der Daten</td><td>Angabe der Datenlänge (Wörter), die übertragen werden soll. Auch wenn in (d0) +7 die byteweise Übertragung gewählt wurde, wird die Anzahl der Daten in der Einheit „Wort“ angegeben.</td><td>1 bis 480</td><td rowspan="2">System</td></tr>
<tr><td>(s2)+1 bis (s2)+n</td><td>Zu übertragende Daten</td><td>Daten, die übertragen werden sollen.</td><td>0000_H bis $FFFF_H$</td></tr>
<tr><td rowspan="4">d1</td><td colspan="5">Bit-Operand, der nach der Ausführung der FWRITE-Anweisung gesetzt wird. Mit (d1)+1 wird die fehlerhafte Beendigung signalisiert.</td><td rowspan="4">Bit</td></tr>
<tr><td>Operand</td><td>Bedeutung</td><td>Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(d1)</td><td>Anweisung ausgeführt</td><td>Zeigt die Beendigung der FWRITE-Anweisung an.
EIN : Anweisung ausgeführt.
AUS : Anweisung nicht ausgeführt</td><td>—</td><td rowspan="2">System</td></tr>
<tr><td>(d1)+1</td><td>Anweisung mit Fehler ausgeführt</td><td>Zeigt an, ob bei der Ausführung der FWRITE-Anweisung ein Fehler aufgetreten ist.
EIN : Anweisung mit Fehler ausgeführt.
AUS : Anweisung ohne Fehler ausgeführt</td><td>—</td></tr>
</table>

HINWEIS

Daten im CSV-Format werden von der Programmier-Software als Dezimalzahlen dargestellt. Zum Beispiel wird „A“ (41_H) als 65 angezeigt. Die angezeigten Werte können zwischen -32768 und 32767 liegen.

Zulässige Werte für die Anfangsadresse in der Zieldatei sind beim Schreiben von binären Wort-Daten der Bereich von 00000000_H bis $7FFFFFFF_H$ und $FFFFFFFF_H$.

Funktionsweise

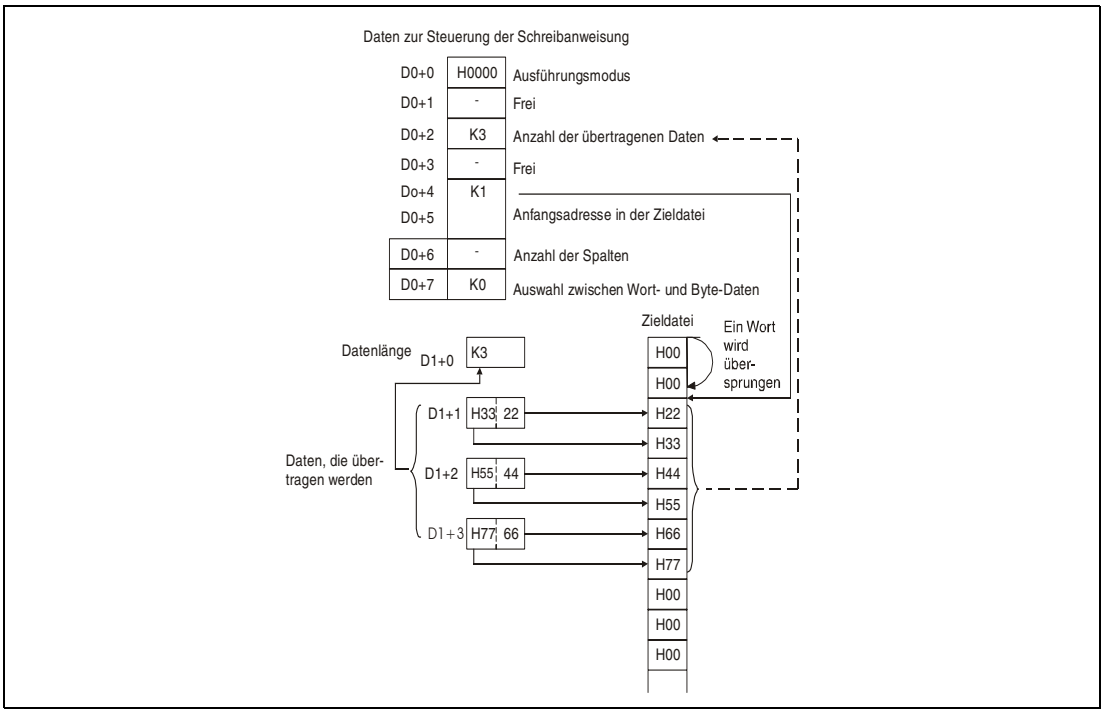
Daten in eine angegebene Datei eintragen

FWRITE Schreibanweisung

Mit der FWRITE-Anweisung wird eine definierte Anzahl von Daten als File auf die ATA-Speicherkarte übertragen. Dabei kann gewählt werden, ob die Daten ohne vorherige Umwandlung im binären Format übertragen werden oder ob die Daten vor der Übertragung in das CSV-Format gewandelt und dann übertragen werden.

Das Bit, das in (d1)+0 angegeben wurde, wird automatisch gesetzt, wenn die Ausführung der FWRITE-Anweisung erkannt und die END-Anweisung ausgeführt wird. Nach der END-Anweisung im nächsten Zyklus wird dieses Bit wieder rückgesetzt. Der Benutzter kann dieses Bit als Signal für die Beendigung der FWRITE-Anweisung verwenden.
Falls bei der Ausführung der Anweisung ein Fehler aufgetreten ist, wird das Bit, das in (d1)+1 angegeben wurde, synchron mit der Fertigmeldung [(d1)+0] gesetzt und rückgesetzt.
Vom Benutzter kann dieses Signal als Fehlermeldung ausgewertet werden.
Während der Ausführung der FWRITE-Anweisung ist SM721 gesetzt. SM721 wird auch bei der Ausführung anderer Anweisungen (S.FREAD, COMRD, PRC) gesetzt. Wenn SM721 gesetzt ist, kann die FWRITE-Anweisung nicht gestartet werden. Falls dies versucht wird, wird die Anweisung nicht ausgeführt.
Falls vor der Ausführung der Anweisung (bevor SM721 gesetzt wird) ein Fehler erkannt wird, werden die Fertigmeldung [(d1)+0], die Fehlermeldung [(d1)+1] und SM721 nicht gesetzt.

Die Angabe der Datenlänge [(s2)+0] erfolgt in der Einheit „Wort“. Die folgende Abbildung zeigt die Übertragung von binären Daten.



Übertragung binärer Daten

Wenn binäre Daten übertragen werden, wird die Erweiterung „.BIN“ an den Dateinamen der Zieldatei angehängt, wenn diese Erweiterung weggelassen wurde. Wenn die Zieldatei nicht existiert, wird eine neue Datei erzeugt und die Daten werden ab dem Anfang der Datei eingetragen. Als Dateiattribute werden die archivierten Attribute verwendet.

Wenn die Datenlänge die Größe der Datei überschreitet, werden die Daten, die nicht mehr in die Datei passen, an deren Ende angehängt.

Wenn zur Speicherung der Daten eine Anfangsadresse angegeben wird, die größer ist als die Zieldatei, wird bei CPUs bis zu einer Seriennummer von „01111“ (die ersten fünf Stellen) eine

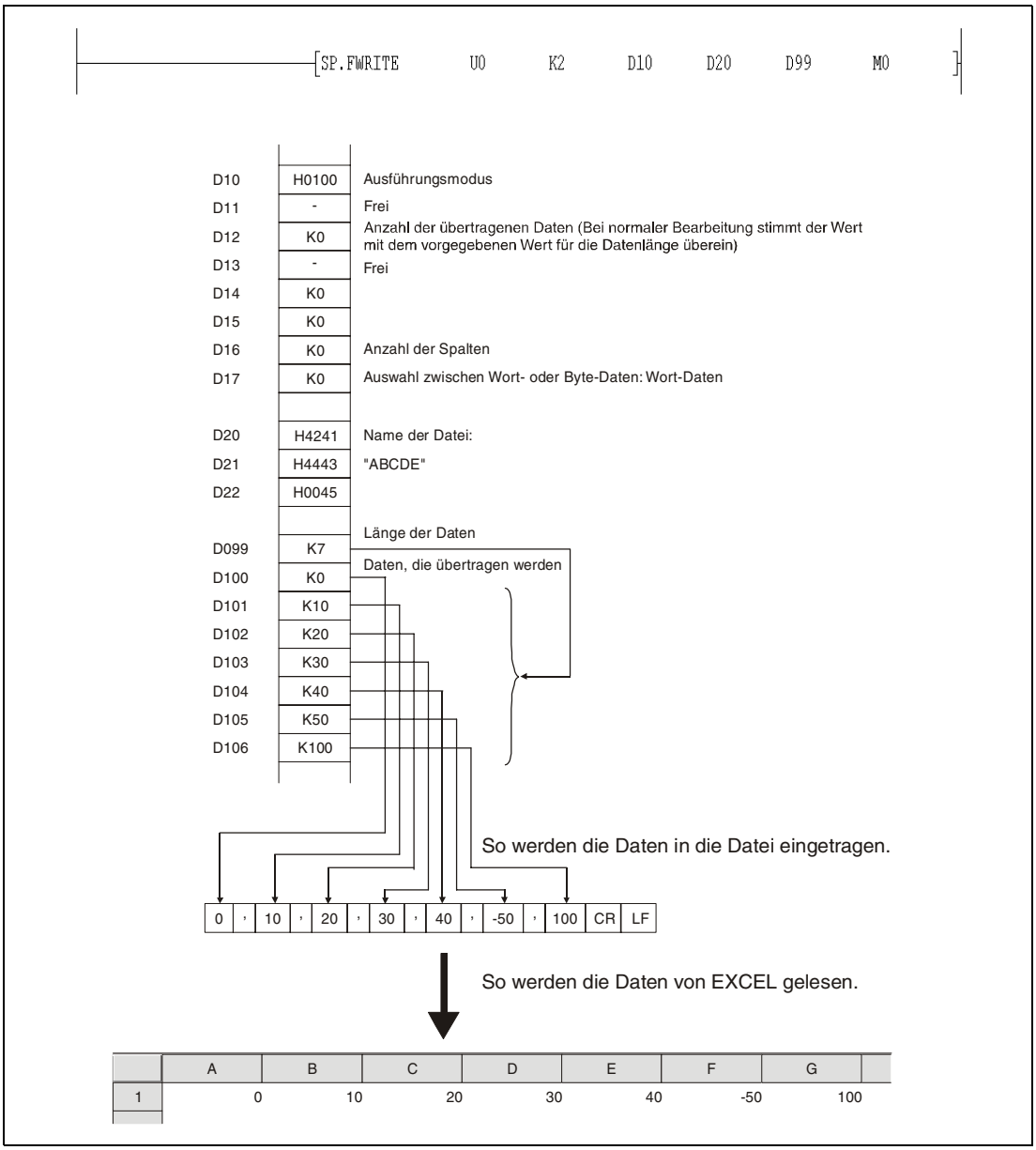
Fehlermeldung ausgegeben. Eine CPU ab der Seriennummer „01112“ überträgt in diesem Fall keine Daten und beendet die Anweisung ohne Fehlermeldung.

Wenn die Speicherkapazität der Speicherkarte überschritten wird, werden so viele Daten wie möglich gespeichert, bevor eine Fehlermeldung ausgegeben wird. Die bis dahin gespeicherten Daten bleiben erhalten.

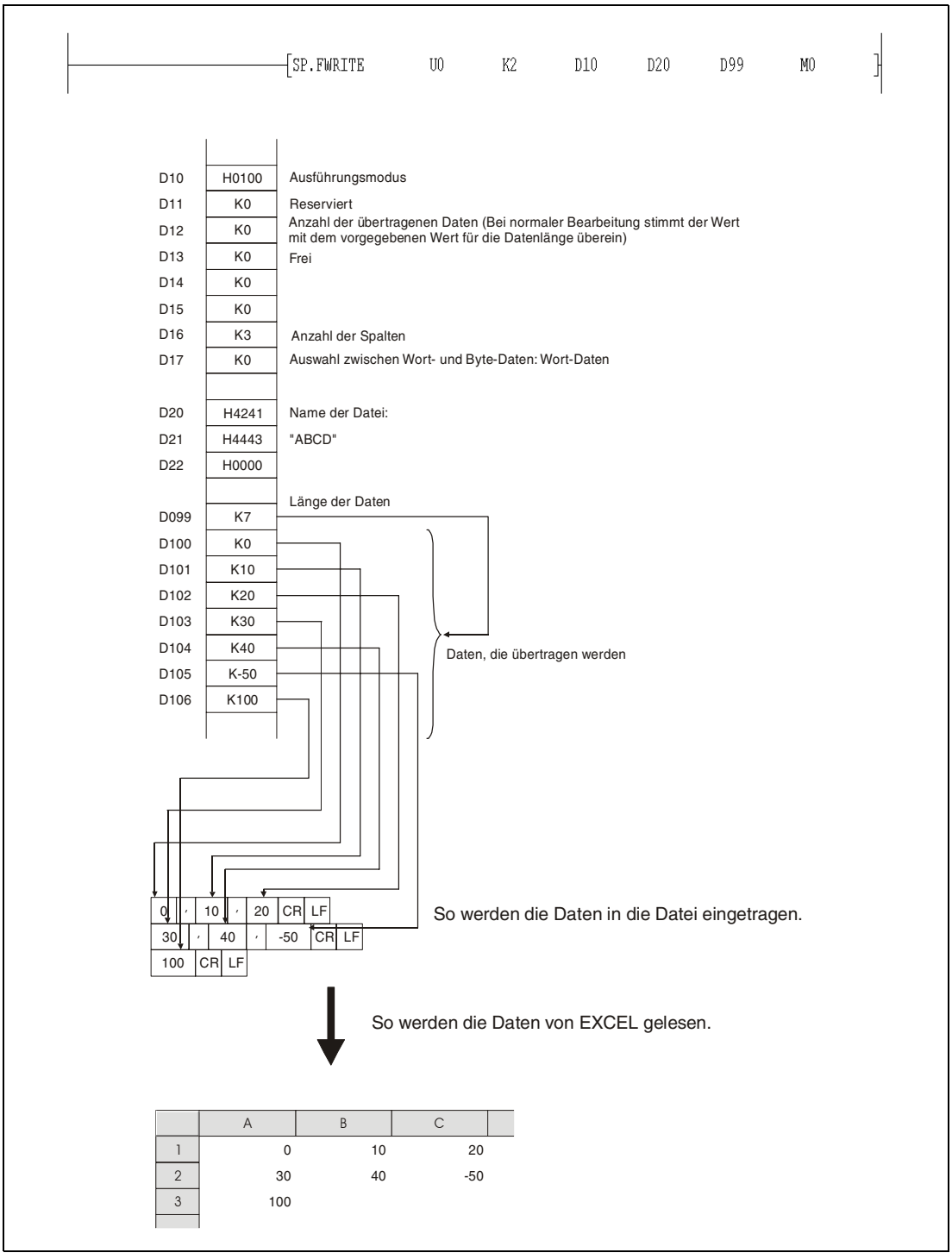
Übertragung von Daten im CSV-Format

Wenn keine Erweiterung des Dateinamens angegeben wurde, wird „.CSV“ angefügt. Wenn als Zieldatei eine bestehende Datei angegeben wurde, wird bei CPUs bis zu einer Seriennummer von „01111“ (die ersten fünf Stellen) der Inhalt dieser Datei gelöscht, bevor die Daten vom Beginn der Datei an eingetragen werden.
Bei CPUs ab der Seriennummer „01112“ wird in diesem Fall ebenfalls der Inhalt dieser Datei gelöscht und anschließend werden die Daten vom Beginn der Datei an eingetragen, wenn in (d0)+4 und (d0)+5 ein anderer Wert als $FFFFFFFF_H$ eingetragen ist. Enthalten (d0)+4 und (d0)+5 den Wert $FFFFFFFF_{H,}$ werden die Daten an das Ende der Datei angefügt.

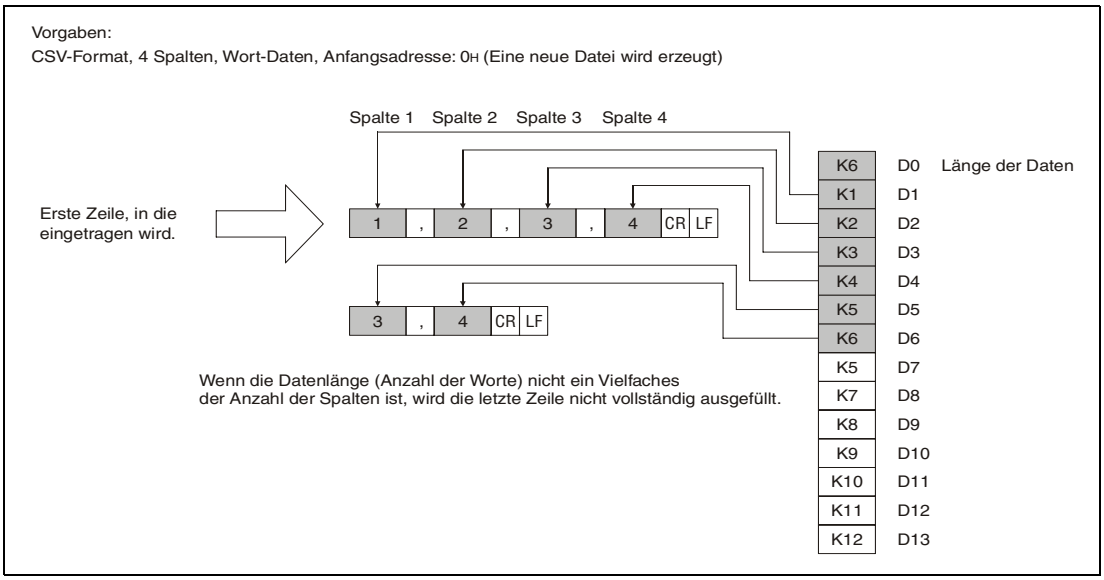
Wenn die Zieldatei nicht existiert, wird eine neue Datei erzeugt und die Daten werden ab dem Anfang der Datei eingetragen. Als Dateiattribute werden die archivierten Attribute verwendet.
Wird die Speicherkapazität der Speicherkarte während des Schreibens überschritten, werden so viele Daten wie möglich gespeichert, bevor eine Fehlermeldung ausgegeben wird. Die bis dahin gespeicherten Daten bleiben erhalten.
Wenn als Anzahl der zu beschreibenen Spalten „0“ vorgegeben wird, werden die Daten in einer Zeile in einer Datei im CSV-Format abgelegt. Die folgende Abbildung verdeutlicht diesen Fall.

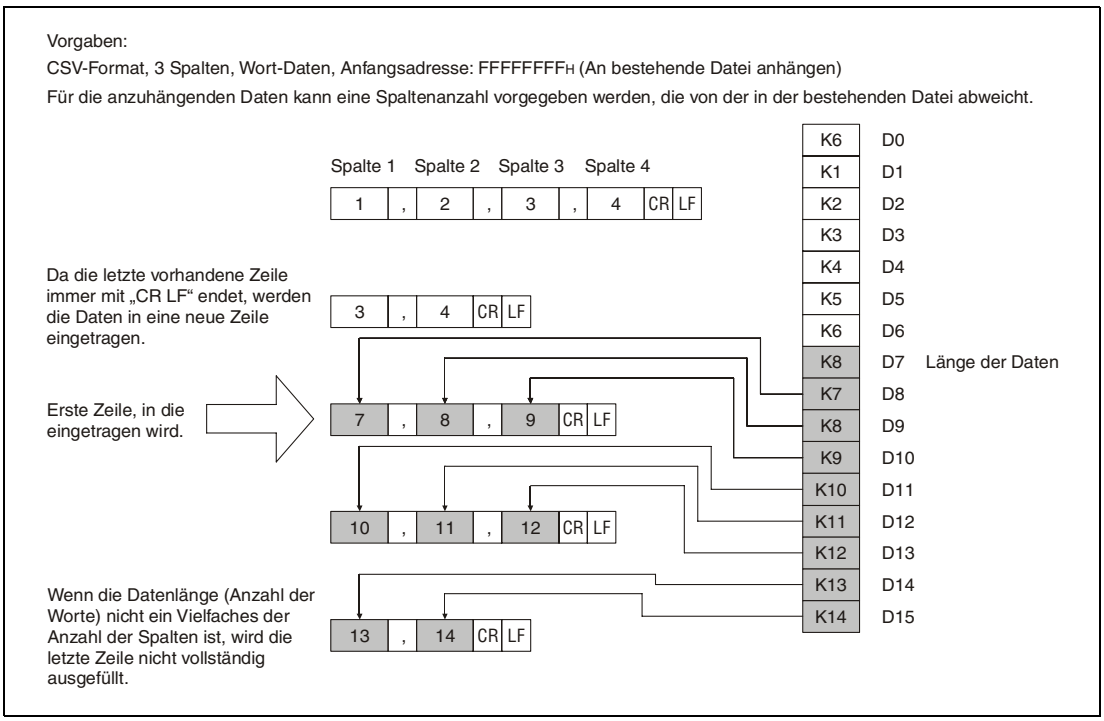


Wenn eine Anzahl von Spalten vorgegeben wird, die größer als „0“ ist, werden die Daten als-Datenfeld mit der entsprechenden Anzahl von Spalten in der Datei im CSV-Format abgelegt. Dies ist in der folgenden Abbildung dargestellt.



Die beiden folgenden Abbildungen zeigen Beispiele, bei denen Daten mit einer CPU ab der Seriennummer „01112“ (die ersten fünf Stellen) eingetragen werden.





HINWEIS Verwenden Sie die FWRITE-Anweisung nicht in einem Interrupt-Programm.

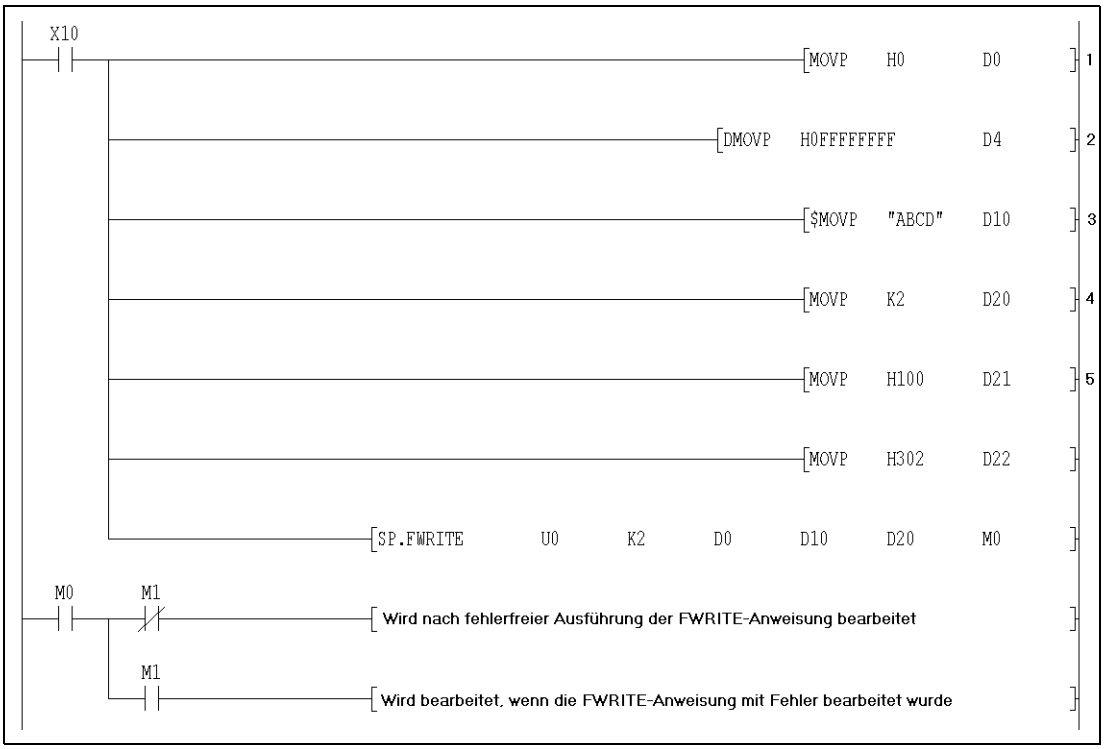
Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Das Laufwerk, das in s0 angegeben wurde, enthält eine Speicherkarte, die keine ATA-Speicherkarte ist (Fehlercode 4100).
- Werte, die als Kontrolldaten eingetragen wurden, liegen ausserhalb der zulässigen Bereiche (Fehlercode 4100).
- Die in (s2)+0 angegebene Länge der Daten überschreitet den zulässigen Bereich oder ist größer als die Datenmenge, die ab (s2)+1 eingetragen ist (Fehlercode 4101).
- Die Speicherkapazität der Speicherkarte ist überschritten (Fehlercode 4100).
- Ein ungültiger Operand wurde angegeben (Fehlercode 4004).

Beispiel 1 FWRITE (GX Developer)

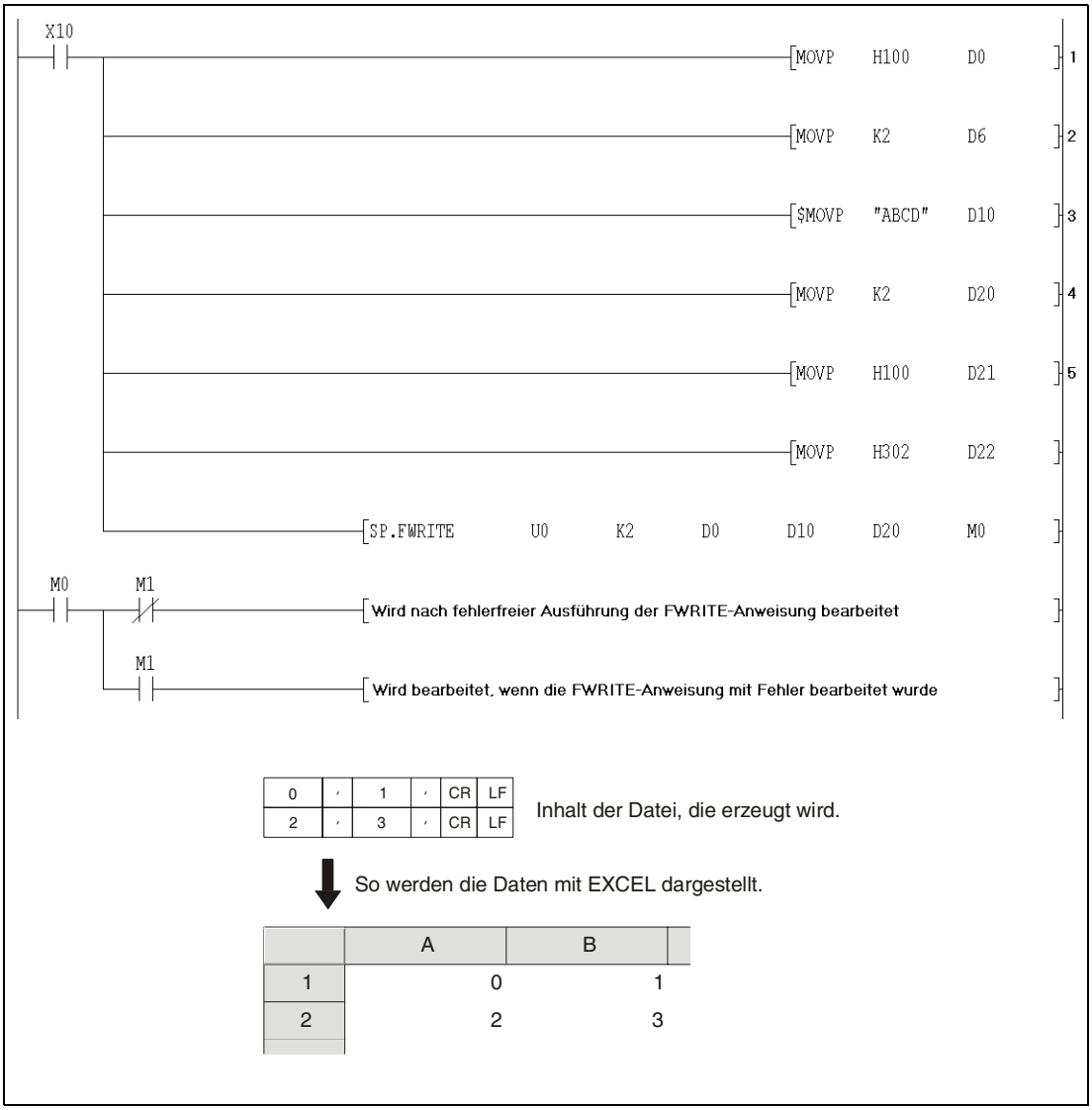
Im folgenden Programm werden vier Bytes binäre Daten mit den Inhalten 00_H, 01_H, 02_H und 03_H an die Datei „ABCD.BIN“ angehängt, wenn X10 gesetzt wird. Die Speicherkarte steckt in Laufwerk 2.
Für die Kontrolldaten werden ab D0 acht Operanden belegt.



[1] Ausführungsmodus festlegen (binäre Daten)
[2] Startadresse festlegen (Daten an das Ende der Datei anfügen)
[3] Dateinamen eintragen (Erweiterung „.BIN“ wird automatisch angefügt)
[4] Anzahl der zu übertragenen Daten
[5] Zu übertragene Daten eintragen

Beispiel 2 FWRITE (GX Developer)

Wenn X10 gesetzt wird, erzeugt das folgende Programm eine Datei mit dem Namen „ABCD.CSV“ auf der Speicherkarte in Laufwerk 2. Dann werden vier Bytes mit den Inhalten 00_H, 01_H, 02_H und 03_H als zweispaltiges Datenfeld im CSV-Format in diese Datei eingetragen. Für die Kontrolldaten werden ab D0 acht Operanden belegt.



[1] Ausführungsmodus festlegen (CSV-Format)

[2] Anzahl der Spalten festlegen

[3] Dateinamen eintragen

[4] Anzahl der zu übertragenen Daten

[5] Zu übertragene Daten eintragen

9.3.2 FREAD

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					●[1]

[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

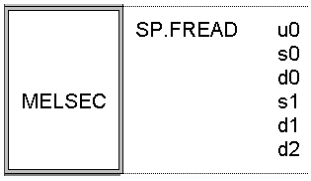
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s0	●	●	●	—	—	—	—	●	—	SM0	11
d0	—	●	●	—	—	—	—	—	—		
s1	—	●	●	—	—	—	—	—	—		
s2	—	●	●	—	—	—	—	—	●		
d1	●*	●*	●*	—	—	—	—	—	—		

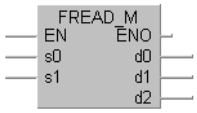
* Lokale Operanden und Operanden, die für Programme reserviert sind, sind nicht zulässig.

GX IEC Developer

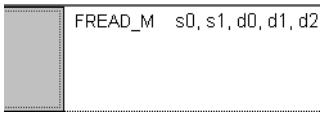
MELSEC-Anweisungsliste



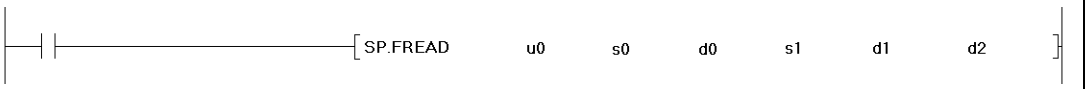
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

Operand	Bedeutung			Werte-bereich	Festle-gung durch	Datentyp
u0	Dummy (Wird nicht benutzt)			—	—	BIN-16-Bit
s0	Angabe des Laufwerkes. Es kann nur ein Laufwerk mit einer ATA-Speicherkarte angegeben werden (Laufwerk 2). Eine ROM-Speicherkarte oder eine normale RAM- oder ROM-Speicherkarte kann nicht benutzt werden.			2	Benutzer	
d0	Erster Operand des Bereiches, in dem die Kontrolldaten gespeichert sind. Die folgenden Kontrolldaten sind notwendig.					BIN-16-Bit
	Operand	**Bedeutung**	**Beschreibung**	**Werte-bereich**	**Festle-gung durch**	
	(d0)	Ausführungs-modus	Legt die Art der Datenübertragung fest: 0000_H: Binäre Daten übertragen 0100_H: Daten vor Übertragung von CSV-Format in binäre Daten wandeln	0000_H 0100_H	Benutzer	
	(d0)+1	Reserviert	Wird vom System benutzt	—	System	
	(d0)+2	Anzahl der zu übertragenen Daten	Angabe der Datenlänge (Wörter), die übertragen werden soll. Auch wenn in (d0) +7 die byteweise Übertragung gewählt wurde, wird die Anzahl der Daten in der Einheit „Wort“ angegeben.	1 bis 480	Benutzer	
	(d0)+3	Nicht benutzt	—	—	—	
	(d0)+4 (d0)+5	Anfangsadresse in Quelldatei	Auswahl, ab welcher Adresse in der Quelldatei Daten ausgelesen werden sollen. Binäre Daten (d0 = 0000): 00000000_H: Ab dem Anfang der Datei 00000001_H bis $FFFFFFFC_H$: Ab der angegebenen Adresse Die Einheit des Wertes wird durch die Festlegung, ob wort- oder byteweise übertragen wird, festgelegt. $FFFFFFFD_H$: Nicht zulässig Wenn Daten nach der Wandlung aus dem CSV-Format übertragen werden (d0 = 0100): Bei CPUs bis zur Seriennummer 01111 muss als Startadresse der Anfang der Datei gewählt werden (00000000_H). Bei CPUs ab der Seriennr. 01112: 00000000_H: Ab dem Anfang der Datei 00000001_H bis $FFFFFFFC_H$: Ab der angegebenen Zeile $FFFFFFFD_H$: Das Lesen wird an der vorherigen Position fortgesetzt.	00000000_H bis $FFFFFFFC_H$ $FFFFFFFD_H$	Benutzer	
	(d0)+6	Anzahl der Spalten	Angabe, in wieviele Spalten Daten im CSV-Format in der Zieldatei eingetragen sind. 0: : Nicht in Spalten eingetragen > 0 : Daten sind in der entsprechenden Anzahl von Spalten eingetragen	0, 1 bis 65535	Benutzer	
	(d0)+7	Festlegung, ob Wort- oder Byte-Daten übertragen werden	0: Wort-Daten 1: Byte-Daten	0, 1	Benutzer	

Variablen

<table>
<tr><th>Operand</th><th colspan="4">Bedeutung</th><th>Werte-bereich</th><th>Festle-gung durch</th><th>Datentyp</th></tr>
<tr><td rowspan="3">s1</td><td colspan="5">Erster Operand des Bereiches, in dem der Dateiname gespeichert ist</td><td rowspan="3">BIN-16-Bit</td></tr>
<tr><td>Operand</td><td>Bedeutung</td><td colspan="2">Beschreibung</td><td>Werte-bereich</td><td>Festle-gung durch</td></tr>
<tr><td>(s1) bis (s1)+n</td><td>Dateibezeichnung</td><td colspan="2">Die Dateibezeichnung besteht aus dem Dateinamen (maximal 8 Zeichen), einem Punkt und der 3-stelligen Erweiterung (ABD.BIN). Die Erweiterung kann entfallen, wenn auch der Punkt weggelassen wird.
Wenn mehr als 8 Zeichen angegeben werden, wird die Erweiterung, auch wenn sie vorhanden ist, nicht berücksichtig. Die Erweiterung BIN oder CSV wird automatisch angefügt.</td><td>Zeichen-folge</td><td>Benutzer</td></tr>
<tr><td rowspan="4">d1</td><td colspan="5">Erster Operand des Bereiches, in dem die Daten gespeichert werden.</td><td rowspan="4">BIN-16-Bit</td></tr>
<tr><td>Operand</td><td>Bedeutung</td><td colspan="2">Beschreibung</td><td>Werte-bereich</td><td>Festle-gung durch</td></tr>
<tr><td>(d1)</td><td>Anzahl der übertragenden Daten</td><td colspan="2">Enthält die Anzahl der tatsächlich übertragenen Daten. Die Einheit des Wertes wird durch die Festlegung, ob wort- oder byteweise übertragen wird, festgelegt.</td><td>0 bis 480</td><td rowspan="2">System</td></tr>
<tr><td>(d1)+1 bis (d1)+n</td><td>Übertragende Daten</td><td colspan="2">Daten, die aus der Quelldatei übertragen wurden.</td><td>0000_H bis $FFFF_H$</td></tr>
<tr><td rowspan="4">d2</td><td colspan="5">Bit-Operand, der nach der Ausführung der FWRITE-Anweisung gesetzt wird. Mit (d2)+1 wird die fehlerhafte Beendigung signalisiert.</td><td rowspan="4">Bit</td></tr>
<tr><td>Operand</td><td>Bedeutung</td><td colspan="2">Beschreibung</td><td>Werte-bereich</td><td>Festle-gung durch</td></tr>
<tr><td>(d2)</td><td>Anweisung ausgeführt</td><td colspan="2">Zeigt die Beendigung der FREAD-Anweisung an.
EIN : Anweisung ausgeführt.
AUS : Anweisung nicht ausgeführt</td><td>—</td><td rowspan="2">System</td></tr>
<tr><td>(d2)+1</td><td>Anweisung mit Fehler ausgeführt</td><td colspan="2">Zeigt an, ob bei der Ausführung der FREAD-Anweisung ein Fehler aufgetreten ist.
EIN : Anweisung mit Fehler ausgeführt.
AUS : Anweisung ohne Fehler ausgeführt</td><td>—</td></tr>
</table>

HINWEIS

Daten im CSV-Format werden von der Programmier-Software als Dezimalzahlen dargestellt. Zum Beispiel wird „A“ (41_H) als 65 angezeigt. Die angezeigten Werte können zwischen -32768 und 32767 liegen.

Zulässige Werte für die Anfangsadresse in der Quelldatei sind beim Lesen von Wort-Daten der Bereich von 00000000_H bis $7FFFFFFF_H$.

Funktions-weise

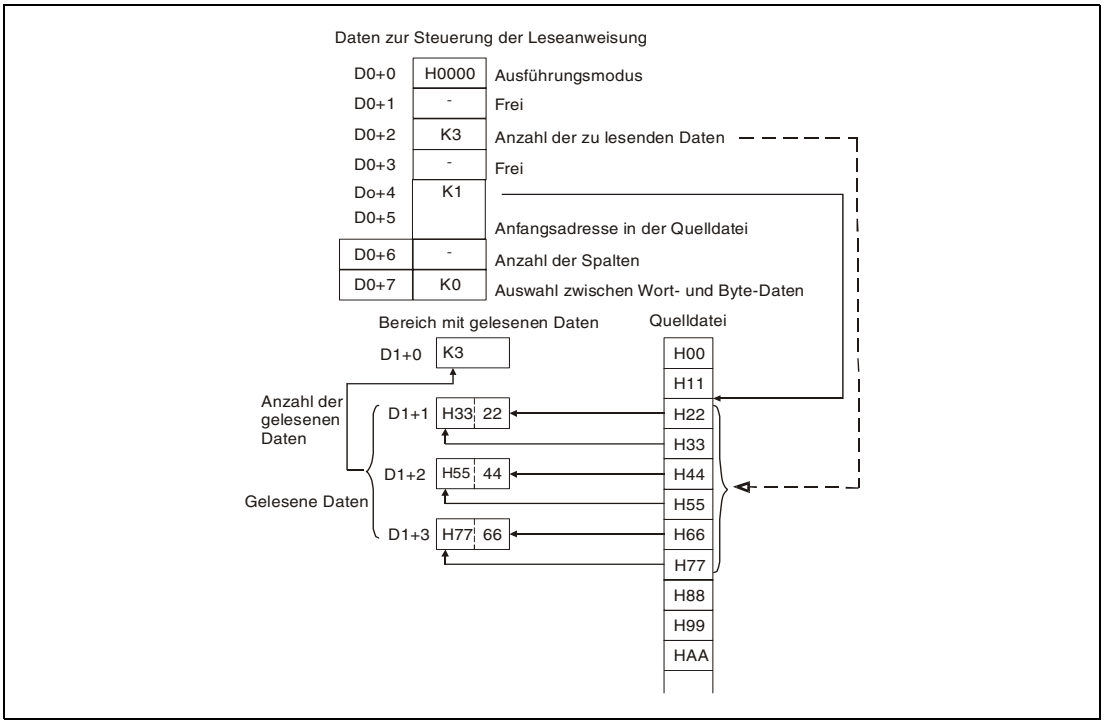
Daten aus einer angegebenen Datei lesen

FREAD Leseanweisung

Mit der FREAD-Anweisung wird eine definierte Anzahl von Daten aus einem File auf der ATA-Speicherkarte gelesen. Dabei kann gewählt werden, ob die Daten ohne vorherige Umwandlung im binären Format übertragen werden oder ob die Daten vor der Übertragung aus dem CSV-Format in binäre Daten gewandelt und dann übertragen werden.

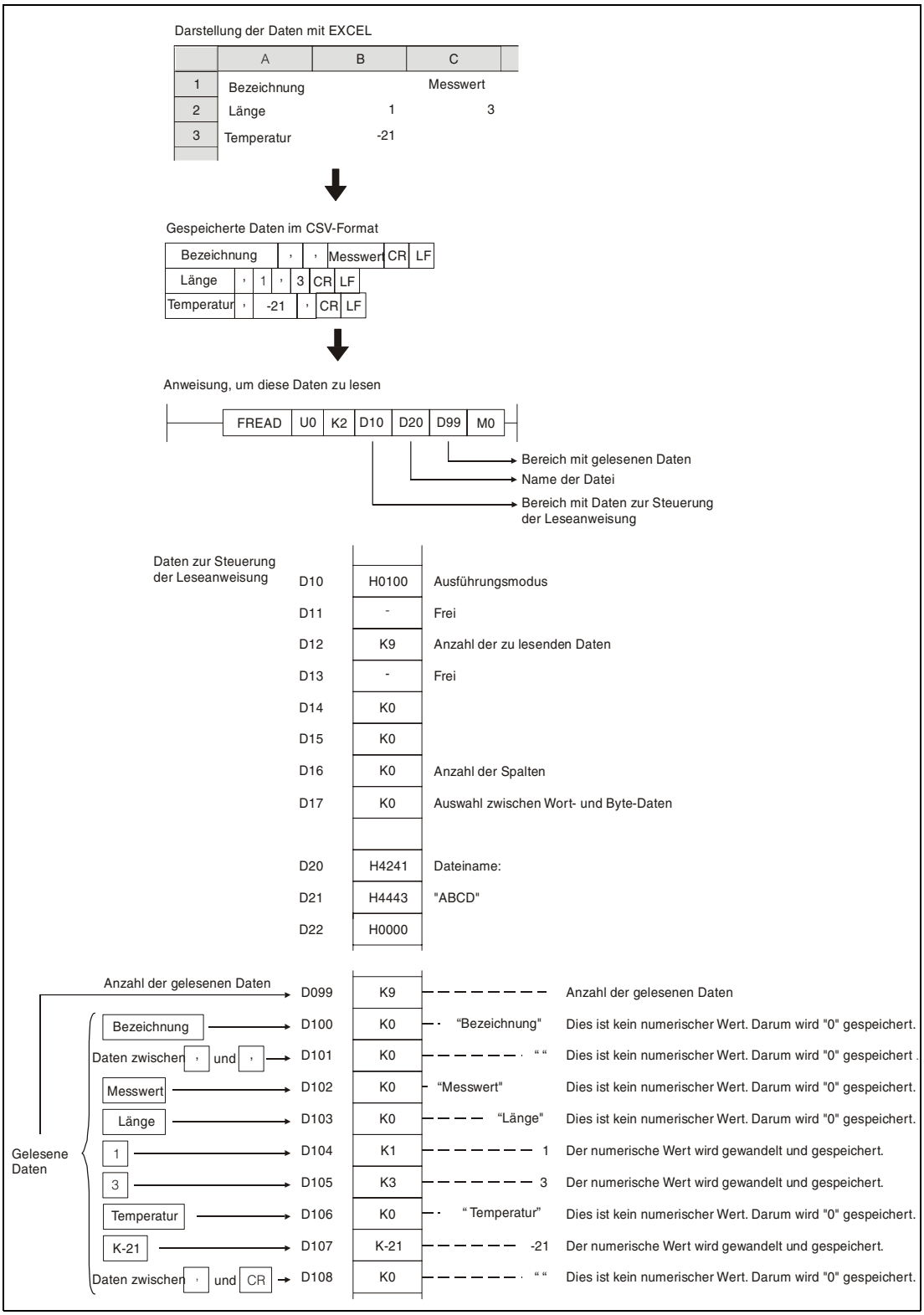
Das Bit, das in (d2)+0 angegeben wurde, wird automatisch gesetzt, wenn die Ausführung der FREAD-Anweisung erkannt und die END-Anweisung ausgeführt wird. Nach der END-Anweisung im nächsten Zyklus wird dieses Bit wieder rückgesetzt. Der Benutzter kann dieses Bit als Signal für die Beendigung der FREAD-Anweisung verwenden.
Falls bei der Ausführung der Anweisung ein Fehler aufgetreten ist, wird das Bit, das in (d2)+1 angegeben wurde, synchron mit der Fertigmeldung [(d2)+0] gesetzt und rückgesetzt.
Vom Benutzter kann dieses Signal als Fehlermeldung ausgewertet werden.
Während der Ausführung der FREAD-Anweisung ist SM721 gesetzt. SM721 wird auch bei der Ausführung anderer Anweisungen (S.WRITE, COMRD, PRC) gesetzt. Wenn SM721 gesetzt ist, kann die FREAD-Anweisung nicht gestartet werden. Falls dies versucht wird, wird die Anweisung nicht ausgeführt.
Falls vor der Ausführung der Anweisung (bevor SM721 gesetzt wird) ein Fehler erkannt wird, werden die Fertigmeldung [(d2)+0], die Fehlermeldung [(d2)+1] und SM721 nicht gesetzt.

Die Angabe der Datenlänge [(d0)+0] erfolgt in der Einheit „Wort“. Die folgende Abbildung zeigt die Übertragung von binären Daten.

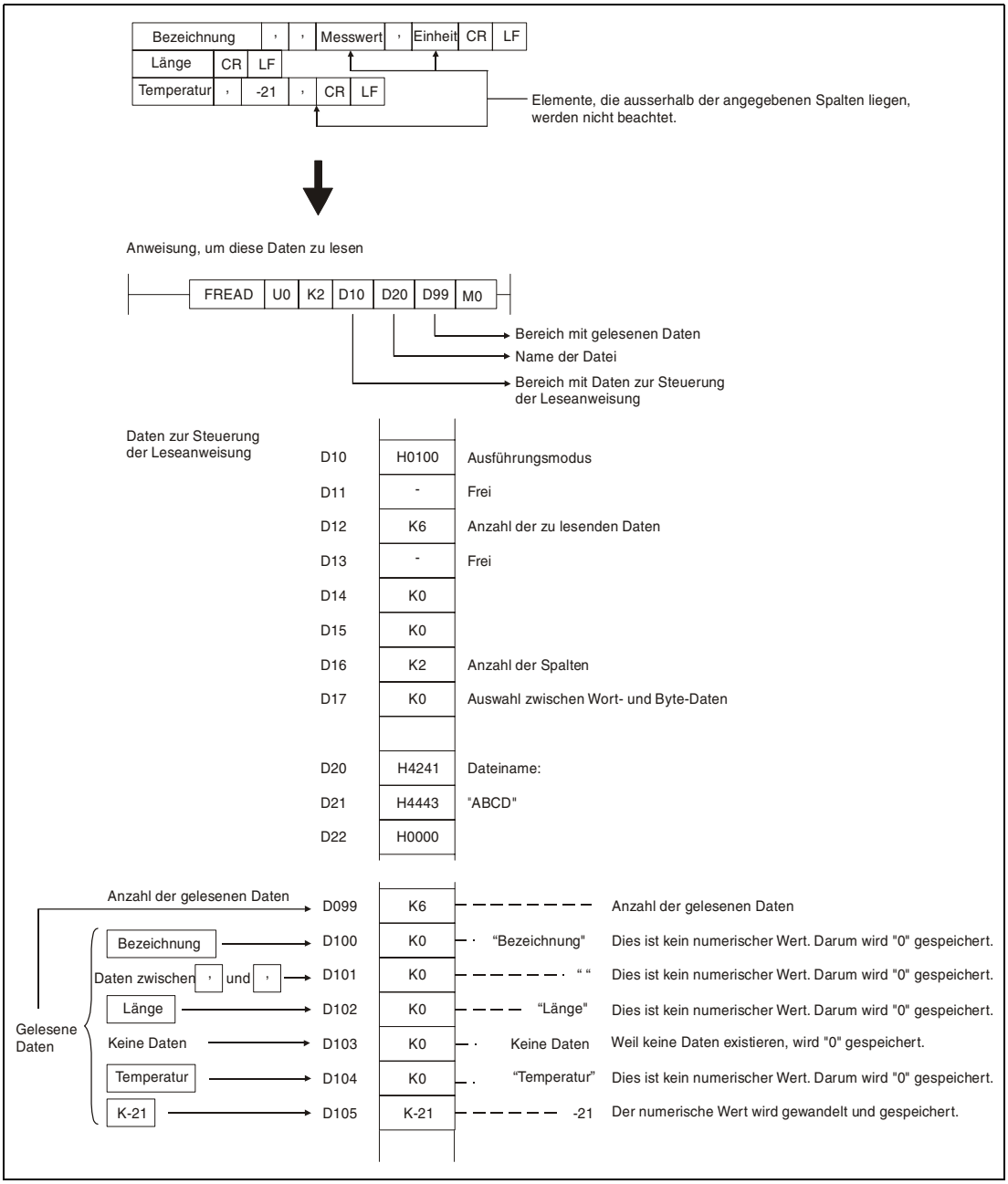


Übertragung binärer Daten:
Wenn binäre Daten übertragen werden, wird die Erweiterung „.BIN“ an den Dateinamen der Quelldatei angehängt. Wenn die Datei nicht existiert, wird eine Fehlermeldung ausgegeben. Wenn eine Anfangsadresse angegeben wird, die den Bereich der Quelldatei überschreitet, wird bei CPUs bis zu einer Seriennummer von „01111“ (die ersten fünf Stellen) eine Fehlermeldung ausgegeben. Eine CPU ab der Seriennummer „01112“ überträgt in diesem Fall keine Daten und beendet die Anweisung ohne Fehlermeldung.

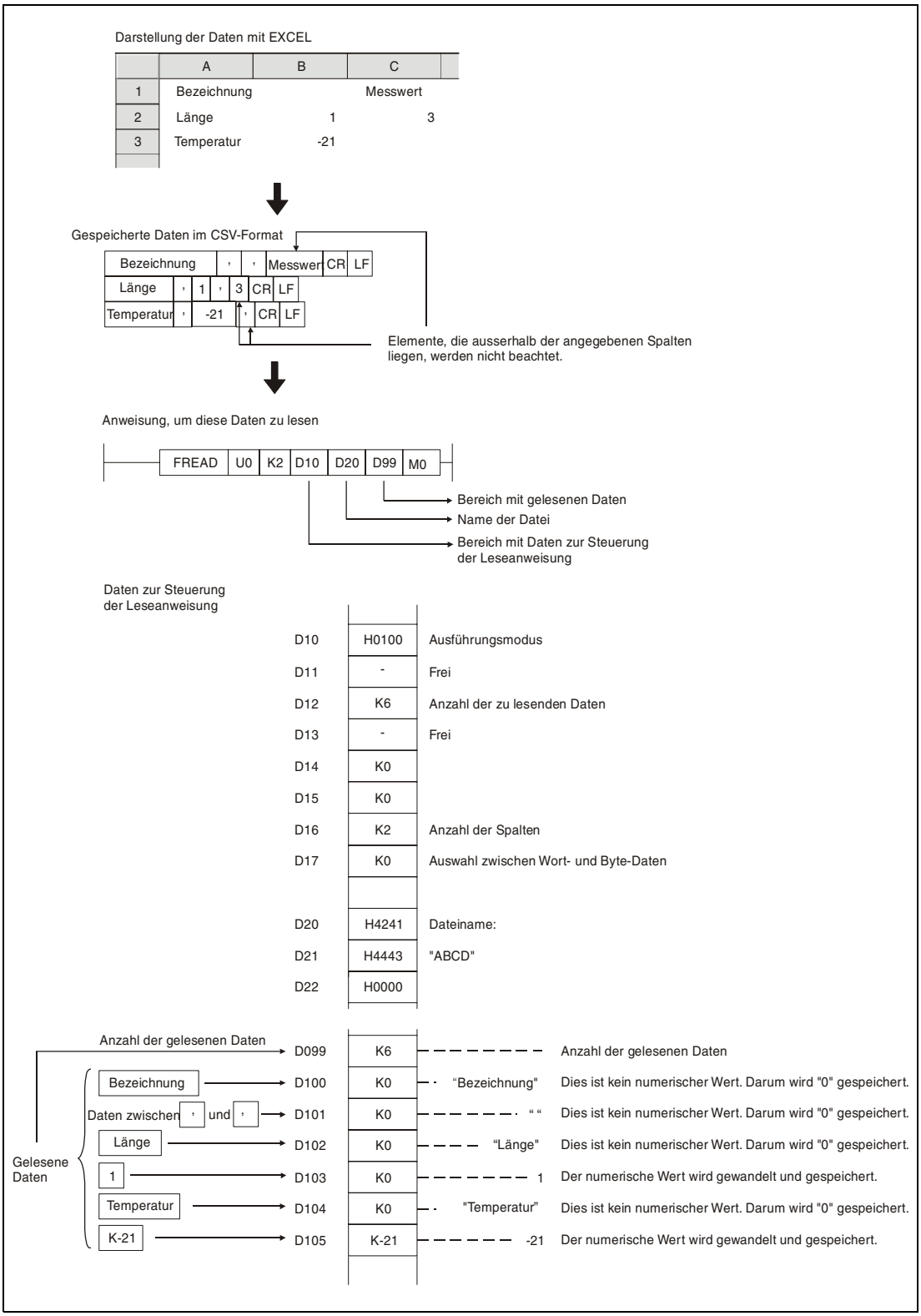
Übertragung von Daten nach Wandlung aus dem CSV-Format:
Die Elemente einer Datei im CSV-Format (Zellen für EXCEL) werden Zeile für Zeile gelesen. Numerische Werte werden in binäre Daten gewandelt und in den angegebenen Operanden gespeichert. Wenn für den Quelldateinamen keine Erweiterung angegeben wurde, wird „.CSV“ angefügt. Wenn die Quelldatei nicht existiert, wird eine Fehlermeldung ausgegeben.
Vom Anfang der Datei an werden so viele Elemente gelesen, wie in [(d0)+2] angegeben wurde. Wenn das Ende der Datei erreicht wird, bevor die angegebene Anzahl gelesen wurde, wird bei CPUs bis zu einer Seriennummer von „01111“ (die ersten fünf Stellen) eine Fehlermeldung ausgegeben. Eine CPU ab der Seriennummer „01112“ liest in diesem Fall die zur Verfügung stehenden Daten.
Wenn als Spaltenzahl „0“ vorgegeben wurde, werden die in der CSV-Datei vorhandenen Zeilen nicht beachtet. In der Abbildung auf der nächsten Seite wird dies verdeutlicht.


Darstellung der Daten mit EXCEL
A
B
C
1
Bezeichnung
Messwert
2
Länge
1
3
3
Temperatur
-21
Gespeicherte Daten im CSV-Format
Bezeichnung , , Messwert CR LF
Länge , 1 , 3 CR LF
Temperatur , -21 , CR LF
Anweisung, um diese Daten zu lesen
FREAD U0 K2 D10 D20 D99 M0
Bereich mit gelesenen Daten
Name der Datei
Bereich mit Daten zur Steuerung der Leseanweisung
Daten zur Steuerung der Leseanweisung
D10 H0100 Ausführungsmodus
D11 - Frei
D12 K9 Anzahl der zu lesenden Daten
D13 - Frei
D14 K0
D15 K0
D16 K0 Anzahl der Spalten
D17 K0 Auswahl zwischen Wort- und Byte-Daten
D20 H4241 Dateiname:
D21 H4443 "ABCD"
D22 H0000
Anzahl der gelesenen Daten
D099 K9 Anzahl der gelesenen Daten
Gelesene Daten
Bezeichnung
D100 K0 "Bezeichnung" Dies ist kein numerischer Wert. Darum wird "0" gespeichert.
Daten zwischen , und ,
D101 K0 " " Dies ist kein numerischer Wert. Darum wird "0" gespeichert.
Messwert
D102 K0 "Messwert" Dies ist kein numerischer Wert. Darum wird "0" gespeichert.
Länge
D103 K0 "Länge" Dies ist kein numerischer Wert. Darum wird "0" gespeichert.
1
D104 K1 1 Der numerische Wert wird gewandelt und gespeichert.
3
D105 K3 3 Der numerische Wert wird gewandelt und gespeichert.
Temperatur
D106 K0 " Temperatur" Dies ist kein numerischer Wert. Darum wird "0" gespeichert.
K-21
D107 K-21 -21 Der numerische Wert wird gewandelt und gespeichert.
Daten zwischen , und CR
D108 K0 " " Dies ist kein numerischer Wert. Darum wird "0" gespeichert.

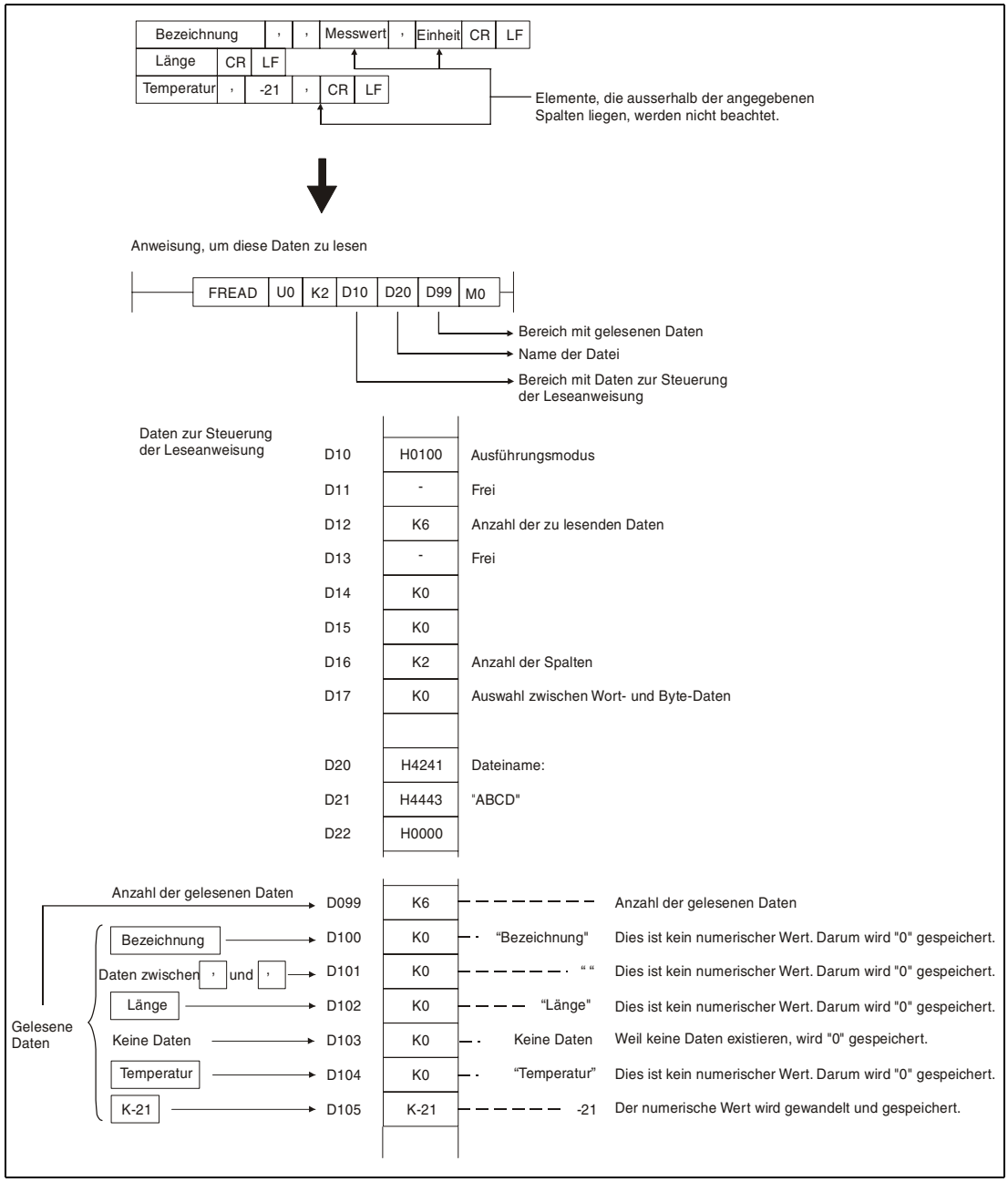
Wenn sich die Anzahl der Spalten in jeder Zeile ändert, werden die Daten ebenfalls ohne Beachtung der Zeilen gelesen. Von EXCEL werden solche Dateien nicht erzeugt, sie können aber nach Modifikationen durch den Benutzer entstehen.



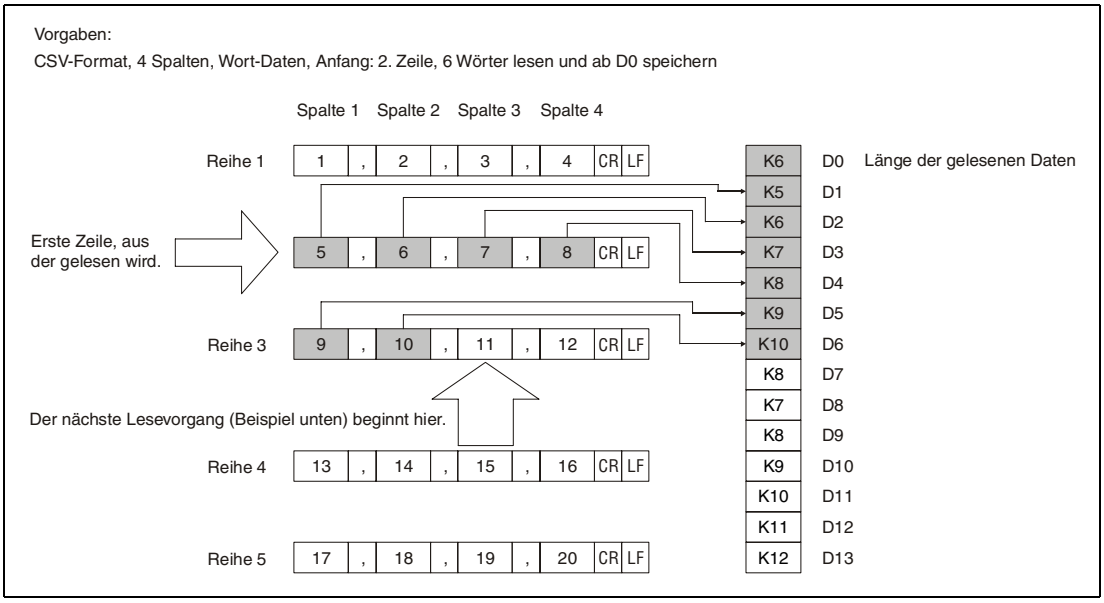
Wenn aus dem CSV-Format gewandelte Dateien gelesen werden und die vorgegebene Anzahl der Spalten ist nicht „0“, wird in der CSV-Datei ein Datenfeld mit der entsprechenden Anzahl von Spalten erwartet. Elemente, die ausserhalb dieser Spalten liegen, werden nicht beachtet. In der folgenden Abbildung ist dieser Fall dargestellt.

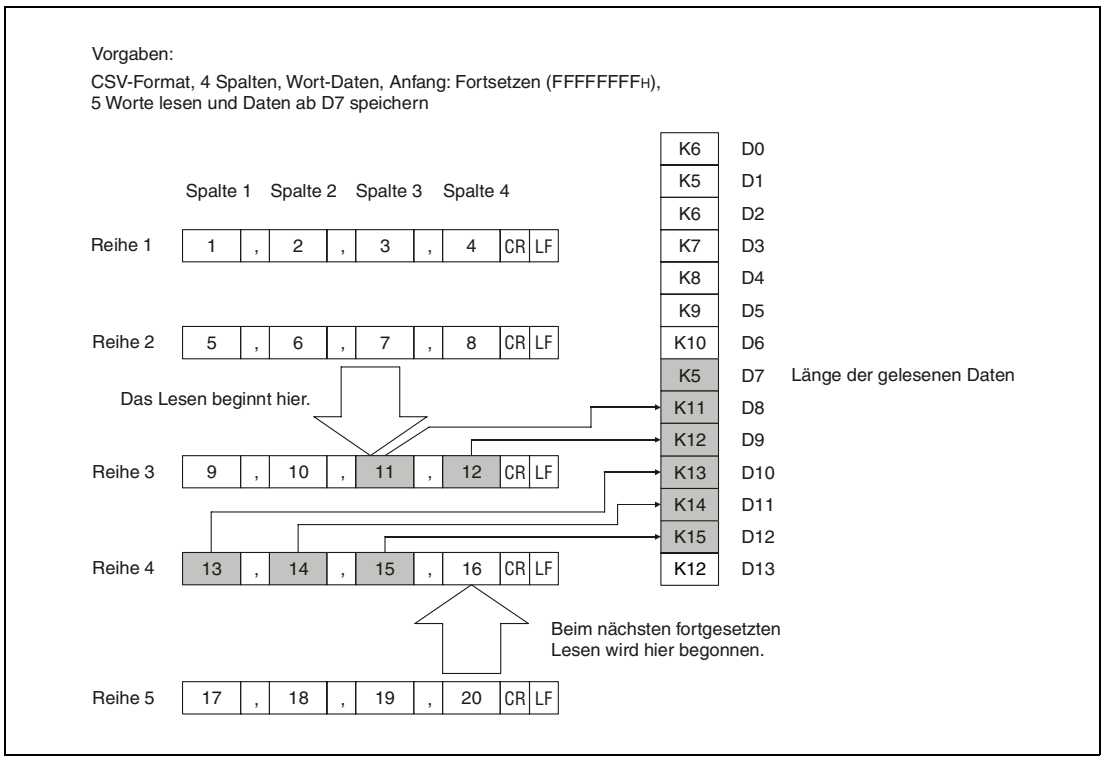


Wenn sich die Anzahl der Spalten in jeder Zeile ändert und eine bestimmte Anzahl Spalten vorgegeben wurde, werden die Elemente, die ausserhalb der Spalten liegen, ignoriert und eine „0“ wird dort eingefügt, wo keine Elemente vorhanden sind.
Wenn die Anzahl der Zeilen niedriger ist, als durch die zu übertragene Datenmenge [(d0)+2] vorgegeben ist, wird dort, wo keine Zeilen mehr vorhanden sind, „0“ eingetragen.



Die beiden folgenden Abbildungen zeigen Beispiele, bei denen mit einer CPU ab der Seriennummer „01112“ (die ersten fünf Stellen) Daten nacheinander aus derselben Datei gelesen werden.





Beim fortgesetztem Lesen dürfen die Einstellungen für Daten-Format, Anzahl der Spalten und die Festlegung, ob Byte- oder Wortdaten übertragen werden müssen, zwischen den einzelnen Lesevorgängen nicht verändert werden.

Während des fortgesetzten Lesens darf keine andere FREAD- oder FWRITE-Anweisung ausgeführt werden.

Numerische Werte im CSV-Format werden, wie in der nachfolgenden Tabelle dargestellt, gewandelt.

Numerische Werte im CSV-Format	Wort-Operand	
	ohne Vorzeichen	Mit Vorzeichen
-32768	32768	-32768
-1	65535	-1
0	0	0
1	1	1
32767	32767	32767
32768	32768	-32768
65535	65535	-1

Für numerische Werte, die ausserhalb des zulässigen Bereiches liegen und für Elemente, die keine numerische Werte sind, wird bei der Umwandlung vom CSV-Format eine „0“ eingesetzt.

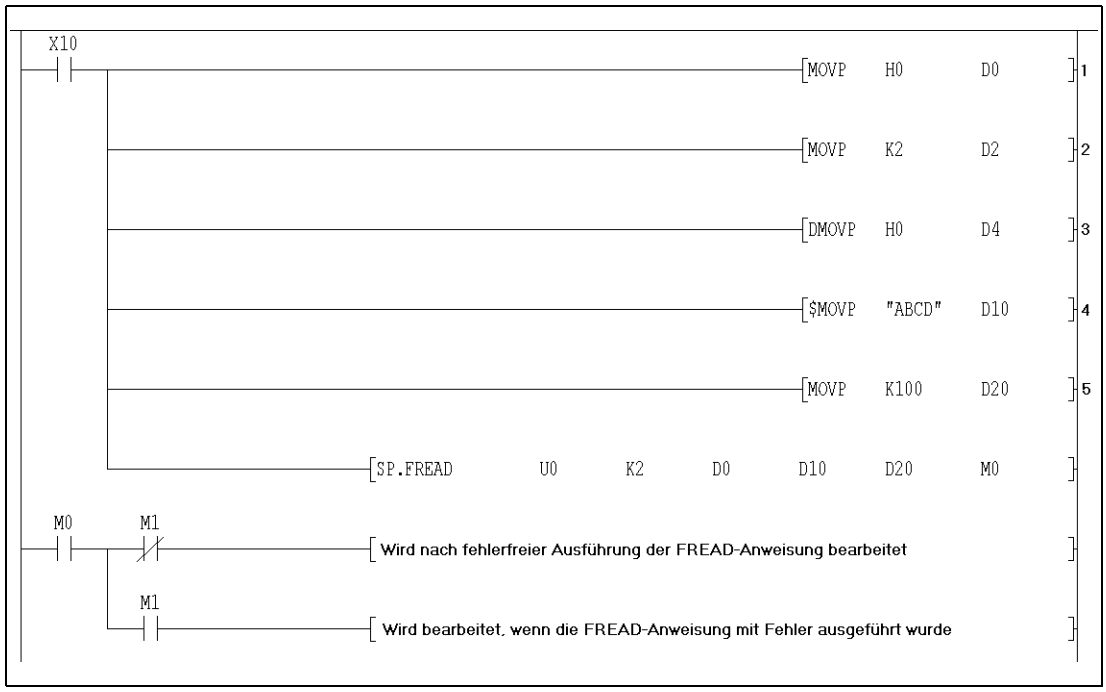
Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Das Laufwerk, das in s0 angegeben wurde, enthält eine Speicherkarte, die keine ATA-Speicherkarte ist (Fehlercode 4100).
- Werte, die als Kontrolldaten eingetragen wurden, liegen ausserhalb der zulässigen Bereiche (Fehlercode 4100).
- Die in (d0)+0 angegebene Länge der Daten überschreitet den zulässigen Bereich (Fehlercode 4101).
- Ein ungültiger Operand wurde angegeben (Fehlercode 4004).
- Eine Datei mit dem unter s1 angegebenen Namen existiert nicht auf dem angegebenem Laufwerk (Fehlercode 2410).
- Die Länge der zu lesenden Daten überschreitet die Größe des Zielbereiches (Fehlercode 4101).
- Beim Lesen binärer Daten ist die Länge der Daten in der Quelldatei kleiner als die Datenmenge, die ab (d0)+2 vorgegeben ist (Fehlercode 4100).

Beispiel 1 FREAD

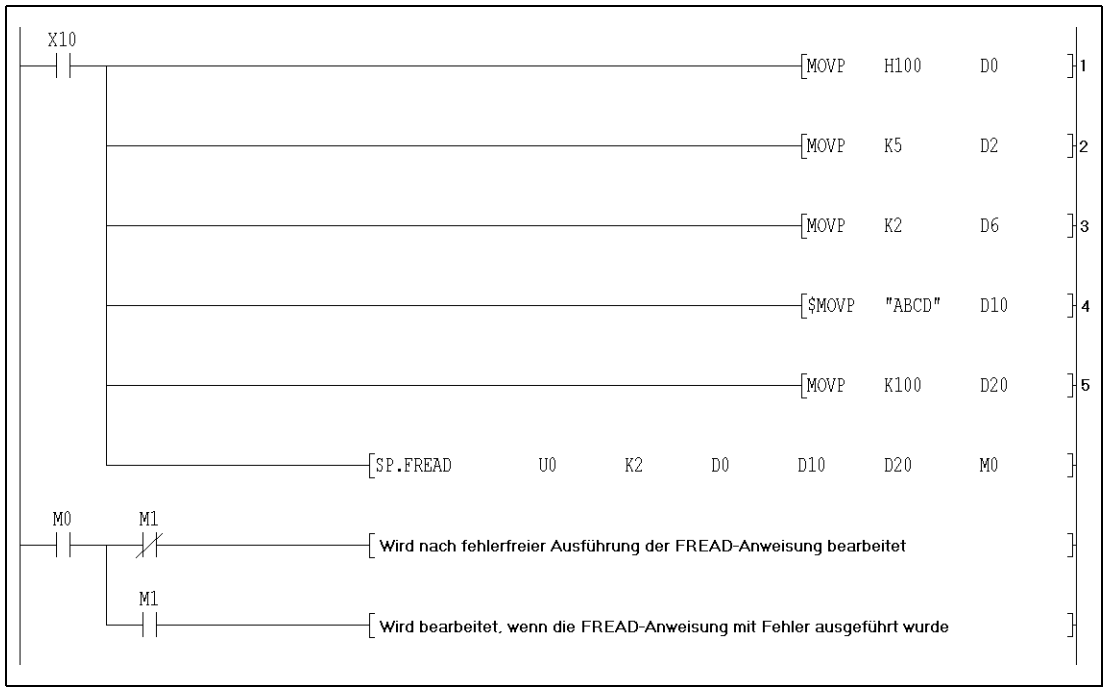
Im folgenden Programm werden vier Bytes binäre Daten aus der Datei „ABCD.BIN“ gelesen, wenn X10 gesetzt wird. Das Lesen beginnt am Anfang der Datei. Die Datei „ABCD.BIN“ ist auf der Speicherkarte, die in Laufwerk 2 steckt, abgelegt.
Für die Kontrolldaten werden ab D0 acht Operanden belegt.
Für die gelesenen Daten sind ab D20 einhundert Bytes reserviert.



[1] Ausführungsmodus festlegen (binäre Daten)
[2] Anzahl der zu lesenden Datenwörter
[3] Anfangsadresse in Quelldatei (am Dateianfang starten)
[4] Dateinamen eintragen
[5] Größe des Bereiches für gelesene Daten festlegen

Beispiel 2 FREAD (GX Developer)

Das folgende Programm liest Daten aus der Datei „ABCD.CSV“, von der Speicherkarte, die in Laufwerk 2 steckt, wenn X10 gesetzt wird. Die Daten bestehen ausschließlich aus numerischen Werten und sind zweispaltig im CSV-Format abgelegt.
Für die Kontrolldaten werden ab D0 acht Operanden belegt.
Für die gelesenen Daten sind ab D20 einhundert Bytes reserviert.



[1] Ausführungsmodus festlegen (CSV-Format)
[2] Anzahl der zu lesenden Datenwörter
[3] Anzahl der Spalten festlegen
[4] Dateinamen eintragen
[5] Größe des Bereiches für gelesene Daten festlegen

9.4 Programmanweisungen

9.4.1 PLOADP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					●[1]

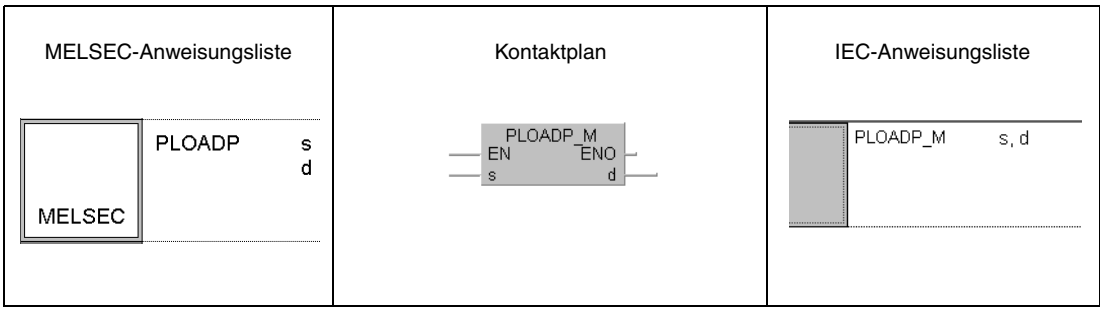
[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	—	—	—	—	●	SM0	3
d	●*	—	—	—	—	—	—	—	—		

* Lokale Operanden sind nicht zulässig.

GX IEC Developer



GX Developer

Variablen

Operand	Befehlswert	Datentyp
s	Nummer des Laufwerkes, auf dem das Programm gespeichert ist und Zeichenfolge mit dem Namen der Datei oder Anfangsadresse des Operanden, in dem der Dateiname als Zeichenfolge gespeichert ist.	BIN-16-Bit
d	Operand, der nach Ausführung der Anweisung für einen Zyklus gesetzt wird.	Bit

HINWEIS *Das File-System wird vom GX IEC Developer nicht unterstützt.*

Funktions-weise

Laden eines Programmes von einer Speicherkarte

PLOADP Programm laden

Mit der PLOADP-Anweisung wird ein Programm von einer Speicherkarte, die in den Laufwerken 1, 2 oder 4 stecken kann, in den internen Speicher (Laufwerk 0) übertragen und in den Standby-Modus geschaltet. Auf Laufwerk 0 muss ausreichend Speicherplatz vorhanden sein.

Bei der Benennung der Programmdatei muss die Erweiterung „.QPG“ nicht angegeben werden.

Der Operand, der unter d eingetragen ist, wird nach der Verarbeitung der END-Anweisung des Zyklusses gesetzt, in dem die PLOADP-Anweisung vollständig ausgeführt wurde. Der Operand wird nach der Ausführung der nächsten END-Anweisung wieder ausgeschaltet.

Die Anweisungen PLOADP, PUNLOADP und PSWAPP können nicht gleichzeitig ausgeführt werden. Die Anweisungen, die aufgerufen werden, nachdem schon eine andere der Programmanweisungen ausgeführt wird, werden nicht bearbeitet. Im Programm sind entsprechende Verriegelungen vorzusehen.

Das geladene Programm erhält die niedrigste freie Programmnummer, die in der CPU zur Verfügung steht. Die Programmnummern können mit dem Programmiergerät überprüft werden. Durch einen Eintrag in SD720 kann dem geladenen Programm eine Nummer zugeordnet werden.

Die PLOADP-Anweisung kann nicht in einem Interruptprogramm ausgeführt werden.

Um das mit der PLOADP-Anweisung geladene Programm auszuführen, wird die PSCAN-Anweisung aufgerufen.

Für das geladene Programm werden alle Einstellungen für File register, Device initial value, Kommentare und lokale Operanden eingestellt als „Use PLC file setting“.

Die Einstellungen zum Auffrischen der Ein- und Ausgänge werden für das geladene Programm als „Disabled“ (gesperrt) eingetragen.

Während die PLOADP-Anweisung ausgeführt wird, werden Programmänderungen, die mit dem Programmiergerät während des Betriebszustandes RUN gemacht werden, nicht übertragen. Erst nach Ausführung der PLOADP-Anweisung werden die Änderungen übertragen. Umgekehrt wird die PLOADP-Anweisung nicht ausgeführt, während vom Programmiergerät Programmänderungen übertragen werden.

Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Das Programm mit dem angegebenem Namen existiert nicht auf dem angegebenem Laufwerk (Fehlercode 2410).
- Die Nummer des angegebenen Laufwerkes ist ungültig (Fehlercode 4100).
- Im internen Speicher (Laufwerk 0) steht nicht genügend Speicherplatz zur Verfügung, um das angegebene Programm zu laden (Fehlercode 2413).
- Die max. Anzahl von Programmen ist bereits im Programmspeicher eingetragen (Fehlercode 4101).
- Ein Programm mit der in SD720 eingetragenen Programmnummer existiert bereits oder die in SD720 angegebene Programmnummer ist größer als die maximal mögliche Programmnummer (Fehlernummer 4101).
- Ein Programm-File mit demselben Namen wie das Programm, das geladen werden soll, existiert bereits (Fehlernummer 2410).
- Der Speicherplatz für die lokalen Operanden kann nicht reserviert werden (Fehlernummer 2401).

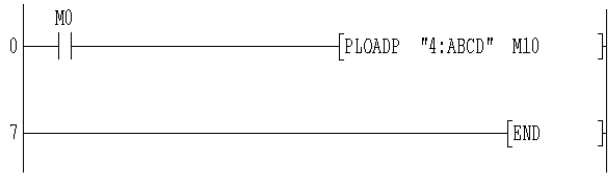
Typ der CPU	Anzahl Programme, die gespeichert werden können	Größte Programmnummer
Q02(H)	28	28
Q06H	60	60
Q12H	124	124
Q25H	124	124

Beispiel

PLOADP (GX Developer)

Wenn im folgenden Beispiel M0 gesetzt wird, wird das Programm „ABCD.QPG“ von Laufwerk 4 nach Laufwerk 0 übertragen und in den Standby-Modus geschaltet.

```
0   LD      M0
1   PLOADP  "4:ABCD" M10
7   END
```



9.4.2 PUNLOADP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					●[1]

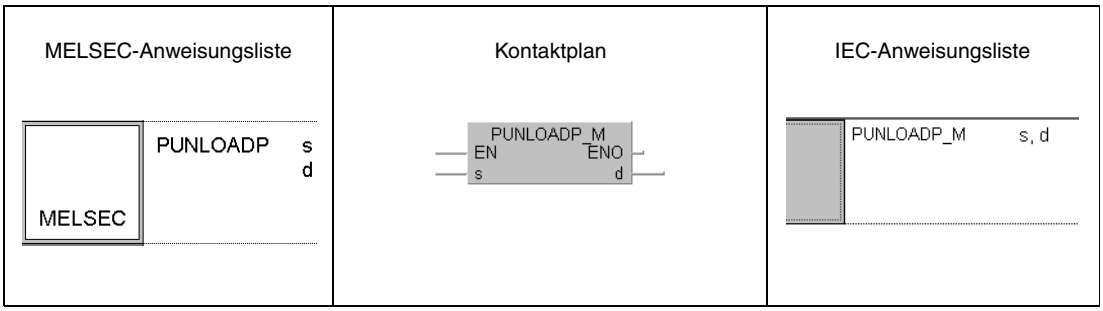
[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	—	—	—	—	●	SM0	3
d	●*	—	—	—	—	—	—	—	—		

* Lokale Operanden sind nicht zulässig.

GX IEC Developer



GX Developer

Variablen

Operand	Befehlswert	Datentyp
s	Zeichenfolge mit dem Namen der Datei, die gelöscht werden soll oder Anfangsadresse des Operanden, in dem der Dateiname als Zeichenfolge gespeichert ist.	BIN-16-Bit
d	Operand, der nach Ausführung der Anweisung für einen Zyklus gesetzt wird.	Bit

Funktionsweise

Löschen eines Programmes aus dem Programmspeicher

PUNLOADP Programm löschen

Mit der PUNLOADP-Anweisung wird ein Programm, das im Programmspeicher (Laufwerk 0) eingetragen und im Standby-Modus ist, gelöscht. Ein Programm im Standby-Modus, das von einer PSCAN-Anweisung angesprochen wird, kann nicht gelöscht werden.

Bei der Benennung der Programmdatei muss die Erweiterung „.QPG“ nicht angegeben werden.

Der Operand, der unter d eingetragen ist, wird nach der Verarbeitung der END-Anweisung des Zyklusses gesetzt, in dem die PUNLOADP-Anweisung vollständig ausgeführt wurde. Der Operand wird nach der Ausführung der nächsten END-Anweisung wieder ausgeschaltet.

Die Anweisungen PUNLOADP, PLOADP und PSWAPP können nicht gleichzeitig ausgeführt werden. Die Anweisungen, die aufgerufen werden, nachdem schon eine andere der Programmanweisungen ausgeführt wird, werden nicht bearbeitet. Im Programm sind entsprechende Verriegelungen vorzusehen.

Wenn nach dem Löschen des Programmes das Netzteil aus- und wieder eingeschaltet oder die CPU zurückgesetzt wird, wird der Fehler „Programmeinstellung fehlerhaft“ mit dem Fehlercode 2400 gemeldet. Um dies zu beheben, muss der Programmname des gelöschten Programmes aus den Programmeinstellungen gelöscht werden.

Die PUNLOADP-Anweisung kann nicht in einem Interruptprogramm ausgeführt werden.

Das Programm, das mit der PUNLOADP-Anweisung gelöscht wird, sollte zuvor mit der PSTOP-Anweisung in den Standby-Modus gebracht werden.

Während die PUNLOADP-Anweisung ausgeführt wird, werden Programmänderungen, die mit dem Programmiergerät während des Betriebszustandes RUN gemacht werden, nicht übertragen. Erst nach Ausführung der PUNLOADP-Anweisung werden die Änderungen übertragen. Umgekehrt wird die PUNLOADP-Anweisung nicht ausgeführt, während vom Programmiergerät Programmänderungen übertragen werden.

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

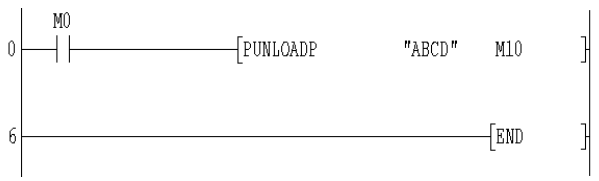
- Das Programm mit dem angegebenem Namen existiert nicht (Fehlercode 2410).
- Das Programm mit dem angegebenem Namen ist nicht im Standby-Modus oder wird ausgeführt (Fehlercode 4101).
- Im Programmspeicher ist nur das angegebene Programm vorhanden (Fehlercode 4101).

Beispiel

PUNLOADP (GX Developer)

Im folgenden Beispiel wird das Programm „ABCD.QPG“ aus dem Laufwerk 0 gelöscht, wenn M0 gesetzt wird.

```
0  LD       M0
1  PUNLOADP "ABCD"   M10
6  END
```



9.4.3 PSWAPP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					●[1]

[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

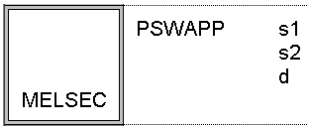
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	—	●	●	—	—	—	—	—	●	SM0	3
s2	—	●	●	—	—	—	—	—	●		
d	●*	—	—	—	—	—	—	—	—		

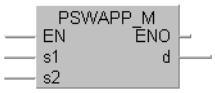
* Lokale Operanden sind nicht zulässig.

GX IEC Developer

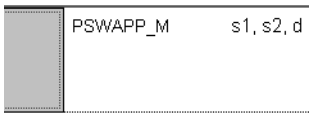
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

Operand	Befehlswert	Datentyp
s1	Zeichenfolge mit dem Namen der Datei, die gelöscht werden soll oder Anfangsadresse des Operanden, in dem der Dateiname als Zeichenfolge gespeichert ist.	BIN-16-Bit
s2	Nummer des Laufwerkes, auf dem das zu ladene Programm gespeichert ist und Zeichenfolge mit dem Namen der Datei oder Anfangsadresse des Operanden, in dem der Dateiname als Zeichenfolge gespeichert ist.	BIN-16-Bit
d	Operand, der nach Ausführung der Anweisung für einen Zyklus gesetzt wird.	Bit

Funktions-weise

Löschen eines Programmes aus dem Programmspeicher und Programm laden

PSWAPP Programm löschen und anderes Programm laden

Mit der PSWAPP-Anweisung wird ein Programm, das im Programmspeicher (Laufwerk 0) eingetragen und sich im Standby-Modus befindet, gelöscht. Ein Programm im Standby-Modus, das von einer PSCAN-Anweisung angesprochen wird, kann nicht gelöscht werden.
Danach wird ein Programm aus Laufwerk 1, 2 oder 4 in den internen Speicher (Laufwerk 0) übertragen und in den Standby-Modus geschaltet. Auf Laufwerk 0 muss ausreichend Speicherplatz vorhanden sein

Bei der Benennung der Programmdatei muss die Erweiterung „.QPG“ nicht angegeben werden.

Der Operand, der unter d eingetragen ist, wird nach der Verarbeitung der END-Anweisung des Zyklusses gesetzt, in dem die PSWAPP-Anweisung vollständig ausgeführt wurde. Der Operand wird nach der Ausführung der nächsten END-Anweisung wieder ausgeschaltet.

Das geladene Programm erhält die Programmnummer des gelöschten Programmes.

Die Anweisungen PUNLOADP, PLOADP und PSWAPP können nicht gleichzeitig ausgeführt werden. Die Anweisungen, die aufgerufen werden, nachdem schon eine andere der Programmanweisungen ausgeführt wird, werden nicht bearbeitet. Im Programm sind entsprechende Verriegelungen vorzusehen.

Wenn nach dem Löschen des Programmes das Netzteil aus- und wieder eingeschaltet oder die CPU zurückgesetzt wird, wird der Fehler „Programmeinstellung fehlerhaft“ mit dem Fehlercode 2400 gemeldet. Um dies zu beheben, kann der Programmname des gelöschten Programmes in den Programmeinstellungen durch den Namen des geladenen Programmes ersetzt werden.

Die PSWAPP-Anweisung kann nicht in einem Interruptprogramm ausgeführt werden.

Für das geladene Programm werden alle Einstellungen für File-Register, Anfangswerte der Operanden, Kommentare und lokale Operanden eingestellt als „Use PLC file setting“.

Die Einstellungen zum Auffrischen der Ein- und Ausgänge werden für das geladene Programm als „Disabled“ (gesperrt) eingetragen.

Während die PSWAPP-Anweisung ausgeführt wird, werden Programmänderungen, die mit dem Programmiergerät während des Betriebszustandes RUN gemacht werden, nicht übertragen. Erst nach Ausführung der PSWAPP-Anweisung werden die Änderungen übertragen. Umgekehrt wird die PSWAPP-Anweisung nicht ausgeführt, während vom Programmiergerät Programmänderungen übertragen werden.

Fehler-quellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Das Laufwerk oder das Programm mit dem angegebenem Namen (s1 oder s2) existiert nicht (Fehlercode 2410).
- Die Nummer des angegebenen Laufwerkes (s2) ist ungültig (Fehlercode 4100).
- Im internen Speicher (Laufwerk 0) steht nicht genügend Speicherplatz zur Verfügung, um das angegebene Programm zu laden (Fehlercode 2413).
- Das Programm mit dem angegebenem Namen (s1) ist nicht im Standby-Modus oder wird ausgeführt (Fehlercode 4101).

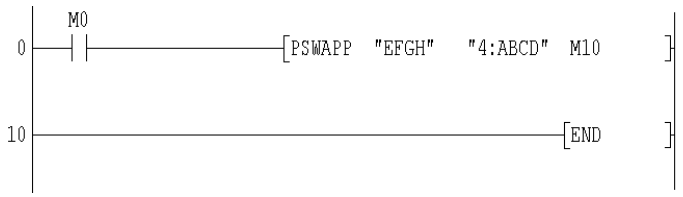
Beispiel PSWAPP (GX Developer)

Wird M0 im folgenden Beispiel gesetzt, wird das Programm „EFGH.QPG“ aus dem Laufwerk 0 gelöscht. Dann wird das Programm „ABCD.QPG“ aus Laufwerk 4 in Laufwerk 0 übertragen und in den Standby-Modus versetzt.

Anweisungsliste

```
  0  LD      M0
  1  PSWAPP  "EFGH"   "4:ABCD" M10
 10  END
```

Kontaktplan



9.5 Transferanweisungen

9.5.1 RBMOV, RBMOVP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					●[1]

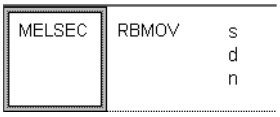
[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

Operanden MELSEC Q

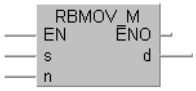
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	●	●	●	●	●	●	—	—	—	SM0	4
d	●	●	●	●	●	●	—	—	—		
n	●	●	●	●	●	●	●	●	—		

GX IEC Developer

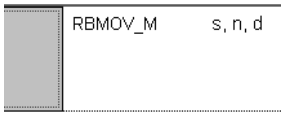
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

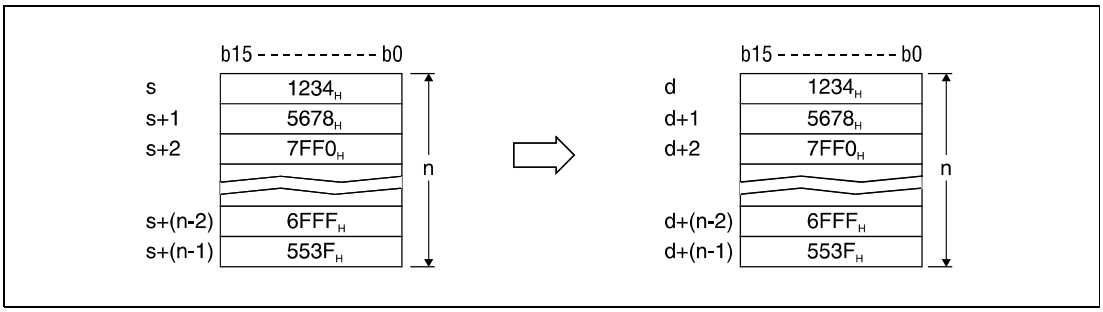
Operand	Befehlswert	Datentyp
s	Anfangsadresse des Operanden, in dem die zu übertragenden Daten gespeichert sind.	BIN-16-Bit
d	Anfangsadresse des Operanden, in dem die Daten gespeichert werden.	
n	Anzahl der zu übertragenden Datenblöcke.	

Funktions-weise

Übertragen von Binärdatenblöcken mit hoher Geschwindigkeit

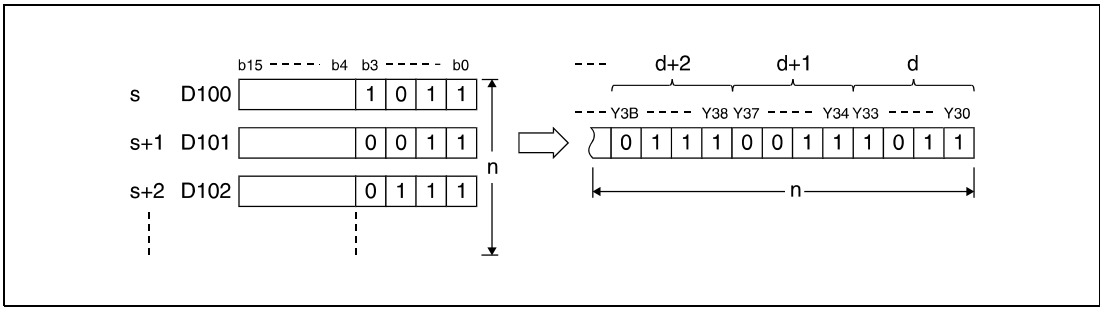
RBMOV/RBMOVP Blockweise Datenübertragung (16 Bit)

Mit Hilfe der RBMOV-Anweisung kann ein Block von aufeinanderfolgenden Adressen gleichzeitig übertragen werden. In s wird die erste zu übertragende Adresse festgelegt. Der Wert in „n“ gibt die Anzahl der aufeinanderfolgenden Adressen an. Die Daten werden in Blöcken zu „n“ Adressen an die Zieladresse beginnend mit d übertragen.



Eine Datenübertragung ist auch dann möglich, wenn Quelle und Ziel die gleichen Adressen enthalten. Die Übertragung zu den Operanden mit der kleineren Adresse beginnt mit s und die Übertragung zu den Operanden mit der höheren Adresse mit s+(n-1).

Ist s ein Wort-Operand und d ein Bit-Operand, werden die über die Bit-Bestimmung angegebenen Stellen des Wort-Operanden an den Bit-Operanden übertragen. Wird zum Beispiel K1Y30 für d gesetzt, werden die niedrigstwertigen 4 Bits der über s festgelegten Wort-Operanden übertragen.



Handelt es sich bei den Adressen um Bit-Operanden, muss die Anzahl der Bits in d und s identisch sein.

HINWEIS *Die RBMOV- und die RBMOVP-Anweisung bieten sich an, wenn bei einer Q(H)-CPU große Mengen an File-Registerdaten übertragen werden sollen. Bei einer Q-CPU gleichen diese Anweisungen den BMOV-Anweisungen. Die folgende Tabellen dienen zum Vergleich der Verarbeitungsgeschwindigkeiten der RBMOV- und der BMOV-Anweisung.*

QnH-CPU				
Speichermedium	**RBMOV-Anweisung**		**BMOV-Anweisung**	
	Zeit (µs) zur Übertragung von		**Zeit (µs) zur Übertragung von**	
	100 Wörtern	**1000 Wörtern**	**100 Wörtern**	**1000 Wörtern**
SRAM	56,30	367,73	44,37	393,14
Eingebautes RAM	44,37	393,14		
Flash ROM	29	308		

Qn-CPU				
Speichermedium	**RBMOV-Anweisung**		**BMOV-Anweisung**	
	Zeit (µs) zur Übertragung von		**Zeit (µs) zur Übertragung von**	
	100 Wörtern	**1000 Wörtern**	**100 Wörtern**	**1000 Wörtern**
SRAM	115,89	579,47	63,83	535,23
Eingebautes RAM				
Flash ROM				

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, und das Error Flag wird gesetzt:

- Die in n angegebene Anzahl von Datenblöcken in s und d liegt außerhalb des für die Speicherung vorgesehenen Bereiches der Operanden (Fehlercode 4101).
- Für s und d ist kein File-Register angegeben (Fehlercode 4101).

Beispiel 1 RBMOVP

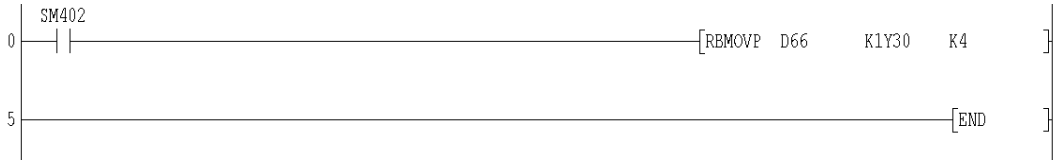
Das folgende Programm überträgt mit positiver Flanke von SM402 die Daten der untersten 4 Bits (von b0 bis b3) von D66 bis D69 an die Ausgänge Y30 bis Y3F. Die Anzahl der zu übertragenden Blöcke (4) ist durch die Konstante K4 angegeben.

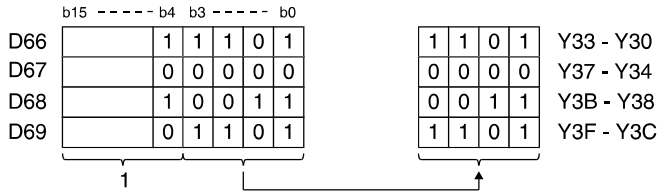
Die Bit-Schemen zeigen die Bit-Struktur jeweils vor und nach der Übertragung.

Anweisungsliste

```
0   LD      SM402
1   RBMOVP  D66     K1Y30   K4
5   END
```

Kontaktplan



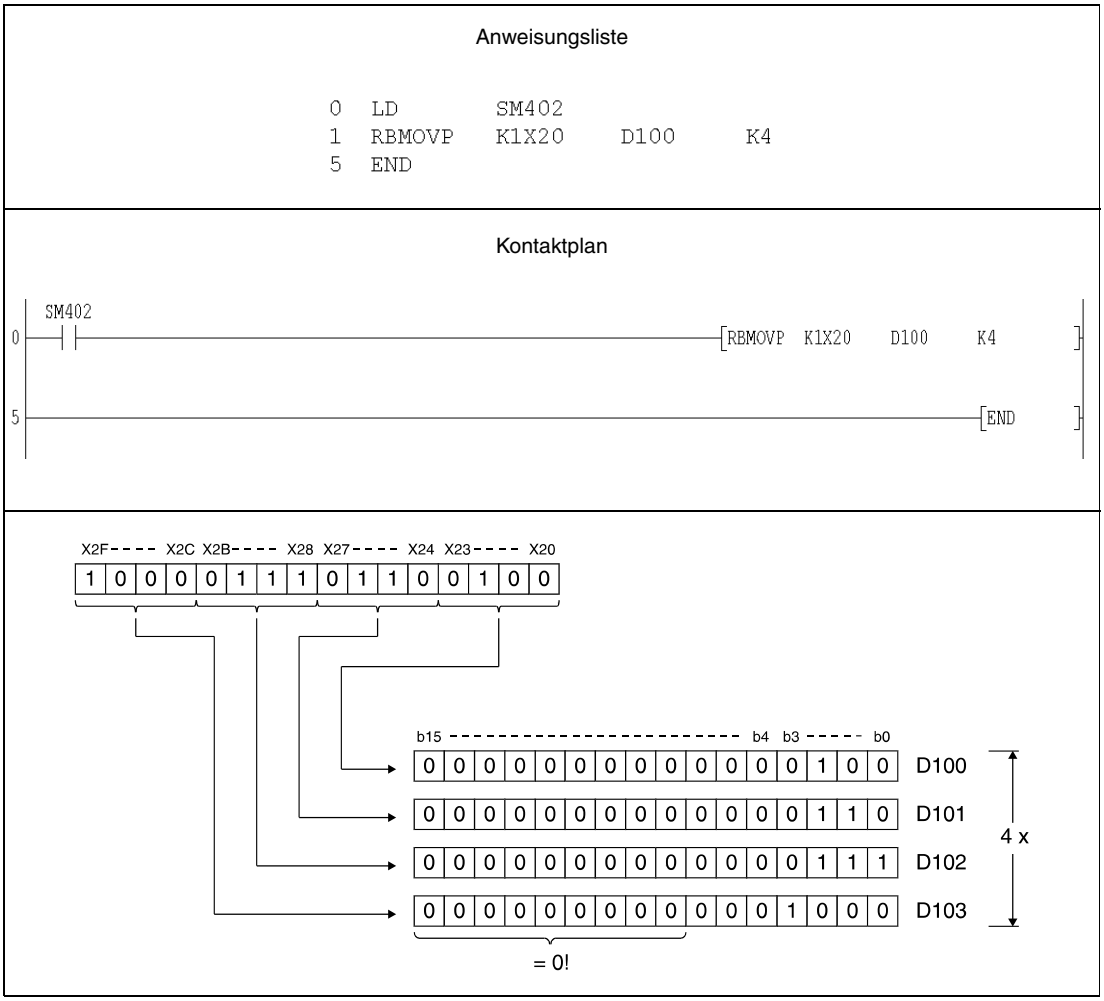


[1] Diese Bits bleiben bei der Operation unberücksichtigt.

Beispiel 2 RBMOVP

Das folgende Programm überträgt mit positiver Flanke von SM402 die Daten von X20 bis X2F an die Register D100 bis D103. Die Anzahl der Datenblöcke (4) ist durch die Konstante K4 angegeben.

Die Bit-Schemen zeigen die Bit-Struktur jeweils vor und nach der Übertragung.



9.6 Anweisungen für den Multi-CPU-Betrieb

9.6.1 S.TO, SP.TO

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					●[1]

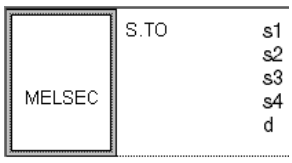
[1] Nur für Q02-, Q02H-, Q06H-, Q12H- und Q25HCPU ab Software-Version B.

Operanden MELSEC Q

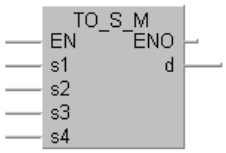
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	—	●	●	—	—	—	—	●	—	SM0	5
s2	—	●	●	—	—	—	—	●	—		
s3	—	●	●	—	—	—	—	—	—		
s4	—	●	●	—	—	—	—	●	—		
d	●	●	●	—	—	—	—	—	—		

GX IEC Developer

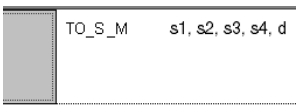
MELSEC-Anweisungsliste



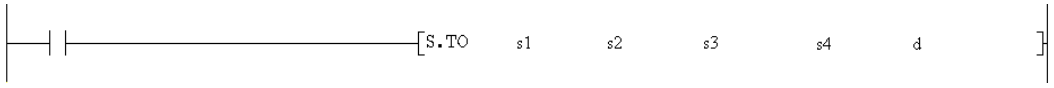
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

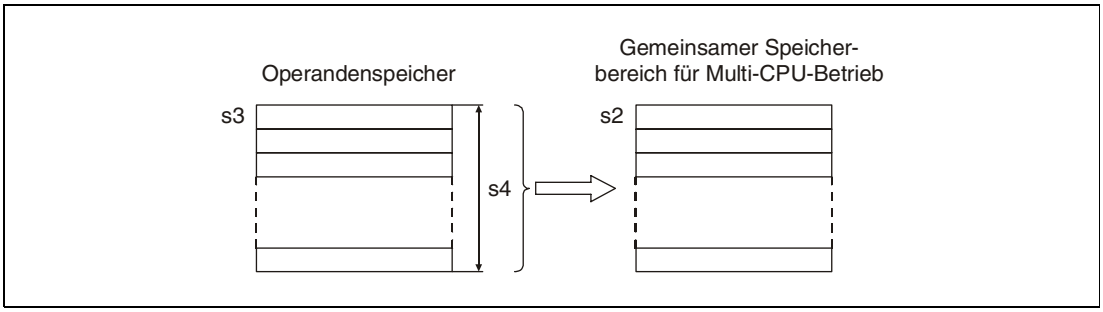
Operand	Befehlswert	Datentyp
s1	Kopfadresse der CPU, in der die S.TO-Anweisung ausgeführt wird	BIN-16-Bit
s2	Anfangsadresse im Anwenderbereich des gemeinsamen Speichers für Multi-CPU-Betrieb	
s3	Startadresse des Bereiches, in dem die zu schreibenden Daten gespeichert sind	
s4	Anzahl der zu schreibenden Datenworte (1 bis 256)	
d	Bitoperand, der nach dem Schreiben für einen Programmzyklus eingeschaltet wird	Bit

Funktionsweise

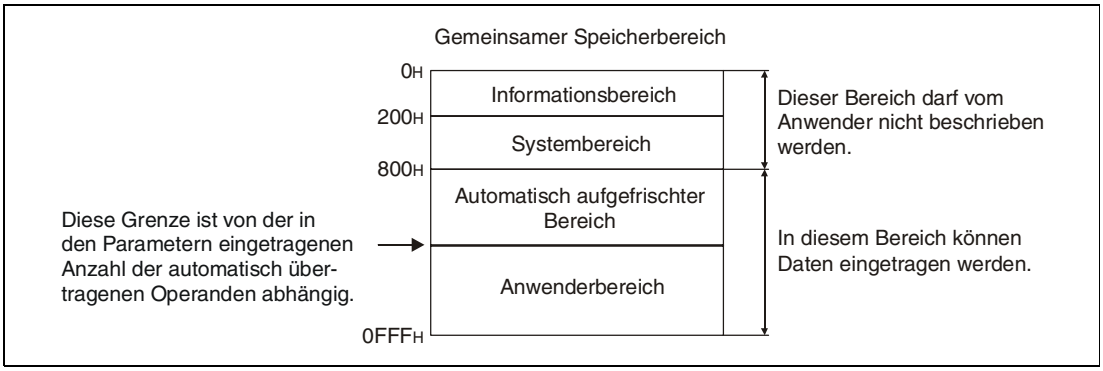
Daten in den gemeinsamen Speicherbereich für Multi-CPU-Betrieb eintragen

S.TO/SP.TO Schreibanweisung

Mit der S.TO-Anweisung wird die in s4 angegebene Anzahl von Datenwörtern in den gemeinsamen Speicher der CPU übertragen, in der die S.TO-Anweisung ausgeführt wird. Die Startadresse des Quellbereiches in derselben CPU wird in s3 angegeben. s2 enthält die Zieladresse der Daten im gemeinsamen Speicherbereich. Mit der S.TO-Anweisung können keine Daten direkt zu anderen CPUs in einem Multi-CPU-System übertragen werden.



Der gemeinsame Speicherbereich einer CPU des System Q dient zum Datenaustausch mit den anderen CPU-Modulen in einem Multi-CPU-System. Ab der Adresse 800H beginnt der automatisch aufgefrischte Bereich, an dem sich der Anwenderbereich anschließt.



Die Kopfadresse der CPU wird durch den Steckplatz der CPU festgelegt. In s1 werden die ersten 3 Stellen der Kopfadresse eingetragen:

Steckplatz auf dem Baugruppenträger	CPU	0	1	2
CPU-Nummer im Multi-CPU-System	1	2	3	4
Kopfadresse	3E00	3E10	3E20	3E30
Inhalt von s1	3E0	3E1	3E2	3E3

Wird in s4 die Anzahl der zu schreibenden Worte mit „0“ angegeben, wird die Anweisung nicht ausgeführt, und der in d angegebene Bitoperand wird nicht gesetzt.

HINWEIS *In jeder CPU darf in einem Programmzyklus nur eine S.TO-Anweisung ausgeführt werden. Interne Verriegelungen bewirken, dass nur die zuerst gestartete S.TO-Anweisung ausgeführt wird, falls in einem Zyklus mehrere S.TO-Anweisungen aufgerufen werden.*

Fehlerquellen

In folgenden Fällen tritt ein Verarbeitungsfehler auf, das Error Flag wird gesetzt und in SD0 wird der Fehlercode eingetragen:

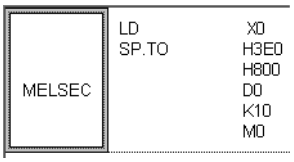
- Die in s4 angegebene Anzahl von Datenwörtern liegt außerhalb des Bereiches von 0 bis 256 (Fehlercode 4101).
- Die in s2 angegebene Anfangsadresse im Anwenderbereich des gemeinsamen Speichers überschreitet den Adressbereich des gemeinsamen Speichers (Fehlercode 4101).
- Durch die in s2 angegebene Anfangsadresse und die in s4 angegebene Anzahl von Datenwörtern wird der Adressbereich des gemeinsamen Speichers überschritten (Fehlercode 4101).
- Durch die in s3 angegebene Startadresse und die in s4 angegebene Anzahl von Datenwörtern wird der Adressbereich des Bereiches überschritten, in dem die zu schreibenden Daten gespeichert sind (Fehlercode 4101).
- In s1 ist nicht die Kopfadresse der CPU angegeben, in der die S.TO-Anweisung ausgeührt wird. (Fehlercode 2107).
- In s1 ist keine zulässige Kopfadresse ($3E0_H$, $3E1_H$, $3E2_H$ oder $3E3_H$) eingetragen (Fehlercode 4100).
- Die angegebene Anweisung ist unzulässig (Fehlercode 4002).
- Die angegebene Anzahl von Operanden ist falsch (Fehlercode 4003).
- Ein unzulässiger Operand wurde angegeben (Fehlercode 4002).

Beispiel

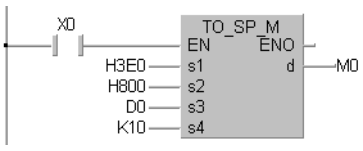
SP.TO

Die Inhalte der Datenregister D0 bis D9 aus CPU1 werden in den Anwenderbereich des gemeinsamen Speichers dieser CPU ab Adresse 800_H eingetragen, wenn X0 eingeschaltet ist.

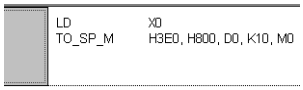
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



9.6.2 FROM, FROMP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					●[1]

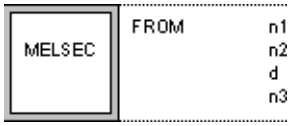
[1] Nur für Q02-, Q02H-, Q06H-, Q12H- und Q25HCPU ab Software-Version B.

Operanden MELSEC Q

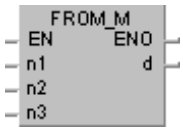
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort				U		
n1	—	●	●	●	●	●	●	●	●	SM0	5
n2	—	●	●	●	●	●	●	—	—		
d	—	●	●	—	—	—	—	—	—		
n3	—	●	●	●	●	●	●	—	—		

GX IEC Developer

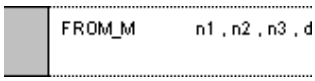
MELSEC-Anweisungsliste



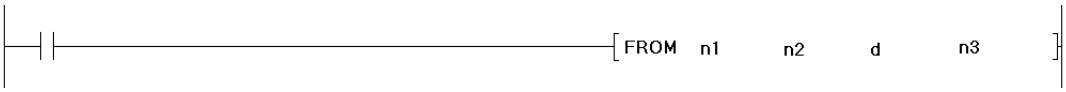
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

Operand	Befehlswert	Datentyp
n1	Kopfadresse der CPU, aus der Daten gelesen werden	BIN-16-Bit
n2	Anfangsadresse im Anwenderbereich des gemeinsamen Speichers, aus dem gelesen wird	
d	Startadresse des Bereiches, in dem die gelesenen Daten gespeichert werden	
n3	Anzahl der zu lesenden Datenworte (1 bis 6144)	

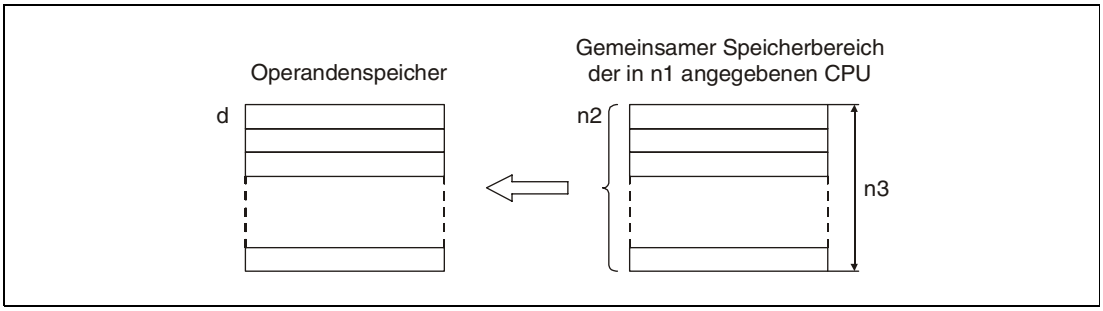
HINWEIS *In Kap. 7.8.1 ist beschrieben, wie die FROM-Anweisung genutzt werden kann, um Daten aus einem Sondermodul zu lesen.*

Funktions-weise

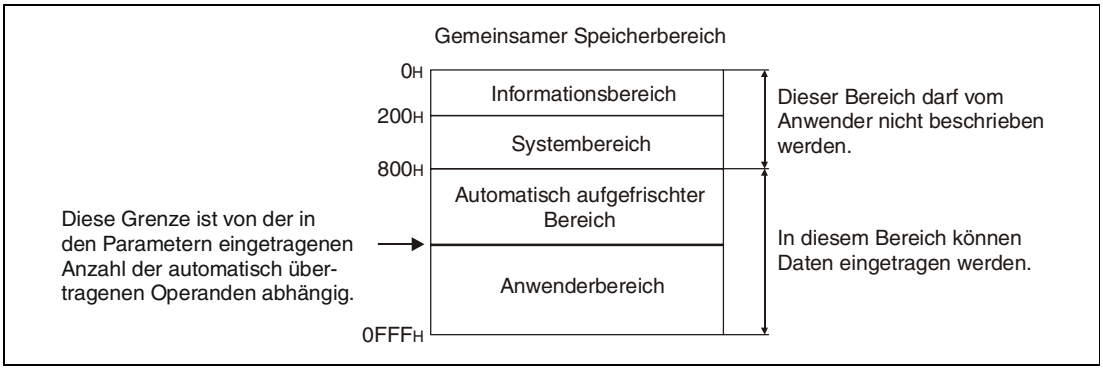
Daten aus dem gemeinsamen Speicherbereich einer anderen CPU lesen

FROM/FROMP Lesen von Wort-Daten

In einem Multi-CPU-System werden mit der FROM-Anweisung Wort-Daten aus dem Anwenderbereich des gemeinsamen Speichers einer anderen CPU gelesen. Die Kopfadresse dieser CPU wird in n1 vorgegeben. In n3 wird die Anzahl der Datenwörtern angegeben, die übertragen werden soll. n2 enthält die Startadresse der Daten in der anderen CPU. In d wird angegeben, wo die Daten in der CPU gespeichert werden sollen, in der die FROM-Anweisung ausgeführt wird.



Der gemeinsame Speicherbereich einer CPU des System Q dient zum Datenaustausch mit den anderen CPU-Modulen in einem Multi-CPU-System. Ab der Adresse 800H beginnt der automatisch aufgefrischte Bereich, an dem sich der Anwenderbereich anschließt.



Die Kopfadresse der CPU wird durch den Steckplatz der CPU festgelegt. In n1 werden die ersten 3 Stellen der Kopfadresse eingetragen:

Steckplatz auf dem Baugruppenträger	CPU	0	1	2
CPU-Nummer im Multi-CPU-System	1	2	3	4
Kopfadresse	3E00	3E10	3E20	3E30
Inhalt von n1	3E0	3E1	3E2	3E3

Nach dem Lesen der Daten wird der Sondermerker SM390 gesetzt. SM390 wird nicht gesetzt, wenn sich die CPU, aus der gelesen werden sollte, im Reset-Status befand. In diesem Fall wird keine Fehlermeldung ausgegeben.

Die Anweisung wird nicht ausgeführt, wenn in n3 die Anzahl der zu lesenden Worte mit „0“ vorgegeben wird.

Fehler-quellen

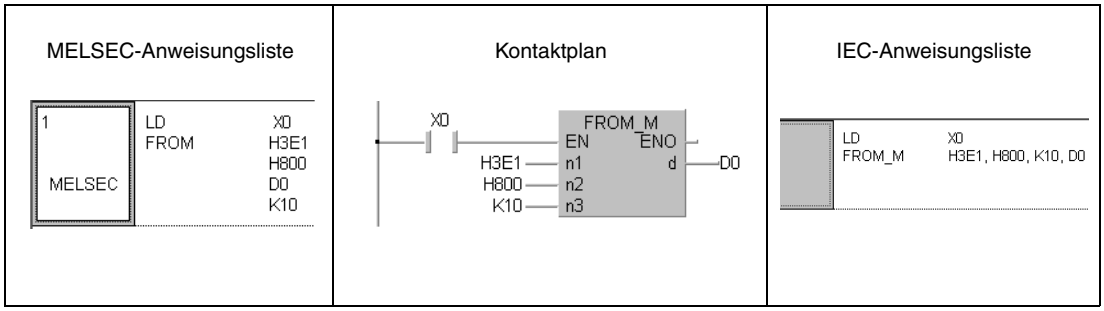
In folgenden Fällen tritt ein Verarbeitungsfehler auf, das Error Flag wird gesetzt und in SD0 wird der Fehlercode eingetragen:

- Die in n2 angegebene Anfangsadresse im Anwenderbereich überschreitet den Adressbereich des gemeinsamen Speichers (Fehlercode 4101).
- Durch die in n2 angegebene Anfangsadresse und die in n3 angegebene Anzahl von Datenwörtern wird der Adressbereich des gemeinsamen Speichers überschritten (Fehlercode 4101).
- Durch die in d angegebene Startadresse und die in n3 angegebene Anzahl von Datenwörtern wird der Adressbereich des Bereiches überschritten, in dem die gelesenen Daten gespeichert werden (Fehlercode 4101).
- In n1 ist die Kopfadresse der CPU angegeben, in der die FROM-Anweisung ausgeführt wird. (Fehlercode 2114).
- Auf dem in n1 mit der Kopfadresse angegebenen Steckplatz befindet sich kein CPU-Modul (Fehlercode 2110).

Beispiel

FROM

Wenn X0 eingeschaltet ist, werden aus dem Anwenderbereich des gemeinsamen Speichers in CPU2 ab der Adresse 800_H zehn Datenworte gelesen und in die Datenregister D0 bis D9 der CPU eingetragen, in der die FROM-Anweisung ausgeführt wird.



10 Anweisungen für eine Q4ARCPU

Mit zwei CPU-Modulen vom Typ Q4ARCPU kann ein redundantes System realisiert werden. Dabei übernimmt bei Ausfall einer CPU die zweite CPU nahtlos die Steuerungsaufgaben und sichert die Fortsetzung des Prozesses. Typische Einsatzgebiete für redundante Systeme sind z. B. Kraftwerke, verfahrenstechnische Anwendungen in der chemischen Industrie oder Wasserwerke.

Die in diesem Kapitel beschriebenen Anweisungen können nur bei einer Q4ARCPU verwendet werden.

Funktion	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor
Anlaufverhalten der CPU wählen	S.STMODE	STMODE_S_M
Verhalten bei Umschaltung der CPU	S.CGMODE	CGMODE_S_M
Daten zur Reserve-CPU transferieren	S.TRUCK	TRUCK_S_M
Daten in Pufferspeichern von Sondermodulen eintragen oder Datem aus Pufferspeicher lesen	S.SPREF	SPRE_S_M

10.1 Anweisungen zur Einstellung der Betriebsart

10.1.1 STMODE

CPU

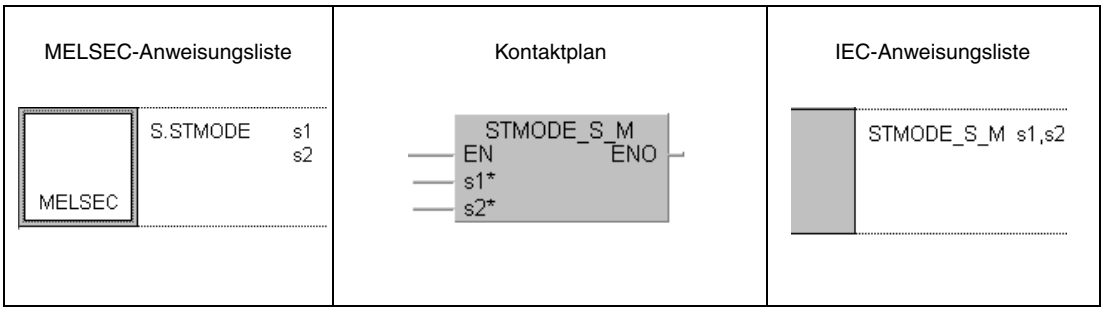
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●[1]	

[1] Nur für Q4AR

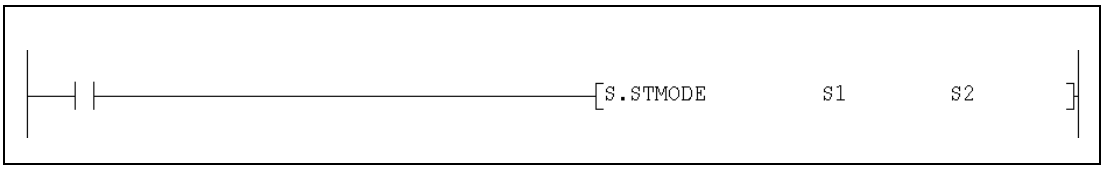
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	—	—	—	—	—	—	—	●	—	—	—
s2	—	—	—	—	—	—	—	●	—	—	

GX IEC Developer



GX Developer



Variablen

Operand	Befehlswert	Datentyp
s1	Einstellung des Anlaufverhaltens der CPU (0 = Neustart, 1 = Wiederanlauf)	BIN-16-Bit
s2	Max. Zeit für Spannungsausfall, nach der ein Neustart ausgeführt wird	BIN-16-Bit

Funktions-weise

Verhalten der CPU beim Einschalten der Versorgungsspannung einstellen

STMODE Anlaufverhalten einstellen

Durch den Inhalt von s1 kann gewählt werden, ob beim Einschalten der Versorgungsspannung und dem anschließenden Start der CPU die Operanden der CPU gelöscht werden (Neustart) oder ob sie ihre Werte behalten sollen (Warmstart).

Wenn die Zeit des Spannungsausfalls die in s2 eingestellte Zeit überschreitet, werden auch bei Wahl eines Warmstarts die Daten automatisch gelöscht und ein Neustart ausgeführt.

Diese Anweisung wird beim Einschalten der Spannung ausgeführt, und auch dann, wenn die Ausführungsbedingung der Anweisung nicht eingeschaltet ist. Die Ausführungsbedingung wird als Dummy betrachtet. Wird die Ausführungsbedingung im normalen Programmzyklus eingeschaltet, wird sie wie eine NOP-Anweisung behandelt.

In jedem System wird diese Anweisung einmal benötigt. Bei mehreren Programmen darf sie nur einmal vorhanden sein. Wird die Anweisung mehrfach programmiert, kann die korrekte Ausführung nicht garantiert werden.

Der Inhalt von s1 kann nur 0 oder 1 sein:
0: Neustart (Operanden außerhalb des Latch-Bereiches werden gelöscht)
1: Warmstart (Die Operanden werden nicht wie beim Neustart gelöscht, Index-Register und Ergebnisse (Signal-Flow) werden jedoch gelöscht und Sondermerker und -register erhalten ihre Voreinstellung.)

Die Zeit in s2 wird in Sekunden angegeben (0 bis 65535). Wird „0“ eingestellt, kann kein Neustart ausgeführt werden. Werte, die 32767 überschreiten, müssen als hexadezimale Zahl vorgegeben werden.

Fehler-quellen

Im folgenden Fall tritt ein Verarbeitungsfehler auf, das Error Flag (SM0) wird gesetzt und im Sonderregister SD0 wird ein Fehlercode eingetragen:

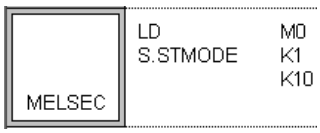
- Wenn für s1 oder s2 Werte angegeben wurden, die außerhalb des zulässigen Bereiches liegen. (Fehlercode: 4104).

Beispiel

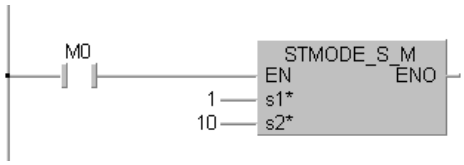
STMODE

Mit der folgenden Programmsequenz wird ein Warmstart angewählt. Fällt die Spannung länger als 10 s aus, wird bei Wiederkehr der Spannung ein Neustart ausgeführt.

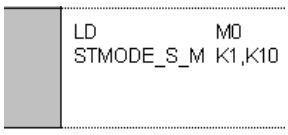
MELSEC-Anweisungsliste



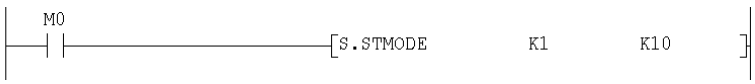
Kontaktplan (GX IEC Developer)



IEC-Anweisungsliste



Kontaktplan (GX Developer)



10.1.2 CGMODE

CPU

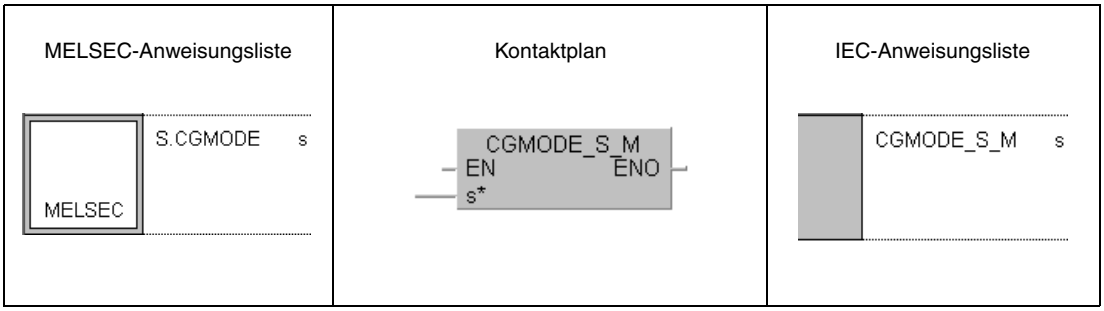
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●[1]	

[1] Nur für Q4AR

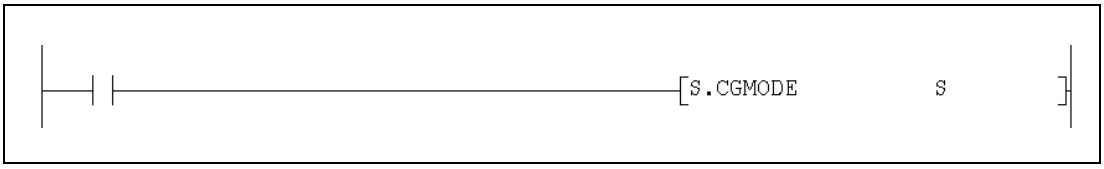
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	—	—	—	—	—	—	—	●	—	—	—

GX IEC Developer



GX Developer



Variablen

Operand	Befehlswert	Datentyp
s	Einstellung der Betriebsart	BIN-16-Bit

Funktions-weise

Verhalten beim Umschalten der CPU-Module einstellen

CGMODE Betriebsart einstellen

Mit dieser Anweisung kann eingestellt werden, ob beim Umschalten auf das Reservesystem die Operanden der CPU gelöscht werden oder ob sie ihre Werte behalten sollen. Die Auswahl wird durch den Inhalt der Variablen s bestimmt.

Diese Anweisung wird beim Einschalten der Spannung ausgeführt, und auch dann, wenn die Ausführungsbedingung der Anweisung nicht eingeschaltet ist. Die Ausführungsbedingung wird als Dummy betrachtet. Wird die Ausführungsbedingung im normalen Programmzyklus eingeschaltet, wird sie wie eine NOP-Anweisung behandelt.

In jedem System kann diese Anweisung nur einmal programmiert werden. Verwenden Sie die Anweisung auch bei mehreren Programmen nur in einem Programm. Wird die Anweisung mehrfach programmiert, kann die korrekte Ausführung nicht garantiert werden.

Der Inhalt von s kann nur 0 oder 1 sein:
0: Neustart (Operanden außerhalb des Latch-Bereiches werden gelöscht)
1: Warmstart (Die Operanden und Ergebnisse werden nicht wie beim Neustart gelöscht, Sondermerker und -register erhalten ihre Voreinstellung).

Fehler-quellen

Im folgenden Fall tritt ein Verarbeitungsfehler auf, das Error Flag (SM0) wird gesetzt und im Sonderregister SD0 wird ein Fehlercode eingetragen:

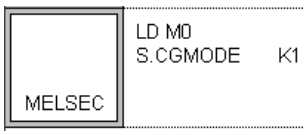
- Wenn für s ein Wert angegeben wurde, der außerhalb des zulässigen Bereichs liegt. (Fehlercode: 4104).

Beispiel

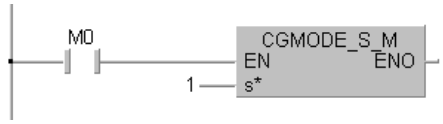
CGMODE

Mit der folgenden Programmsequenz wird eingestellt, dass die CPU beim Umschalten auf das Reservesystem einen Warmstart ausführt.

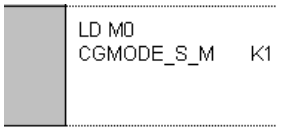
MELSEC-Anweisungsliste



Kontaktplan (GX IEC Developer)



IEC-Anweisungsliste



Kontaktplan (GX Developer)

10.2 Anweisungen für den Datentransfer

10.2.1 TRUCK

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●[1]	

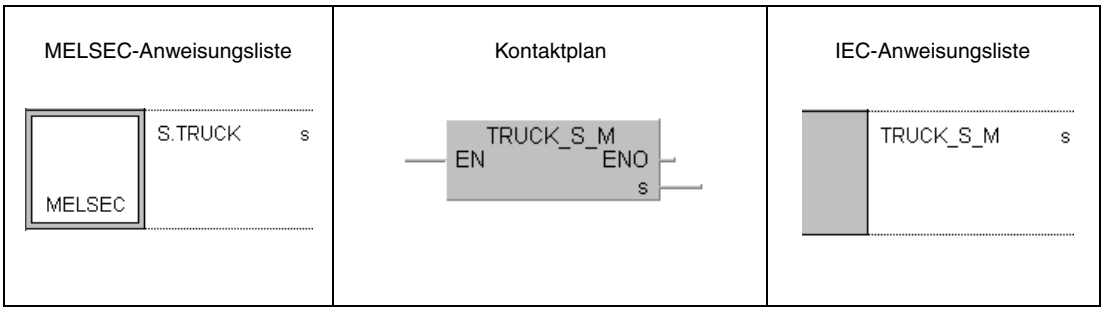
[1] Nur für Q4AR

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	—	●[1]	●	—	—	—	—	—	—	—	—

[1] Es sind nur Operanden aus dem Latch-Bereich zulässig.

GX IEC Developer



GX Developer

Variablen

Operand	Befehlswert	Datentyp
s	Anfangsadresse des Parameterblocks	BIN-16-Bit

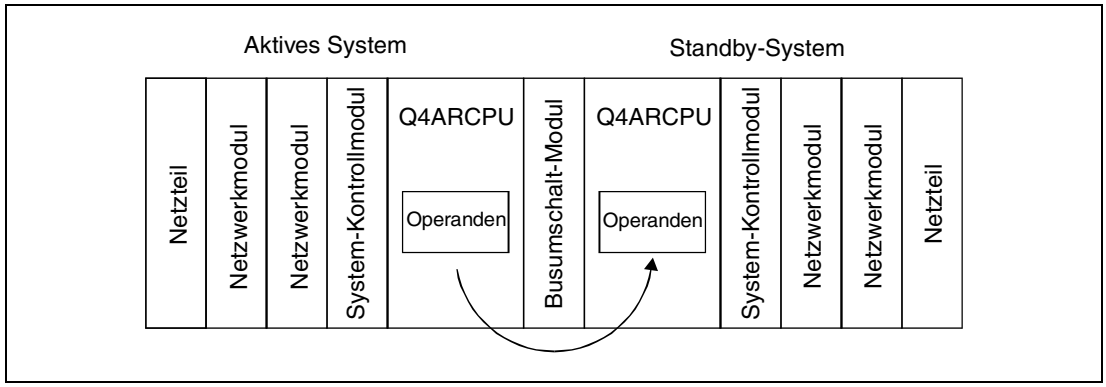
Funktionsweise

Datenaustausch zwischen den CPU-Modulen eines redundanten Systems

TRUCK Daten in die Reserve-CPU übertragen

In einem redundanten System werden am Ende eines Programmzyklus (während der END-Verarbeitung) Daten aus der aktiven Q4ARCPU zur Reserve-CPU (ebenfalls eine Q4ARCPU) übertragen.

Mit der TRUCK-Anweisung wird festgelegt, welche Daten übertragen werden. Die Variable s enthält die Anfangsadresse eines Parameterblocks, in dem wiederum der Typ und die Anzahl der zu übertragenden Operanden angegeben ist.

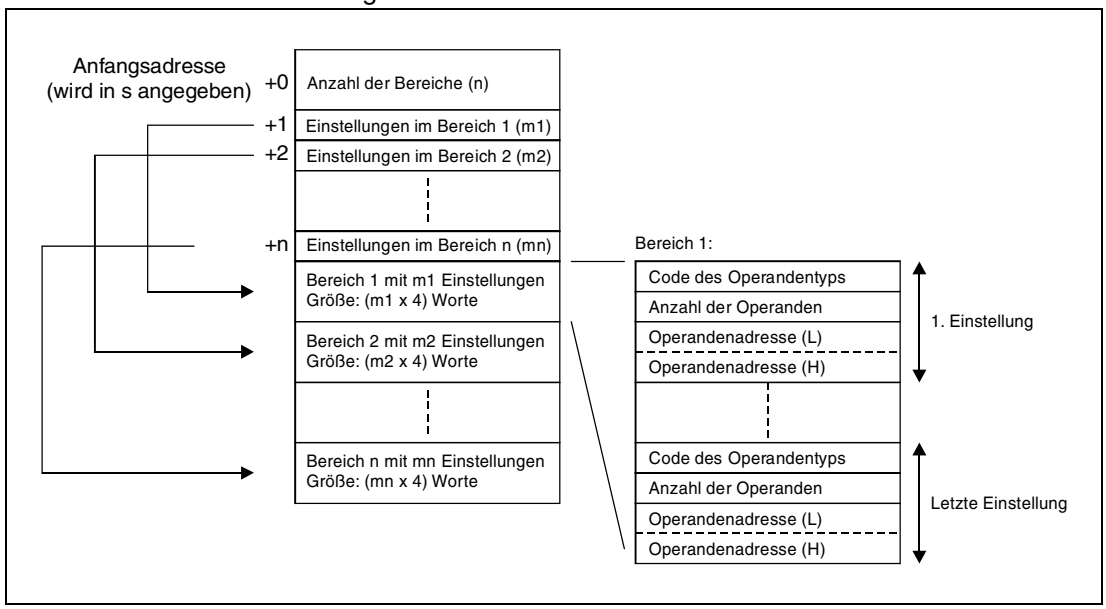


In jedem System kann diese Anweisung nur einmal programmiert werden. Verwenden Sie die Anweisung auch bei mehreren vorhandenen Programmen nur in einem Programm. Wird die Anweisung mehrfach programmiert, kann die korrekte Ausführung nicht garantiert werden.

Die Q4ARCPU liest den Inhalt des Parameterblocks beim Einschalten der Versorgungsspannung oder bei einem Reset. (Beachten Sie, das dass System neu gestartet werden muss, wenn der Inhalt des Parameterblocks verändert wurde.) Der Parameterblock ist in mehrere Bereiche unterteilt. Die Übertragung der Bereiche wird durch die Sondermerker SM1520 bis SM1583 gesteuert. Mit SM1520 wird der erste Bereich übertragen, mit SM1521 der zweite usw. Die S.TRUCK-Anweisung muss ausgeführt werden, nachdem die jeweiligen Sondermerker gesetzt wurden.

HINWEIS *Zur Übertragung der Daten mit der S.TRUCK-Anweisung können dieselben Sondermerker (SM1520 bis SM1583) verwendet werden, wie zur Auffrischung des Pufferspeichers mit der S.PREF-Anweisung.*

Der Parameterblock hat die folgende Struktur:



Inhalt des Parameterblocks:

– Anzahl der Bereiche (n)
Innerhalb des Parameterblocks können mehrere Bereiche existieren, in denen z. B. die Anfangsadressen und die Anzahl der Operanden eingetragen sind, die übertragen werden sollen. Geben Sie hier die Anzahl der Bereiche an.

– Anzahl der Einstellungen pro Bereich (m1 bis mn)
Jeder Bereich kann mehrere Einstellungen zum Datentransfer enthalten. Jede Einstellung besteht aus dem Operandentyp, der Anzahl der Operanden und der Anfangsadresse.

– Bereiche mit Einstellungen
Jede Einstellung zum Datenaustausch belegt 4 Worte:
1. Wort: Codierte Angabe des Operandentyps (siehe folgende Tabelle)

Operand	Code	Operand	Code	Operand	Code	Operand	Code
X	0	B	5	C[1]	10	Z	15
Y	1	F	6	D	11	SB	16
M	2	V	7	W	12	SW	17
L	3	ST	8	R	13	SM	18
S	4	T[1]	9	ZR	14	SD	19

[1] Bei Timern (T) und Zählern (C) ist der Kontakt, die Spule und der aktuelle Wert enthalten.

HINWEIS *Operanden, die als lokale Operanden deklariert wurden, werden nicht transferiert.*

2. Wort: Anzahl der Operanden
Die Anzahl der Operanden kann dezimal oder hexadezimal angegeben werden. Bei Bit-Operanden muss die angegebene Anzahl durch 16 teilbar sein.

3. und 4. Wort: Anfangsadresse, niederwertiges (L) und höherwertiges Byte (H)
Die Anfangsadresse kann dezimal oder hexadezimal angegeben werden. Bei Bit-Operanden muss die angegebene Adresse entweder 0 oder eine durch 16 teilbare Zahl sein (0, 16, 32, ...).

HINWEISE *Beachten Sie bitte die folgenden Einschränkungen:*

– Innerhalb eines Parameterblocks können maximal 64 Bereiche angegeben werden (n <= 64).

– Die Summe der Einstellungen darf 2048 nicht überschreiten (m1+m2+...mn <= 2048).

– Wird als Anzahl der Einstellungen eines Bereiches (m1 bis mn) eine „0“ eingetragen, wird die Anzahl der Bereiche ebenfalls auf „0“ gesetzt. Dadurch wird der Bereich übersprungen.

– Am Ende eines Programmzyklus können max. 48 k Worte übertragen werden. Wird diese Datenmenge überschritten, tritt ein Fehler auf und der Datentransfer wird nicht ausgeführt.

– Bei der Angabe von Bit-Operanden muss die Anzahl der Operanden und die Anfangsadresse ein Vielfaches von 16 sein.

– Bei der Angabe von Timern oder Zählern wird die tatsächlich übertragene Anzahl der Operanden nach der folgenden Formel berechnet:

Übertragene Anzahl der Operanden = Eingestellte Anzahl der Operanden × (1 + 1/8)

Die „1“ in der Klammer steht für den Istwert des Timers/Zählers (Wort), der Wert „1/8“ für das Bit des Kontakts oder der Spule.

Steuerung der Ausführung der TRUCK-Anweisung
Mit dem Sondermerker SM1518 kann die Ausführung der TRUCK-Anweisung beeinflusst werden. Die Auswahl ist ab der END-Bearbeitung des Programmzyklus gültig, in dem SM1518 gesetzt oder zurückgesetzt wird.

a) Übertragung mit Wartezeit (SM1518 = 0)

Falls das Reservesystem zu dem Zeitpunkt auf den Speicherbereich zugreift, in dem die Daten

transferiert werden sollen, wartet die aktive CPU, bis das Reservesystem fertig ist. Diese Wartezeit der aktiven CPU verlängert die Zykluszeit.

b) Wiederholung der Übertragung (SM1518 =1)

Falls das Reservesystem zu dem Zeitpunkt auf den Speicherbereich zugreift, in dem die Daten transferiert werden sollen, führt die aktive CPU die folgende END-Bearbeitung aus, ohne die Daten zu transferieren. Während der Wiederholung des Datentransfers werden weitere Anforderungen zur Datenübertragung ignoriert. Die Zykluszeit der aktiven CPU wird nicht verlängert.

Anzeige der Ausführung der TRUCK-Anweisung
Nach der Übertragung wird für jeden übertragenen Bereich ein Sondermerker (SM1712 bis SM1775) für die Dauer eines Zyklus gesetzt. (Bereich 1: SM1712, Bereich 2: SM1713 .... Bereich 64: SM1775)

Fehlerquellen

In den folgenden Fällen tritt ein Verarbeitungsfehler auf, das Error Flag SM0 wird gesetzt und im Sonderregister SD0 wird ein Fehlercode eingetragen:

- Die Datei mit dem File-Register existiert nicht, obwohl das File-Register R im Parameterblock angegeben wurde. (Fehlercode: 2402)
- Wenn ein Wert angegeben wurde, der außerhalb des zulässigen Bereichs liegt. (Fehlercode: 4104)
- Die Menge der zu übertragenen Daten überschreitet 48 kWorte. (Fehlercode: 4104)

Beispiel TRUCK

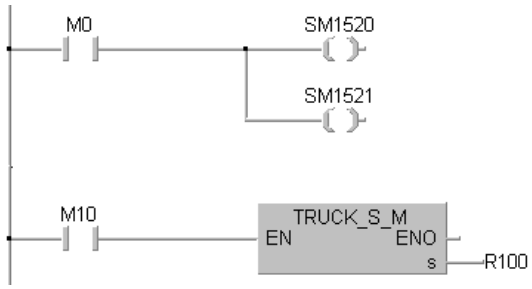
Die Zustände der Merker M0 bis M95 und M320 bis M639 werden ebenso wie die Inhalte der Datenregister D0 bis D29 und D600 bis D699 in die Reserve-CPU übertragen. Der Parameterblock beginnt bei R100 und enthält zwei Bereiche, die durch die Sondermerker SM1520 und SM1521 gesteuert werden: Im ersten Bereich sind die Merker und im zweiten sind die Datenregister angegeben, die übertragen werden sollen:

Parameterblock		Beschreibung	Bemerkung	
Adresse	Inhalt			
R100	2	Anzahl der Bereiche	—	
R101	2	Einstellungen in Bereich 1	—	
R102	2	Einstellungen in Bereich 2	—	
R103	2	Operandencode (2 = M)	Bereich 1	Einstellung 1
R104	96	Anzahl der Operanden		
R105	0	Anfangsadresse der Operanden (M0)		
R106	0			
R107	2	Operandencode (2 = M)		Einstellung 2
R108	320	Anzahl der Operanden		
R109	320	Anfangsadresse der Operanden (M320)		
R110	0			
R111	11	Operandencode (11 = R)	Bereich 2	Einstellung 1
R112	30	Anzahl der Operanden		
R113	0	Anfangsadresse der Operanden (D0)		
R114	0			
R115	11	Operandencode (11 = R)		Einstellung 2
R116	100	Anzahl der Operanden		
R117	600	Anfangsadresse der Operanden (D600)		
R118	0			

MELSEC-Anweisungsliste

```
MELSEC   LD    M0
         OUT   SM1520
         OUT   SM1521
MELSEC   LD    M10
         S.TRUCK R100
```

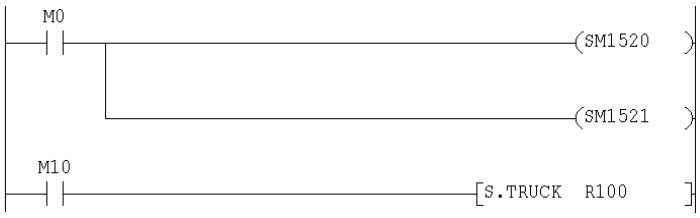
Kontaktplan (GX IEC Developer)



IEC-Anweisungsliste

```
         LD    M0
         ST    SM1520
         ST    SM1521
         LD    M10
         TRUCK_S_M R100
```

Kontaktplan (GX Developer)



10.2.2 SPREF

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●[1]	

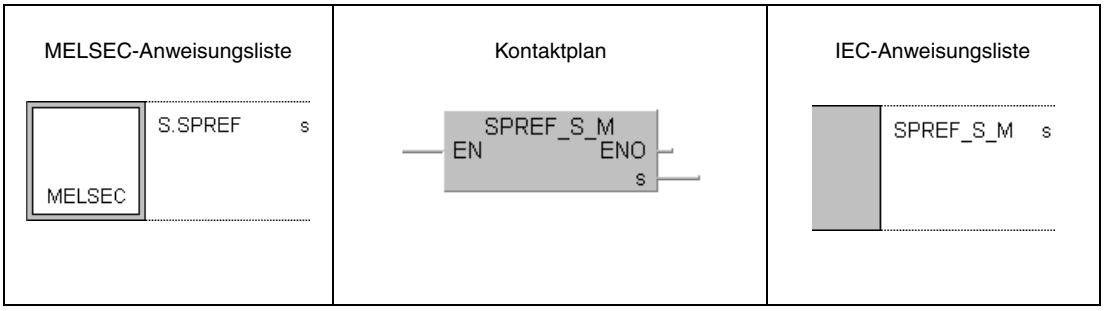
[1] Nur für Q4AR

Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	—	●[1]	●	—	—	—	—	—	—	—	—

[1] Es sind nur Operanden aus dem Latch-Bereich zulässig.

GX IEC Developer



GX Developer

Variablen

Operand	Befehlswert	Datentyp
s	Anfangsadresse des Parameterblocks	BIN-16-Bit

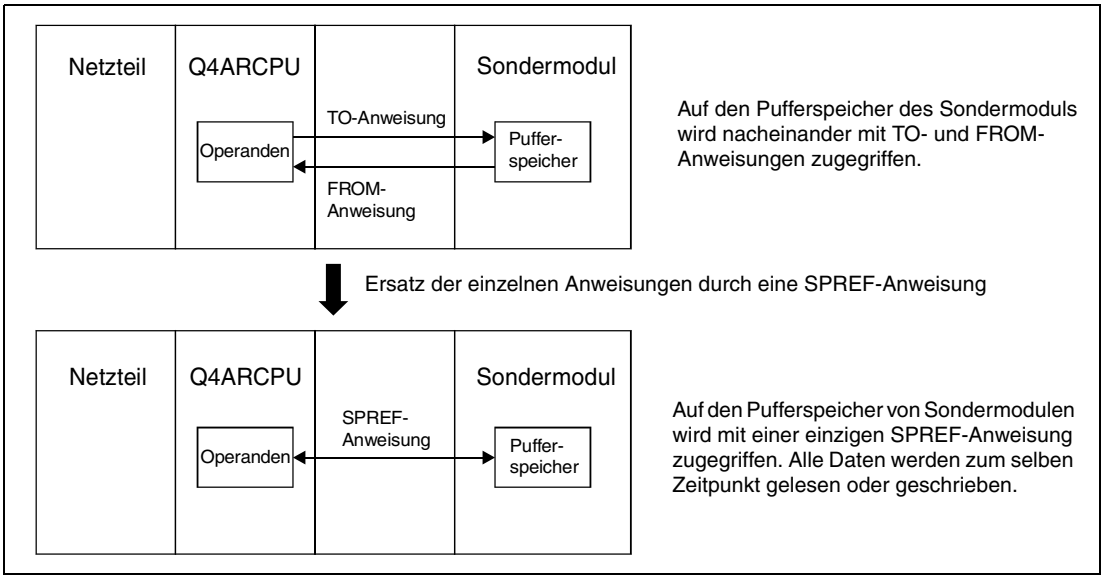
Funktionsweise

Datenaustausch zwischen CPU und den Pufferspeichern von Sondermodulen

S.SPREF Datenaustausch mit Pufferspeichern

Mit der SPREF-Anweisung werden Daten zwischen einer Q4ARCPU und dem Pufferspeicher von einen oder mehreren Sondermodulen ausgetauscht.

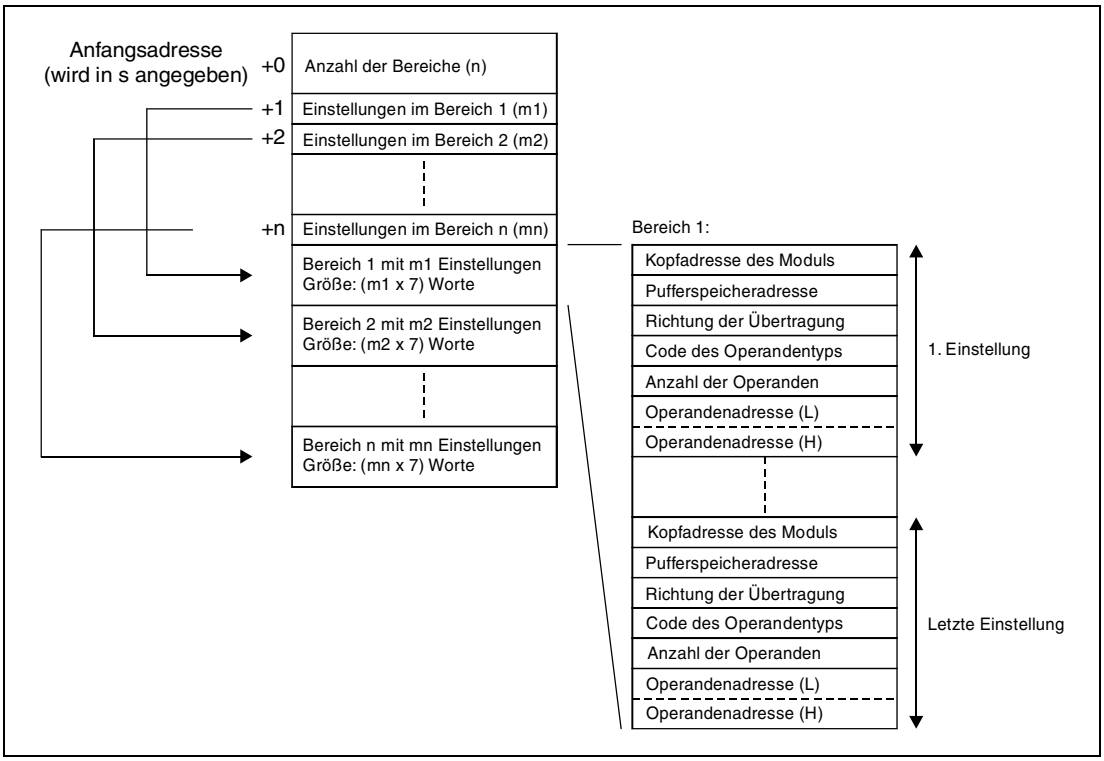
HINWEIS *Die S.PREF-Anweisung kann nicht für Sondermodule in dezentralen E/A-Stationen des MELSECNET (II), /B, oder /10 und des MELSECNET/MINI-S3 verwendet werden.*



Die Variable s enthält die Anfangsadresse eines Parameterblocks mit Einstellungen zum Datenaustausch. Der Inhalt des Parameterblocks muss vor der Ausführung der SPREF-Anweisung festgelegt werden. Der Parameterblock ist in mehrere Bereiche unterteilt. Die Übertragung der Bereiche wird durch die Sondermerker SM1520 bis SM1583 gesteuert. Mit SM1520 wird der erste Bereich übertragen, mit SM1521 der zweite usw. Die jeweiligen Sondermerker müssen vor der Ausführung der SPREF-Anweisung gesetzt werden.

HINWEIS *Zur Übertragung der Daten mit der SPREF-Anweisung können dieselben Sondermerker (SM1520 bis SM1583) verwendet werden, wie für die TRUCK-Anweisung.*

Der Parameterblock hat die folgende Struktur:



Inhalt des Parameterblocks:

– Anzahl der Bereiche (n)
Innerhalb des Parameterblocks können mehrere Bereiche existieren, in denen z. B. die Anfangsadresse des Sondermoduls und die Anzahl der Operanden eingetragen ist, die übertragen werden sollen. Geben Sie hier die Anzahl dieser Bereiche an.

– Anzahl der Einstellungen pro Bereich (m1 bis mn)
Jeder Bereich kann mehrere Einstellungen zum Datenaustausch enthalten. Jede Einstellung besteht aus den nachfolgend aufgeführten Elementen.

– Bereiche mit Einstellungen
Jede Einstellung belegt 7 Worte eines Bereiches und enthält die Parameter für eine Datenübertragung.

1. Wort: Kopfadresse des Sondermoduls auf dem Baugruppenträger
Es werden nur die ersten beiden Stellen der 3-stelligen Adresse angegeben (Beispiel: Eine Anfangsadresse von X/Y100 wird als 10H eingetragen).

2. Wort: Pufferspeicheradresse
Die Angabe der Pufferspeicheradresse kann dezimal oder hexadezimal erfolgen.

3. Wort: Datenrichtung
Festlegung, ob die Daten aus dem Sondermodul gelesen oder in das Sondermodul geschrieben werden.
0 = Lesen (vom Pufferspeicher in die CPU), 1 = Schreiben (von der CPU in das Sondermodul)

4. Wort: Codierte Angabe des Operandentyps (siehe folgende Tabelle)

Operand	Code	Operand	Code	Operand	Code	Operand	Code
X	0	B	5	C[1]	10	Z	15
Y	1	F	6	D	11	SB	16
M	2	–	–	W	12	SW	17
L	3	ST	8	R	13	SM	18
–	–	T[1]	9	ZR	14	SD	19

[1] Bei Timern (T) und Zählern (C) wird nur der aktuelle Wert übertragen.

5. Wort: Anzahl der Operanden
Die Anzahl der Operanden kann dezimal oder hexadezimal angegeben werden. Bei Bit-Operanden muss die angegebene Anzahl durch 16 teilbar sein.

6. und 7. Wort: Anfangsadresse der Operanden, niederwertiges (L), höherwertiges Byte (H)
Die Anfangsadresse der Operanden belegt zwei Worte und kann dezimal oder hexadezimal angegeben werden. Bei Bit-Operanden muss die angegebene Adresse entweder 0 oder eine durch 16 teilbare Zahl sein (0, 16, 32, ...).

HINWEISE *Beachten Sie bitte die folgenden Einschränkungen:*

- Innerhalb eines Parameterblocks können maximal 64 Bereiche angegeben werden (n <= 64).

- Die Summe der Einstellungen darf 2048 nicht überschreiten (m1+m2+...mn <= 2048).

- Wird als Anzahl der Einstellungen eines Bereiches (m1 bis mn) eine „0“ eingetragen, wird die Anzahl der Bereiche ebenfalls auf „0“ gesetzt. Dadurch wird der Bereich übersprungen.

Fehlerquellen

Im folgenden Fall tritt ein Verarbeitungsfehler auf, das Error Flag SM0 wird gesetzt und im Sonderregister SD0 wird ein Fehlercode eingetragen:

- Wenn ein Wert angegeben wurde, der außerhalb des zulässigen Bereichs liegt. (Fehlercode: 4104)

Beispiel SPREF

Im folgenden Beispiel werden zwei Sondermodule angesprochen. Für jedes Sondermodul ist im Parameterblock, der ab dem File-Register R100 gespeichert ist, ein Bereich eingerichtet:

- 1. Bereich: Zugriff auf das Sondermodul mit der Kopfadresse X/Y20

  Dieser Bereich wird übertragen, wenn SM1520 gesetzt ist.

  Der Inhalt der Pufferspeicheradressen 0 bis 3 wird aus dem Sondermodul in die File-Register R0 bis R3 übertragen (gelesen).

  Der Inhalt der File-Register R10 und R11 wird in die Pufferspeicheradressen 10 und 11 des Sondermoduls übertragen (geschrieben).

- 2. Bereich: Zugriff auf das Sondermodul mit der Kopfadresse X/Y100

  Dieser Bereich wird übertragen, wenn SM1521 gesetzt ist.

  Der Inhalt der Pufferspeicheradressen 110 bis 119 wird aus dem Sondermodul in die Daten-Register D110 bis D113 übertragen (gelesen)

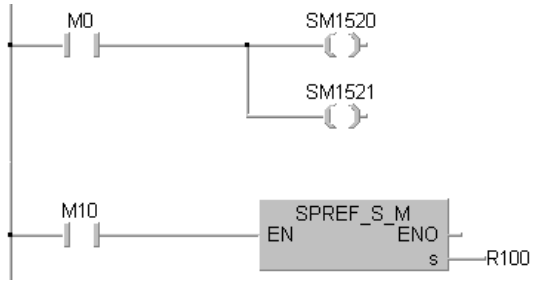
Der Parameterblock für dieses Beispielprogramm enthält die folgenden Konstanten:

Parameterblock		Beschreibung	Bemerkung	
Adresse	Inhalt			
R100	2	Anzahl der Bereiche	—	
R101	2	Einstellungen in Bereich 1	—	
R102	1	Einstellungen in Bereich 2	—	
R103	2	Kopfadresse des Sondermoduls (X/Y20)	Bereich 1	Einstellung 1
R104	0	Startadresse im Pufferspeicher		
R105	0	Datenrichtung (0 = Lesen)		
R106	13	Operandencode (13 = R)		
R107	4	Anzahl der Operanden		
R108	0	Anfangsadresse der Operanden (R0)		
R109	0			
R110	2	Kopfadresse des Sondermoduls (X/Y20)		Einstellung 2
R111	10	Startadresse im Pufferspeicher		
R112	1	Datenrichtung (1 = Schreiben)		
R113	13	Operandencode (13 = R)		
R114	2	Anzahl der Operanden		
R115	10	Anfangsadresse der Operanden (R10)		
R116	0			
R117	10	Kopfadresse des Sondermoduls (X/Y100)	Bereich 2	Einstellung 1
R118	110	Startadresse im Pufferspeicher		
R119	0	Datenrichtung (0 = Lesen)		
R120	11	Operandencode (11 = D)		
R121	10	Anzahl der Operanden		
R122	110	Anfangsadresse der Operanden (D110)		
R123	0			

MELSEC-Anweisungsliste

```
MELSEC   LD    M0
         OUT   SM1520
         OUT   SM1521
MELSEC   LD    M10
         S.SPREF R100
```

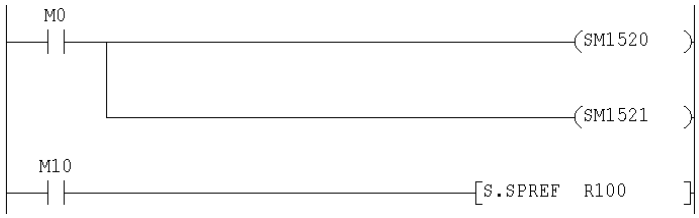
Kontaktplan (GX IEC Developer)



IEC-Anweisungsliste

```
         LD       M0
         ST       SM1520
         ST       SM1521
         LD       M10
         SPREF_S_M R100
```

Kontaktplan (GX Developer)



11 Anweisungen für Sondermodule

Einteilung	Beschreibung
Anweisungen für serielle Schnittstellenmodule	Empfangene Daten in einem Interrupt-Programm in die SPS-CPU eintragen, Anwenderdefinierte Datenrahmen lesen, einstellen oder löschen, Daten mittels anwenderdefinierter Datenrahmen versenden
Anweisungen für PROFIBUS/DP-Module	Daten aus dem Pufferspeicher von PROFIBUS/DP-Modulen lesen oder Daten dort eintragen
Anweisungen für ETHERNET-Module	Daten in feste Puffer eintragen oder aus festen Puffern lesen, Verbindungen öffnen und schließen, Fehlerspeicher lesen oder löschen, ETHERNET-Modul neu initialisieren
Anweisungen für MELSECNET/10	Stationen für paarweisen Betrieb (Duplexbetrieb) festlegen
Anweisungen für CC-Link	Übertragung der Netzwerkparameter, Parameter für automatische Aktualisierung einstellen, Daten aus dem Pufferspeicher einer anderen CC-Link-Station oder der SPS-CPU dieser Station lesen, Daten in den Pufferspeicher einer anderen CC-Link-Station oder der SPS-CPU dieser Station schreiben, Daten in den automatisch aktualisierten Speicher eintragen oder Daten aus diesem Speicher lesen.

11.1 Anweisungen für serielle Schnittstellenmodule

Funktion	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor
Empfangene Daten in einem Interrupt-Programm vom QJ71C24 in die SPS-CPU übertragen	Z.BUFRCVS	BUFRCVS_M
Anwenderdefinierten Datenrahmen aus Schnittstellenmodul lesen	G.GETE	GETE_M
	GP.GETE	GETEP_M
Anwenderdefinierten Datenrahmen in Schnittstellenmodul eintragen oder löschen	G.PUTE	PUTE_M
	GP.PUTE	PUTEP_M
Daten mittels anwenderdefinierten Datenrahmen versenden	G.PRR	PRR_M
	GP.PRR	PRRP_M

11.1.1 BUFRCVS

CPU

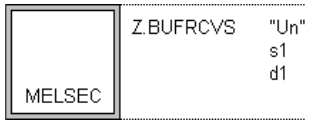
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					●

Operanden MELSEC Q

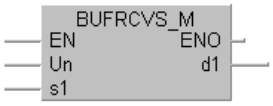
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□N□		Sondermodule U□NG□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	—	●	●	—	—	—	—	—	—	SM0	
d1	●	●	●	—	—	—	—	—	—		

GX IEC Developer

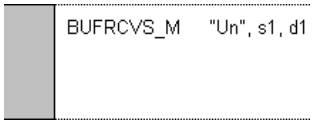
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

Operand	Bedeutung			Werte-bereich	Festlegung des Inhalts durch	Datentyp
„Un“	Kopfadresse des Schnittstellenmoduls auf dem Baugruppenträger (Es werden nur die ersten beiden Stellen der 3-stelligen Adresse angegeben, z. B. wird die Kopfadresse X/Y100 als „U10“ eingetragen.)			0 bis FE_H	Anwender	BIN-16-Bit
s1	Angabe der Schnittstelle, über die Daten empfangen werden 1: Schnittstelle 1 (CH1) 2: Schnittstelle 2 (CH2)			1 oder 2		
d1	Erster Operand des Bereiches, in dem die empfangenen Daten gespeichert werden.					Adresse
	Operand	**Bedeutung**	**Beschreibung**	**Werte-bereich**	**Festlegung durch**	
	(d1)+0	Datenlänge	Anzahl der Daten, die empfangen wurden Die Einheit (Bytes oder Worte) hängt von der Parametrierung ab.	—	System	
	(d1)+1 bis (d1)+n	Empfangene Daten	In diesem Bereich werden die aus dem Empfangsbereich des Pufferspeichers gelesenen Daten nacheinander in aufsteigender Reihenfolge eingetragen.			

Funktionsweise

Empfangene Daten aus QJ71C24 lesen

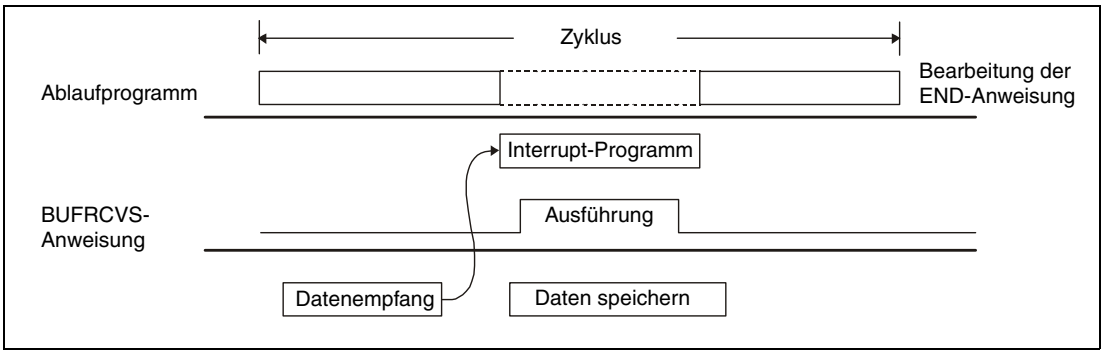
BUFRCVS Daten lesen

Mit der BUFRCVS-Anweisung werden Daten, die von einem externen Gerät an ein Schnittstellenmodul QJ71C24 gesendet wurden, aus dem Modul gelesen und in der SPS-CPU gespeichert.

Die BUFRCVS-Anweisung erkennt dabei selbständig, wo die Daten im Pufferspeicher des QJ71C24 abgelegt sind und überträgt sie in dem mit d1 angegebnene Operandenbereich.

Nach der Datenübertragung wird automatisch das Anforderungssignal zum Lesen der empfangenen Daten (X3/XA) oder das Eingangssignal „Fehler beim Empfang von Daten“ (X4/XB) zurückgesetzt. Das Ausgangssignal, das dem Schnittstellenmodul anzeigt, dass die Daten gelesen wurden (Y1/Y8), muss bei Verwendung der BUFRCVS-Anweisung nicht gesetzt werden.

Die BUFRCVS-Anweisung wird in einem Interrupt-Programm verwendet und wird in einem Zyklus abgeschlossen. Die folgende Abbildung zeigt den Signalverlauf bei Ausführung der BUFRCVS-Anweisung:



HINWEISE

Falls empfangene Daten mit einer BUFRCVS-Anweisung innerhalb eines Interrupt-Programms gelesen werden, können die Daten derselben Schnittstelle nicht mehr im Hauptprogramm gelesen werden. Dadurch kann die BUFRCVS-Anweisung nicht zusammen mit den folgenden Anweisungen verwendet werden:

- der INPUT-Anweisung

- der BIDIN-Anweisung

- der FROM-Anweisung in Verbindung mit den Ein- und Ausgangssignalen des Schnittstellenmoduls

Eine BUFRCVS- und eine CSET-Anweisung können nicht gleichzeitig ausgeführt werden.

Der in d1 angegebene Operandenbereich in der SPS-CPU, in dem die Daten gespeichert werden, muss so groß sein, dass er alle Daten aufnehmen kann, die vom Partnergerät gesendet werden. Ist der Bereich zu klein, gehen die Daten, die nicht gespeichert werden können, verloren.

Fehlerquellen

Wenn bei der Ausführung der BUFRCVS-Anweisung ein Fehler aufgetreten ist, wird das Error Flag SM0 gesetzt und in SD0 ein Fehlercode eingetragen. Detaillierte Angaben zu den Fehlercodes entnehmen Sie bitte den folgenden Anleitungen:

- Bei einem Fehlercode bis $4FFF_H$ finden Sie Hinweise zur Fehlerbehebung in diesem Handbuch (Kapitel 13).
- Bei einem Fehlercode ab 7000_H finden Sie detaillierte Angaben in der Bedienungsanleitung zum Schnittstellenmodul QJ71C24.

Bei einem fehlerhaften Datenempfang (angezeigt durch die Eingangssignale X4 bzw. XB) kann der Fehlercode, der in diesem Fall in den Pufferspeicheradressen 258_H und 268_H des Schnittstellenmoduls eingetragen wird, ausgewertet werden.

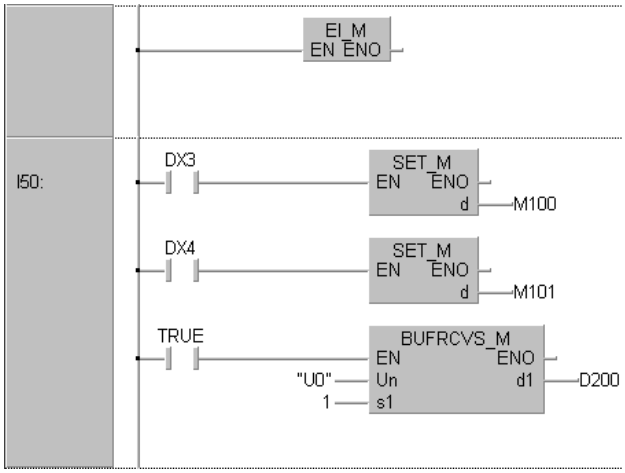
Beispiel BUFRCVS

Das folgende Programm liest die über die Schnittstelle 1 eines QJ71C24 (Kopfadresse X/Y0) empfangenen Daten und speichert sie ab D200. Das Modul ist so parametriert, dass nur die Schnittstelle 1 einen Interrupt auslöst. Beim Empfang von Daten wird das Interrupt-Programm 50 (I50) bearbeitet. Als Schnittstelle zum Hauptprogramm dienen die Merker M100 und M101. Wurden Daten fehlerfrei empfangen, wird das dem Hauptprogramm mit M100 angezeigt, während M101 bei fehlerhaften Datenempfang gesetzt wird. Im Hauptprogramm werden diese beiden Merker nach der Auswertung zurückgesetzt.

MELSEC-Anweisungsliste

```
        EI
        FEND
MELSEC
I50     LD      DX3
        SET     M100
MELSEC  LD      DX4
        SET     M101
        LD      M400
        Z.BUFRCVS "U0"
                K1
                D200
        IRET
```

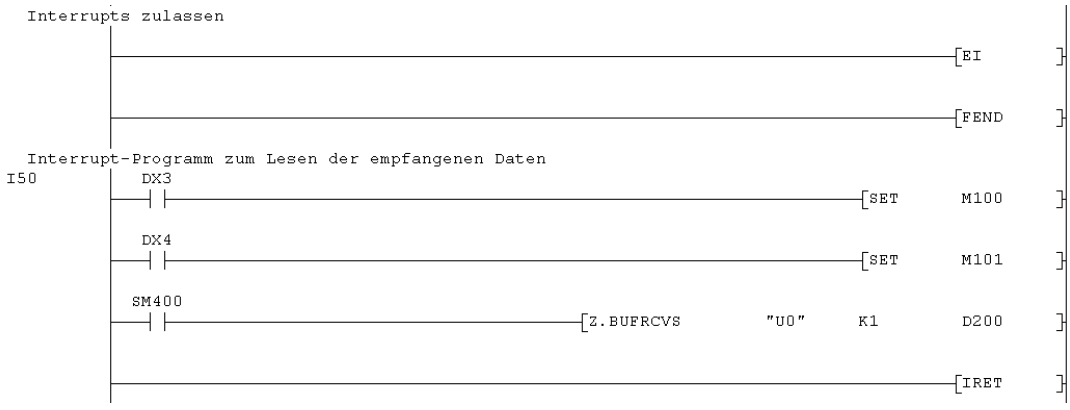
Kontaktplan (GX IEC Developer)



IEC-Anweisungsliste

```
        LD      TRUE
        EI_M
I50:    LD      DX3
        SET_M   M100
        LD      DX4
        SET_M   M101
        LD      TRUE
        BUFRCVS_M "U0",  1,  D200
```

Kontaktplan (GX Developer)



11.1.2 GETE, GETEP

CPU

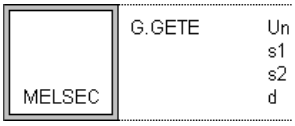
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Operanden MELSEC Q

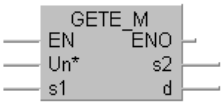
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□N□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	—	●	●	—	—	—	—	—	—	SM0	
s2	—	●	●	—	—	—	—	—	—		
d	●	●	●	—	—	—	—	—	—		

GX IEC Developer

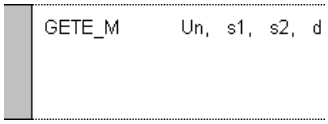
MELSEC-Anweisungsliste



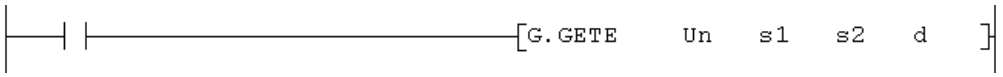
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

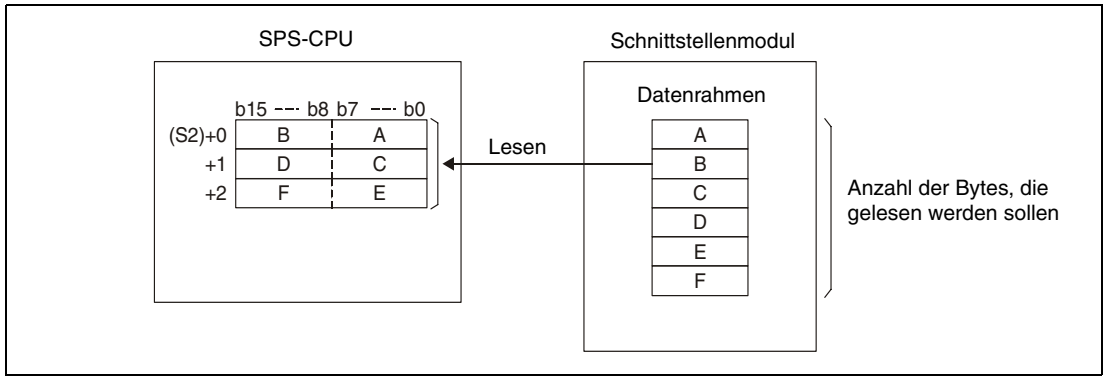
<table>
<tr><th>Operand</th><th colspan="4">Bedeutung</th><th>Werte-bereich</th><th>Festlegung des Inhalts durch</th><th>Datentyp</th></tr>
<tr><td>Un</td><td colspan="4">Kopfadresse des Schnittstellenmoduls auf dem Baugruppenträger (Es werden nur die ersten beiden Stellen der 3-stelligen Adresse angegeben, z. B. wird die Kopfadresse X/Y100 als „10“ eingetragen.)</td><td>0 bis FE_H</td><td>Anwender</td><td>BIN-16-Bit</td></tr>
<tr><td rowspan="6">s1</td><td colspan="5">Erster Operand des Bereiches mit Informationen zur Ausführung der Anweisung</td><td rowspan="6">BIN-16-Bit</td></tr>
<tr><td>Operand</td><td>Bedeutung</td><td colspan="2">Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(s1)+0</td><td>Dummy</td><td colspan="2">Wird vom System verwendet</td><td>0</td><td>—</td></tr>
<tr><td>(s1)+1</td><td>Ausführungs-status der Anweisung</td><td colspan="2">Zeigt an, ob bei der Bearbeitung der Anweisung ein Fehler aufgetreten ist:
0000_H: Fehlerfreie Bearbeitung
Jeder andere Wert als 0000_H: Bei der Bearbeitung ist ein Fehler aufgetreten. Der eingetragene Wert ist ein Fehlercode.</td><td></td><td>System</td></tr>
<tr><td>(s1)+2</td><td>Angesprochener Datenrahmen</td><td colspan="2">Nummer des anwenderdefinierten Datenrahmens</td><td>1000 bis 1199</td><td rowspan="2">Anwender</td></tr>
<tr><td rowspan="2">(s1)+3</td><td>Anzahl der auszulesenden Bytes</td><td colspan="2">Max. Anzahl Bytes des Datenrahmens, die in s2 gespeichert werden können</td><td>1 bis 80</td></tr>
<tr><td>Anzahl der ausgelesenden Bytes</td><td colspan="2">Anzahl der Bytes des Datenrahmens, die tatsächlich ausgelesen wurden</td><td>1 bis 80</td><td>System</td></tr>
<tr><td>s2</td><td colspan="5">Erster Operand des Bereiches, in dem die gelesenen Daten gespeichert werden</td><td>Anwender
System</td><td>Adresse</td></tr>
<tr><td rowspan="4">d</td><td colspan="5">Bit-Operand, der nach der Ausführung der GETE-Anweisung für einen Zyklus gesetzt wird. Mit (d)+1 wird die fehlerhafte Beendigung signalisiert.</td><td rowspan="4">Bit</td></tr>
<tr><td>Operand</td><td>Bedeutung</td><td colspan="2">Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(d)+0</td><td>Anweisung ausgeführt</td><td colspan="2">Zeigt die Beendigung der GETE-Anweisung an.
EIN : Anweisung ausgeführt
AUS : Anweisung nicht ausgeführt</td><td>—</td><td rowspan="2">System</td></tr>
<tr><td>(d)+1</td><td>Anweisung mit Fehler ausgeführt</td><td colspan="2">Zeigt an, ob bei der Ausführung der GETE-Anweisung ein Fehler aufgetreten ist.
EIN: Anweisung mit Fehler ausgeführt
AUS: Anweisung ohne Fehler ausgeführt</td><td>—</td></tr>
</table>

Funktions-weise

Auslesen der anwenderdefinierten Datenrahmen

GETE Daten lesen

Mit der GETE-Anweisung werden anwenderdefinierte Datenrahmen aus einem seriellen Schnittstellenmodul gelesen und in der SPS-CPU gespeichert. Die Kopfadresse des Schnittstellenmoduls wird mit Un angegeben.

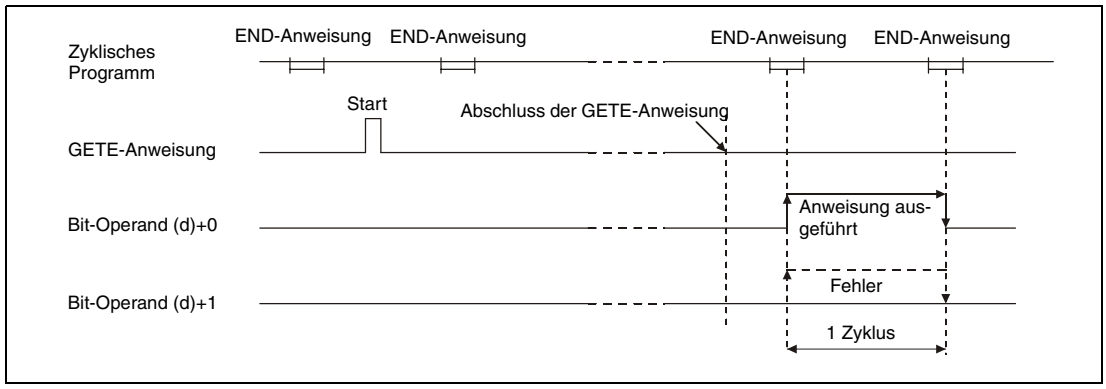


Während eine GETE-Anweisung ausgeführt wird, kann keine weitere GETE-Anweisung ausgeführt werden. Die Ausführung einer PUTE-Anweisung ist ebenfalls nicht möglich. Wird während der Bearbeitung einer GETE-Anweisung eine weitere GETE- oder eine PUTE-Anweisung gestartet, wartet das System mit der Ausführung dieser Anweisungen, bis die momentan bearbeitete GETE-Anweisung vollständig ausgeführt wurde.

Ob die Ausführung der GETE-Anweisung beendet ist, kann anhand der Bit-Operanden (d)+0 und (d)+1 überprüft werden:

- Der Bit-Operand (d)+0 wird gesetzt, wenn die END-Anweisung des Zyklus ausgeführt wird, in dem die GETE-Anweisung beendet wurde. Bei der nächsten Bearbeitung der END-Anweisung wird der mit (d)+0 angegebene Bit-Operand wieder zurückgesetzt.
- Der Bit-Operand (d)+1 zeigt einen Fehler bei der Ausführung der GETE-Anweisung an. Bei fehlerfreier Ausführung bleibt dieser Bit-Operand zurückgesetzt. Bei einem Fehler dagegen wird (d)+1 gesetzt, wenn die END-Anweisung des Zyklus ausgeführt wird, in dem die GETE-Anweisung beendet wurde. Bei der nächsten Bearbeitung der END-Anweisung wird dieser Bit-Operand wieder zurückgesetzt.

Die folgende Abbildung zeigt den zeitlichen Ablauf bei Ausführung der GETE-Anweisung:



Fehler-quellen

Wenn bei der Ausführung der GETE-Anweisung ein Fehler aufgetreten ist, wird (d)+1 gesetzt und in (s1)+1 ein Fehlercode eingetragen. Detaillierte Angaben zu den Fehlercodes entnehmen Sie bitte den folgenden Anleitungen:

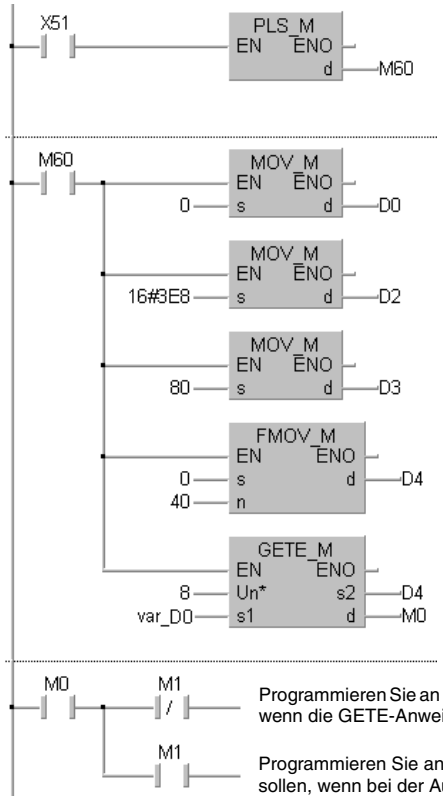
- Bei einem Fehlercode bis $4FFF_H$ finden Sie Hinweise zur Fehlerbehebung in diesem Handbuch (Kapitel 13).
- Bei einem Fehlercode ab 7000_H finden Sie detaillierte Angaben in der Bedienungsanleitung zum Schnittstellenmodul.

Beispiel GETE

Das folgende Programm liest die Daten des anwenderdefinierten Datenrahmens mit der Nummer $3E8_H$ aus einem QJ71C24 und speichert sie in der CPU des System Q ab Datenregister D4. Das Schnittstellenmodul belegt den Adressbereich von X/Y80 bis X/Y9F.

- IEC-Editoren

Kontaktplan (GX IEC Developer)



IEC-Anweisungsliste

```
LD      X51
PLS_M   M60

LD      M60
MOV_M   0,      D0
MOV_M   16#3E8, D2
MOV_M   80,     D3
FMOV_M  0,      40,     D4
GETE_M  8,      var_D0, D4,     M0

LD      M0
ANDN    M1
```

Die Operanden und Anweisungen sind oben beim Programmbeispiel für den Kontaktplan erläutert.

Ausführung der hier programmierten Anweisung, wenn die GETE-Anweisung ohne Fehler beendet wurde

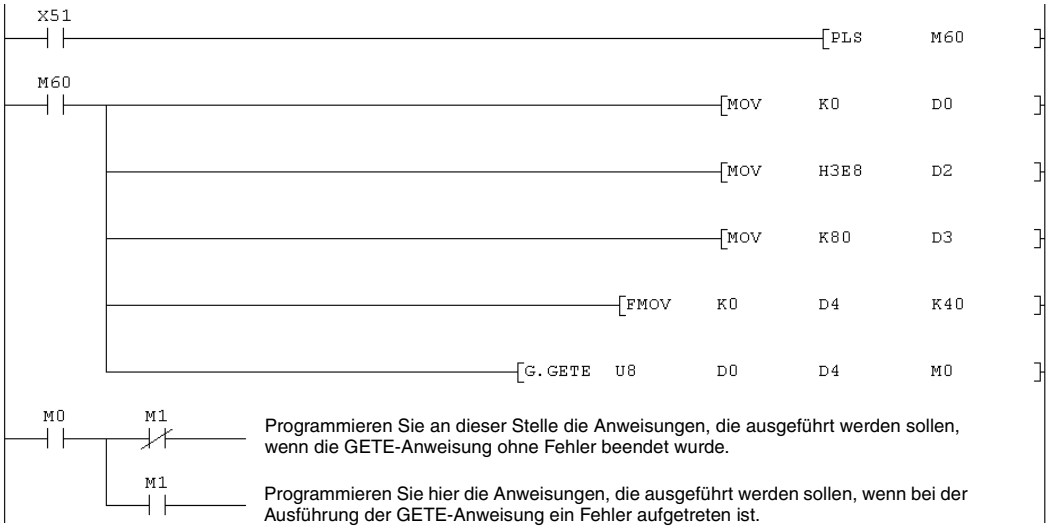
```
LD      M0
AND     M1
```

Ausführung der an dieser Stelle programmierten Anweisung, wenn bei der Bearbeitung der GETE-Anweisung ein Fehler aufgetreten ist.

HINWEIS *Bei den IEC-Editoren des GX IEC Developers müssen die Variablen im Header der Programmorganisationseinheit (POE) definiert werden. Ohne Variablendefinition werden beim Überprüfen oder Kompilieren des Programms Fehler gemeldet.*
Weitere Informationen finden Sie im Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung.

- MELSEC-Anweisungliste und Kontaktplan des GX Developers
  Die Operanden und Anweisungen sind auf der vorherigen Seite beim Programmbeispiel für den Kontaktplan des GX IEC Developers erläutert.

Kontaktplan (GX Developer)



Programmieren Sie an dieser Stelle die Anweisungen, die ausgeführt werden sollen, wenn die GETE-Anweisung ohne Fehler beendet wurde.

Programmieren Sie hier die Anweisungen, die ausgeführt werden sollen, wenn bei der Ausführung der GETE-Anweisung ein Fehler aufgetreten ist.

MELSEC-Anweisungsliste

MELSEC

```
LD      X51
PLS     M60
```

MELSEC

```
LD      M60
MOV     K0    D0
MOV     H3E8  D2
MOV     K80   D3
FMOV    K0    D4  K40
G.GETE  U8    D0  D4   M0
```

MELSEC

```
LD      M0
MPS
ANI     M1
```

Ausführung der hier programmierten Anweisung, wenn die GETE-Anweisung ohne Fehler beendet wurde

```
MPP
AND     M1
```

Programmieren Sie an dieser Stelle die Anweisungen, die ausgeführt werden sollen, wenn bei der Ausführung der GETE-Anweisung ein Fehler aufgetreten ist.

11.1.3 PUTE, PUTEP

CPU

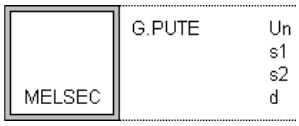
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Operanden MELSEC Q

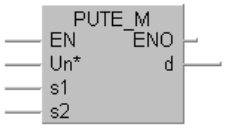
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□N□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	—	●	●	—	—	—	—	—	—	SM0	
s2	—	●	●	—	—	—	—	—	—		
d	●	●	●	—	—	—	—	—	—		

GX IEC Developer

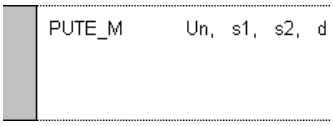
MELSEC-Anweisungsliste



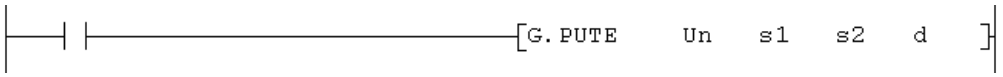
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

<table>
<tr><th>Operand</th><th colspan="4">Bedeutung</th><th>Werte-bereich</th><th>Festlegung des Inhalts durch</th><th>Datentyp</th></tr>
<tr><td>Un</td><td colspan="4">Kopfadresse des Schnittstellenmoduls auf dem Baugruppenträger (Nur die ersten beiden Stellen der 3-stelligen Adresse werden angegeben, z. B. wird die Kopfadresse X/Y100 als „10“ eingetragen.)</td><td>0 bis FE_H</td><td>Anwender</td><td>BIN-16-Bit</td></tr>
<tr><td rowspan="6">s1</td><td colspan="5">Erster Operand des Bereiches mit Informationen zur Ausführung der Anweisung</td><td rowspan="6">BIN-16-Bit</td></tr>
<tr><td>Operand</td><td>Bedeutung</td><td>Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(s1)+0</td><td>Auswahl: Eintragen oder löschen</td><td>Legen Sie hier fest, ob der mit (s1)+2 angebene Datenrahmen gelöscht oder eingestellt werden soll:
• 1: Daten eintragen
• 3: Daten löschen</td><td>1 oder 3</td><td>Anwender</td></tr>
<tr><td>(s1)+1</td><td>Ausführungs-status der Anweisung</td><td>Zeigt an, ob bei der Bearbeitung der Anweisung ein Fehler aufgetreten ist.
0000_H: Fehlerfreie Bearbeitung
Jeder andere Wert als 0000_H: Bei der Bearbeitung ist ein Fehler aufgetreten. Der eingetragene Wert ist ein Fehlercode.</td><td>—</td><td>System</td></tr>
<tr><td>(s1)+2</td><td>Angesprochener Datenrahmen</td><td>Nummer des anwenderdefinierten Datenrahmens</td><td>1000 bis 1199</td><td rowspan="2">Anwender</td></tr>
<tr><td>(s1)+3</td><td>Anzahl der einzutragenen Bytes</td><td>Angabe, wieviele Bytes in den Datenrahmen übertragen werden sollen. Geben Sie auch beim Löschen von Daten [(s1)+0 = 3] einen beliebigen Wert zwischen 1 und 80 an.</td><td>1 bis 80</td></tr>
<tr><td>s2</td><td colspan="4">Erster Operand des Bereiches, in dem die zu übertragenden Daten gespeichert sind</td><td>Anwender</td><td>Adresse</td></tr>
<tr><td rowspan="4">d</td><td colspan="5">Bit-Operand, der nach der Ausführung der PUTE-Anweisung für einen Zyklus gesetzt wird. Mit (d)+1 wird die fehlerhafte Beendigung signalisiert.</td><td rowspan="4">Bit</td></tr>
<tr><td>Operand</td><td>Bedeutung</td><td>Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(d)+0</td><td>Anweisung ausgeführt</td><td>Zeigt die Beendigung der PUTE-Anweisung an.
EIN : Anweisung ausgeführt
AUS : Anweisung nicht ausgeführt</td><td>—</td><td rowspan="2">System</td></tr>
<tr><td>(d)+1</td><td>Anweisung mit Fehler ausgeführt</td><td>Zeigt an, ob bei der Ausführung der PUTE-Anweisung ein Fehler aufgetreten ist.
EIN: Anweisung mit Fehler ausgeführt
AUS: Anweisung ohne Fehler ausgeführt</td><td>—</td></tr>
</table>

Funktionsweise

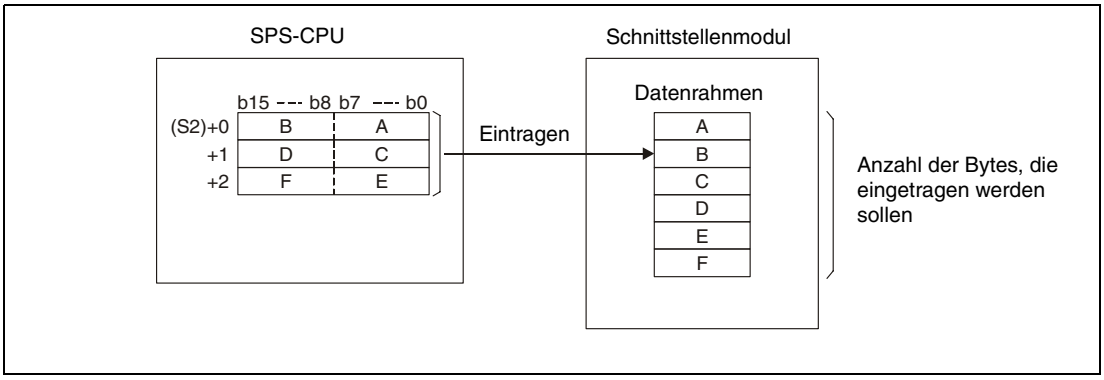
Einstellen oder Löschen anwenderdefinierter Datenrahmen

PUTE Einstell- und Löschanweisung

Mit der PUTE-Anweisung werden anwenderdefinierte Datenrahmen in ein serielles Schnittstellenmodul eingetragen oder gelöscht. Die Kopfadresse des Schnittstellenmoduls wird mit Un angegeben.

Eintragen eines anwenderdefinierten Datenrahmens

Zum Eintragen eines anwenderdefinierten Datenrahmens wird in dem mit (s1)+0 angegebenen Operanden der Wert „1“ geschrieben. Die Daten aus den mit s2 angegebenen Operandenbereich werden zum Schnittstellenmodul übertragen. Jeder dieser Operanden kann zwei Bytes aufnehmen, die Anzahl der benötigten Operanden entspricht also der halben Anzahl der Datenbytes.
Sollen z. B. sechs Bytes in einen Datenrahmen eingetragen werden, müssen ab s2 noch zwei weitere Operanden reserviert werden:



Löschen eines anwenderdefinierten Datenrahmens

Zum Löschen des in (s1)+2 angegebenen anwenderdefinierten Datenrahmens tragen Sie in dem mit (s1)+0 angegebenen Operanden den Wert „3“ ein.

Beim Löschen werden die Datenlänge [(s1)+3] und der Operandenbereich s2 nicht berücksichtigt, zur korrekten Ausführung der Anweisung müssen sie aber angegeben werden. Tragen Sie in den für (s1)+3 angegebenen Operanden einen beliebigen Wert zwischen 1 und 80 ein und wählen Sie für s2 einen Dummy-Operanden.

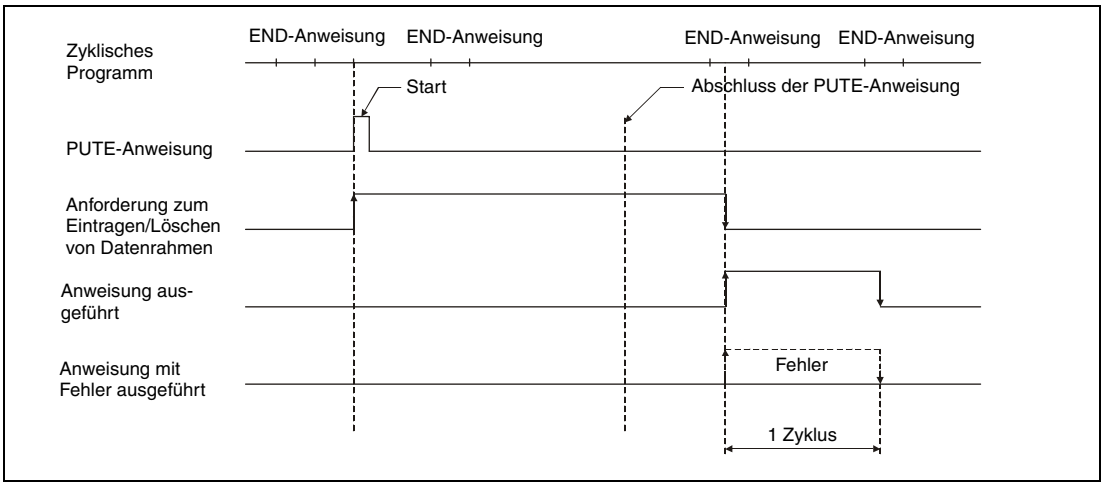
Ausführungsbedingungen

Während eine PUTE-Anweisung ausgeführt wird, kann keine weitere PUTE-Anweisung ausgeführt werden. Die Ausführung einer GETE-Anweisung ist ebenfalls nicht möglich. Wird während der Bearbeitung einer PUTE-Anweisung eine weitere PUTE- oder eine GETE-Anweisung gestartet, wartet das System mit der Ausführung dieser Anweisungen, bis die momentan bearbeitete PUTE-Anweisung vollständig ausgeführt wurde.

Ob die Ausführung der PUTE-Anweisung beendet ist, kann mit den Bit-Operanden (d)+0 und (d)+1 überprüft werden:

- Der Bit-Operand (d)+0 wird gesetzt, wenn die END-Anweisung des Zyklus ausgeführt wird, in dem die PUTE-Anweisung beendet wurde. Bei der nächsten Bearbeitung der END-Anweisung wird der mit d+0 angegebene Bit-Operand wieder zurückgesetzt.
- Der Bit-Operand (d)+1 zeigt einen Fehler bei der Ausführung der PUTE-Anweisung an. Bei fehlerfreier Ausführung bleibt dieser Bit-Operand zurückgesetzt. Bei einem Fehler dagegen wird (d)+1 gesetzt, wenn die END-Anweisung des Zyklus ausgeführt wird, in dem die PUTE-Anweisung beendet wurde. Bei der nächsten Bearbeitung der END-Anweisung wird der in (d)+1 angegebene Bit-Operand wieder zurückgesetzt.

Die folgende Abbildung zeigt den zeitlichen Ablauf bei Ausführung der PUTE-Anweisung:



Fehlerquellen

Wenn bei der Ausführung der PUTE-Anweisung ein Fehler aufgetreten ist, wird (d)+1 gesetzt und in (s1)+1 ein Fehlercode eingetragen. Detaillierte Angaben zu den Fehlercodes entnehmen Sie bitte den folgenden Anleitungen:

- Bei einem Fehlercode bis $4FFF_H$ finden Sie Hinweise zur Fehlerbehebung in diesem Handbuch (Kapitel 13).
- Bei einem Fehlercode ab 7000_H finden Sie detaillierte Angaben in der Bedienungsanleitung zum Schnittstellenmodul.

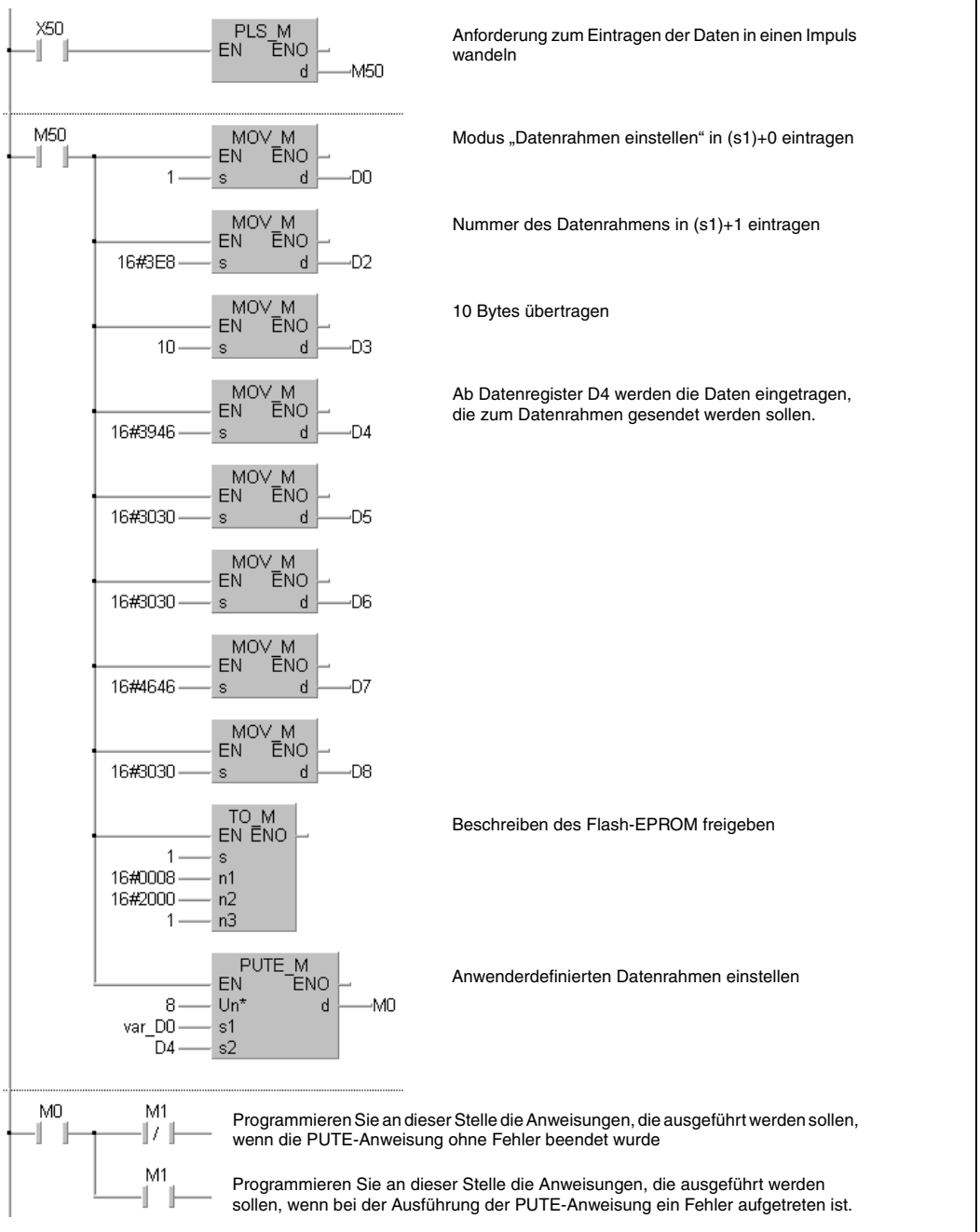
Beispiel PUTE

Das folgende Programm überträgt Einstellungen des anwenderdefinierten Datenrahmens mit der Nummer $3E8_H$. Das verwendete Schnittstellenmodul QJ71C24 belegt den Adressbereich von X/Y80 bis X/Y9F.

HINWEIS *Bei den IEC-Editoren des GX IEC Developers müssen die Variablen im Header der Programmorganisationseinheit (POE) definiert werden. Ohne Variablendefinition werden beim Überprüfen oder Kompilieren des Programms Fehler gemeldet.*
Weitere Informationen finden Sie im Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung.

- IEC-Editoren

Kontaktplan (GX IEC Developer)



IEC-Anweisungsliste

```
LD          X50
PLS_M       M50
```

```
LD          M50
MOV_M       1,          D0
MOV_M       16#3E8,     D2
MOV_M       10,         D3
MOV_M       16#3946,    D4
MOV_M       16#3030,    D5
MOV_M       16#3030,    D6
MOV_M       16#4646,    D7
MOV_M       16#3030,    D8
TO_M        1,          16#0008,  16#2000,  1
PUTE_M      8,          var_D0,   D4,       M0
```

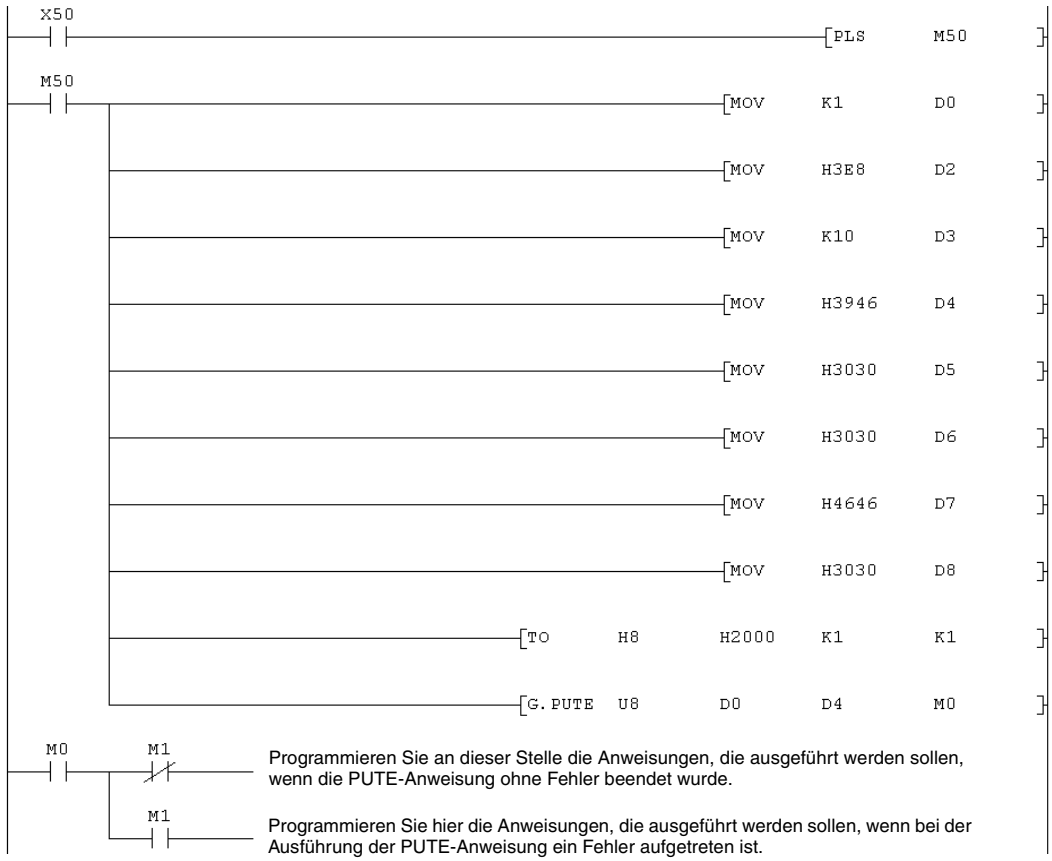
Die Operanden und Anweisungen sind beim Programmbeispiel für den Kontaktplan erläutert.

```
LD          M0
ANDN        M1
```
Ausführung der hier programmierten Anweisung, wenn die PUTE-Anweisung ohne Fehler beendet wurde

```
LD          M0
AND         M1
```
Ausführung der an dieser Stelle programmierten Anweisung, wenn bei der Bearbeitung der PUTE-Anweisung ein Fehler aufgetreten ist.

- MELSEC-Anweisungliste und Kontaktplan des GX Developers
  Die Operanden und Anweisungen sind auf der vorherigen Seite beim Programmbeispiel für den Kontaktplan des GX IEC Developers erläutert.

Kontaktplan (GX Developer)



Programmieren Sie an dieser Stelle die Anweisungen, die ausgeführt werden sollen, wenn die PUTE-Anweisung ohne Fehler beendet wurde.

Programmieren Sie hier die Anweisungen, die ausgeführt werden sollen, wenn bei der Ausführung der PUTE-Anweisung ein Fehler aufgetreten ist.

MELSEC-Anweisungsliste

```
MELSEC
        LD      X50
        PLS     M50

MELSEC
        LD      M50
        MOV     K1      D0
        MOV     H3E8    D2
        MOV     K10     D3
        MOV     H3946   D4
        MOV     H3030   D5
        MOV     H3030   D6
        MOV     H4646   D7
        MOV     H3030   D8
        TO      H8      H2000   K1      K1
        G.PUTE  U8      D0      D4      M0

MELSEC
        LD      M0
        MPS
        ANI     M1
```

Ausführung der hier programmierten Anweisung, wenn die PUTE-Anweisung ohne Fehler beendet wurde

```
        MPP
        AND     M1
```

Programmieren Sie an dieser Stelle die Anweisungen, die ausgeführt werden sollen, wenn bei der Ausführung der PUTE-Anweisung ein Fehler aufgetreten ist.

11.1.4 PRR, PRRP

CPU

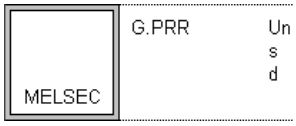
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Operanden MELSEC Q

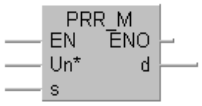
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□N□		Sonder-module U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	—	●	●	—	—	—	—	—	—	SM0	
d1	●	●	●	—	—	—	—	—	—		

GX IEC Developer

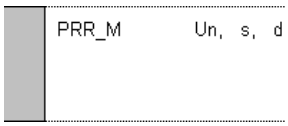
MELSEC-Anweisungsliste



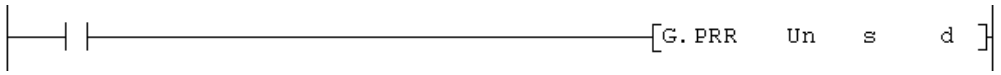
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

<table>
<tr><th>Operand</th><th colspan="4">Bedeutung</th><th>Werte-bereich</th><th>Festlegung des Inhalts durch</th><th>Datentyp</th></tr>
<tr><td>Un</td><td colspan="4">Kopfadresse des Schnittstellenmoduls auf dem Baugruppenträger (Nur die ersten beiden Stellen der 3-stelligen Adresse werden angegeben, z. B. wird die Kopfadresse X/Y100 als „10“ eingetragen.)</td><td>0 bis FE_H</td><td>Anwender</td><td>BIN-16-Bit</td></tr>
<tr><td rowspan="8">s</td><td colspan="5">Erster Operand des Bereiches mit Informationen zur Ausführung der Anweisung</td><td rowspan="8">Adresse</td></tr>
<tr><td>Operand</td><td colspan="2">Bedeutung</td><td>Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(s)+0</td><td colspan="2">Übertragungs-kanal</td><td>Angabe der Schnittstelle, über die Daten gesendet werden
1: Schnittstelle 1 (CH1)
2: Schnittstelle 2 (CH2)</td><td>1 oder 2</td><td>Anwender</td></tr>
<tr><td>(s)+1</td><td colspan="2">Ausführungs-status der Anweisung</td><td>Zeigt an, ob bei der Bearbeitung der Anweisung ein Fehler aufgetreten ist.
0000_H: Fehlerfreie Bearbeitung
Jeder andere Wert als 0000_H: Bei der Bearbeitung ist ein Fehler aufgetreten. Der eingetragene Wert ist ein Fehlercode.</td><td>—</td><td>System</td></tr>
<tr><td>(s)+2</td><td colspan="2">Anfügen von CR/LF</td><td>Festlegung, ob den gesendeten Daten die Zeichen CR/LF angefügt werden
0: CR/LF nicht anfügen
1: CR/LF anfügen</td><td>0 oder 1</td><td rowspan="3">Anwender</td></tr>
<tr><td>(s)+3</td><td colspan="2">Datenzeiger</td><td>Zeiger auf die erste Adresse des Operandenbereichs, in dem die Sende-Daten gespeichert sind.</td><td>1 bis 100</td></tr>
<tr><td>(s)+4</td><td colspan="2">Anzahl der zu übertragenen Datenrahmen</td><td>Angabe, wieviele Datenrahmen übertragen werden sollen.</td><td>1 bis 100</td></tr>
<tr><td rowspan="4">d</td><td colspan="5">Bit-Operand, der nach der Ausführung der PRR-Anweisung für einen Zyklus gesetzt wird. Mit (d)+1 wird die fehlerhafte Beendigung signalisiert.</td><td rowspan="4">Bit</td></tr>
<tr><td>Operand</td><td colspan="2">Bedeutung</td><td>Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(d)+0</td><td colspan="2">Anweisung ausgeführt</td><td>Zeigt die Beendigung der PRR-Anweisung an.
EIN : Anweisung ausgeführt.
AUS : Anweisung nicht ausgeführt</td><td>—</td><td rowspan="2">System</td></tr>
<tr><td>(d)+1</td><td colspan="2">Anweisung mit Fehler ausgeführt</td><td>Zeigt an, ob bei der Ausführung der PRR-Anweisung ein Fehler aufgetreten ist.
EIN: Anweisung mit Fehler ausgeführt
AUS: Anweisung ohne Fehler ausgeführt</td><td>—</td></tr>
</table>

Funktionsweise

Übertragung von anwenderdefinierten Datenrahmen

PRR Übertragungsanweisung

Mit einer PRR-Anweisung werden Daten unter Verwendung anwenderdefinierter Datenrahmen zu dem Schnittstellenmodul übertragen, das mit Un angegeben ist. Ab dem mit s angegebenen Operanden sind Einstellungen zur Ausführung der Anweisung gespeichert. Die Inhalte der Datenrahmen müssen vor der Ausführung der PRR-Anweisung in dem Schnittstellenmodul eingetragen werden.

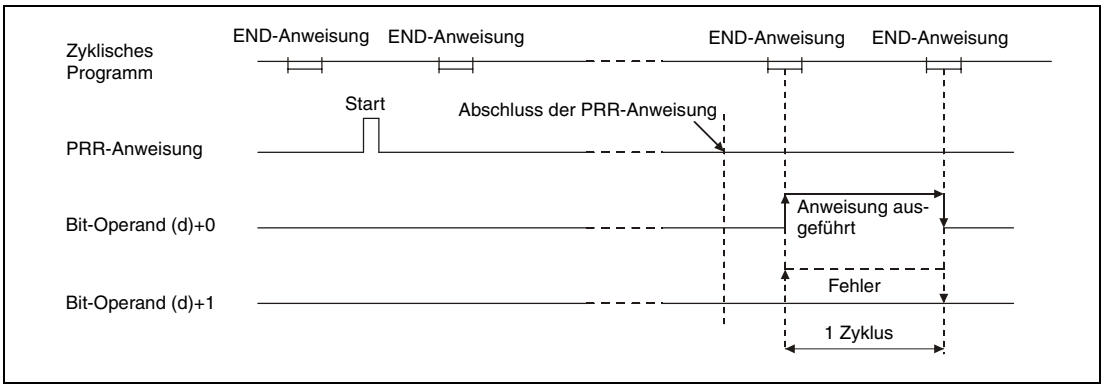
Während der Ausführung einer PRR-Anweisung können für dieselbe Schnittstelle die folgenden Anweisungen nicht ausgeführt werden:
OUTPUT, ONDEMAND, BIDOUT und weitere PRR-Anweisungen

Wird während der Bearbeitung einer PRR-Anweisung eine dieser Anweisungen gestartet, wartet das System mit deren Ausführung, bis die PRR-Anweisung vollständig ausgeführt wurde.

Ob die Ausführung der PRR-Anweisung beendet ist, kann mit den Bit-Operanden (d)+0 und (d)+1 überprüft werden:

- Der Bit-Operand (d)+0 wird gesetzt, wenn die END-Anweisung des Zyklus ausgeführt wird, in dem die PRR-Anweisung beendet wurde. Bei der nächsten Bearbeitung der END-Anweisung wird der mit (d)+0 angegebene Bit-Operand wieder zurückgesetzt.
- Der Bit-Operand (d)+1 zeigt einen Fehler bei der Ausführung der PRR-Anweisung an. Bei fehlerfreier Ausführung bleibt dieser Bit-Operand zurückgesetzt. Bei einem Fehler dagegen wird (d)+1 gesetzt, wenn die END-Anweisung des Zyklus ausgeführt wird, in dem die PRR-Anweisung beendet wurde. Bei der nächsten Bearbeitung der END-Anweisung wird dieser Bit-Operand wieder zurückgesetzt.

Die folgende Abbildung zeigt den zeitlichen Ablauf bei Ausführung der PRR-Anweisung:



Fehlerquellen

Wenn bei der Ausführung der PRR-Anweisung ein Fehler aufgetreten ist, wird (d)+1 gesetzt und in s+1 ein Fehlercode eingetragen. Detaillierte Angaben zu den Fehlercodes entnehmen Sie bitte den folgenden Anleitungen:

- Bei einem Fehlercode bis $4FFF_H$ finden Sie Hinweise zur Fehlerbehebung in diesem Handbuch (Kapitel 13).
- Bei einem Fehlercode ab 7000_H finden Sie detaillierte Angaben in der Bedienungsanleitung zum Schnittstellenmodul.

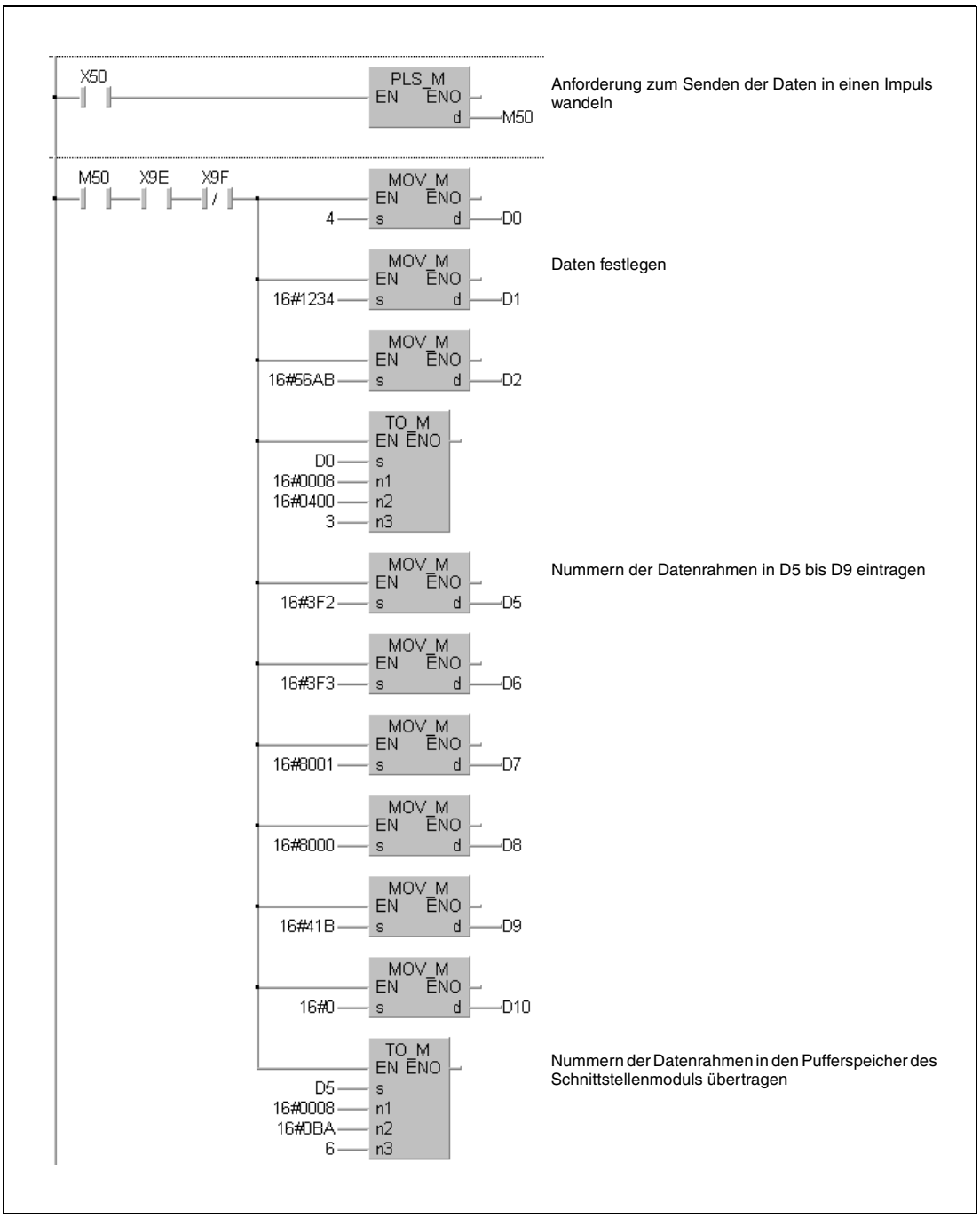
Beispiel PRR

Das Beispielprogramm überträgt Daten und die ersten fünf anwenderdefinierten Datenrahmen. Das verwendete Schnittstellenmodul QJ71C24 belegt den Adressbereich von X/Y80 bis X/Y9F. Die in der folgenden Tabelle aufgeführten Datenregister werden verwendet:

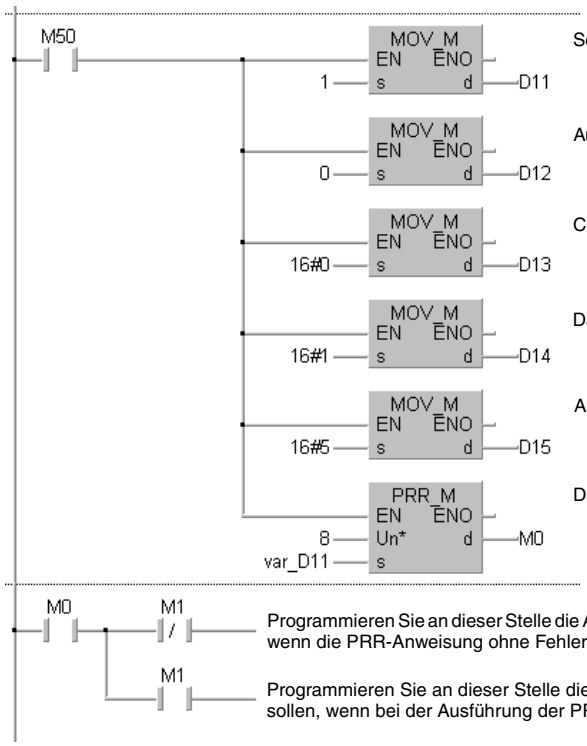
Datenregister	Inhalt	Bedeutung	
D0	0004_H	Anzahl der Bytes, die gesendet werden	
D1	3412_H	Daten, die gesendet werden	
D2	$AB56_H$		
D5	$03F2_H$	Nummern der Datenrahmen	
D6	$03F3_H$		
D7	8001_H		
D8	8000_H		
D9	$041B_H$		
D10	0000_H		
D11	0001_H	(s)+0	Schnittstelle: CH1
D12	0000_H oder Fehlercode	(s)+1	Ausführungsergebnis
D13	0000_H	(s)+2	CR/LF nicht anfügen
D14	0001_H	(s)+3	Datenzeiger
D15	0005_H	(s)+4	Anzahl der zu übertragenen Datenrahmen

HINWEIS *Bei den IEC-Editoren des GX IEC Developers müssen die Variablen im Header der Programmorganisationseinheit (POE) definiert werden. Ohne Variablendefinition werden beim Überprüfen oder Kompilieren des Programms Fehler gemeldet.*
Weitere Informationen finden Sie im Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung.

- IEC-Editoren
  Kontaktpan des GX IEC Developers (Teil 1)



Kontaktpan des GX IEC Developers (Fortsetzung)



IEC-Anweisungsliste

```
LD      X50
PLS_M   M50

LD      M50
AND     X9E
ANDN    X9F
MOV_M   4,      D0
MOV_M   16#1234, D1
MOV_M   16#56AB, D2
TO_M    D0,     16#0008, 16#0400,3
MOV_M   16#3F2,  D5
MOV_M   16#3F3,  D6
MOV_M   16#8001, D7
MOV_M   16#8000, D8
MOV_M   16#41B,  D9
MOV_M   16#0,    D10
TO_M    D5,     16#0008, 16#0BA,  6

LD      M50
MOV_M   1,              D11
MOV_M   0,              D12
MOV_M   16#0,           D13
MOV_M   16#1,           D14
MOV_M   16#5,           D15
PRR_M   8,              var_D11,      M0

LD      M0
ANDN    M1
```

Ausführung der hier programmierten Anweisung, wenn die PRR-Anweisung ohne Fehler beendet wurde

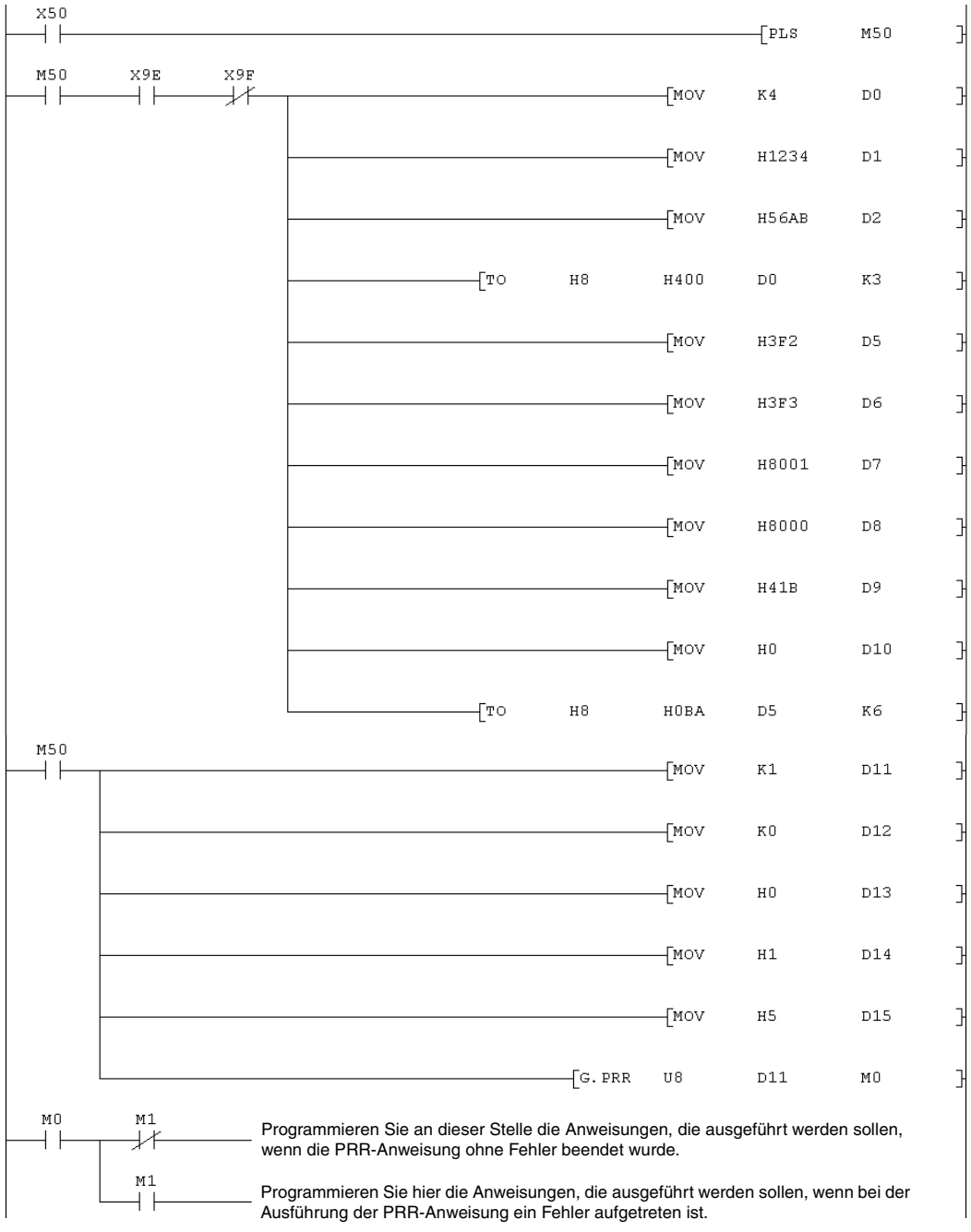
```
LD      M0
AND     M1
```

Ausführung der an dieser Stelle programmierten Anweisung, wenn bei der Bearbeitung der PRR-Anweisung ein Fehler aufgetreten ist.

Die Operanden und Anweisungen sind beim Programmbeispiel für den Kontaktplan erläutert.

- MELSEC-Anweisungliste und Kontaktplan des GX Developers
  Die Operanden und Anweisungen sind beim Programmbeispiel für den Kontaktplan des GX IEC Developers erläutert.

Kontaktplan (GX Developer)



MELSEC-Anweisungsliste

```
MELSEC
        LD      X50
        PLS     M50

MELSEC
        LD      M50
        AND     X9E
        ANI     X9F
        MOV     K4      D0
        MOV     H1234   D1
        MOV     H56AB   D2
        TO      H8      H400    D0      K3
        MOV     H3F2    D5
        MOV     H3F3    D6
        MOV     H8001   D7
        MOV     H8000   D8
        MOV     H41B    D9
        MOV     H0      D10
        TO      H8      H0BA    D5      K6

MELSEC
        LD      M50
        MOV     K1      D11
        MOV     K0      D12
        MOV     H0      D13
        MOV     H1      D14
        MOV     H5      D15
        G.PRR   U8      D11     M0

MELSEC
        LD      M0
        MPS
        ANI     M1
Ausführung der hier programmierten Anweisung, wenn die PRR-Anweisung ohne Fehler beendet wurde
        MPP
        AND     M1
Programmieren Sie an dieser Stelle die Anweisungen, die ausgeführt werden sollen, wenn bei der
Ausführung der PRR-Anweisung ein Fehler aufgetreten ist.
```

11.2 Anweisungen für PROFIBUS/DP-Module

Funktion	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor
Daten aus den Pufferspeicher von PROFIBUS/DP-Modulen lesen	G.BBLKRD	BBLKRD_M
	GP.BBLKRD	BBLKRDP_M
Daten in den Pufferspeicher von PROFIBUS/DP-Modulen schreiben	G.BBLKWR	BBLKWR_M
	GP.BBLKWR	BBLKWRP_M

11.2.1 BBLKRD, BBLKRDP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					●

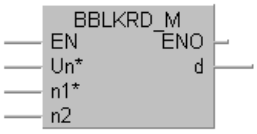
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□N□		Sondermodule U□NG□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
n1	—	●	●	—	—	—	—	●	—	SM0	
d	—	●	●	—	—	—	—	—	—		
n2	—	●	●	—	—	—	—	●	—		

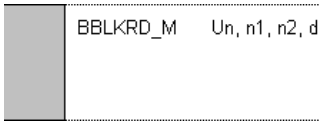
GX IEC Developer

MELSEC-Anweisungsliste

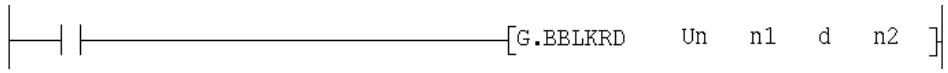
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

Operand	Bedeutung	Datentyp
Un	Kopfadresse des PROFIBUS-Moduls auf dem Baugruppenträger	BIN-16-Bit
n1	Anfangsadresse im Pufferspeicher des PROFIBUS-Moduls, ab der die Daten gelesen werden	
d	Anfangsadresse des Operandenbereichs, in dem die gelesenen Daten gespeichert werden	Operandenbezeichnung
n2	Anzahl der zu lesenden Daten	BIN-16-Bit

Funktions-weise

Daten aus dem Pufferspeicher eines PROFIBUS-Moduls lesen

BBLKRD / BBLKRDP Daten lesen

Die BBLKRD-Anweisung dient zum Auslesen des Pufferspeicherinhalts der PROFIBUS-Module QJ71PB92D und QJ71PB93D. Die gelesenen Daten die z. B. aus dem Ausgangsbereich des Moduls stammen können, werden zusammenhängend zur CPU übertragen.

Das QJ71PB93 muss durch Setzen des Ausgangssignals Y0A auf die BBLKRD-Anweisung vorbereitet werden. Setzt das PROFIBUS-Modul daraufhin den Eingang X0A, kann die BBLKRD-Anweisung ausgeführt werden. Nach dem Lesen des Pufferspeichers muss das Ausgangssignal Y0A wieder zurückgesetzt werden.

Wertebereiche und Angabe der Operanden:

- Un (Kopfadresse des PROFIBUS-Moduls auf dem Baugruppenträger): 0 bis FF_H
  (Es werden nur die ersten beiden Stellen der 3-stelligen Adresse angegeben, z. B. wird die Kopfadresse X/Y100 als 10_H eingetragen.)
- n1 (Anfangsadresse im Pufferspeicher): Die angegebene Adresse muss vorhanden sein.
- d (Anfangsadresse der Ziel-Operandenbereichs): Der hier angegebene Operand muss existieren
- n2 (Anzahl der zu lesenden Daten)
  Beim QJ71PB92D: 1 bis 960 Worte (1 bis $3C0_H$)
  Beim QJ71PB93D: 1 bis 122 Worte (1 bis $7A_H$)

HINWEISE

In einem Programmzyklus darf nur eine BBLKRD-Anweisung ausgeführt werden.

Die BBLKRD-Anweisung und die BBLKWR-Anweisung (Kapitel 11.2.2) arbeiten unabhängig voneinander.

Bei Verwendung der BBLKRD-Anweisung wird die Verzögerungszeit für die Übertragung verlängert.

Falls in den Parametern der Master-Station kein Ausgangsmodul eingetragen wurde, wird die BBLKRD-Anweisung nicht ausgeführt.

Fehler-quellen

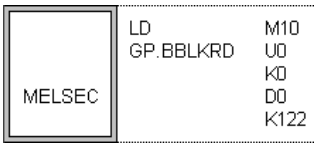
In den folgenden Fällen tritt ein Verarbeitungsfehler auf, das Error Flag SM0 wird gesetzt und in SD0 wird der Fehlercode eingetragen:

- Der eingestellte Wert liegt außerhalb des zulässigen Bereichs (Fehlercode: 4101).
- Durch die Addition der in n1 angegebenen Anfangsadresse und die in n2 angegebene Anzahl von Datenwörtern wird der Adressbereich des Pufferspeichers überschritten (Fehlercode 4101).
- Die in n2 angegebene Anzahl von Datenwörtern ist größer als der zur Verfügung stehende Operandenbereich für die gelesenen Daten, dessen Anfangsadresse in d angegeben ist. (Fehlercode 4101).

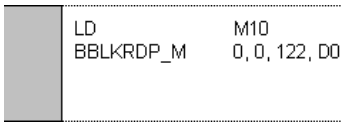
Beispiel BBLKRDP

Wenn der Merker M10 gesetzt wird, werden aus dem PROFIBUS-Modul QJ71PB93D mit der Kopfadresse X/Y0 122 Worte ab der Pufferspeicheradresse 0 gelesen und in der CPU ab Datenregister D0 gespeichert.

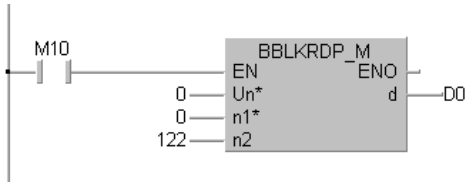
MELSEC-Anweisungsliste



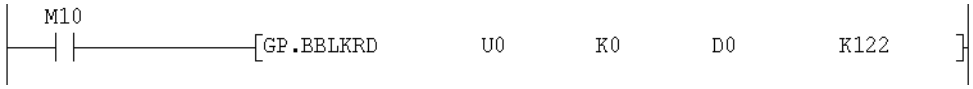
IEC-Anweisungsliste



Kontaktplan (GX IEC Developer)



Kontaktplan (GX Developer)



11.2.2 BBLKWR, BBLKWRP

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					●

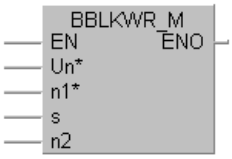
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□N□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
n1	—	●	●	—	—	—	—	●	—	SM0	
s	—	●	●	—	—	—	—	—	—		
n2	—	●	●	—	—	—	—	●	—		

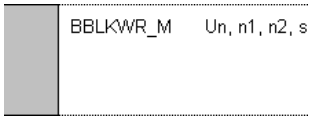
GX IEC Developer

MELSEC-Anweisungsliste

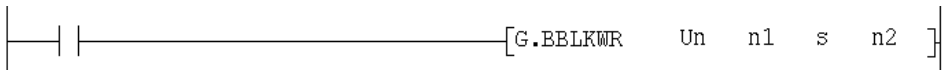
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

Operand	Bedeutung	Datentyp
Un	Kopfadresse des PROFIBUS-Moduls auf dem Baugruppenträger	BIN-16-Bit
n1	Anfangsadresse im Pufferspeicher des PROFIBUS-Moduls, ab der die Daten eingetragen werden	
s	Anfangsadresse des Operandenbereichs, in dem die Daten gespeichert sind, die zum PROFIBUS-Modul übertragen werden	Operandenbezeichnung
n2	Anzahl der zu übertragenden Daten	BIN-16-Bit

Funktions-weise

Daten in den Pufferspeicher eines PROFIBUS-Moduls schreiben

BBLKWR / BBLKWRP Daten schreiben

Mit der BBLKWR-Anweisung werden Daten in den Pufferspeicher der PROFIBUS/DP-Module QJ71PB92D und QJ71PB93D eingetragen. Dabei ist eine konsistente Übertragung, z. B. in den Eingangsbereich des Moduls, gewährleistet.

Das QJ71PB93 muss durch Setzen des Ausgangssignals Y0B auf die BBLKWR-Anweisung vorbereitet werden. Setzt das PROFIBUS-Modul daraufhin zur Bestätigung den Eingang X0B, kann die BBLKWR-Anweisung ausgeführt werden. Nach der Übertragung der Daten muss das Ausgangssignal Y0B wieder zurückgesetzt werden.

Wertebereiche und Angabe der Operanden:

- Un (Kopfadresse des PROFIBUS-Moduls auf dem Baugruppenträger): 0 bis FF_H
  (Es werden nur die ersten beiden Stellen der 3-stelligen Adresse angegeben, z. B. wird die Kopfadresse X/Y100 als 10_H eingetragen.)
- n1 (Anfangsadresse im Pufferspeicher): Die angegebene Adresse muss vorhanden sein.

  Beim QJ71PB93 ist die Anfangsadresse mit einem Offset von 100_H versehen. Bei der Angabe von n1 muss daher von der gewünschten Anfangsadresse der Wert 100_H abgezogen werden. Zum Beispiel wird die Anfangsadresse 100_H als „0_H“ und die Anfangsadresse 120_H als „20_H“ angegeben.
- s (Anfangsadresse des Quell-Operandenbereichs): Der hier angegebene Operand muss existieren
- n2 (Anzahl der zu lesenden Daten)
  Beim QJ71PB92D: 1 bis 960 Worte (1 bis $3C0_H$)
  Beim QJ71PB93D: 1 bis 122 Worte (1 bis $7A_H$)

HINWEISE

In einem Programmzyklus darf nur eine BBLKWR-Anweisung ausgeführt werden.

Die BBLKWR-Anweisung und die BBLKRD-Anweisung (Kapitel 11.2.1) arbeiten unabhängig voneinander.

Bei Verwendung der BBLKWR-Anweisung wird die Verzögerungszeit für die Übertragung verlängert.

Falls in den Parametern der Master-Station kein Eingangsmodul eingetragen wurde, wird die BBLKWR-Anweisung nicht ausgeführt.

Fehler-quellen

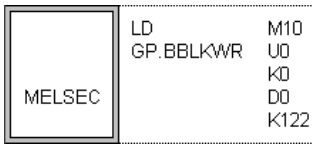
In den folgenden Fällen tritt ein Verarbeitungsfehler auf, das Error Flag SM0 wird gesetzt und in SD0 wird der Fehlercode eingetragen:

- Der eingestellte Wert liegt außerhalb des zulässigen Bereichs (Fehlercode: 4101).
- Durch die Addition der in n1 angegebenen Anfangsadresse und die in n2 angegebene Anzahl von Datenwörtern wird der Adressbereich des Pufferspeichers überschritten (Fehlercode 4101).
- Die in n2 angegebene Anzahl von Datenwörtern ist größer als der zur Verfügung stehende Operandenbereich für die zu übertragenen Daten, dessen Anfangsadresse in s angegeben ist. (Fehlercode 4101).

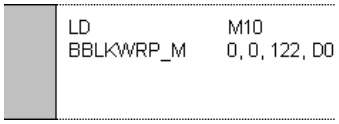
Beispiel BBLKWRP

Nach dem Setzen des Merkers M10 werden 122 Worte aus dem Operandenbereich D0 bis D121 der CPU in den Eingangsbereich des PROFIBUS/DP-Slave-Moduls QJ71PB93D übertragen. Der Eingangsbereich beginnt ab der Pufferspeicheradresse 100_H. Beachten Sie, dass die Anfangsadresse in n1 in diesem Fall mit „0_H“ angegeben wird. Das PROFIBUS/DP-Modul hat die Kopfadresse X/Y0.

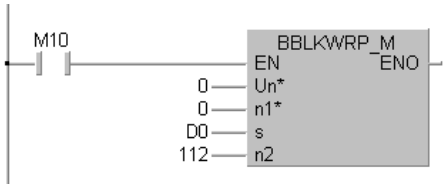
MELSEC-Anweisungsliste



IEC-Anweisungsliste



Kontaktplan (GX IEC Developer)



Kontaktplan (GX Developer)

11.3 Anweisungen für ETHERNET-Module

Funktion	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor
Daten aus den festen Puffern lesen	ZP.BUFRCV	BUFRCV_M
	Z.BUFRCVS	BUFRCVS_M
Daten in feste Puffer transferieren	ZP.BUFSND	BUFSND_M
Verbindung aufbauen (öffnen)	ZP.OPEN	OPEN_M
Verbindung abbauen (schließen)	ZP.CLOSE	CLOSE_M
Fehlerspeicher löschen	ZP.ERRCLR	ERRCLR_M
Fehlercode auslesen	ZP.ERRRD	ERRRD_M
ETHERNET-Modul initialisieren	ZP.UINI	UINI_M

11.3.1 BUFRCV

CPU

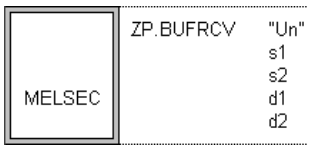
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					●

Operanden MELSEC Q

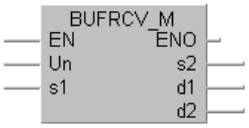
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□N□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	—	●	●	—	—	—	—	●	—		
s2	—	●	●	—	—	—	—	—	—		
d1	—	●	●	—	—	—	—	—	—		
d2	●	●	●	—	—	—	—	—	—		

GX IEC Developer

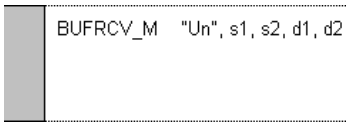
MELSEC-Anweisungsliste



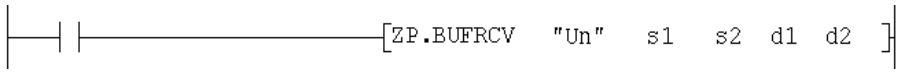
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

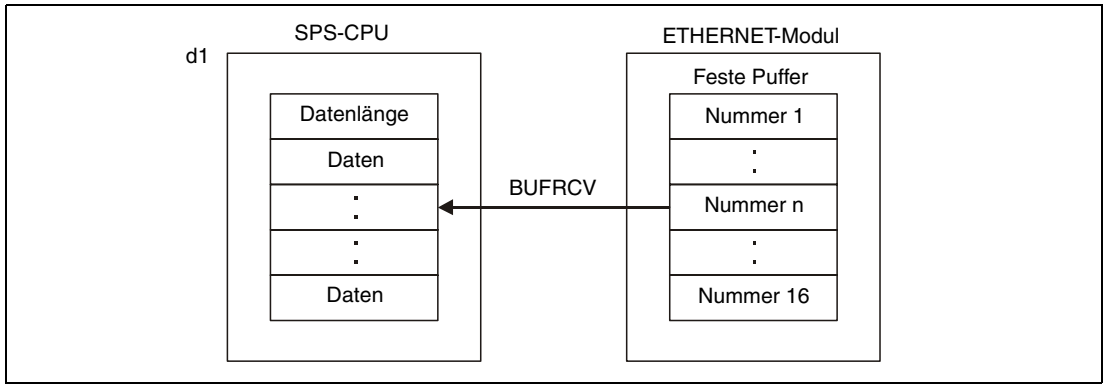
<table>
<tr><th>Operand</th><th colspan="4">Bedeutung</th><th>Werte-bereich</th><th>Festlegung des Inhalts durch</th><th>Datentyp</th></tr>
<tr><td>„Un“</td><td colspan="4">Kopfadresse des ETHERNET-Moduls auf dem Baugruppenträger (Es werden nur die ersten beiden Stellen der 3-stelligen Adresse angegeben, z. B. wird die Kopfadresse X/Y100 als „U10“ eingetragen.)</td><td>0 bis FE_H</td><td rowspan="2">Anwender</td><td rowspan="2">BIN-16-Bit</td></tr>
<tr><td>s1</td><td colspan="4">Nummer der Verbindung</td><td>1 bis 16</td></tr>
<tr><td rowspan="4">s2</td><td colspan="6">Erster Operand des Bereiches mit Informationen zur Ausführung der Anweisung</td><td rowspan="4">BIN-16-Bit</td></tr>
<tr><td>Operand</td><td colspan="2">Bedeutung</td><td>Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(s2)+0</td><td colspan="2">Systembereich</td><td>Wird vom System verwendet</td><td rowspan="2">—</td><td rowspan="2">System</td></tr>
<tr><td>(s2)+1</td><td colspan="2">Ausführungs-status der Anweisung</td><td>Zeigt an, ob bei der Bearbeitung der Anweisung ein Fehler aufgetreten ist:
0000_H: Fehlerfreie Bearbeitung
Jeder andere Wert als 0000_H: Bei der Bearbeitung ist ein Fehler aufgetreten. Der eingetragene Wert ist ein Fehlercode der entweder in der Bedienungsanleitung des ETHERNET-Moduls oder in Kapitel 13 dieses Handbuchs beschrieben ist.</td></tr>
<tr><td rowspan="5">d1</td><td colspan="6">Erster Operand des Bereiches, in dem die empfangenen Daten gespeichert werden.</td><td rowspan="5">BIN-16-Bit</td></tr>
<tr><td>Operand</td><td colspan="2">Bedeutung</td><td>Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td rowspan="3">(d1)+0</td><td colspan="2" rowspan="3">Länge der empfangenen Daten</td><td>Mit Prozedur (Binäre Daten): Anzahl der Daten, die aus dem festen Puffer gelesen wurden</td><td>1 bis 1017 Worte</td><td rowspan="3">System</td></tr>
<tr><td>Mit Prozedur (ASCII-Daten): Anzahl der Daten, die aus dem festen Puffer gelesen wurden.</td><td>1 bis 508 Worte</td></tr>
<tr><td>Ohne Prozedur (Binäre Daten): Anzahl der Daten, die aus dem festen Puffer gelesen wurden.</td><td>1 bis 2016 Bytes</td></tr>
<tr><td>(d1)+1 bis (d1)+n</td><td colspan="2">Empfangene Daten</td><td>In diesem Bereich werden die aus dem festen Puffer gelesenen Daten nacheinander in aufsteigender Reihenfolge eingetragen.</td><td>—</td></tr>
<tr><td rowspan="4">d2</td><td colspan="6">Bit-Operand, der nach der Ausführung der BUFRCV-Anweisung für einen Zyklus gesetzt wird. Mit (d2)+1 wird die fehlerhafte Beendigung signalisiert.</td><td rowspan="4">Bit</td></tr>
<tr><td>Operand</td><td colspan="2">Bedeutung</td><td>Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(d2)+0</td><td colspan="2">Anweisung ausgeführt</td><td>Zeigt die Beendigung der BUFRCV-Anweisung an.
EIN : Anweisung ausgeführt
AUS : Anweisung nicht ausgeführt</td><td>—</td><td rowspan="2">System</td></tr>
<tr><td>(d2)+1</td><td colspan="2">Anweisung mit Fehler ausgeführt</td><td>Zeigt an, ob bei der Ausführung der BUFRCV-Anweisung ein Fehler aufgetreten ist.
EIN: Anweisung mit Fehler ausgeführt
AUS: Anweisung ohne Fehler ausgeführt</td><td>—</td></tr>
</table>

Funktionsweise

Daten aus feste Puffer lesen (Aufruf der Anweisung im Hauptprogramm)

BUFRCV Daten lesen

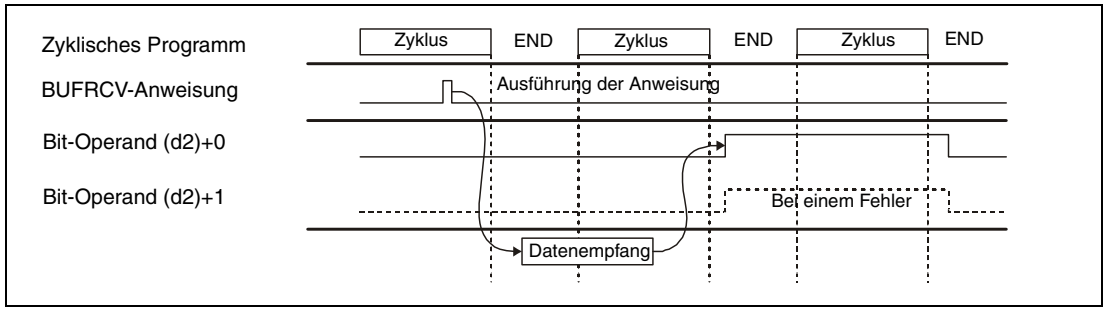
Daten, die bei der Kommunikation über feste Puffer von einer Partnerstation an ein ETHERNET-Modul gesendet wurden, können mit der BUFRCV-Anweisung aus dem ETHERNET-Modul gelesen und in der SPS-CPU gespeichert werden. Die BUFRCV-Anweisung wird im Gegensatz zur BUFRCVS-Anweisung im Hauptprogramm verwendet. Mit d1 wird angegeben, wo die Daten abgelegt werden sollen:



Ob die Ausführung der BUFRCV-Anweisung beendet ist, kann anhand der Bit-Operanden (d2)+0 und (d2)+1 überprüft werden:

- Der Bit-Operand (d2)+0 wird gesetzt, wenn die END-Anweisung des Zyklus ausgeführt wird, in dem die BUFRCV-Anweisung beendet wurde. Bei der nächsten Bearbeitung der END-Anweisung wird dieser Bit-Operand wieder zurückgesetzt.
- Der Bit-Operand (d2)+1 zeigt einen Fehler bei der Ausführung der BUFRCV-Anweisung an. Bei fehlerfreier Ausführung bleibt dieser Bit-Operand zurückgesetzt. Bei einem Fehler dagegen wird (d2)+1 gesetzt, wenn die END-Anweisung des Zyklus ausgeführt wird, in dem die BUFRCV-Anweisung beendet wurde. Bei der nächsten Bearbeitung der END-Anweisung wird der in (d2)+1 angegebene Bit-Operand wieder zurückgesetzt.

Die folgende Abbildung zeigt den Signalverlauf bei Ausführung der BUFRCV-Anweisung:



Die BUFRCV-Anweisung kann ausgeführt werden, wenn durch das ETHERNET-Modul angezeigt wird, dass Daten empfangen wurden. Dazu wird in der Pufferspeicheradresse 5005_H des ETHERNET-Moduls ein Bit gesetzt. Für jede der 16 möglichen Verbindungen ist hier ein Bit reserviert.

HINWEIS *Empfangene Daten derselben Verbindung können nicht gleichzeitig mit der BUFRCV- (für Hauptprogramme) und der BUFRCVS-Anweisung (für Interrupt-Programme) gelesen werden.*

Fehlerquellen

Wenn die BUFRCV-Anweisung fehlerhaft ausgeführt wurde, wird der Operand (d2)+1 gesetzt und der entsprechende Fehlercode wird in (s2)+1 gespeichert. Detaillierte Angaben zu den einzelnen Fehlercodes entnehmen Sie bitte den folgenden Anleitungen:

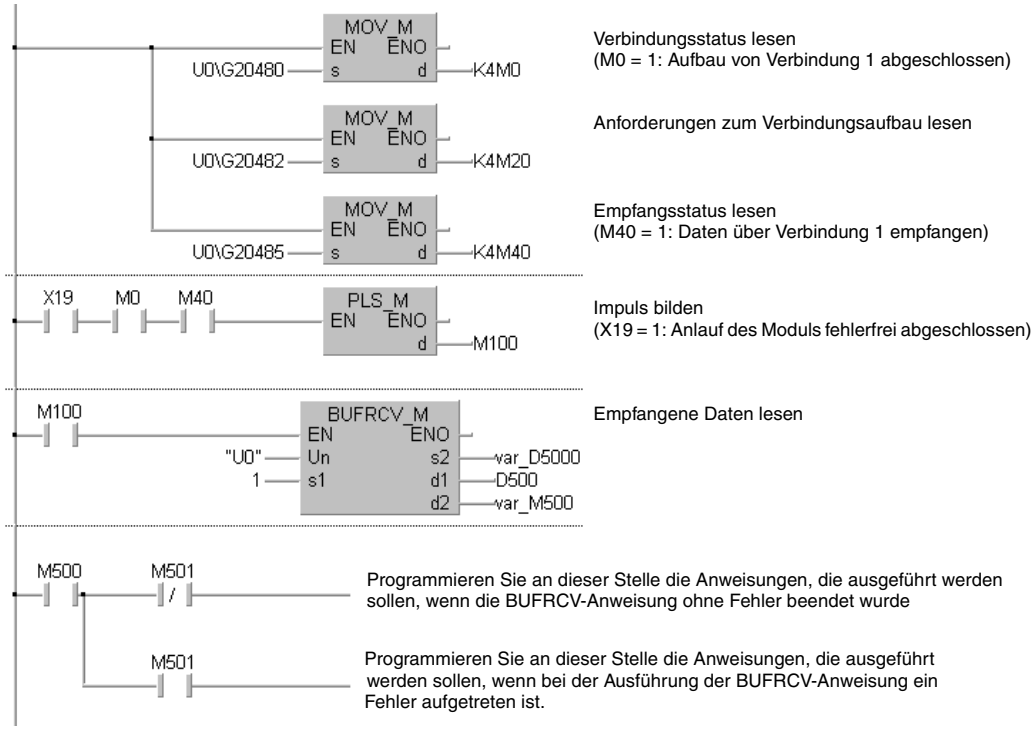
- Bei einem Fehlercode bis $4FFF_H$ finden Sie Hinweise zur Fehlerbehebung in diesem Handbuch (Kapitel 13).
- Bei einem Fehlercode ab $C001_H$ finden Sie detaillierte Angaben in der Bedienungsanleitung zu den ETHERNET-Modulen des System Q.

Beispiel BUFRCV

Das folgende Programm liest die über Verbindung 1 empfangenden Daten aus den festen Puffern. Das ETHERNET-Modul belegt in der SPS den Adressbereich von X/Y0 bis X/Y1F.

- IEC-Editoren (Auf der nächsten Seite ist dieses Beispiel in der MELSEC-Anweisungliste und dem Kontaktplan des GX Developers dargestellt.)

Kontaktplan (GX IEC Developer)



IEC-Anweisungsliste

```
LD TRUE
MOV_M       U0\G20480, K4M0
MOV_M       U0\G20482, K4M20
MOV_M       U0\G20485, K4M40
LD          X19
AND         M0
AND         M40
PLS_M       M100
LD          M100
BUFRCV_M    "U0", K1, var_D5000, D500, var_M500
```

Die Operanden und Anweisungen sind oben beim Programmbeispiel für den Kontaktplan erläutert.

```
LD          M500
ANDN        M501
```

Programmieren Sie an dieser Stelle die Anweisungen, die ausgeführt werden sollen, wenn die BUFRCV-Anweisung ohne Fehler beendet wurde

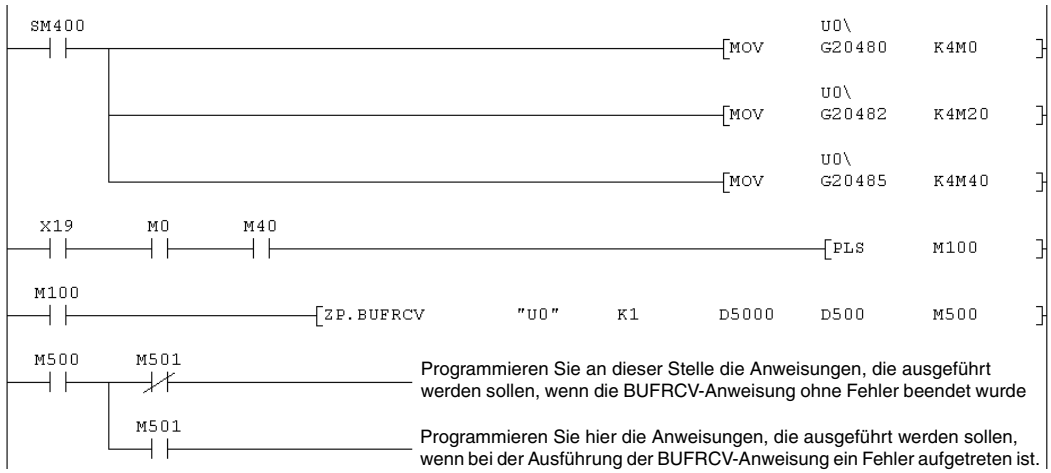
```
LD          M500
AND         M501
```

Programmieren Sie an dieser Stelle die Anweisungen, die ausgeführt werden sollen, wenn bei der Ausführung der BUFRCV-Anweisung ein Fehler aufgetreten ist.

HINWEIS *Bei den IEC-Editoren des GX IEC Developers müssen die Variablen im Header der Programmorganisationseinheit (POE) definiert werden. Ohne Variablendefinition werden beim Überprüfen oder Kompilieren des Programms Fehler gemeldet.*
Weitere Informationen finden Sie im Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung.

- MELSEC-Anweisungliste und Kontaktplan des GX Developers
  Die Operanden und Anweisungen sind auf der vorherigen Seite beim Programmbeispiel für den Kontaktplan des GX IEC Developers erläutert.

Kontaktplan (GX Developer)



MELSEC-Anweisungsliste

MELSEC	LD	SM400				
	MOV	U0\G20480	K4M0			
	MOV	U0\G20482	K4M20			
	MOV	U0\G20485	K4M40			
	LD	X19				
	AND	M0				
	AND	M40				
	PLS	M100				
	LD	M100				
	ZP.BUFRCV	"U0"	K1	D5000	D500	M500
MELSEC	LD	M500				
	ANI	M501				
	Programmieren Sie an dieser Stelle die Anweisungen, die ausgeführt werden sollen, wenn die BUFRCV-Anweisung ohne Fehler beendet wurde					
MELSEC	LD	M500				
	AND	M501				
	Programmieren Sie an dieser Stelle die Anweisungen, die ausgeführt werden sollen, wenn bei der Ausführung der BUFRCV-Anweisung ein Fehler aufgetreten ist.					

11.3.2 BUFRCVS

CPU

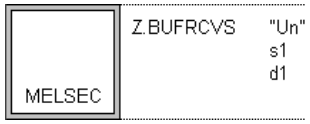
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					●

Operanden MELSEC Q

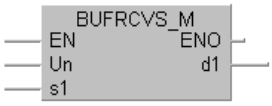
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□N□		Sondermodule U□NG□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	—	●	●	—	—	—	—	●	—	SM0	
d1	—	●	●	—	—	—	—	—	—		

GX IEC Developer

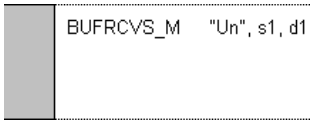
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

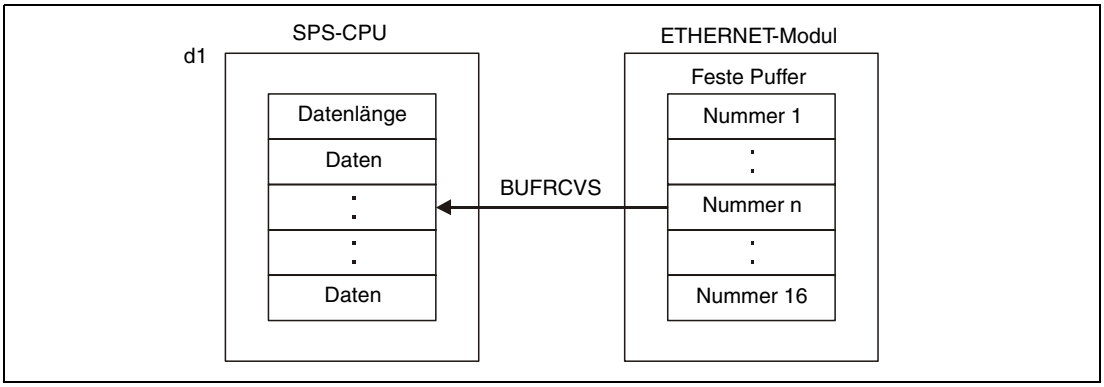
Operand	Bedeutung			Werte-bereich	Festlegung des Inhalts durch	Datentyp
„Un“	Kopfadresse des ETHERNET-Moduls auf dem Baugruppenträger (Es werden nur die ersten beiden Stellen der 3-stelligen Adresse angegeben, z. B. wird die Kopfadresse X/Y100 als „U10“ eingetragen.)			0 bis FE_H	Anwender	BIN-16-Bit
s1	Nummer der Verbindung			1 bis 16		
d1	Erster Operand des Bereiches, in dem die empfangenen Daten gespeichert werden.					BIN-16-Bit
	Operand	**Bedeutung**	**Beschreibung**	**Werte-bereich**	**Festlegung durch**	
	(d1)+0	Länge der empfangenen Daten (Anzahl der Daten, die aus dem festen Puffer gelesen wurden)	Mit Prozedur (Binäre Daten):	1 bis 1017 worte	System	
			Mit Prozedur (ASCII-Daten):	1 bis 508 Worte		
			Ohne Prozedur (Binäre Daten):	1 bis 2016 Bytes		
	(d1)+1 bis (d1)+n	Empfangene Daten	In diesem Bereich werden die aus dem festen Puffer gelesenen Daten nacheinander in aufsteigender Reihenfolge eingetragen.	—		

Funktionsweise

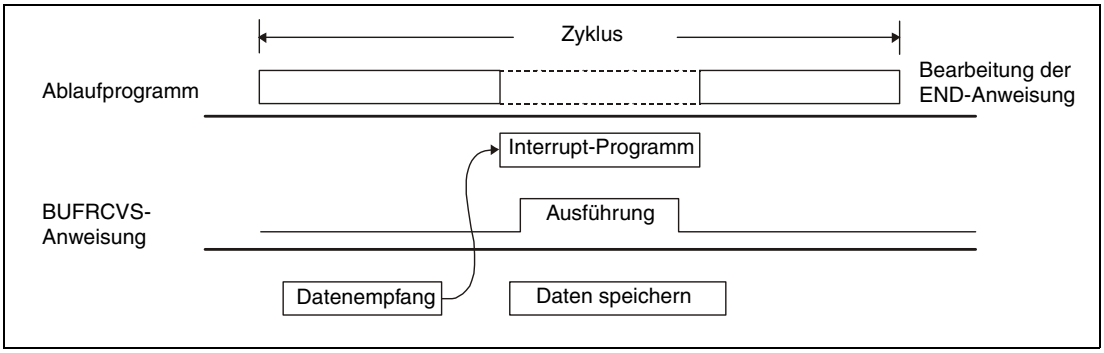
Daten aus feste Puffer lesen (Aufruf der Anweisung in einem Interrupt-Programm)

BUFRCVS Daten lesen

Daten, die bei der Kommunikation über feste Puffer von einer Partnerstation an ein ETHERNET-Modul gesendet wurden, können mit der BUFRCVS-Anweisung aus dem ETHERNET-Modul gelesen und in der SPS-CPU gespeichert werden. Die BUFRCVS-Anweisung wird im Gegensatz zur BUFRCV-Anweisung in einem Interrupt-Programm verwendet. Mit d1 wird angegeben, wo die Daten abgelegt werden sollen:



Die Übertragung der Daten wird in einem Zylus abgeschlossen. Die folgende Abbildung zeigt den Signalverlauf bei Ausführung der BUFRCVS-Anweisung:



Um Empfangsdaten über ein Interrupt-Programm auszulesen, ist es notwendig, dass die Interrupt-Einstellungen und die Interrupt-Pointer mit dem GX (IEC) Developer parametriert werden.

HINWEISE

Empfangene Daten derselben Verbindung können nicht gleichzeitig mit der BUFRCV- (für Hauptprogramme) und der BUFRCVS-Anweisung (für Interrupt-Programme) gelesen werden.

Die BUFRCVS-Anweisung kann auch in Verbindung mit einem Schnittstellenmodul QJ71C24 eingesetzt werden.

Fehlerquellen

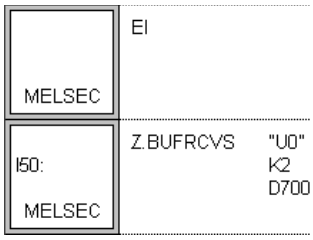
Wenn bei der Ausführung der BUFRCVS-Anweisung ein Fehler aufgetreten ist, wird das Error Flag SM0 gesetzt und in SD0 ein Fehlercode eingetragen. Detaillierte Angaben zu den Fehlercodes entnehmen Sie bitte den folgenden Anleitungen:

- Bei einem Fehlercode bis $4FFF_H$ finden Sie Hinweise zur Fehlerbehebung in diesem Handbuch (Kapitel 13).
- Bei einem Fehlercode ab $C001_H$ finden Sie detaillierte Angaben in der Bedienungsanleitung zu den ETHERNET-Modulen des System Q.

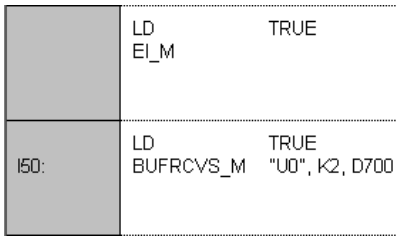
Beispiel BUFRCVS

Das folgende Programm liest die über Verbindung 2 empfangenden Daten aus den festen Puffern des ETHERNET-Moduls mit der Kopfadresse X/Y0.

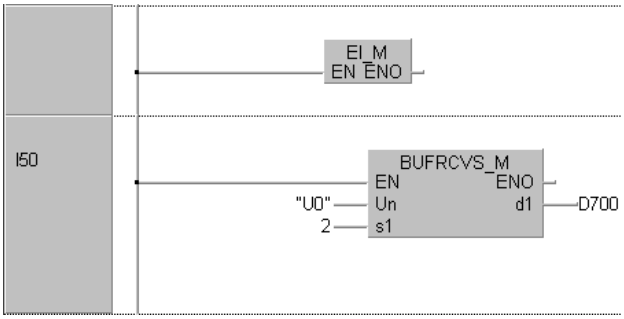
MELSEC-Anweisungsliste



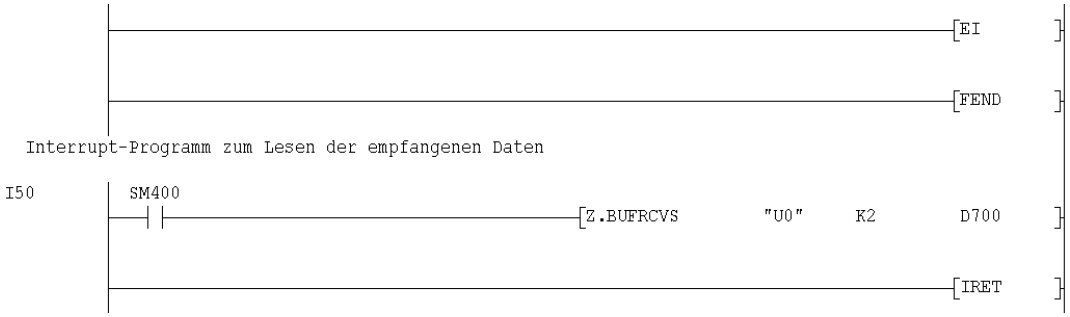
IEC-Anweisungsliste



Kontaktplan (GX IEC Developer)



Kontaktplan (GX Developer)



11.3.3 BUFSND

CPU

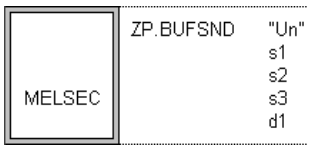
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					●

Operanden MELSEC Q

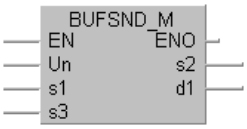
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□N□		Sondermodule U□NG□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	—	●	●	—	—	—	—	●	—		
s2	—	●	●	—	—	—	—	—	—		
s3	—	●	●	—	—	—	—	—	—		
d1	●	●	●	—	—	—	—	—	—		

GX IEC Developer

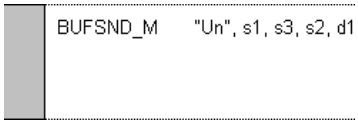
MELSEC-Anweisungsliste



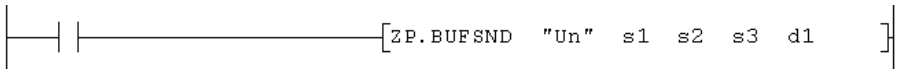
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

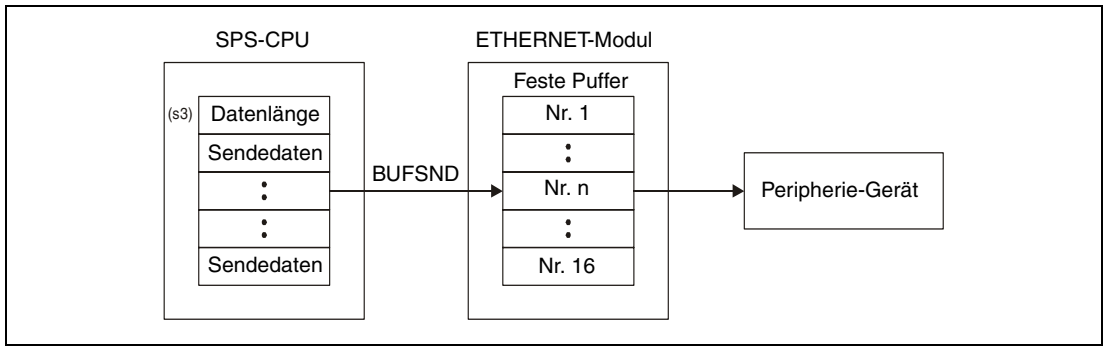
Operand	Bedeutung			Werte-bereich	Festlegung des Inhalts durch	Datentyp
„Un“	Kopfadresse des ETHERNET-Moduls auf dem Baugruppenträger (Es werden nur die ersten beiden Stellen der 3-stelligen Adresse angegeben, z. B. wird die Kopfadresse X/Y100 als „U10“ eingetragen.)			0 bis FE_H	Anwender	BIN-16-Bit
s1	Nummer der Verbindung			1 bis 16		
s2	Erster Operand des Bereiches mit Informationen zur Ausführung der Anweisung					BIN-16-Bit
	Operand	**Bedeutung**	**Beschreibung**	**Werte-bereich**	**Festlegung durch**	
	(s2)+0	Systembereich	Wird vom System verwendet	—	System	
	(s2)+1	Ausführungs-status der Anweisung	Zeigt an, ob bei der Bearbeitung der Anweisung ein Fehler aufgetreten ist: 0000_H: Fehlerfreie Bearbeitung Jeder andere Wert als 0000_H: Bei der Bearbeitung ist ein Fehler aufgetreten. Der eingetragene Wert ist ein Fehlercode, der entweder in der Bedienungsanleitung des ETHERNET-Moduls oder in Kapitel 13 dieses Handbuchs beschrieben ist.			
s3	Erster Operand des Bereiches, in dem die zu sendenden Daten gespeichert sind					BIN-16-Bit
	Operand	**Bedeutung**	**Beschreibung**	**Werte-bereich**	**Festlegung durch**	
	(s3)+0	Länge der zu sendenden Daten	Angabe der Anzahl der Daten, die bei der Übertragung mit Prozedur (binäre Daten) in den festen Puffer eingetragen werden sollen.	1 bis 1017 Worte	Anwender	
			Angabe der Anzahl der Daten, die bei der Übertragung mit Prozedur (ASCII-Daten) in den festen Puffer eingetragen werden sollen.	1 bis 508 Worte		
			Angabe der Anzahl der Daten, die bei der Übertragung ohne Prozedur (binäre Daten) in den festen Puffer eingetragen werden sollen.	1 bis 2046 Bytes		
	(s3)+1 bis (s3)+n	Sendedaten	Daten, die zum ETHERNET-Modul übertragen werden	—		
d1	Bit-Operand, der nach der Ausführung der BUFSND-Anweisung für einen Zyklus gesetzt wird. Mit (d1)+1 wird die fehlerhafte Beendigung signalisiert.					Bit
	Operand	**Bedeutung**	**Beschreibung**	**Werte-bereich**	**Festlegung durch**	
	(d1)+0	Anweisung ausgeführt	Zeigt die Beendigung der BUFSND-Anweisung an. EIN : Anweisung ausgeführt AUS : Anweisung nicht ausgeführt	—	System	
	(d1)+1	Anweisung mit Fehler ausgeführt	Zeigt an, ob bei der Ausführung der BUFSND-Anweisung ein Fehler aufgetreten ist. EIN: Anweisung mit Fehler ausgeführt AUS: Anweisung ohne Fehler ausgeführt	—		

Funktionsweise

Daten in feste Puffer eintragen

BUFSND Daten schreiben

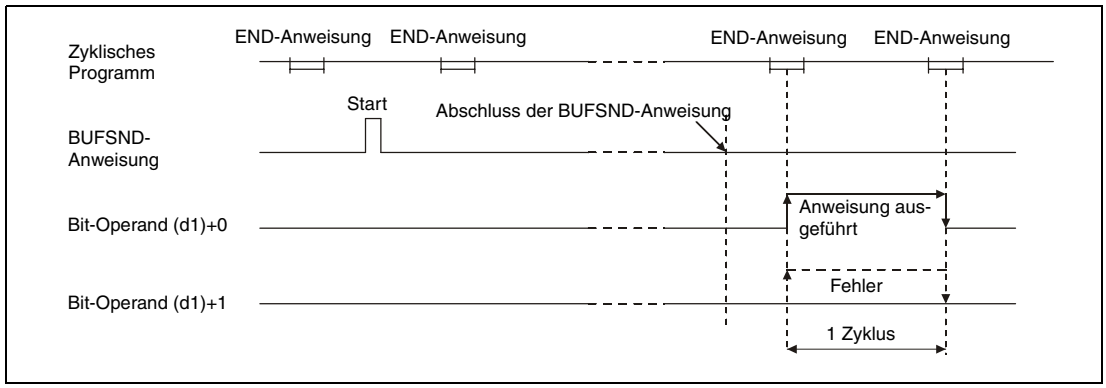
Daten, die bei der Kommunikation über feste Puffer an eine an ein ETHERNET-Modul angeschlossene Partnerstation gesendet werden sollen, können mit der BUFSND-Anweisung an das ETHERNET-Modul übertragen werden. Die Daten sind in der CPU ab dem unter (s3)+1 angegebenen Operanden gespeichert:



Ob die Ausführung der BUFSND-Anweisung beendet ist, kann anhand der Bit-Operanden (d1)+0 und (d1)+1 überprüft werden:

- Der Bit-Operand (d1)+0 wird gesetzt, wenn die END-Anweisung des Zyklus ausgeführt wird, in dem die BUFSND-Anweisung beendet wurde. Bei der nächsten Bearbeitung der END-Anweisung wird der in (d1)+0 angegebene Bit-Operand wieder zurückgesetzt.
- Der Bit-Operand (d1)+1 zeigt einen Fehler bei der Ausführung der BUFSND-Anweisung an. Bei fehlerfreier Ausführung bleibt dieser Bit-Operand zurückgesetzt. Bei einem Fehler jedoch wird (d1)+1 gesetzt, wenn die END-Anweisung des Zyklus ausgeführt wird, in dem die BUFSND-Anweisung beendet wurde. Bei der nächsten Bearbeitung der END-Anweisung wird der in (d1)+1 angegebene Bit-Operand wieder zurückgesetzt.

Die folgende Abbildung zeigt den Signalverlauf bei Ausführung der BUFSND-Anweisung:



Die BUFSND-Anweisung wird ausgeführt, wenn die Eingangsbedingung der Anweisung erfüllt ist.

Fehlerquellen

Wenn die BUSND-Anweisung fehlerhaft ausgeführt wurde, wird der Operand (d1)+1 gesetzt und der entsprechende Fehlercode wird in (s2)+1 gespeichert. Detaillierte Angaben zu den einzelnen Fehlercodes entnehmen Sie bitte den folgenden Anleitungen:

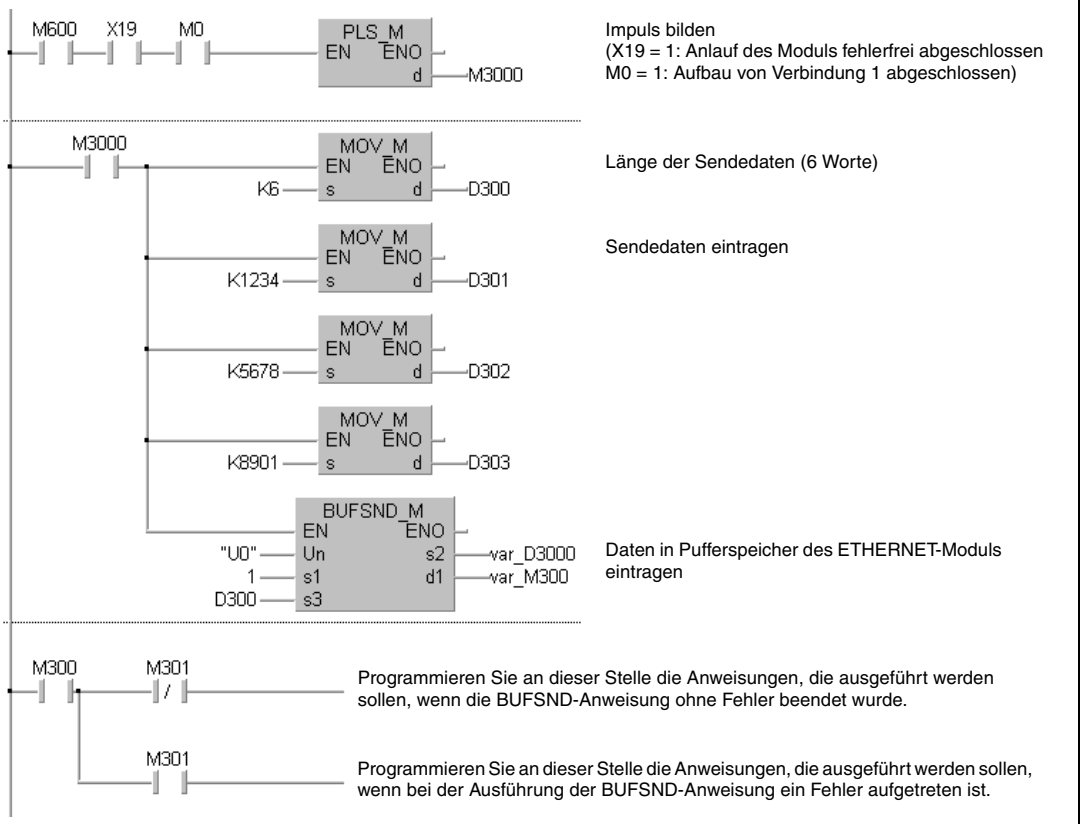
- Bei einem Fehlercode bis $4FFF_H$ finden Sie Hinweise zur Fehlerbehebung in diesem Handbuch (Kapitel 13).
- Bei einem Fehlercode ab $C001_H$ finden Sie detaillierte Angaben in der Bedienungsanleitung zu den ETHERNET-Modulen des System Q.

Beispiel BUFSND

Das folgende Programm trägt Daten in den festen Puffer der Verbindung 1 ein. Das ETHERNET-Moduls hat die Kopfadresse X/Y0.

- IEC-Editoren (Auf der nächsten Seite ist dieses Beispiel in der MELSEC-Anweisungliste und dem Kontaktplan des GX Developers dargestellt.)

Kontaktplan (GX IEC Developer)



IEC-Anweisungsliste

```
LD          M600
AND         X19
AND         M0
PLS_M       M3000
LD          M3000
MOV_M       K6, D300
MOV_M       K1234, D301
MOV_M       K5678, D302
MOV_M       K8901, D303
BUFSND_M    "U0", K1, D300, var_D3000, var_M300
```

Die Operanden und Anweisungen sind oben beim Programmbeispiel für den Kontaktplan erläutert.

```
LD          M300
ANDN        M301
```

Programmieren Sie an dieser Stelle die Anweisungen, die ausgeführt werden sollen, wenn die BUFSND-Anweisung ohne Fehler beendet wurde.

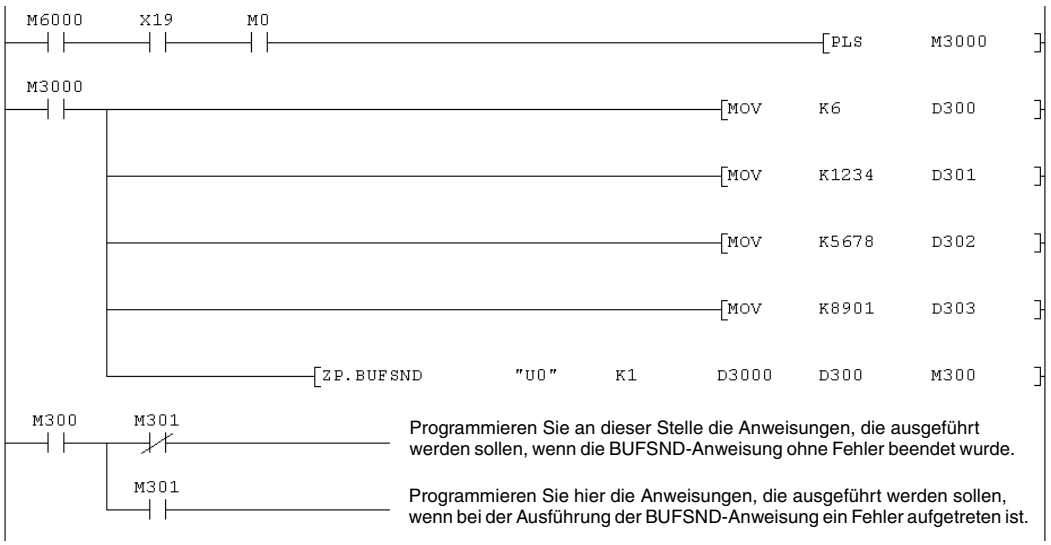
```
LD          M300
AND         M301
```

Programmieren Sie an dieser Stelle die Anweisungen, die ausgeführt werden sollen, wenn bei der Ausführung der BUFSND-Anweisung ein Fehler aufgetreten ist.

HINWEIS *Bei den IEC-Editoren des GX IEC Developers müssen die Variablen im Header der Programmorganisationseinheit (POE) definiert werden. Ohne Variablendefinition werden beim Überprüfen oder Kompilieren des Programms Fehler gemeldet.*
Weitere Informationen finden Sie im Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung.

- MELSEC-Anweisungliste und Kontaktplan des GX Developers
  Die Operanden und Anweisungen sind auf der vorherigen Seite beim Programmbeispiel für den Kontaktplan des GX IEC Developers erläutert.

Kontaktplan (GX Developer)



MELSEC-Anweisungsliste

```
MELSEC   LD        M6000
         AND       X19
         AND       M0
         PLS       M3000
         LD        M3000
         MOV       K6        D300
         MOV       K1234     D301
         MOV       K5678     D302
         MOV       K8901     D303
         ZP.BUFSND "U0"      K1        D3000   D300    M300
```

```
MELSEC   LD        M300
         ANI       M301
```

Programmieren Sie an dieser Stelle die Anweisungen, die ausgeführt werden sollen, wenn die BUFSND-Anweisung ohne Fehler beendet wurde.

```
MELSEC   LD        M300
         AND       M301
```

Programmieren Sie an dieser Stelle die Anweisungen, die ausgeführt werden sollen, wenn bei der Ausführung der BUFSND-Anweisung ein Fehler aufgetreten ist.

11.3.4 OPEN

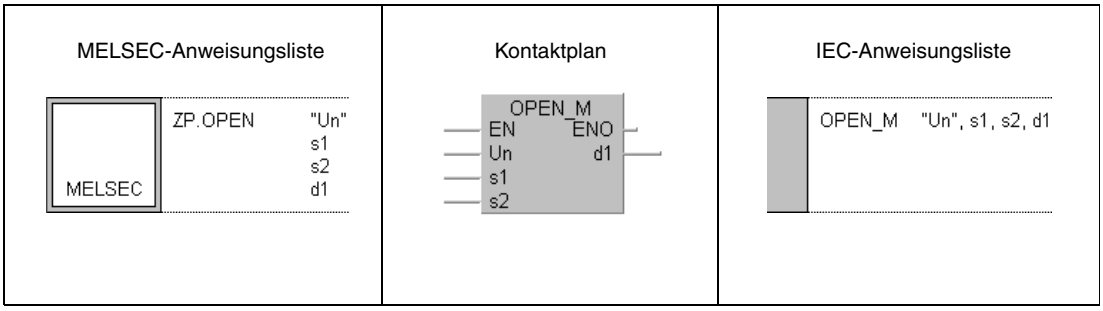
CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					●

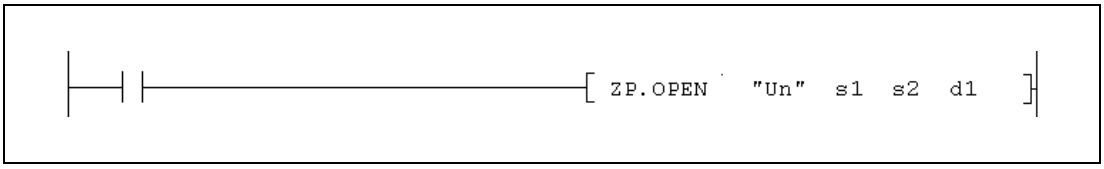
Operanden MELSEC Q

	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□N□		Sondermodule U□G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	—	●	●	—	—	—	—	●	—		
s2	—	●	●	—	—	—	—	—	—		
d1	●	●	●	—	—	—	—	—	—		

GX IEC Developer



GX Developer



Variablen

Operand	Bedeutung	Wertebereich	Festlegung des Inhalts durch	Datentyp
„Un“	Kopfadresse des ETHERNET-Moduls auf dem Baugruppenträger (Es werden nur die ersten beiden Stellen der 3-stelligen Adresse angegeben, z. B. wird die Kopfadresse X/Y100 als „U10“ eingetragen.)	0 bis FE_H	Anwender	BIN-16-Bit
s1	Nummer der Verbindung	1 bis 16		

Variablen

<table>
<tr><th>Operand</th><th colspan="3">Bedeutung</th><th>Werte-bereich</th><th>Festlegung des Inhalts durch</th><th>Datentyp</th></tr>
<tr><td rowspan="5">s2</td><td colspan="5">Erster Operand des Bereiches mit Informationen zur Ausführung der Anweisung</td><td rowspan="5">BIN-16-Bit</td></tr>
<tr><td>Operand</td><td>Bedeutung</td><td>Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(s2)+0</td><td>Parameterquelle</td><td>Angabe, welche Parameter zum Aufbau der Verbindung verwendet werden sollen.
• 0000_H: Die Verbindung wird mit den im GX (IEC) Developer angegebenen Parametern aufgebaut.
• 8000_H: Die Verbindung wird mit den Parametern aufgebaut, die in den Operanden (s2)+2 bis (s2)+9 gespeichert sind.</td><td>0000_H oder 8000_H</td><td>Anwender</td></tr>
<tr><td>(s2)+1</td><td>Ausführungs-status der Anweisung</td><td>Zeigt an, ob bei der Bearbeitung der Anweisung ein Fehler aufgetreten ist.
0000_H: Fehlerfreie Bearbeitung
Jeder andere Wert als 0000_H: Bei der Bearbeitung ist ein Fehler aufgetreten. Der eingetragene Wert ist ein Fehlercode, der entweder in der Bedienungsanleitung des ETHERNET-Moduls oder in Kapitel 13 dieses Handbuchs beschrieben ist.</td><td>—</td><td>System</td></tr>
<tr><td>(s2)+2</td><td>Verbindungsart</td><td>Die einzelnen Bits dieses Operanden dienen zur Einstellung von Verbindungsmerkmalen.
• Bit 0: Verwendung fester Puffer
0: Puffer dient als Sendepuffer, die Übertragung fester Puffer ist abgeschaltet
1: Puffer dient zum Empfang von Daten
• Bit 1: Prüfung, ob die Partnerstation existiert
0: Keine Existenzprüfung
1: Prüfung wird ausgeführt
• Bit 7: Paarweises Öffnen der Verbindung
0: Keine paarige Verbindung
1: Paarige Verbindung
• Bit 8: Wahl des Übertragungsprotokolls
0: TCP/IP
1: UDP/IP
• Bit 9: Prozedur bei der Übertragung fester Puffer
0: Mit Prozedur
1: Ohne Prozedur
• Bit 13: Interrupt-Auslösung beim Empfang fester Puffer
0: Kein Interrupt
1: Interrupt wird ausgelöst
• Bits 14 und 15:Aktiver oder passiver Verbindungsaufbau
Bit 15/14 = 00: Verbindung wird aktiv geöffnet oder UDP/IP
Bit 15/14 = 10: Verbindung wird „unpassive“ geöffnet
Bit 15/14 = 11: Verbindung wird „full passive“ geöffnet</td><td>Die Werte ergeben sich durch Setzen der einzelnen Bits.</td><td>Anwender</td></tr>
</table>

Variablen

<table>
<tr><th>Operand</th><th colspan="3">Bedeutung</th><th>Werte-bereich</th><th>Festlegung des Inhalts durch</th><th>Datentyp</th></tr>
<tr><td rowspan="5">s2</td><td colspan="5">Erster Operand des Bereiches mit Informationen zur Ausführung der Anweisung</td><td rowspan="5">BIN-16-Bit</td></tr>
<tr><td>Operand</td><td>Bedeutung</td><td>Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(s2)+0</td><td>Parameterquelle</td><td>Angabe, welche Parameter zum Aufbau der Verbindung verwendet werden sollen.
• 0000_H: Die Verbindung wird mit den im GX (IEC) Developer angegebenen Parametern aufgebaut.
• 8000_H: Die Verbindung wird mit den Parametern aufgebaut, die in den Operanden (s2)+2 bis (s2)+9 gespeichert sind.</td><td>0000_H oder 8000_H</td><td>Anwender</td></tr>
<tr><td>(s2)+1</td><td>Ausführungs-status der Anweisung</td><td>Zeigt an, ob bei der Bearbeitung der Anweisung ein Fehler aufgetreten ist.
0000_H: Fehlerfreie Bearbeitung
Jeder andere Wert als 0000_H: Bei der Bearbeitung ist ein Fehler aufgetreten. Der eingetragene Wert ist ein Fehlercode, der entweder in der Bedienungsanleitung des ETHERNET-Moduls oder in Kapitel 13 dieses Handbuchs beschrieben ist.</td><td>—</td><td>System</td></tr>
<tr><td>(s2)+2</td><td>Verbindungsart</td><td>Die einzelnen Bits dieses Operanden dienen zur Einstellung von Verbindungsmerkmalen.
• Bit 0: Verwendung fester Puffer
0: Puffer dient als Sendepuffer, die Übertragung fester Puffer ist abgeschaltet
1: Puffer dient zum Empfang von Daten
• Bit 1: Prüfung, ob die Partnerstation existiert
0: Keine Existenzprüfung
1: Prüfung wird ausgeführt
• Bit 7: Paarweises Öffnen der Verbindung
0: Keine paarige Verbindung
1: Paarige Verbindung
• Bit 8: Wahl des Übertragungsprotokolls
0: TCP/IP
1: UDP/IP
• Bit 9: Prozedur bei der Übertragung fester Puffer
0: Mit Prozedur
1: Ohne Prozedur
• Bit 13: Interrupt-Auslösung beim Empfang fester Puffer
0: Kein Interrupt
1: Interrupt wird ausgelöst
• Bits 14 und 15:Aktiver oder passiver Verbindungsaufbau
Bit 15/14 = 00: Verbindung wird aktiv geöffnet oder UDP/IP
Bit 15/14 = 10: Verbindung wird „unpassive“ geöffnet
Bit 15/14 = 11: Verbindung wird „full passive“ geöffnet</td><td>Die Werte ergeben sich durch Setzen der einzelnen Bits.</td><td>Anwender</td></tr>
</table>

Variablen

<table>
<tr><th>Operand</th><th colspan="3">Bedeutung</th><th>Werte-bereich</th><th>Festlegung des Inhalts durch</th><th>Datentyp</th></tr>
<tr><td rowspan="5">s2</td><td colspan="5">Erster Operand des Bereiches mit Informationen zur Ausführung der Anweisung</td><td rowspan="5">BIN-16-Bit</td></tr>
<tr><td>Operand</td><td>Bedeutung</td><td>Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(s2)+0</td><td>Parameterquelle</td><td>Angabe, welche Parameter zum Aufbau der Verbindung verwendet werden sollen.
• 0000_H: Die Verbindung wird mit den im GX (IEC) Developer angegebenen Parametern aufgebaut.
• 8000_H: Die Verbindung wird mit den Parametern aufgebaut, die in den Operanden (s2)+2 bis (s2)+9 gespeichert sind.</td><td>0000_H oder 8000_H</td><td>Anwender</td></tr>
<tr><td>(s2)+1</td><td>Ausführungs-status der Anweisung</td><td>Zeigt an, ob bei der Bearbeitung der Anweisung ein Fehler aufgetreten ist.
0000_H: Fehlerfreie Bearbeitung
Jeder andere Wert als 0000_H: Bei der Bearbeitung ist ein Fehler aufgetreten. Der eingetragene Wert ist ein Fehlercode, der entweder in der Bedienungsanleitung des ETHERNET-Moduls oder in Kapitel 13 dieses Handbuchs beschrieben ist.</td><td>—</td><td>System</td></tr>
<tr><td>(s2)+2</td><td>Verbindungsart</td><td>Die einzelnen Bits dieses Operanden dienen zur Einstellung von Verbindungsmerkmalen.
• Bit 0: Verwendung fester Puffer
0: Puffer dient als Sendepuffer, die Übertragung fester Puffer ist abgeschaltet
1: Puffer dient zum Empfang von Daten
• Bit 1: Prüfung, ob die Partnerstation existiert
0: Keine Existenzprüfung
1: Prüfung wird ausgeführt
• Bit 7: Paarweises Öffnen der Verbindung
0: Keine paarige Verbindung
1: Paarige Verbindung
• Bit 8: Wahl des Übertragungsprotokolls
0: TCP/IP
1: UDP/IP
• Bit 9: Prozedur bei der Übertragung fester Puffer
0: Mit Prozedur
1: Ohne Prozedur
• Bit 13: Interrupt-Auslösung beim Empfang fester Puffer
0: Kein Interrupt
1: Interrupt wird ausgelöst
• Bits 14 und 15:Aktiver oder passiver Verbindungsaufbau
Bit 15/14 = 00: Verbindung wird aktiv geöffnet oder UDP/IP
Bit 15/14 = 10: Verbindung wird „unpassive“ geöffnet
Bit 15/14 = 11: Verbindung wird „full passive“ geöffnet</td><td>Die Werte ergeben sich durch Setzen der einzelnen Bits.</td><td>Anwender</td></tr>
</table>

Variablen

<table>
<tr><th>Operand</th><th colspan="3">Bedeutung</th><th>Werte-bereich</th><th>Festlegung des Inhalts durch</th><th>Datentyp</th></tr>
<tr><td rowspan="6">s2</td><td>Operand</td><td>Bedeutung</td><td>Beschreibung</td><td rowspan="2"></td><td rowspan="2"></td><td rowspan="2"></td></tr>
<tr><td>(s2)+3</td><td>Port-Nr. des ETHERNET-Moduls</td><td>Geben Sie hier die Port-Nummer des ETHERNET-Moduls an.</td></tr>
<tr><td>(s2)+4
(s2)+5</td><td>IP-Adresse der Ziel-Station</td><td>IP-Adresse der Partnerstation, mit der kommuniziert wird.
Bei einer IP-Adresse von $FFFFFFFF_H$ werden Daten im Broadcast-Verfahren ausgetauscht.</td><td>1_H bis $FFFFFFFF_H$</td><td rowspan="4">Anwender</td><td rowspan="4">BIN-16-Bit</td></tr>
<tr><td>(s2)+6</td><td>Port-Nr. der Ziel-Station</td><td>Port-Nummer der Partnerstation, mit der kommuniziert wird.
($FFFF_H$ = Broadcast)</td><td>401_H bis $FFFF_H$</td></tr>
<tr><td>(s2)+7
bis
(s2)+9</td><td>Ethernet-Adresse der Ziel-Station</td><td>Tragen Sie bei einer Partnerstation mit ARP-Funktion als Wert entweder 000000000000_H oder $FFFFFFFFFFFF_H$ ein. Bei einer Partnerstation ohne ARP-Funktion geben Sie die individuelle ETHERNET-Adresse an.</td><td>siehe nebenstehende Erläuterung</td></tr>
<tr><td rowspan="4">d1</td><td colspan="6">Bit-Operand, der nach der Ausführung der OPEN-Anweisung für einen Zyklus gesetzt wird. Mit dem folgenden Operanden (d1)+1 wird die fehlerhafte Beendigung signalisiert.</td></tr>
<tr><td>Operand</td><td>Bedeutung</td><td>Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(d1)+0</td><td>Anweisung ausgeführt</td><td>Zeigt die Beendigung der OPEN-Anweisung an.
EIN: Anweisung ausgeführt
AUS: Anweisung nicht ausgeführt</td><td>—</td><td rowspan="2">System</td><td rowspan="2">Bit</td></tr>
<tr><td>(d1)+1</td><td>Anweisung mit Fehler ausgeführt</td><td>Zeigt an, ob bei der Ausführung der OPEN-Anweisung ein Fehler aufgetreten ist.
EIN: Anweisung mit Fehler ausgeführt
AUS: Anweisung ohne Fehler ausgeführt</td><td>—</td></tr>
</table>

Funktions-weise

Verbindung aufbauen

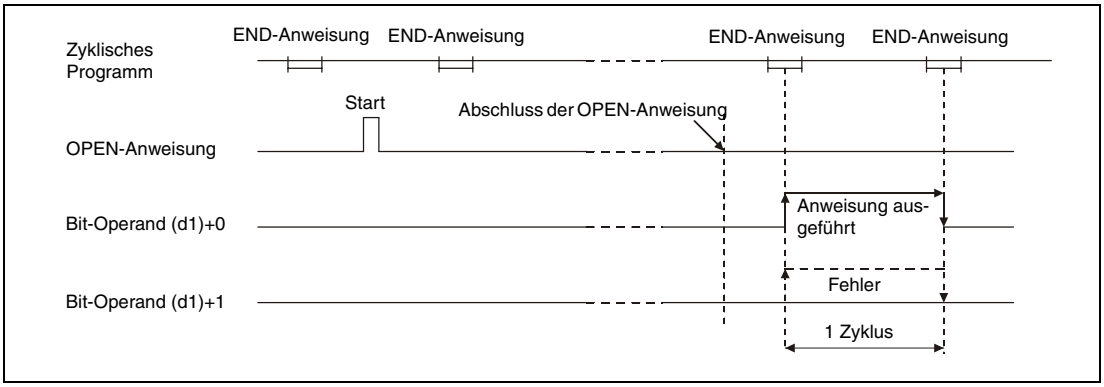
OPEN Verbindung öffnen

Mit dieser Anweisung kann die in s1 angegebene Verbindung des ETHERNET-Moduls mit der Kopfadresse Un geöffnet (aufgebaut) werden.

Die Ausführung der OPEN-Anweisung kann mit den Bit-Operanden (d1)+0 und (d1)+1 überprüft werden:

- Der Bit-Operand (d1)+0 wird gesetzt, wenn die END-Anweisung des Zyklus ausgeführt wird, in dem die OPEN-Anweisung beendet wurde. Bei der nächsten Bearbeitung der END-Anweisung wird dieser Bit-Operand wieder zurückgesetzt.
- Der Bit-Operand (d1)+1 zeigt einen Fehler bei der Ausführung der OPEN-Anweisung an. Bei fehlerfreier Ausführung bleibt dieser Bit-Operand zurückgesetzt. Bei einem Fehler jedoch wird (d1)+1 gesetzt, wenn die END-Anweisung des Zyklus ausgeführt wird, in dem die OPEN-Anweisung beendet wurde. Bei der nächsten Bearbeitung der END-Anweisung wird der in (d1)+1 angegebene Bit-Operand wieder zurückgesetzt.

Die folgende Abbildung zeigt den Signalverlauf bei Ausführung der OPEN-Anweisung:



Die OPEN-Anweisung wird ausgeführt, wenn die Eingangsbedingung der Anweisung erfüllt ist.

HINWEIS *Öffnen oder schließen Sie eine Verbindung nicht gleichzeitig über die E/A-Signale und die Anweisungen OPEN bzw. CLOSE. Dies kann es zu Fehlfunktionen des Moduls führen.*

Fehler-quellen

Wenn die OPEN-Anweisung fehlerhaft ausgeführt wurde, wird der Operand (d1)+1 gesetzt und in (s2)+1 wird ein Fehlercode gespeichert. Detaillierte Angaben zu den Fehlercodes entnehmen Sie bitte den folgenden Anleitungen:

- Bei einem Fehlercode bis $4FFF_H$ finden Sie Hinweise zur Fehlerbehebung in diesem Handbuch (Kapitel 13).
- Bei einem Fehlercode ab $C001_H$ finden Sie detaillierte Angaben in der Bedienungsanleitung zu den ETHERNET-Modulen des System Q.

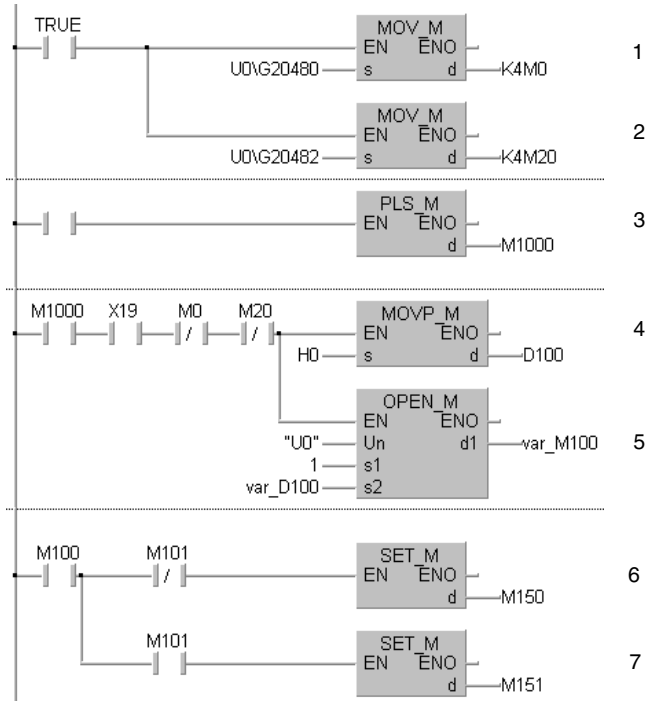
Beispiel OPEN

Das folgende Programm öffnet die Verbindung 1 des ETHERNET-Moduls mit der Kopfadresse X/Y0 für TCP/IP-Kommunikation.

HINWEIS *Bei den IEC-Editoren des GX IEC Developers müssen die Variablen im Header der Programmorganisationseinheit (POE) definiert werden. Ohne Variablendefinition werden beim Überprüfen oder Kompilieren des Programms Fehler gemeldet.*
Weitere Informationen finden Sie im Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung.

- Kontaktplan (GX IEC Developer)
  Im folgenden Beispiel müssen die Parameter vorab mit einem Programmierwerkzeug eingestellt werden. Ein Beispiel, bei dem die Parameter mit der OPEN-Anweisung übergeben werden, finden Sie auf der nächsten Seite.

Parametervorgabe durch GX Developer / GX IEC Developer



[1] Verbindungsstatus lesen (M0 = 1: Aufbau von Verbindung 1 abgeschlossen)

[2] Anforderungen zum Verbindungsaufbau lesen (M20 = 1: Aufbau von Verbindung 1 angefordert)

[3] Aus dem Signal zum Öffnen der Verbindung wird ein Impuls gebildet.

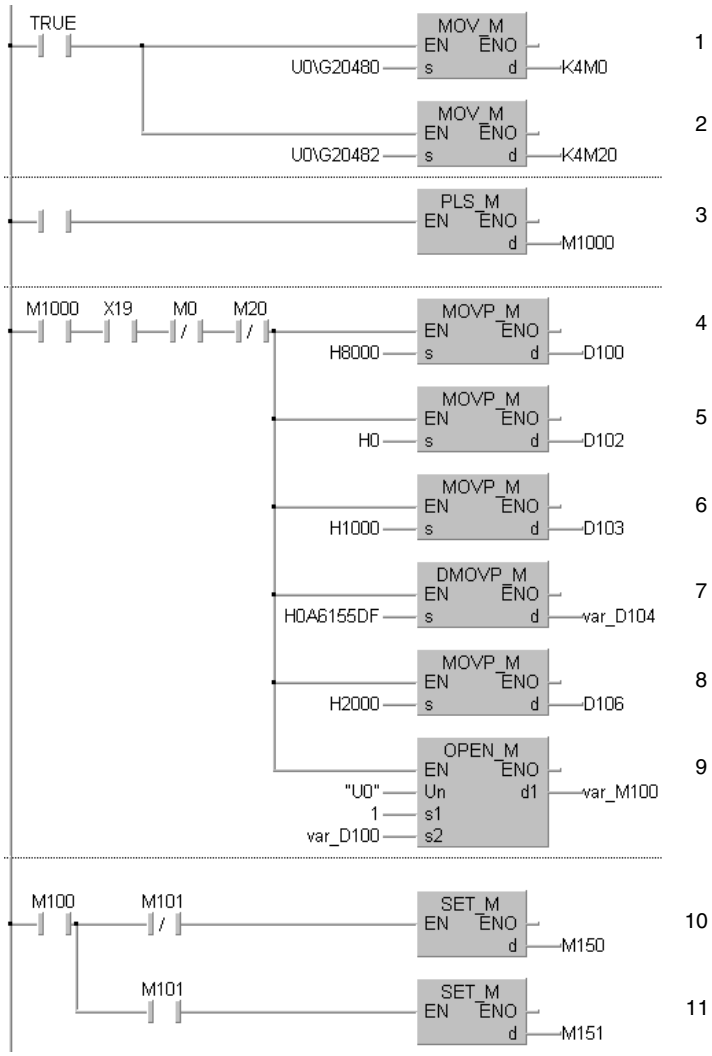
[4] Parameterquelle angeben (0000_H = Externe Vorgabe)

[5] Verbindung 1 öffnen

[6] M150 wird gesetzt, wenn die Verbindung ohne Fehler geöffnet wurde.

[7] M151 wird gesetzt, wenn beim Öffnen der Verbindung ein Fehler aufgetreten ist.

Angabe der Parameter in den Operanden (s2)+2 bis (s2)+9



[1] Verbindungsstatus lesen (M0 = 1: Aufbau von Verbindung 1 abgeschlossen)

[2] Anforderungen zum Verbindungsaufbau lesen (M20 = 1: Aufbau von Verbindung 1 angefordert)

[3] Aus dem Signal zum Öffnen der Verbindung wird ein Impuls gebildet.

[4] Parameterquelle angeben (8000_H = Parameter in Operanden (s2)+2 bis (s2)+9))

[5] Verbindungsart in (s2)+2 eintragen

[6] Port-Nr. des ETHERNET-Moduls in (s2)+3 eintragen

[7] IP-Adresse (10.97.85.223) der Partnerstation in (s2)+4 und (s2)+5 eintragen

[8] Port-Nr. der Partnerstation in (s2)+6 eintragen

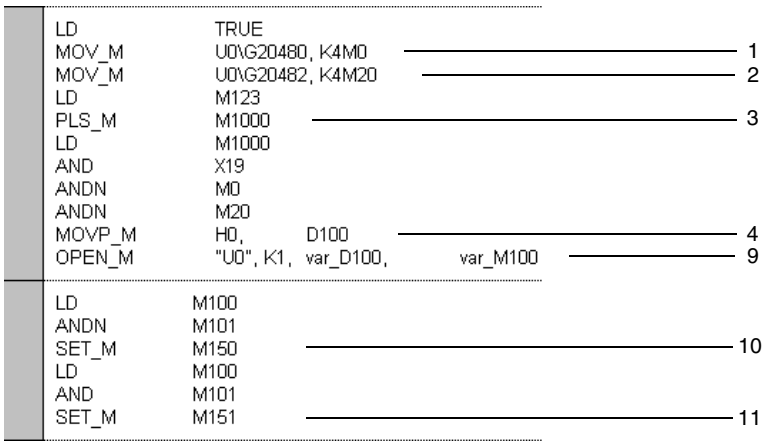
[9] Verbindung 1 öffnen

[10] M150 wird gesetzt, wenn die Verbindung ohne Fehler geöffnet wurde.

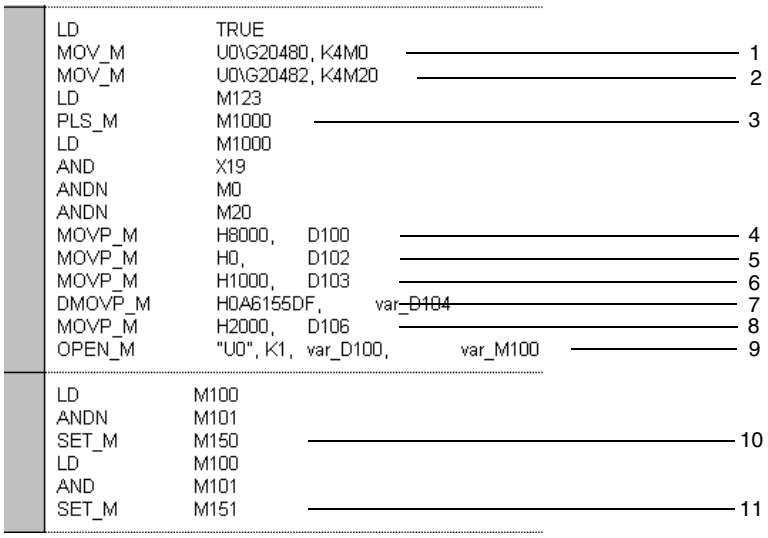
[11] M151 wird gesetzt, wenn beim Öffnen der Verbindung ein Fehler aufgetreten ist.

- IEC-Anweisungliste

Parametervorgabe durch GX Developer / GX IEC Developer



Angabe der Parameter in den Operanden (s2)+2 bis (s2)+9



[1] Verbindungsstatus lesen (M0 = 1: Aufbau von Verbindung 1 abgeschlossen)

[2] Anforderungen zum Verbindungsaufbau lesen (M20 = 1: Aufbau von Verbindung 1 angefordert)

[3] Aus dem Signal zum Öffnen der Verbindung wird ein Impuls gebildet.

[4] Parameterquelle angeben (0000_H = Extern, 8000_H = Operanden (s2)+2 bis (s2)+9))

[5] Verbindungsart in (s2)+2 eintragen

[6] Port-Nr. des ETHERNET-Moduls in (s2)+3 eintragen

[7] IP-Adresse (10.97.85.223) der Partnerstation in (s2)+4 und (s2)+5 eintragen

[8] Port-Nr. der Partnerstation in (s2)+6 eintragen

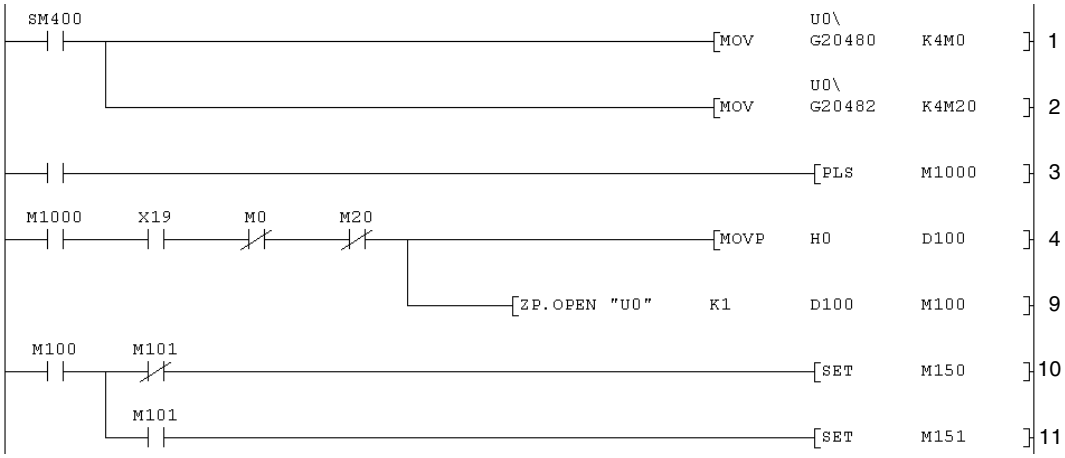
[9] Verbindung 1 öffnen

[10] M150 wird gesetzt, wenn die Verbindung ohne Fehler geöffnet wurde.

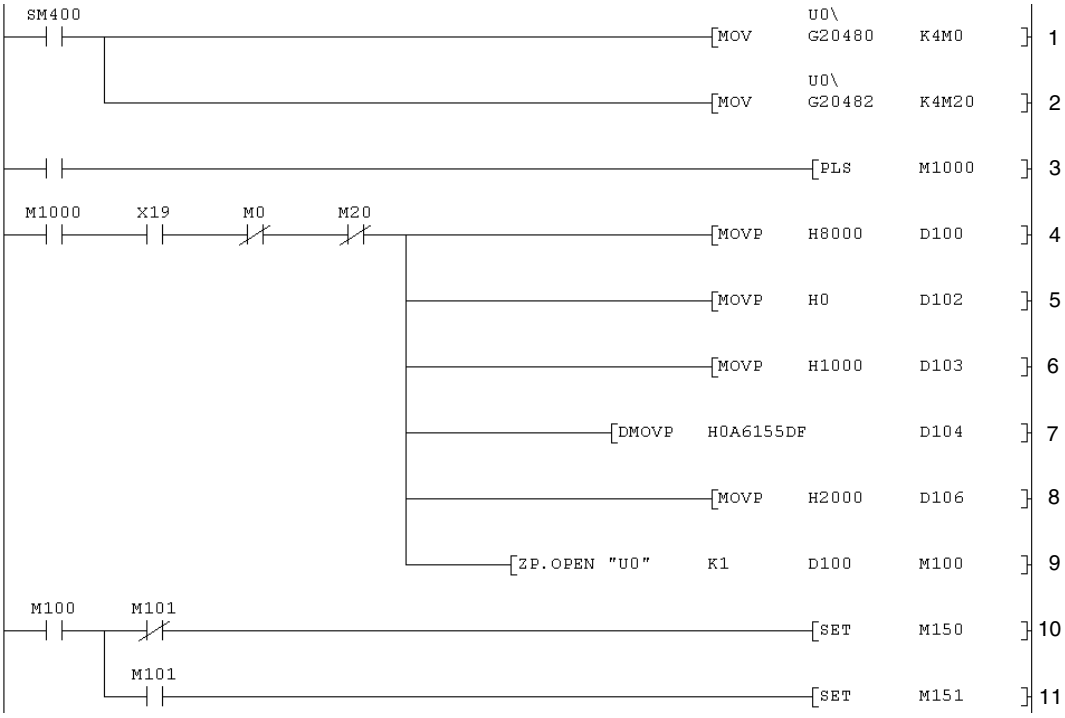
[11] M151 wird gesetzt, wenn beim Öffnen der Verbindung ein Fehler aufgetreten ist.

- Kontaktplan (GX Developer)

Parametervorgabe durch GX Developer / GX IEC Developer



Parameter werden in den Operanden (s2)+2 bis (s2)+9 angegeben



[1] Verbindungsstatus lesen (M0 = 1: Aufbau von Verbindung 1 abgeschlossen)

[2] Anforderungen zum Verbindungsaufbau lesen (M20 = 1: Aufbau von Verbindung 1 angefordert)

[3] Aus dem Signal zum Öffnen der Verbindung wird ein Impuls gebildet.

[4] Parameterquelle angeben (0000_H = Extern, 8000_H = Operanden (s2)+2 bis (s2)+9))

[5] Verbindungsart in (s2)+2 eintragen

[6] Port-Nr. des ETHERNET-Moduls in (s2)+3 eintragen

[7] IP-Adresse (10.97.85.223) der Partnerstation in (s2)+4 und (s2)+5 eintragen.

[8] Port-Nr. der Partnerstation in (s2)+6 eintragen

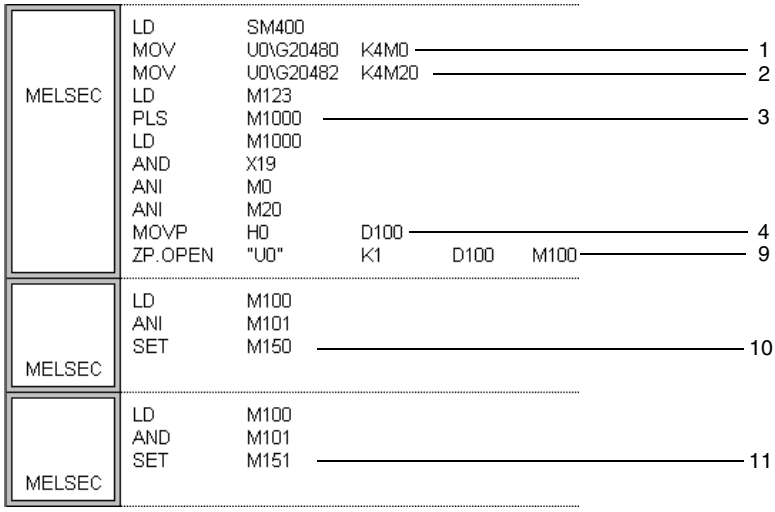
[9] Verbindung 1 öffnen

[10] M150 wird gesetzt, wenn die Verbindung ohne Fehler geöffnet wurde.

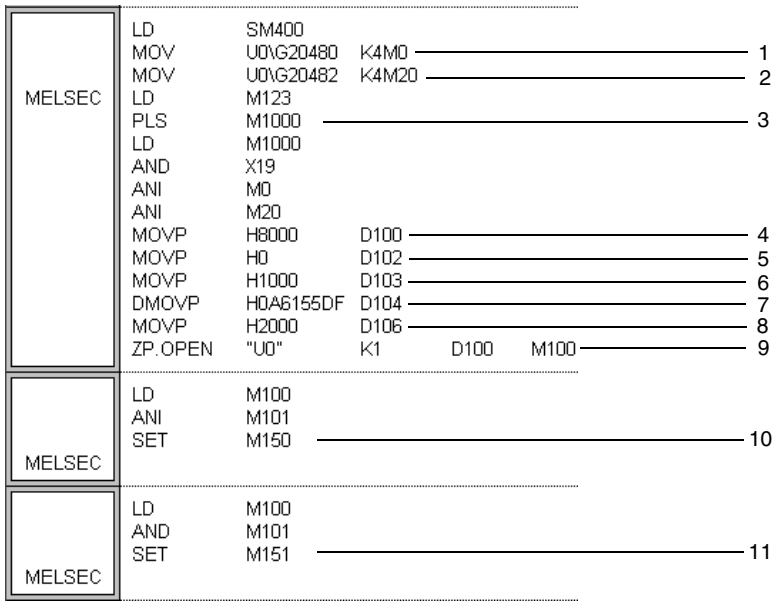
[11] M151 wird gesetzt, wenn beim Öffnen der Verbindung ein Fehler aufgetreten ist.

- MELSEC-Anweisungliste

Parametervorgabe durch GX Developer / GX IEC Developer



Parameter werden in den Operanden (s2)+2 bis (s2)+9 angegeben



[1] Verbindungsstatus lesen (M0 = 1: Aufbau von Verbindung 1 abgeschlossen)

[2] Anforderungen zum Verbindungsaufbau lesen (M20 = 1: Aufbau von Verbindung 1 angefordert)

[3] Aus dem Signal zum Öffnen der Verbindung wird ein Impuls gebildet.

[4] Parameterquelle angeben (0000_H = Extern, 8000_H = Operanden (s2)+2 bis (s2)+9)

[5] Verbindungsart in (s2)+2 eintragen

[6] Port-Nr. des ETHERNET-Moduls in (s2)+3 eintragen

[7] IP-Adresse (10.97.85.223) der Partnerstation in (s2)+4 und (s2)+5 eintragen

[8] Port-Nr. der Partnerstation in (s2)+6 eintragen

[9] Verbindung 1 öffnen

[10] M150 wird gesetzt, wenn die Verbindung ohne Fehler geöffnet wurde.

[11] M151 wird gesetzt, wenn beim Öffnen der Verbindung ein Fehler aufgetreten ist.

11.3.5 CLOSE

CPU

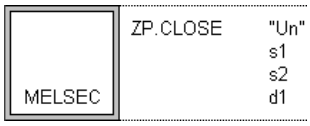
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					●

Operanden MELSEC Q

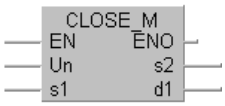
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□N□		Sondermodule U□NG□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	—	●	●	—	—	—	—	●	—		
s2	—	●	●	—	—	—	—	—	—		
d1	●	●	●	—	—	—	—	—	—		

GX IEC Developer

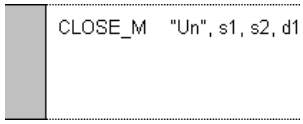
MELSEC-Anweisungsliste



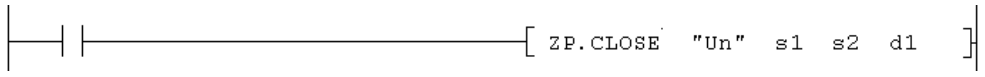
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

<table>
<tr><th>Operand</th><th colspan="4">Bedeutung</th><th>Werte-bereich</th><th>Festlegung des Inhalts durch</th><th>Datentyp</th></tr>
<tr><td>„Un“</td><td colspan="4">Kopfadresse des ETHERNET-Moduls auf dem Baugruppenträger (Es werden nur die ersten beiden Stellen der 3-stelligen Adresse angegeben, z. B. wird die Kopfadresse X/Y100 als „U10“ eingetragen.)</td><td>0 bis FE_H</td><td rowspan="2">Anwender</td><td rowspan="2">BIN-16-Bit</td></tr>
<tr><td>s1</td><td colspan="4">Nummer der Verbindung</td><td>1 bis 16</td></tr>
<tr><td rowspan="4">s2</td><td colspan="6">Erster Operand des Bereiches mit Informationen zur Ausführung der Anweisung</td></tr>
<tr><td>Operand</td><td>Bedeutung</td><td colspan="2">Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(s2)+0</td><td>Systembereich</td><td colspan="2">Wird vom System verwendet</td><td rowspan="2">—</td><td rowspan="2">System</td></tr>
<tr><td>(s2)+1</td><td>Ausführungs-status der Anweisung</td><td colspan="2">Zeigt an, ob bei der Bearbeitung der Anweisung ein Fehler aufgetreten ist:
0000_H: Fehlerfreie Bearbeitung
Jeder andere Wert als 0000_H: Bei der Bearbeitung ist ein Fehler aufgetreten. Der eingetragene Wert ist ein Fehlercode, der entweder in der Bedienungsanleitung des ETHERNET-Moduls oder in Kapitel 13 dieses Handbuchs beschrieben ist.</td></tr>
<tr><td rowspan="4">d1</td><td colspan="6">Bit-Operand, der nach der Ausführung der CLOSE-Anweisung für einen Zyklus gesetzt wird. Mit dem folgenden Operanden (d1)+1 wird die fehlerhafte Beendigung signalisiert.</td></tr>
<tr><td>Operand</td><td>Bedeutung</td><td colspan="2">Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(d1)+0</td><td>Anweisung ausgeführt</td><td colspan="2">Zeigt die Beendigung der CLOSE-Anweisung an.
EIN: Anweisung ausgeführt
AUS: Anweisung nicht ausgeführt</td><td>—</td><td rowspan="2">System</td></tr>
<tr><td>(d1)+1</td><td>Anweisung mit Fehler ausgeführt</td><td colspan="2">Zeigt an, ob bei der Ausführung der CLOSE-Anweisung ein Fehler aufgetreten ist.
EIN: Anweisung mit Fehler ausgeführt
AUS: Anweisung ohne Fehler ausgeführt</td><td>—</td></tr>
</table>

Funktions-weise

Verbindung schließen

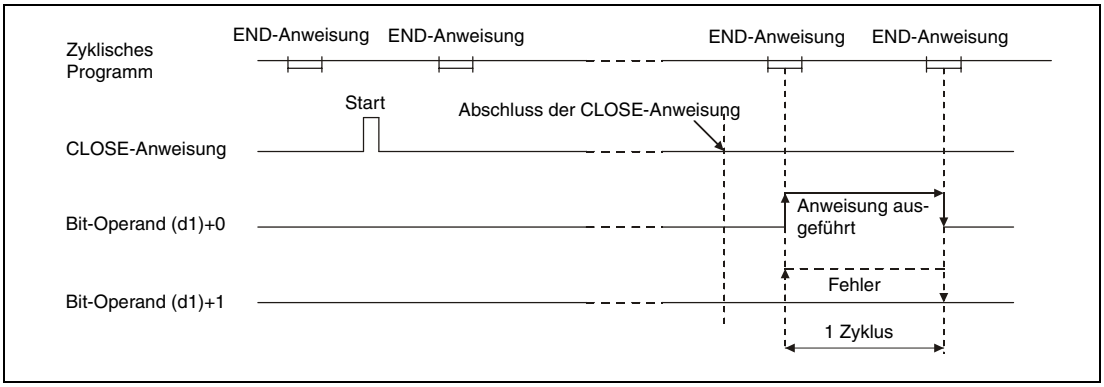
CLOSE Verbindung schließen

Mit dieser Anweisung kann die in s1 angegebene Verbindung des ETHERNET-Moduls mit der Kopfadresse Un geschlossen werden. Die Verbindung wird abgebaut.

Ob die Ausführung der CLOSE-Anweisung beendet ist, kann anhand der Bit-Operanden (d1)+0 und (d1)+1 überprüft werden:

- Der Bit-Operand (d1)+0 wird gesetzt, wenn die END-Anweisung des Zyklus ausgeführt wird, in dem die CLOSE-Anweisung beendet wurde. Bei der nächsten Bearbeitung der END-Anweisung wird dieser Bit-Operand wieder zurückgesetzt.
- Der Bit-Operand (d1)+1 zeigt einen Fehler bei der Ausführung der CLOSE-Anweisung an. Bei fehlerfreier Ausführung bleibt dieser Bit-Operand zurückgesetzt. Bei einem Fehler jedoch wird (d1)+1 gesetzt, wenn die END-Anweisung des Zyklus ausgeführt wird, in dem die CLOSE-Anweisung beendet wurde. Bei der nächsten Bearbeitung der END-Anweisung wird der in (d1)+1 angegebene Bit-Operand wieder zurückgesetzt.

Die folgende Abbildung zeigt den Signalverlauf bei Ausführung der CLOSE-Anweisung:



Die CLOSE-Anweisung wird ausgeführt, wenn die Eingangsbedingung der Anweisung erfüllt ist.

HINWEIS *Öffnen oder Schließen Sie eine Verbindung nicht gleichzeitig über die E/A-Signale und die Anweisungen OPEN bzw. CLOSE. Dies kann zu Fehlfunktionen des Moduls führen.*

Fehler-quellen

Wenn die CLOSE-Anweisung fehlerhaft ausgeführt wurde, wird der Operand (d1)+1 gesetzt und ein Fehlercode in (s2)+1 gespeichert. Detaillierte Angaben zu den Fehlercodes entnehmen Sie bitte den folgenden Anleitungen:

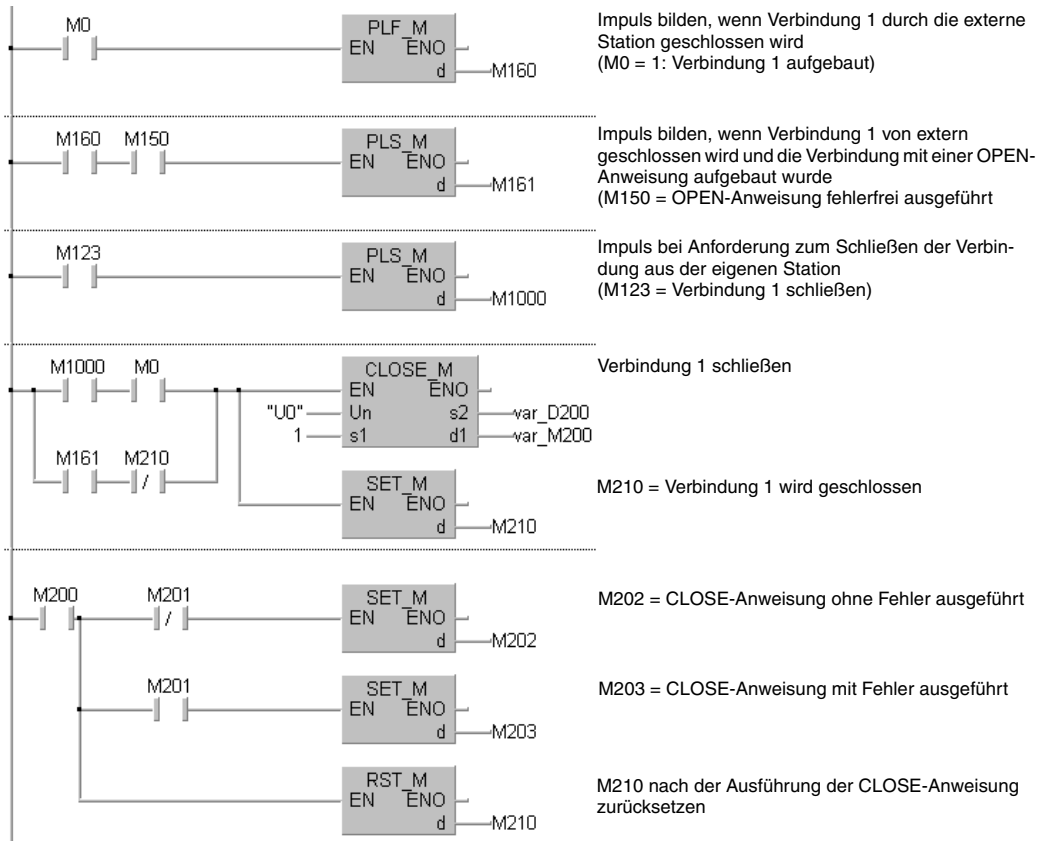
- Bei einem Fehlercode bis $4FFF_H$ finden Sie Hinweise zur Fehlerbehebung in diesem Handbuch (Kapitel 13).
- Bei einem Fehlercode ab $C001_H$ finden Sie detaillierte Angaben in der Bedienungsanleitung zu den ETHERNET-Modulen des System Q.

Beispiel CLOSE

Das folgende Programm schließt die Verbindung 1 des ETHERNET-Moduls mit der Kopfadresse X/Y0.

- IEC-Editoren (Auf der nächsten Seite ist dieses Beispiel in der MELSEC-Anweisungliste und dem Kontaktplan des GX Developers dargestellt.)

Kontaktplan (GX IEC Developer)



IEC-Anweisungsliste

```
LD      M0
PLF_M   M160
LD      M160
AND     M150
PLS_M   M161
LD      M123
PLS_M   M1000
LD      M1000
AND     M0
OR(
        M161
ANDN    M210
)
CLOSE_M "U0", K1, var_D200, var_M200
SET_M   M210
```

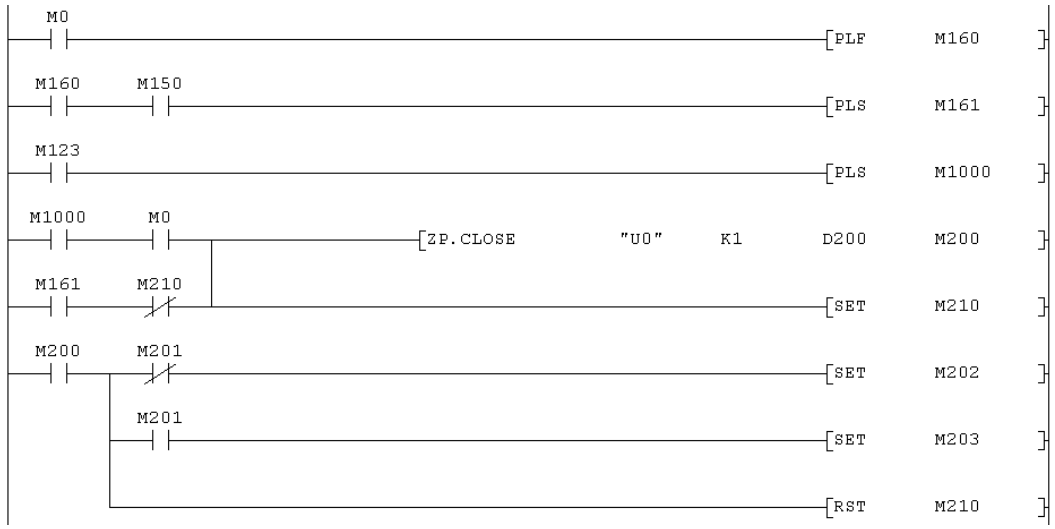
```
LD      M201
ANDN    M201
SET_M   M202
LD      M201
AND     M202
SET_M   M203
LD      M200
RST_M   M210
```

Die Operanden und Anweisungen sind oben beim Programmbeispiel für den Kontaktplan erläutert.

HINWEIS *Bei den IEC-Editoren des GX IEC Developers müssen die Variablen im Header der Programmorganisationseinheit (POE) definiert werden. Ohne Variablendefinition werden beim Überprüfen oder Kompilieren des Programms Fehler gemeldet.*
Weitere Informationen finden Sie im Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung.

- MELSEC-Anweisungliste und Kontaktplan des GX Developers
  Die Operanden und Anweisungen sind auf der vorherigen Seite beim Programmbeispiel für den Kontaktplan des GX IEC Developers erläutert.

Kontaktplan (GX Developer)



MELSEC-Anweisungsliste

```
MELSEC  LD      M0
        PLF     M160
        LD      M160
        AND     M150
        PLS     M161
        LD      M123
        PLS     M1000
        LD      M1000
        AND     M0
        LD      M161
        ANI     M210
        ORB
        ZP.CLOSE "U0"   K1      D200    M200
        SET     M210

MELSEC  LD      M200
        MPS
        ANI     M201
        SET     M202
        MRD
        AND     M201
        SET     M203
        MPP
        RST     M210
```

11.3.6 ERRCLR

CPU

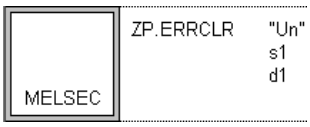
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					●

Operanden MELSEC Q

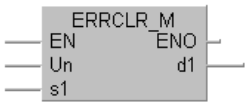
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□N□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	—	●	●	—	—	—	—	—	—	SM0	
d1	●	●	●	—	—	—	—	—	—		

GX IEC Developer

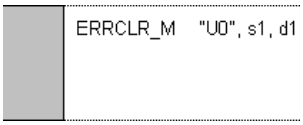
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

<table>
<tr><th>Operand</th><th colspan="4">Bedeutung</th><th>Werte-bereich</th><th>Festlegung des Inhalts durch</th><th>Datentyp</th></tr>
<tr><td>„Un“</td><td colspan="4">Kopfadresse des ETHERNET-Moduls auf dem Baugruppenträger (Es werden nur die ersten beiden Stellen der 3-stelligen Adresse angegeben, z. B. wird die Kopfadresse X/Y100 als „U10“ eingetragen.)</td><td>0 bis FE_H</td><td>Anwender</td><td>BIN-16-Bit</td></tr>
<tr><td rowspan="9">s1</td><td colspan="4">Erster Operand des Bereiches mit Informationen zur Ausführung der Anweisung</td><td></td><td></td><td rowspan="9">BIN-16-Bit</td></tr>
<tr><td>Operand</td><td colspan="2">Bedeutung</td><td>Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(s1)+0</td><td colspan="2">Systembereich</td><td>Wird vom System verwendet</td><td rowspan="2">—</td><td rowspan="2">System</td></tr>
<tr><td>(s1)+1</td><td colspan="2">Ausführungs-status der Anweisung</td><td>Zeigt an, ob bei der Bearbeitung der Anweisung ein Fehler aufgetreten ist:
0000_H: Fehlerfreie Bearbeitung
Jeder andere Wert als 0000_H: Bei der Bearbeitung ist ein Fehler aufgetreten. Der eingetragene Wert ist ein Fehlercode, der entweder in der Bedienungsanleitung des ETHERNET-Moduls oder in Kapitel 13 dieses Handbuchs beschrieben ist.</td></tr>
<tr><td>(s1)+2</td><td colspan="2">Zu löschender Fehler</td><td>Abhängig vom eingetragenen Wert wird ein im Pufferspeicher des Moduls gespeicherter Fehlercode gelöscht und/oder die LED „ERR.“ des ETHERNET-Moduls gelöscht. Eine ausführliche Beschreibung finden Sie weiter unten.</td><td>0000_H
0001_H bis 0016_H
0100_H
0101_H
0102_H
0103_H
$FFFF_H$</td><td rowspan="2">Anwender</td></tr>
<tr><td>(s1)+3</td><td colspan="2">Funktion</td><td>Wählen Sie zwischen dem Ausschalten der LED oder dem Löschen des Fehlercodes.
Die Funktionen sind weiter unten detailliert beschrieben.</td><td>0000_H
oder
$FFFF_H$</td></tr>
<tr><td>(s1)+4 bis (s1)+7</td><td colspan="2">Systembereich</td><td>Wird vom System verwendet</td><td>—</td><td>System</td></tr>
<tr><td rowspan="4">d1</td><td colspan="4">Bit-Operand, der nach der Ausführung der ERRCLR-Anweisung für einen Zyklus gesetzt wird. Mit (d1)+1 wird ein Fehler bei der Ausführung angezeigt.</td><td></td><td></td><td rowspan="4">Bit</td></tr>
<tr><td>Operand</td><td colspan="2">Bedeutung</td><td>Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(d1)+0</td><td colspan="2">Anweisung ausgeführt</td><td>Zeigt die Beendigung der ERRCLR-Anweisung an.
EIN: Anweisung ausgeführt
AUS: Anweisung nicht ausgeführt</td><td>—</td><td rowspan="2">System</td></tr>
<tr><td>(d1)+1</td><td colspan="2">Anweisung mit Fehler ausgeführt</td><td>Zeigt an, ob bei der Ausführung der ERRCLR-Anweisung ein Fehler aufgetreten ist.
EIN: Anweisung mit Fehler ausgeführt
AUS: Anweisung ohne Fehler ausgeführt</td><td>—</td></tr>
</table>

Funktionsweise

Fehler im Pufferspeicher löschen und „ERR.“-LED ausschalten

ERRCLR Fehler löschen

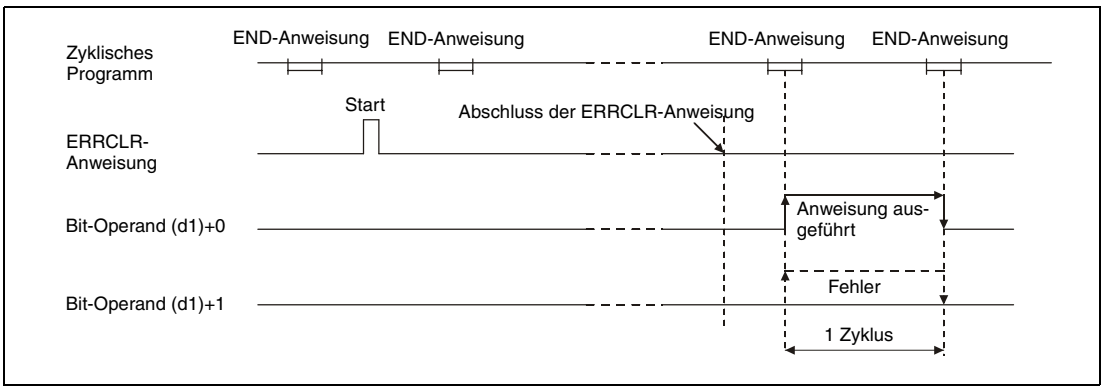
Die ERRCLR-Anweisung löscht einen im Pufferspeicher des ETHERNET-Moduls gespeicherten Fehlercode. Falls die „ERR.“-Leuchtdiode an der Vorderseite des Moduls leuchtet, wird diese LED nach Ausführung der ERRCLR-Anweisung ausgeschaltet. Zusätzlich können mit dieser Anweisung die Pufferspeicherbereiche, in denen der Status der Kommunikation abgelegt ist, gelöscht werden.

Der Inhalt der Operanden (s1)+2 und (s1)+3 bestimmt den Bereich im Pufferspeicher, der gelöscht wird:

Fehler oder Statusbereich		Inhalt der Operanden		Ausgeführte Aktion
		(s1)+2	(s2)+3	
Fehler beim Anlauf des Moduls		0000_H	0000_H	• Die Pufferspeicheradresse 69_H wird gelöscht. • Die „ERR.“-LED wird ausgeschaltet.
Fehler beim Öffnen einer Verbindung		0001_H bis 0016_H (Nummer der Verbindung)		• Löschen der Pufferspeicheradresse, in der für eine Verbindung ein Fehlercode eingetragen wird ($7C_H$, 86_H ...). • Die „ERR.“-LED wird ausgeschaltet.
Fehlerspeicher		0100_H	$FFFF_H$	• Der Fehlerspeicher (Adressen $E3_H$ bis 174_H) wird gelöscht.
Kommunikationsstatus	Status der Protokolle	0101_H		• Der Pufferspeicherbereich von 178_H bis $1FF_H$ wird gelöscht.
	Status des E-Mail-Empfangs	0102_H		• Löschen des Pufferspeicherbereichs von 5871_H bis $5B38_H$.
	Status der E-Mail-Sendung	0103_H		• Löschen des Pufferspeicherbereichs von $5B39_H$ bis $5CA0_H$.
Alle gespeicherten Fehler oder Status-Bereiche		$FFFF_H$		• Alle zuvor genannten Pufferspeicheradressen werden gelöscht. • Die „ERR.“-LED wird ausgeschaltet.

Die Ausführung der ERRCLR-Anweisung kann mit den Bit-Operanden (d1)+0 und (d1)+1 überprüft werden:

- Der Bit-Operand (d1)+0 wird gesetzt, wenn die END-Anweisung des Zyklus ausgeführt wird, in dem die ERRCLR-Anweisung beendet wurde. Bei der nächsten Bearbeitung der END-Anweisung wird der in (d1)+0 angegebene Bit-Operand wieder zurückgesetzt.
- Der Bit-Operand (d1)+1 zeigt einen Fehler bei der Ausführung der ERRCLR-Anweisung an. Bei fehlerfreier Ausführung bleibt dieser Bit-Operand zurückgesetzt. Bei einem Fehler jedoch wird (d1)+1 gesetzt, wenn die END-Anweisung des Zyklus ausgeführt wird, in dem die ERRCLR-Anweisung beendet wurde. Bei der nächsten Bearbeitung der END-Anweisung wird dieser Operand wieder zurückgesetzt.



Fehler-quellen

Wenn bei der Ausführung der ERRCRL-Anweisung ein Fehler auftritt, wird der Operand (d1)+1 gesetzt und ein Fehlercode in (s1)+1 gespeichert. Detaillierte Angaben zu den Fehlercodes entnehmen Sie bitte den folgenden Anleitungen:

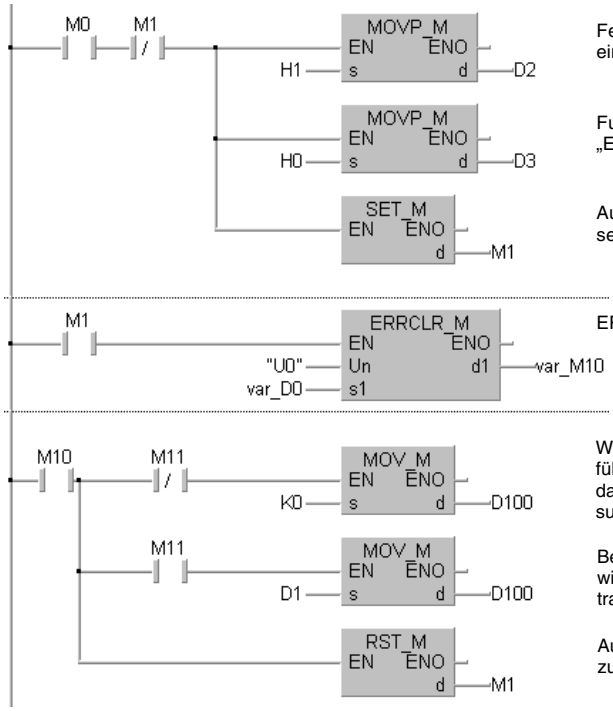
- Bei einem Fehlercode bis $4FFF_H$ finden Sie Hinweise zur Fehlerbehebung in diesem Handbuch (Kapitel 13).
- Bei einem Fehlercode ab $C001_H$ finden Sie detaillierte Angaben in der Bedienungsanleitung zu den ETHERNET-Modulen des System Q.

Beispiel ERRCLR

Mit dem folgenden Programm wird ein für Verbindung 1 eingetragener Fehlercode gelöscht. Das ETHERNET-Modul hat die Kopfadresse X/Y0.

- IEC-Editoren (Auf der nächsten Seite ist dieses Beispiel in der MELSEC-Anweisungliste und dem Kontaktplan des GX Developers dargestellt.)

Kontaktplan (GX IEC Developer)



IEC-Anweisungsliste

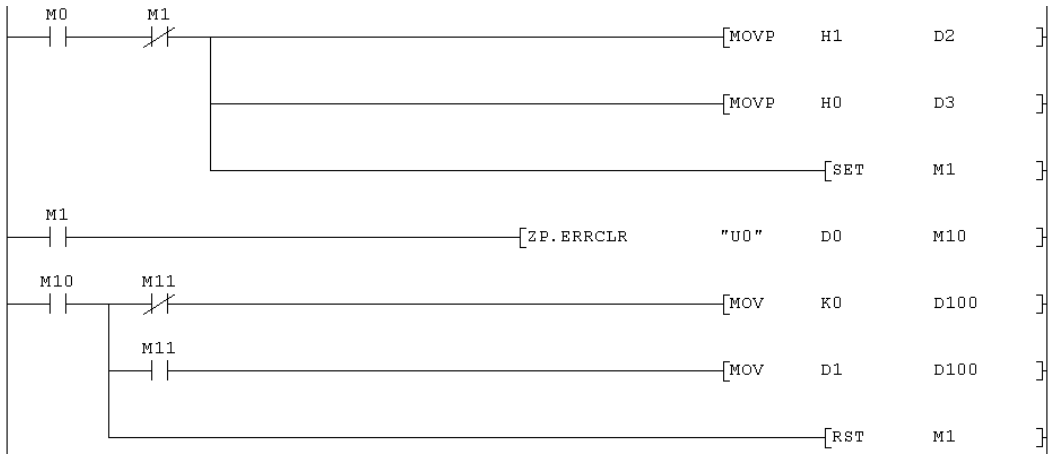
```
LD          M0
ANDN        M1
MOVP_M      H1,     D2
MOVP_M      H0,     D3
SET_M       M1
LD          M1
ERRCLR_M    "U0",   var_D0,   var_M10
LD          M10
ANDN        M11
MOV_M       K0,     D100
LD          M10
AND         M11
MOV_M       D1,     D100
LD          M10
RST_M       M1
```

Die Operanden und Anweisungen sind oben beim Programmbeispiel für den Kontaktplan erläutert.

HINWEIS *Bei den IEC-Editoren des GX IEC Developers müssen die Variablen im Header der Programmorganisationseinheit (POE) definiert werden. Ohne Variablendefinition werden beim Überprüfen oder Kompilieren des Programms Fehler gemeldet.*
Weitere Informationen finden Sie im Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer" dieser Programmieranleitung.

- MELSEC-Anweisungliste und Kontaktplan des GX Developers
  Die Operanden und Anweisungen sind auf der vorherigen Seite beim Programmbeispiel für den Kontaktplan des GX IEC Developers erläutert.

Kontaktplan (GX Developer)



MELSEC-Anweisungsliste

```
MELSEC  LD        M0
        ANI       M1
        MOVP      H1        D2
        MOVP      H0        D3
        SET       M1
        LD        M1
        ZP.ERRCLR "U0"      D0        M10

MELSEC  LD        M10
        MPS
        ANI       M11
        MOV       K0              D100
        MRD
        AND       M11
        MOV       D1              D100
        MPP
        RST       M1
```

11.3.7 ERRRD

CPU

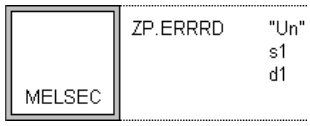
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					●

Operanden MELSEC Q

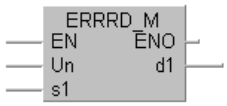
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□N□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	—	●	●	—	—	—	—	—	—	SM0	
d1	●	●	●	—	—	—	—	—	—		

GX IEC Developer

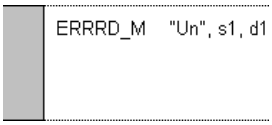
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

<table>
<tr><th>Operand</th><th colspan="4">Bedeutung</th><th>Werte-bereich</th><th>Festlegung des Inhalts durch</th><th>Datentyp</th></tr>
<tr><td>„Un“</td><td colspan="4">Kopfadresse des ETHERNET-Moduls auf dem Baugruppenträger (Es werden nur die ersten beiden Stellen der 3-stelligen Adresse angegeben, z. B. wird die Kopfadresse X/Y100 als „U10“ eingetragen.)</td><td>0 bis FE_H</td><td>Anwender</td><td>BIN-16-Bit</td></tr>
<tr><td rowspan="9">s1</td><td colspan="4">Erster Operand des Bereiches mit Informationen zur Ausführung der Anweisung</td><td></td><td></td><td rowspan="9">BIN-16-Bit</td></tr>
<tr><td>Operand</td><td>Bedeutung</td><td colspan="2">Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(s1)+0</td><td>Systembereich</td><td colspan="2">Wird vom System verwendet</td><td rowspan="2">—</td><td rowspan="2">System</td></tr>
<tr><td>(s1)+1</td><td>Ausführungs-status der Anweisung</td><td colspan="2">Zeigt an, ob bei der Bearbeitung der Anweisung ein Fehler aufgetreten ist:
0000_H: Fehlerfreie Bearbeitung
Jeder andere Wert als 0000_H: Bei der Bearbeitung ist ein Fehler aufgetreten. Der eingetragene Wert ist ein Fehlercode, der entweder in der Bedienungsanleitung des ETHERNET-Moduls oder in Kapitel 13 dieses Handbuchs beschrieben ist.</td></tr>
<tr><td>(s1)+2</td><td>Zu lesender Fehler Code</td><td colspan="2">Abhängig vom eingetragenen Wert wird ein im Pufferspeicher des ETHERNET-Moduls gespeicherter Fehlercode gelesen:
• 0000_H: Code, der beim fehlerhaften Anlauf des Moduls in die Pufferspeicheradr. 69_H eingetragen wird
• 0001_H bis 0016_H: Fehlercode für die jeweilige Verbindung (Pufferspeicheradr. $7C_H$, 86_H ...)</td><td>0000_H
0001_H
bis
0016_H</td><td>Anwender</td></tr>
<tr><td>(s1)+3</td><td>Funktion</td><td colspan="2">Lesen des zuletzt aufgetretenen Fehlercodes</td><td>0000_H</td></tr>
<tr><td>(s1)+4</td><td>Ausgelesener Fehlercode</td><td colspan="2">Enthält den vom ETHERNET-Modul übermittelten Fehlercode.
0000_H = Kein Fehler
Andere Werte als 0000_H: Fehlercode</td><td>—</td><td>System</td></tr>
<tr><td>(s1)+5
bis
(s1)+7</td><td>Systembereich</td><td colspan="2">Wird vom System verwendet</td><td>—</td><td>System</td></tr>
<tr><td rowspan="4">d1</td><td colspan="4">Bit-Operand, der nach der Ausführung der ERRRD-Anweisung für einen Zyklus gesetzt wird. Mit (d1)+1 wird ein Fehler bei der Ausführung angezeigt.</td><td></td><td></td><td rowspan="4">Bit</td></tr>
<tr><td>Operand</td><td>Bedeutung</td><td colspan="2">Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(d1)+0</td><td>Anweisung ausgeführt</td><td colspan="2">Zeigt die Beendigung der ERRRD-Anweisung an.
EIN: Anweisung ausgeführt
AUS: Anweisung nicht ausgeführt</td><td>—</td><td rowspan="2">System</td></tr>
<tr><td>(d1)+1</td><td>Anweisung mit Fehler ausgeführt</td><td colspan="2">Zeigt an, ob bei der Ausführung der ERRRD-Anweisung ein Fehler aufgetreten ist.
EIN: Anweisung mit Fehler ausgeführt
AUS: Anweisung ohne Fehler ausgeführt</td><td>—</td></tr>
</table>

Funktions-weise

Fehlercode aus ETHERNET-Modul lesen

ERRRD Fehlercode lesen

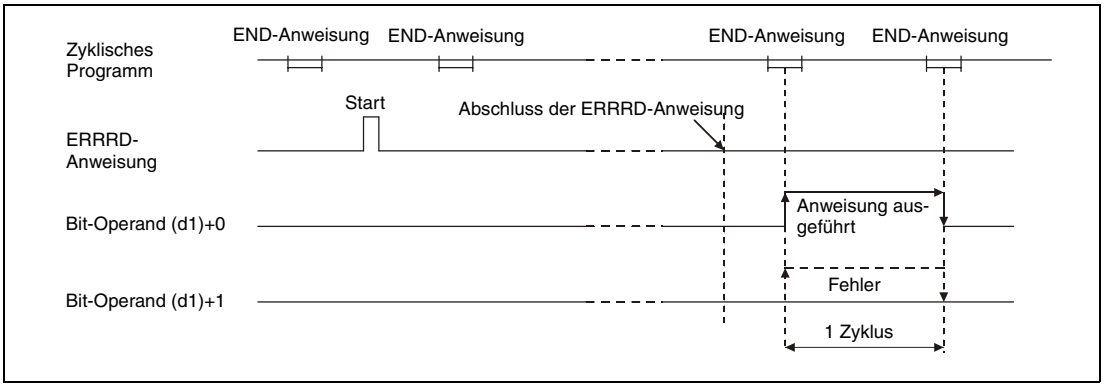
Die ERRCLR-Anweisung liest einen im Pufferspeicher eines ETHERNET-Moduls gespeicherten Fehlercode.

Im Operanden (s1)+2 wird angegeben, aus welcher Adresse des Pufferspeichers der Fehlercode gelesen werden soll. Mit „Un“ wird die Kopfadresse des ETHERNET-Moduls angegeben.

Die Ausführung der ERRRD-Anweisung kann mit den Bit-Operanden (d1)+0 und (d1)+1 überprüft werden:

- Der Bit-Operand (d1)+0 wird gesetzt, wenn die END-Anweisung des Zyklus ausgeführt wird, in dem die ERRRD-Anweisung beendet wurde. Bei der nächsten Bearbeitung der END-Anweisung wird dieser Bit-Operand wieder zurückgesetzt.
- Der Bit-Operand (d1)+1 zeigt einen Fehler bei der Ausführung der ERRRD-Anweisung an. Bei fehlerfreier Ausführung bleibt dieser Bit-Operand zurückgesetzt. Bei einem Fehler jedoch wird (d1)+1 gesetzt, wenn die END-Anweisung des Zyklus ausgeführt wird, in dem die ERRRD-Anweisung beendet wurde. Bei der nächsten Bearbeitung der END-Anweisung wird dieser Operand wieder zurückgesetzt.

Die folgende Abbildung zeigt den Signalverlauf bei Ausführung der ERRRD-Anweisung:



Fehler-quellen

Tritt bei der Ausführung der ERRRD-Anweisung ein Fehler auf, wird der Operand (d1)+1 gesetzt und ein Fehlercode in (s1)+1 gespeichert. Detaillierte Angaben zu den Fehlercodes entnehmen Sie bitte den folgenden Anleitungen:

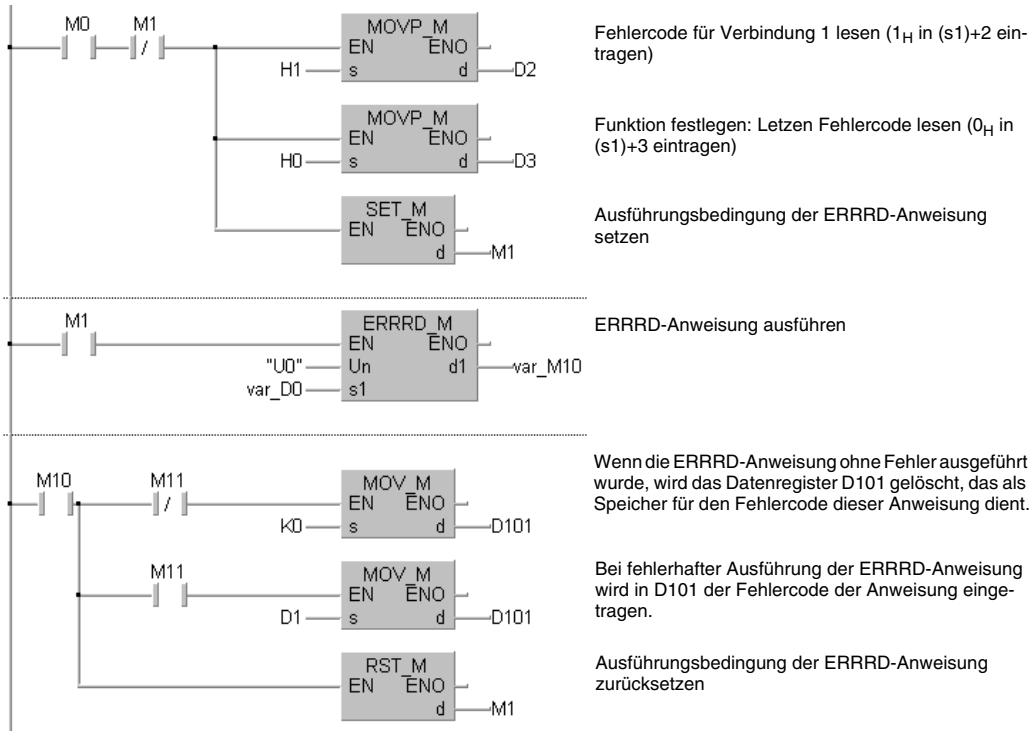
- Bei einem Fehlercode bis $4FFF_H$ finden Sie Hinweise zur Fehlerbehebung in diesem Handbuch (Kapitel 13).
- Bei einem Fehlercode ab $C001_H$ finden Sie detaillierte Angaben in der Bedienungsanleitung zu den ETHERNET-Modulen des System Q.

Beispiel ERRRD

Mit dem folgenden Programm wird der Fehlercode gelesen, der eingetragen wird, falls beim Öffnen von Verbindung 1 ein Fehler auftritt. Das ETHERNET-Modul hat die Kopfadresse X/Y0.

- IEC-Editoren (Auf der nächsten Seite ist dieses Beispiel in der MELSEC-Anweisungliste und dem Kontaktplan des GX Developers dargestellt.)

Kontaktplan (GX IEC Developer)



IEC-Anweisungsliste

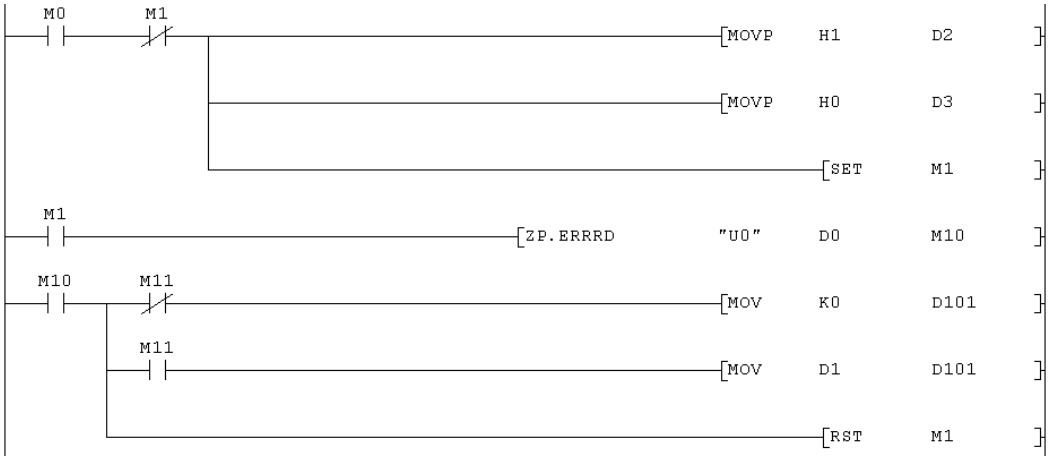
```
LD          M0
ANDN        M1
MOVP_M      H1,     D2
MOVP_M      H0,     D3
SET_M       M1
LD          M1
ERRRD_M     "U0",   var_D0,   var_M10

LD          M10
ANDN        M11
MOV_M       K0,     D101
LD          M10
AND         M11
MOV_M       D1,     D101
LD          M10
RST_M       M1
```

Die Operanden und Anweisungen sind oben beim Programmbeispiel für den Kontaktplan erläutert.

HINWEIS *Bei den IEC-Editoren des GX IEC Developers müssen die Variablen im Header der Programmorganisationseinheit (POE) definiert werden. Ohne Variablendefinition werden beim Überprüfen oder Kompilieren des Programms Fehler gemeldet.*
Weitere Informationen finden Sie im Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung.

- MELSEC-Anweisungliste und Kontaktplan des GX Developers
  Die Operanden und Anweisungen sind auf der vorherigen Seite beim Programmbeispiel für den Kontaktplan des GX IEC Developers erläutert.

Kontaktplan (GX Developer)



MELSEC-Anweisungsliste

MELSEC	LD	M0		
	ANI	M1		
	MOVP	H1	D2	
	MOVP	H0	D3	
	SET	M1		
	LD	M1		
	ZP.ERRRD	"U0"	D0	M10
MELSEC	LD	M10		
	MPS			
	ANI	M11		
	MOV	K0	D101	
	MRD			
	AND	M11		
	MOV	D1	D101	
	MPP			
	RST	M1		

11.3.8 UINI

CPU

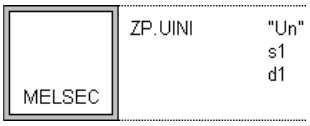
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					●

Operanden MELSEC Q

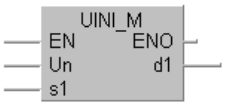
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□N□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	—	●	●	—	—	—	—	—	—	SM0	
d1	●	●	●	—	—	—	—	—	—		

GX IEC Developer

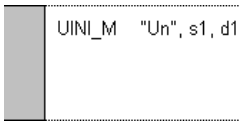
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

Operand	Bedeutung	Wertebereich	Festlegung des Inhalts durch	Datentyp
„Un“	Kopfadresse des ETHERNET-Moduls auf dem Baugruppenträger (Es werden nur die ersten beiden Stellen der 3-stelligen Adresse angegeben, z. B. wird die Kopfadresse X/Y100 als „U10“ eingetragen.)	0 bis FE_H	Anwender	BIN-16-Bit

Variablen

<table>
<tr><th>Operand</th><th colspan="3">Bedeutung</th><th>Werte-bereich</th><th>Festlegung des Inhalts durch</th><th>Datentyp</th></tr>
<tr><td rowspan="8">s1</td><td colspan="5">Erster Operand des Bereiches mit Informationen zur Ausführung der Anweisung</td><td rowspan="8">BIN-16-Bit</td></tr>
<tr><td>Operand</td><td>Bedeutung</td><td>Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(s1)+0</td><td>Systembereich</td><td>Wird vom System verwendet</td><td rowspan="2">—</td><td rowspan="2">System</td></tr>
<tr><td>(s1)+1</td><td>Ausführungs-status der Anweisung</td><td>Zeigt an, ob bei der Bearbeitung der Anweisung ein Fehler aufgetreten ist:
0000_H: Fehlerfreie Bearbeitung
Jeder andere Wert als 0000_H: Bei der Bearbeitung ist ein Fehler aufgetreten. Der eingetragene Wert ist ein Fehlercode, der entweder in der Bedienungsanleitung des ETHERNET-Moduls oder in Kapitel 13 dieses Handbuchs beschrieben ist.</td></tr>
<tr><td>(s1)+2</td><td>Auswahl der Änderungen</td><td>Die Bits 0 und 1 dieses Wort-Operanden dienen zur Einstellung von Verbindungsmerkmalen.
• Bit 0: Änderung der IP-Adresse der lokalen Station (Die neue Adresse wird in (s1)+3 und (s1)+4 angegeben.)
0: IP-Adresse nicht ändern
1: IP-Adresse ändern
• Bit 1: Änderung der Betriebseinstellungen (Die neuen Einstellungen werden in (s1)+5 angegeben.)
0: Einstellungen nicht ändern
1: Einstellungen ändern
Die übrigen Bits (b2 bis b15) müssen auf „0“ gesetzt werden.</td><td>0000_H
bis
0003_H</td><td rowspan="3">Anwender</td></tr>
<tr><td>(s1)+3
(s1)+4</td><td>IP-Adresse der lokalen Station</td><td>Neue IP-Adresse der lokalen Station</td><td>00000001_H
bis
$FFFFFFFE_H$</td></tr>
<tr><td>(s1)+5</td><td>Betriebsein-stellungen</td><td>Mit den einzelnen Bits dieses Wort-Operanden werden die Einstellungen festgelegt.
• Bit 1: Codierung der Daten bei der Kommunikation
0: Binärcode
1: ASCII-Code
• Bit 5: Senderahmen
0: ETHERNET-Rahmen
1: Rahmen nach IEEE802.3
• Bit 6: Programmänderungen der CPU in der Betriebsart RUN freigeben
0: Änderungen nicht zulassen
1: Änderungen zulassen
• Bit 8: Wartezeit
0: Nicht auf das Öffnen einer Verbindung warten (Bei gestoppter CPU kann nicht kommuniziert werden)
1: Immer auf das Öffnen einer Verbindung warten (Kommunikation bei gestoppter CPU ist möglich)
Die übrigen Bits dieses Operanden müssen auf „0“ gesetzt werden.</td><td>—</td></tr>
</table>

Variablen

<table>
<tr><th>Operand</th><th colspan="3">Bedeutung</th><th>Werte-bereich</th><th>Festlegung des Inhalts durch</th><th>Datentyp</th></tr>
<tr><td rowspan="4">d1</td><td colspan="5">Bit-Operand, der nach der Ausführung der UINI-Anweisung für einen Zyklus gesetzt wird. Mit (d1)+1 wird ein Fehler bei der Ausführung angezeigt.</td><td rowspan="4">Bit</td></tr>
<tr><td>Operand</td><td>Bedeutung</td><td>Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(d1)+0</td><td>Anweisung ausgeführt</td><td>Zeigt die Beendigung der UINI-Anweisung an.
EIN: Anweisung ausgeführt
AUS: Anweisung nicht ausgeführt</td><td>—</td><td rowspan="2">System</td></tr>
<tr><td>(d1)+1</td><td>Anweisung mit Fehler ausgeführt</td><td>Zeigt an, ob bei der Ausführung der UINI-Anweisung ein Fehler aufgetreten ist.
EIN: Anweisung mit Fehler ausgeführt
AUS: Anweisung ohne Fehler ausgeführt</td><td>—</td></tr>
</table>

HINWEIS

Falls das ETHERNET-Modul nur erneut initialisiert werden soll (ohne Änderung der IP-Adresse der lokalen Station oder der Betriebseinstellungen), muss vor der Ausführung der UINI-Anweisung in den Operanden (s1)+2 der Wert „0“ eingetragen werden.
Bei der Initialisierung werden Adressinformationen anderer Stationen im ETHERNET-Modul gelöscht und der Datenaustausch wieder ermöglicht. Der Eingang X19 wird nach Abschluss der Initialisierung eingeschaltet.

Funktions-weise

ETHERNET-Modul erneut initialisieren

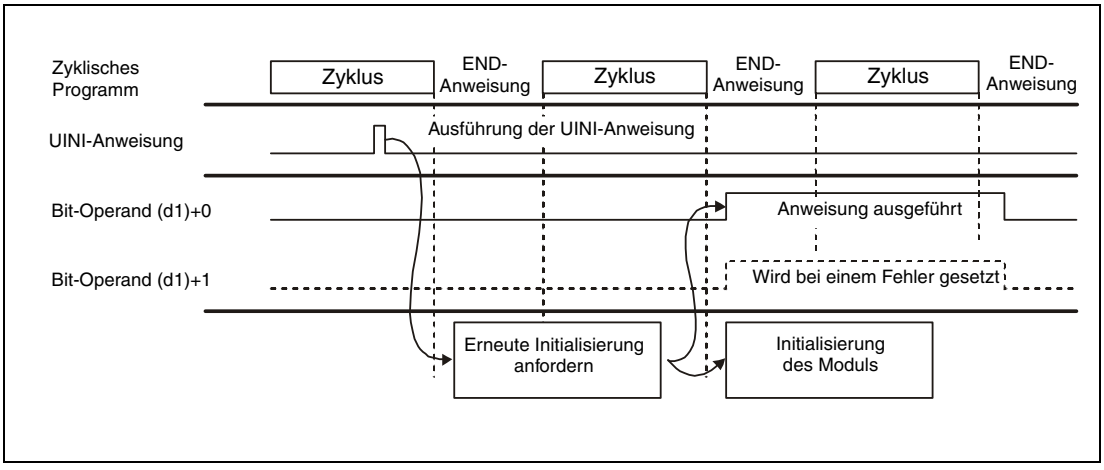
UINI Initialisierung starten

Mit der UINI-Anweisung wird das in Un angegebene Ethernet-Modul erneut initialisiert.

Mit den Bit-Operanden (d1)+0 und (d1)+1 kann die Ausführung der UINI-Anweisung überprüft werden:

- Der Bit-Operand (d1)+0 wird gesetzt, wenn die END-Anweisung des Zyklus ausgeführt wird, in dem die UINI-Anweisung beendet wurde. Bei der nächsten Bearbeitung der END-Anweisung wird der in d1 angegebene Bit-Operand wieder zurückgesetzt.
- Der Bit-Operand (d1)+1 zeigt einen Fehler bei der Ausführung der UINI-Anweisung an. Bei fehlerfreier Ausführung bleibt dieser Bit-Operand zurückgesetzt. Bei einem Fehler jedoch wird (d1)+1 gesetzt, wenn die END-Anweisung des Zyklus ausgeführt wird, in dem die UINI-Anweisung beendet wurde. Bei der nächsten Bearbeitung der END-Anweisung wird dieser Operand wieder zurückgesetzt.

Die folgende Abbildung zeigt den Signalverlauf bei Ausführung der UINI-Anweisung:



HINWEISE *Bitte beachten Sie die folgenden Punkte bei der erneuten Initialisierung des ETHERNET-Moduls. (Andernfalls können Fehler bei der Datenkommunikation mit externen Modulen auftreten.)*

- Stellen Sie sicher, dass alle laufenden Kommunikationen mit externen Modulen beendet sind und schließen Sie alle Verbindungen, bevor Sie das ETHERNET-Modul erneut initialisieren.

- Während der Ausführung der UINI-Anweisung dürfen keine Daten (z. B. mit einer TO-Anweisung) direkt in den Pufferspeicher geschrieben werden.
Fordern Sie keine weitere erneute Initialisierung an, während bereits eine UINI-Anweisung ausgeführt wird.

- Falls die IP-Adresse des ETHERNET-Modul geändert wird, müssen externe Module zurückgesetzt werden. (Wenn ein externes Modul die ETHERNET-Adresse eines Moduls, mit dem es kommuniziert, speichert, kann die Kommunikation nach der Änderung der IP-Adresse nicht fortgesetzt werden.)

Fehlerquellen

Wenn bei der Ausführung der UINI-Anweisung ein Fehler auftritt, wird der Operand (d1)+1 gesetzt und ein Fehlercode in (s1)+1 gespeichert. Detaillierte Angaben zu den Fehlercodes finden Sie in den folgenden Anleitungen:

- Bei einem Fehlercode bis $4FFF_H$ finden Sie Hinweise zur Fehlerbehebung in diesem Handbuch (Kapitel 13).
- Bei einem Fehlercode ab $C001_H$ finden Sie detaillierte Angaben in der Bedienungsanleitung zu den ETHERNET-Modulen des System Q.

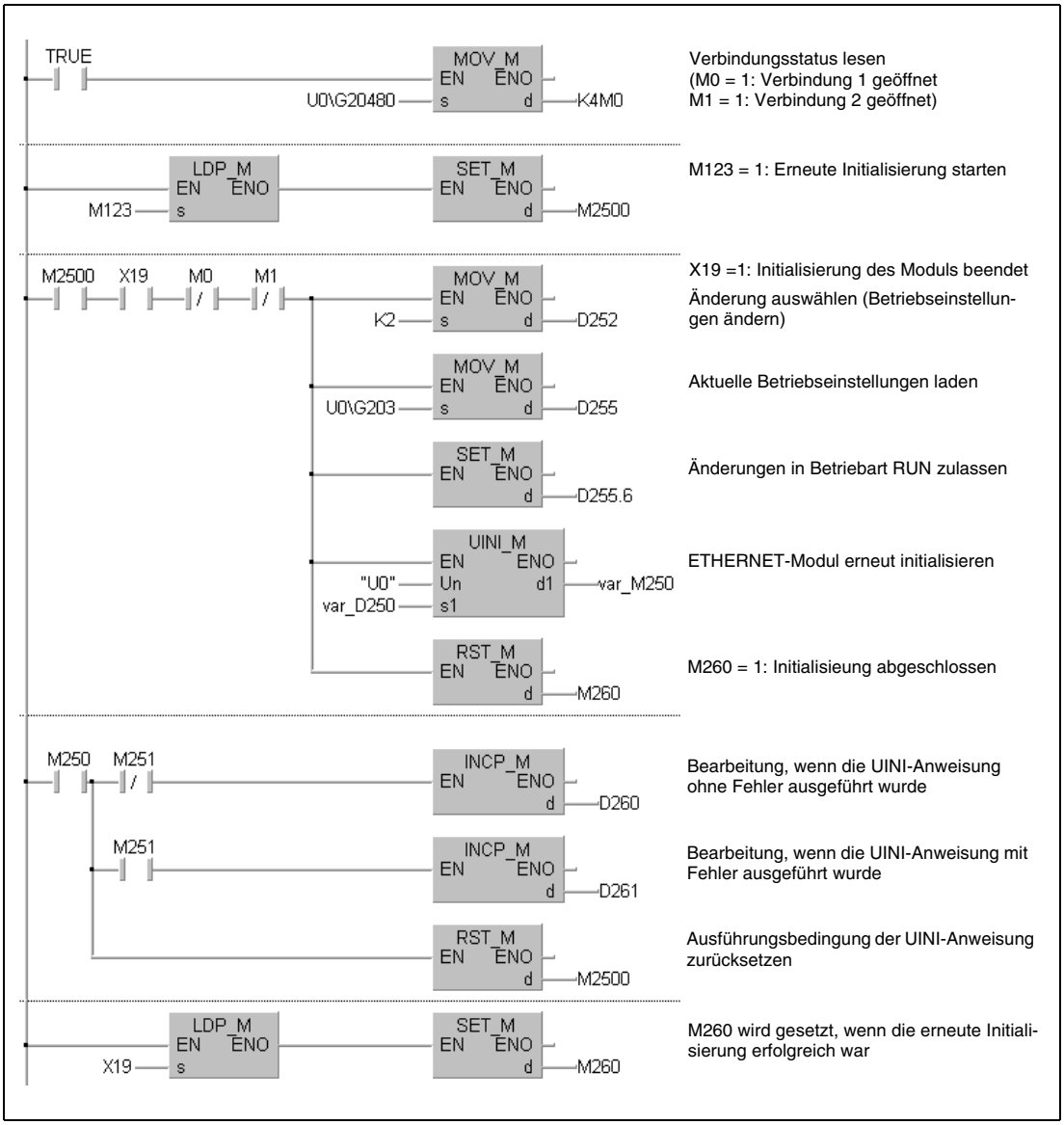
Beispiel UINI

Mit dem folgenden Programm wird das ETHERNET-Modul mit der Kopfadresse X/Y0 (Adressbereich X/Y0 bis X/Y1F) erneut initialisiert.

HINWEISE *Bei diesem Beispielprogramm werden nur die Verbindungen 1 und 2 verwendet. Bei anderen Verbindungen müssen im Programm die entsprechenden Signale verarbeitet werden.*

Bei den IEC-Editoren des GX IEC Developers müssen die Variablen im Header der Programmorganisationseinheit (POE) definiert werden. Ohne Variablendefinition werden beim Überprüfen oder Kompilieren des Programms Fehler gemeldet.
Weitere Informationen finden Sie im Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung.

- Kontaktplan (GX IEC Developer)



- IEC- und MELSEC-Anweisungliste
  Die Operanden und Anweisungen sind beim Programmbeispiel für den Kontaktplan des GX IEC Developers erläutert.

IEC-Anweisungsliste

```
LD      TRUE
MOV_M   U0\G20480, K4M0
PLS_M   M123
SET_M   M2500
AND     M2500
AND     X19
ANDN    M0
ANDN    M1
MOV_M   K2,        D252
MOV_M   U0\G203,   D255
SET_M   D255.6
UINI_M  "U0",      var_D250, var_M250
RST_M   M260
```

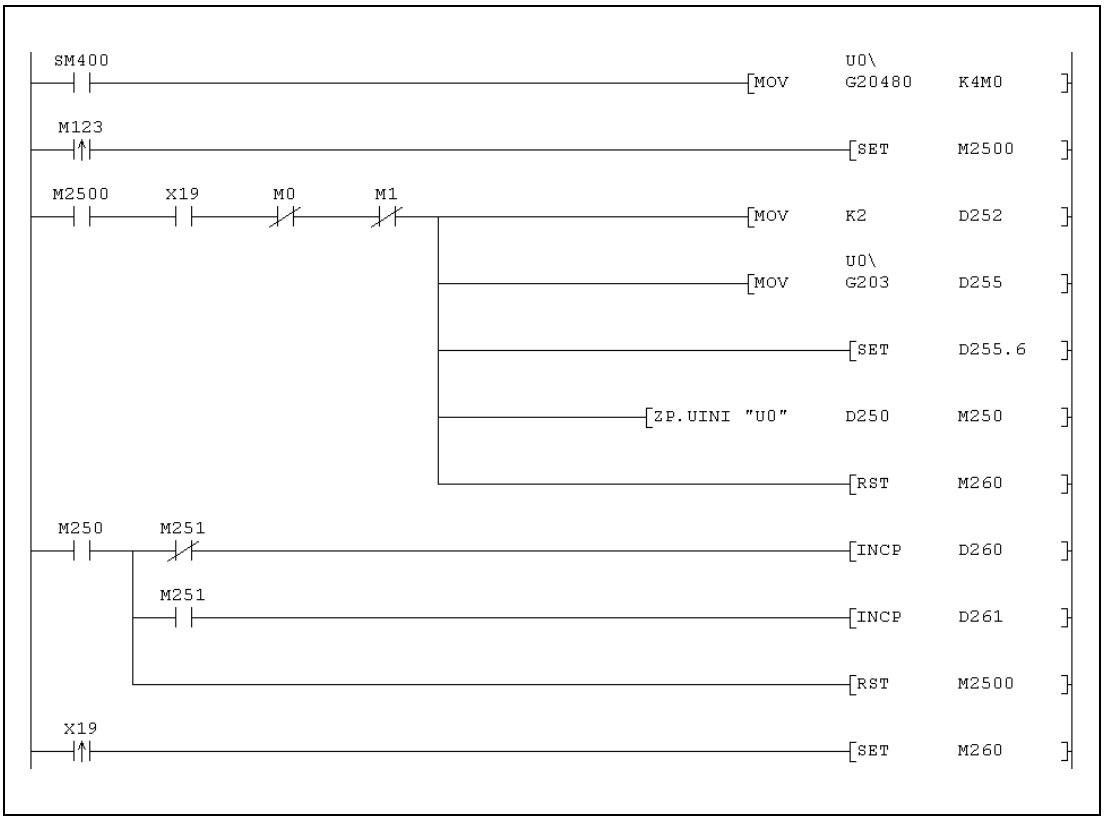
```
LD      M250
ANDN    M251
INCP_M  D260
LD      M250
AND     M251
INCP_M  D261
LD      M250
RST_M   M2500
PLS_M   X19
SET_M   M260
```

MELSEC-Anweisungsliste

```
MELSEC  LD      SM400
        MOV     U0\G20480  K4M0
        LDP     M123
        SET     M2500
        LD      M2500
        AND     X19
        ANI     M0
        ANI     M1
        MOV     K2         D252
        MOV     U0\G203    D255
        SET     D255.6
        ZP.UINI "U0"       D250    M250
        RST     M260
```

```
MELSEC  LD      M250
        MPS
        ANI     M251
        INCP    D260
        MRD
        AND     M251
        INCP    D261
        MPP
        RST     M2500
        LDP     X19
        SET     M260
```

- Kontaktplan (GX Developer)
  Die Operanden und Anweisungen sind beim Programmbeispiel für den Kontaktplan des GX IEC Developers erläutert.



11.4 Anweisungen für MELSECNET/10

Funktion	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor
Stationen für Duplexbetrieb festlegen	J.PAIRSET	PAIRSET_M

11.4.1 PAIRSET

CPU

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●[1]	

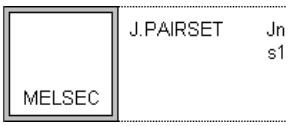
[1] Nur für Q4AR

Operanden MELSEC Q

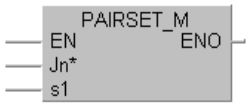
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□\□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	—	●	●	—	—	—	—	—	—	SM0	

GX IEC Developer

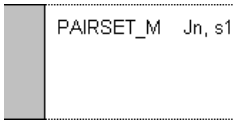
MELSEC-Anweisungsliste



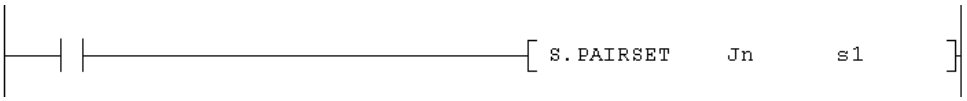
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

Operand	Befehlswert	Datentyp
Jn	Nummer des Netzwerks (1 bis 239)	BIN-16-Bit
s1	Anfangsadresse des Operandenbereichs, in dem die Einstellungen für Paarungen gespeichert werden Es können File-Register (R, ZR) oder die Operanden T, ST, C, D und W aus dem Latch-Bereich verwendet werden. Bei File-Registern ist eine Speicherkarte erforderlich.	

Funktions-weise

Stationen paarweise zusammenfassen

PAIRSET Parrungen festlegen

Mit dieser Anweisung wird der Operandenbereich angegeben, in dem festgelegt ist, welche Stationen beim Duplexbetrieb verbunden sind.

Struktur des Operandenbereiches mit den Einstellungen

- Die Einstellung der Stationen in dem durch s1 bezeichneten Operandenbereich ist nicht mit einem Ablaufprogramm möglich. Die Daten müssen vorher mit Hilfe eines Programmiergerätes in der SPS-CPU gespeichert werden.
- Es werden unabhängig von der Anzahl der angeschlossenen Stationen vier Worte belegt.
- Es können nur zwei Stationen mit aufeinanderfolgenden Stationsnummern „gepaart“ werden. Das Bit, dass in dem durch s1 bezeichneten Operandenbereich die Station mit der höheren Stationsnummer angibt, muss für eine Paarung gesetzt werden.
- Jedes Bit im Operandenbereich (s1)+0 bis (s1)+3 steht für eine Stationsnummer zwischen 1 und 64:

Operand	Bits															
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
(s1)+0	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
(s1)+1	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
(s1)+2	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33
(s1)+3	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49

HINWEISE

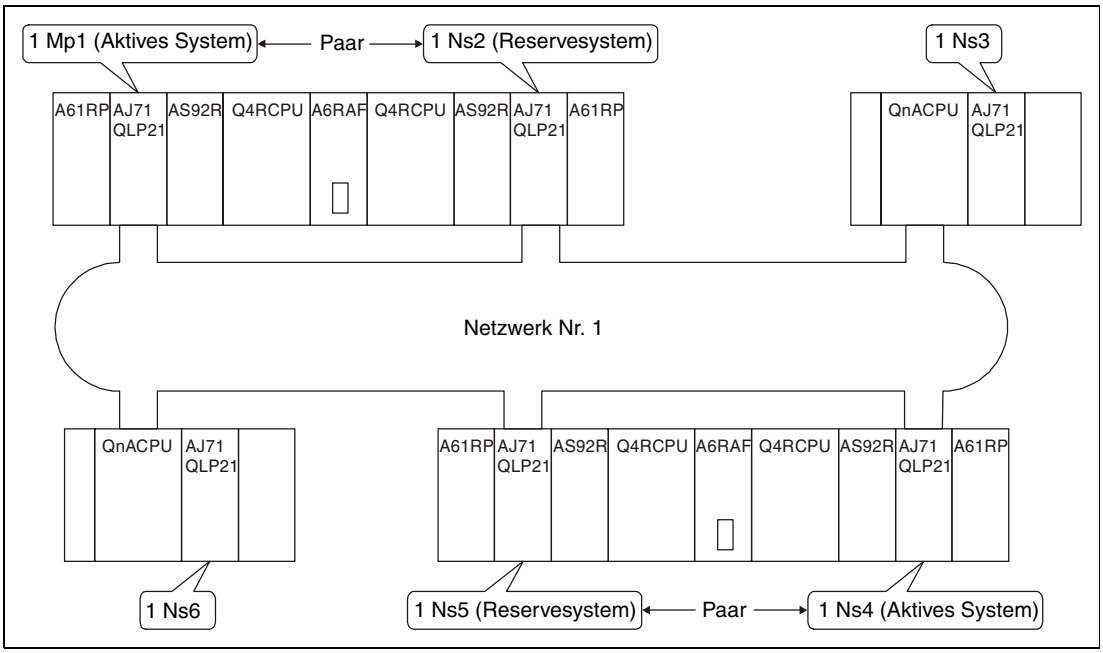
Die PAIRSET-Anweisung ist nur bei Kontrollstationen gültig. Einstellungen an einer Normalstation sind ungültig.

Kann in einer redundanten Steuerung mit Q4ARCPUs das Netzwerkmodul des aktiven Systems wegen einer Leitungsunterbrechung nicht mehr kommunizieren, wird nur auf das Reservesystem umgeschaltet, wenn die PAIRSET-Anweisung ausgeführt wurde.

Beispiel

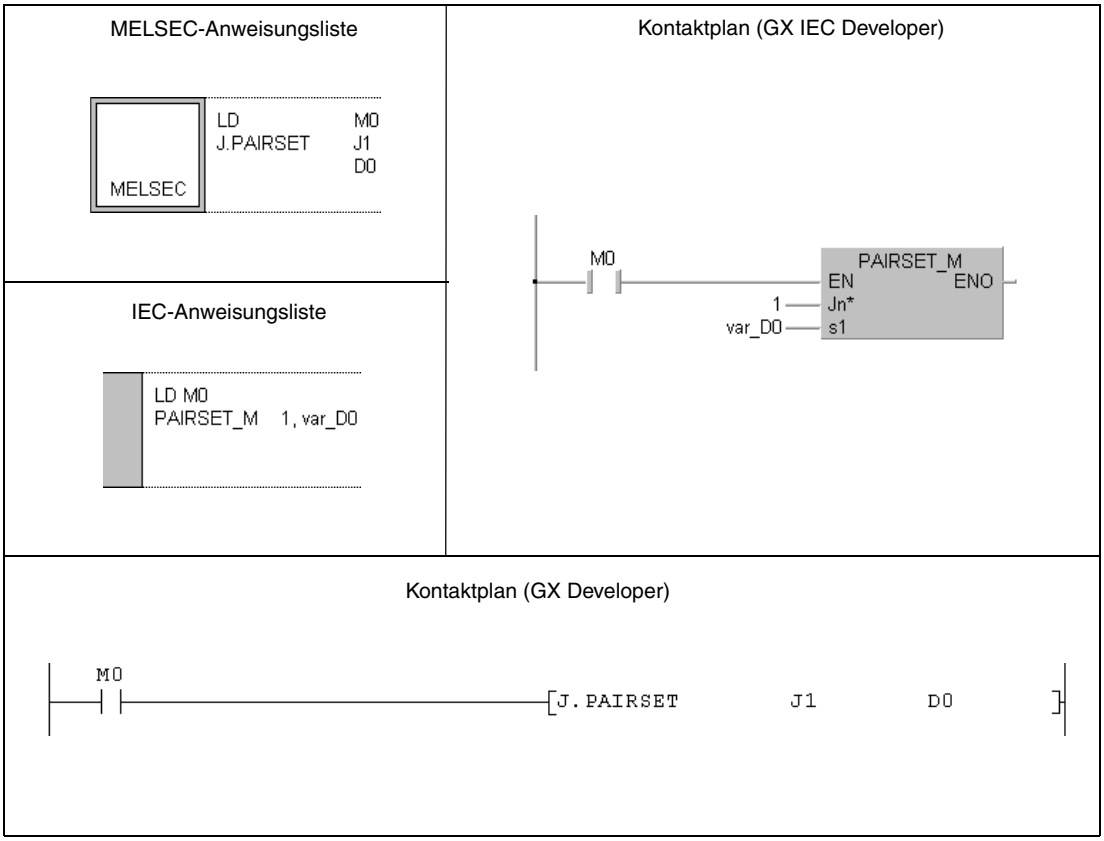
PAIRSET

Die Stationen 1 und 2 sowie 4 und 5 eines redundanten Systems sollen paarweise zusammengefasst werden:



Die Einstellungen für die Paarungen sind in den Datenregistern D0 bis D3 gespeichert. In D0 wird b1 (für die Verbindung der Stationen 1 und 2) und b4 (für die Verbindung der Stationen 4 und 5) gesetzt:

Operand	Bits															
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
D0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
D1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



HINWEIS *Bei den IEC-Editoren des GX IEC Developers müssen die Variablen im Header der Programmorganisationseinheit (POE) definiert werden. Ohne Variablendefinition werden beim Überprüfen oder Kompilieren des Programms Fehler gemeldet.*
Weitere Informationen finden Sie im Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung.

11.5 Anweisungen für CC-Link

Funktion	MELSEC-Anweisung im MELSEC-Editor	MELSEC-Anweisung im IEC-Editor
Netzwerkparameter in die Master-Station übertragen (A-Serie)	G.RLPA	RLPA_MD
	GP.RLPA	RLPA_P_MD
Netzwerkparameter in die Master-Station übertragen und Datenaustausch starten (System Q)	G.RLPASET	RLPASET_MD
	GP.RLPASET	RLPASET_P_MD
Parameter für automatische Aktualisierung übertragen (A-Serie)	G.RRPA	RRPA_MD
	G.RRPA	RRPA_P_MD
Daten aus dem Pufferspeicher eines CC-Link-Moduls einer anderen Station oder aus der SPS-CPU dieser Station lesen	G.RIRD	RIRD_MD
	GP.RIRD	RIRD_P_MD
Daten in den Pufferspeicher eines CC-Link-Moduls einer anderen Station oder in die SPS-CPU dieser Station schreiben	G.RIWT	RIWT_MD
	GP.RIWT	RIWT_P_MD
Daten unter Verwendung eines Handshake aus dem Pufferspeicher einer intelligenten Station lesen	G.RIRCV	RIRCV_MD
	GP.RIRCV	RIRCV_P_MD
Daten unter Abwicklung eines Handshake in den Pufferspeicher einer intelligenten Station eintragen	G.RISEND	RISEND_MD
	GP.RISEND	RISEND_P_MD
Daten in den automatisch aktualisierten Speicher schreiben	G.RITO	RITO_MD
	GP.RITO	RITO_P_MD
Daten aus dem automatisch aktualisierten Speicher lesen	G.RIFR	RIFR_MD
	GP.RIFR	RIFR_P_MD

11.5.1 RLPA (A-Serie)

CPU

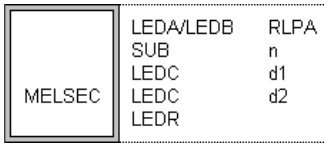
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●		

Operanden MELSEC A

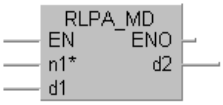
	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9011
n1																	●	●								
d1								●	●	●	●	●											23			●
d2		●	●	●	●	●																				

GX IEC Developer

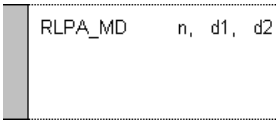
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



GX Developer

Hinweise zur Programmierung der erweiterten Anweisungen in den MELSEC-Editoren finden Sie im Kapitel 3.3 dieses Handbuchs.

Variablen

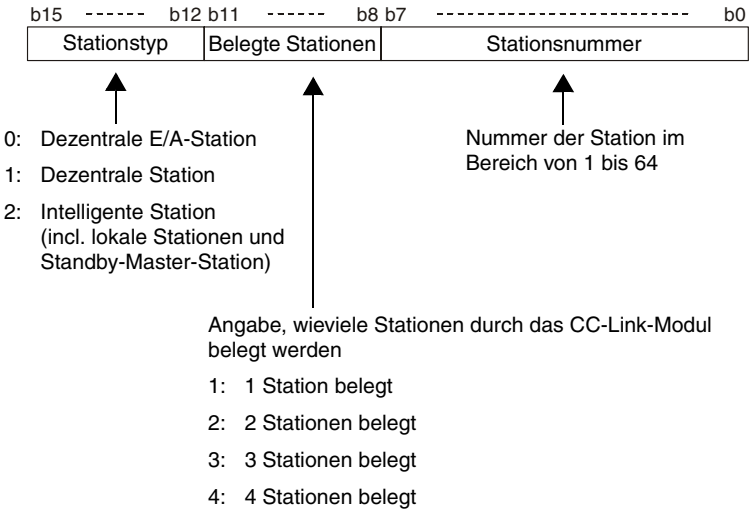
<table>
<tr><th>Operand</th><th colspan="4">Bedeutung</th><th>Werte-bereich</th><th>Festlegung des Inhalts durch</th><th>Datentyp</th></tr>
<tr><td>n</td><td colspan="4">Kopfadresse des CC-Link-Moduls auf dem Baugruppenträger (Es werden nur die ersten beiden Stellen der 3-stelligen Adresse angegeben. Beispiel: Kopfadresse X/Y100 -> 10_H)</td><td>0 bis FE_H</td><td>Anwender</td><td>BIN-16-Bit</td></tr>
<tr><td rowspan="15">d1</td><td colspan="4">Erster Operand des Bereiches mit Netzwerkparametern</td><td></td><td></td><td rowspan="15">BIN-16-Bit</td></tr>
<tr><td>Operand</td><td colspan="2">Bedeutung</td><td>Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(d1)+0</td><td colspan="2">Synchronbetrieb</td><td>• [(d)1+0]= 0: Kein Synchron-betrieb
• [(d1)+0] = 1: Synchronbetrieb</td><td>0 oder 1</td><td>Anwender</td></tr>
<tr><td>(d1)+1</td><td colspan="2">Anzahl der angeschlossenen Stationen</td><td>Geben Sie hier an, wieviele Slave-Stationen an der CC-Link-Master-Station angeschlossen sind.</td><td>1 bis 64</td><td rowspan="13">Anwender</td></tr>
<tr><td>(d1)+2</td><td colspan="2">Stations-informationen (1. Station)</td><td colspan="2" rowspan="2">Siehe Tabelle auf der nächsten Seite</td></tr>
<tr><td>(d1)+3</td><td colspan="2">Stations-informationen (2. Station)</td></tr>
<tr><td>•
•</td><td>•
•</td><td>•
•</td><td colspan="2">•
•</td></tr>
<tr><td></td><td colspan="2">Stations-informationen (Letzte Station)</td><td colspan="2">Siehe Tabelle auf der nächsten Seite</td></tr>
<tr><td></td><td rowspan="3">1. lokale Station</td><td>Größe des Sendepuffers</td><td rowspan="2">Anzahl der Operanden, die zwischen der Master-Station und einer lokalen oder intelligenten Station ausgetauscht werden.</td><td rowspan="3">Abhängig vom verwendeten Modul</td></tr>
<tr><td></td><td>Größe des Empfangs-puffers</td></tr>
<tr><td></td><td>Größe des automatisch aktualisierten Puffers</td><td>Anzahl der Operanden des automatisch aktualisierten Puffers die für die Kommunikation zwischen Master- und lokaler/ intelligenter Station verwendet werden.</td></tr>
<tr><td>•
•</td><td>•
•</td><td>•
•</td><td>•
•</td><td>•
•</td></tr>
<tr><td>(d1)+(n-2)</td><td rowspan="3">Letzte Station</td><td>Größe des Sendepuffers</td><td rowspan="3">Wie bei Station 1</td><td rowspan="3">Abhängig vom verwendeten Modul</td></tr>
<tr><td>(d1)+(n-1)</td><td>Größe des Empfangs-puffers</td></tr>
<tr><td>(d1)+n</td><td>Größe des automatisch aktualisierten Puffers</td></tr>
<tr><td>d2</td><td colspan="4">Bit-Operand, der nach der Ausführung der RLPA-Anweisung für einen Zyklus gesetzt wird.</td><td>0 oder 1</td><td>System</td><td>Bit</td></tr>
</table>

Anzahl der Operanden für d1:

Bei d1 werden für die Anwahl des Synchronbetriebs in (d1)+0 und der Angabe der Anzahl der angeschlossenen Stationen in (d1)+1 **zwei** Operanden belegt. Jede Station benötigt **einen** Operanden für die Stationsinformationen. Zusätzlich müssen für jede lokale oder intelligente Station **drei** Operanden für die Festlegung der Puffergröße reserviert werden.

Stationsinformationen

Für jede Station ist innerhalb der Parameter ein Wort ((d1)+2, (d1)+6, (d1)+10, ...) mit detaillierten Angaben zur Art der Station reserviert:

Bedeutung	Beschreibung	Wertebereich
Stations-informationen		b0 bis b7: 1 – 64 (01_H – 40_H) b8 bis b11: 1 – 4 b12 bis b15: 0 – 2

Funktionsweise

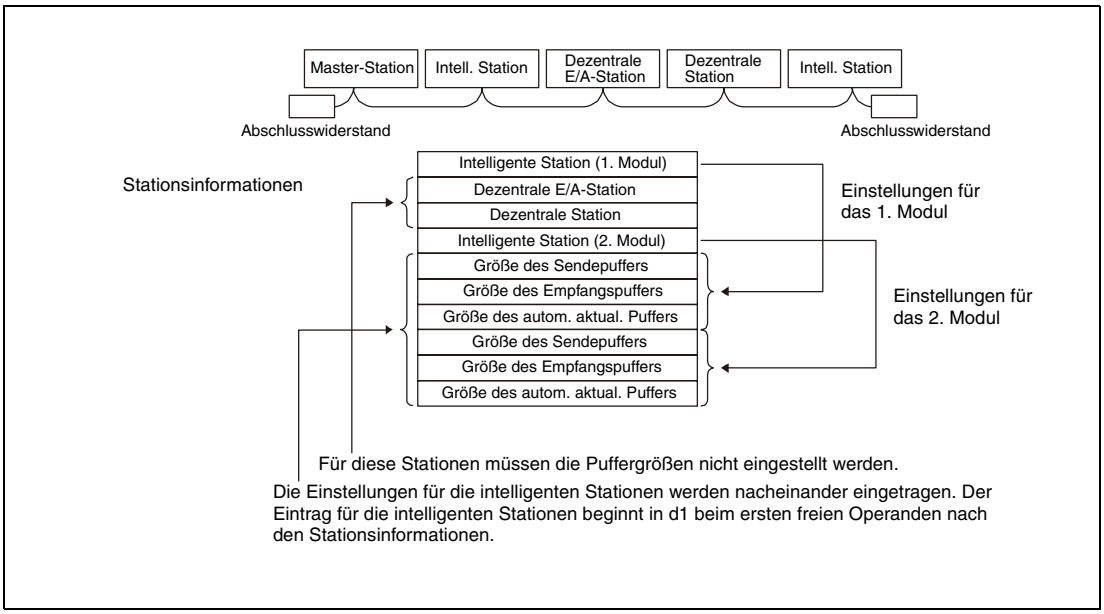
Parameter eines CC-Link-Netzwerks einstellen

RLPA Parameter einstellen

Mit der RLPA-Anweisung werden Einstellungen (d1) zum CC-Link-Netzwerk in die Masterstation (n) übertragen.

HINWEIS *Verwenden Sie die RLPA-Anweisung nur, um den Synchronbetrieb, die Anzahl der angeschlossenen Stationen, die Stationsinformationen oder die Größe der Sende- und Empfangspuffer sowie des automatisch aktualisierten Puffers einzustellen.*
Alle anderen Parameter werden bei der Ausführung der RLPA-Anweisung auf ihre Standardwerte eingestellt.
Werden Parameter gleichzeitig mit der RLPA-Anweisung und TO-Anweisungen eingestellt, werden die mit TO-Anweisungen übertragenen Einstellungen nicht beachtet.

Wurde die Slave-Station als lokale/intelligente Station definiert, muss die Größe der Sende- und Empfangspuffer und des automatisch aktualisierten Puffers in d1 festgelegt werden. Bei einer dezentralen E/A- oder dezentralen Station müssen diese Angaben nicht gemacht werden. Die folgende Abbildung zeigt ein Beispiel:



HINWEISE *Geben Sie für die Größe des Sende-/Empfangspuffer 7 Worte mehr an, als für den Datenaustausch benötigt werden.*

Dem automatisch aktualisierten Puffer wird die vom Sondermodul benötigte Größe zugewiesen.

Bei intelligenten Stationen, die die automatische Aktualisierung nicht unterstützen oder bei denen sie nicht verwendet wird, muss für die Größe des automatisch aktualisierten Puffers der Wert „0“ eingetragen werden.

Falls nach der Einstellung der Netzwerkparameter die RLPA-Anweisung erneut in der Betriebsart RUN ausgeführt wird, um die Netzwerkparameter zu ändern, werden diese neuen Einstellungen nicht für die Kommunikation mit den Slave-Stationen verwendet.
Erst nachdem die CPU der A-Serie gestoppt (STOP/PAUSE-Modus) und wieder in die Betriebsart RUN geschaltet wurde, wird der Datenaustausch mit den Slave-Stationen mit den neuen Parametern abgewickelt.

Die Ausführung der RLPA-Anweisung startet automatisch die Datenübertragung.

Wird die RLPA-Anweisung ausgeführt, muss im Programm eine Verriegelung mit Hilfe der Eingangssignale Xn0 (Modul-Fehler) und XnF (Modul bereit) vorgesehen werden.

Ausführungsbedingungen

Wird die RLPA-Anweisung in Verbindung mit einer LEDA-Anweisung programmiert, wird die RLPA-Anweisung ausgeführt, solange die Ausführungsbedingung der LEDA-Anweisung eingeschaltet ist. Bei Anwendung einer LEDB-Anweisung dagegen wird der RLPA-Anweisung nur bei der steigenden Flanke der Ausführungsbedingung ausgeführt.

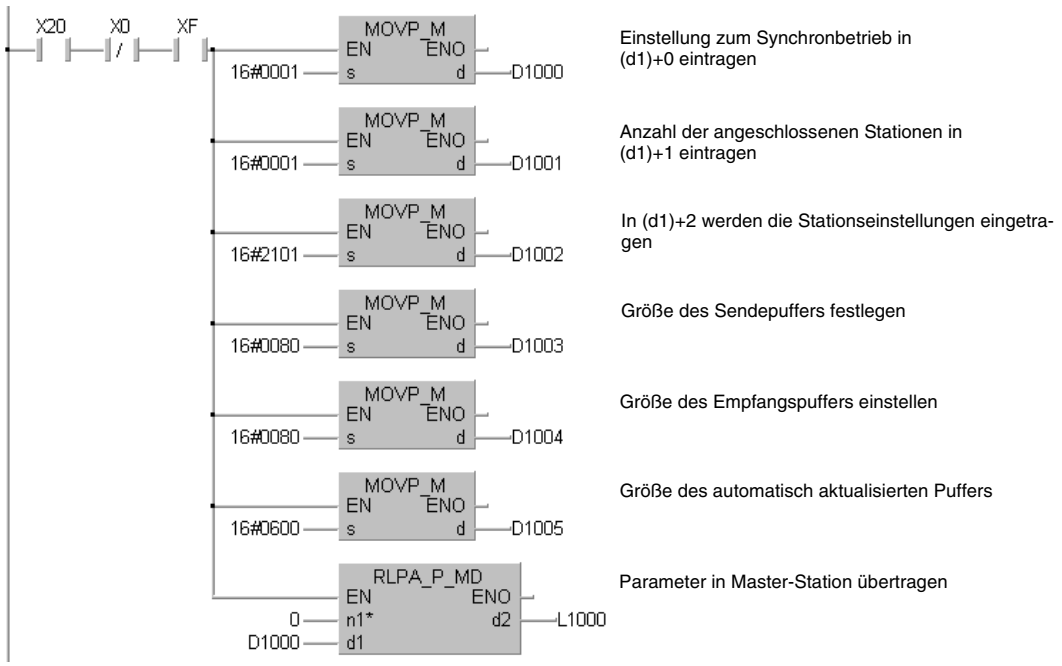
Beispiel RLPA

Das Beispiel-Programm überträgt die folgenden Netzwerkparameter in die Master-Station mit der E/A-Kopfadresse X/Y000.

Parameter		Einstellung	Wert	Verwendeter Operand
Synchronbetrieb		Aktiviert	1	D1000
Anzahl der angeschlossenen Stationen		1 Modul	1	D1001
Stationsinformationen	Stationsart	Intelligente Station	2	D1002
	Belegte Stationen	1 Station	1	
	Stationsnummer	Nr. 1	1	
Größe des Sendepuffers		128 Worte	80_H	D1003
Größe des Empfangspuffers		128 Worte	80_H	D1004
Größe des automatisch aktualisierten Puffers		960 Worte	600_H	D1005

- IEC-Editoren (Auf der nächsten Seite ist dieses Beispiel in der MELSEC-Anweisungliste und dem Kontaktplan des GX Developers dargestellt.)

Kontaktplan (GX IEC Developer)



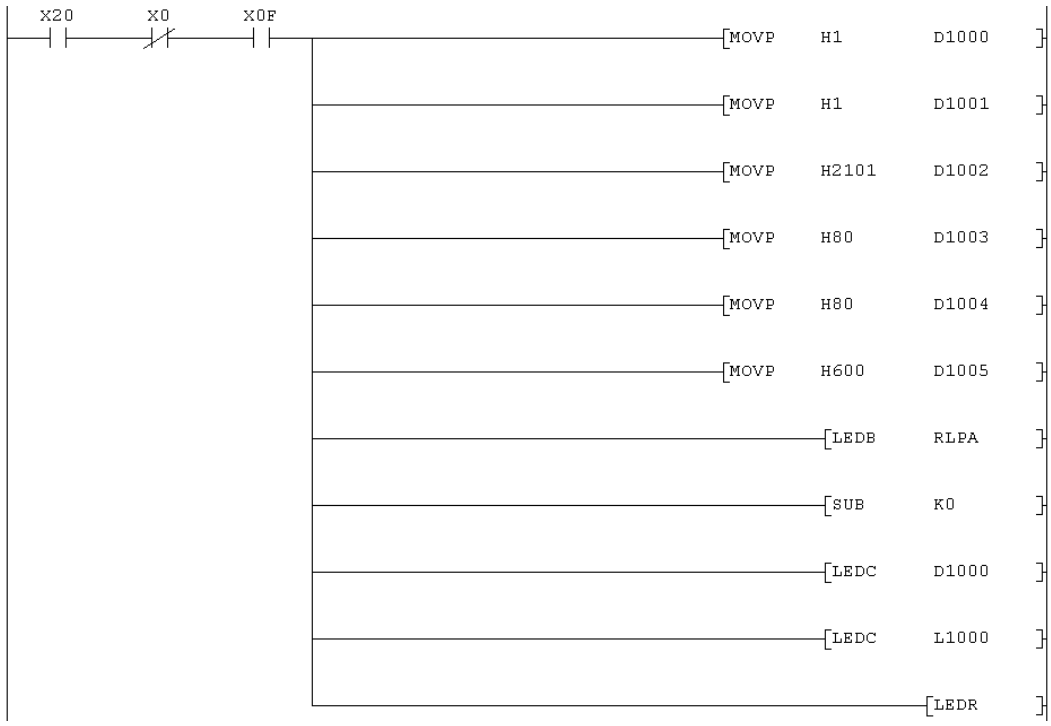
IEC-Anweisungsliste

```
LD          X20
ANDN        X0
AND         XF
MOVP_M      16#0001,    D1000
MOVP_M      16#0001,    D1001
MOVP_M      16#2101,    D1002
MOVP_M      16#0080,    D1003
MOVP_M      16#0080,    D1004
MOVP_M      16#0600,    D1005
RLPA_P_MD   0,          D1000, L1000
```

Die Operanden und Anweisungen sind oben beim Programmbeispiel für den Kontaktplan erläutert.

- MELSEC-Anweisungliste und Kontaktplan des GX Developers
  Die Operanden und Anweisungen sind auf der vorherigen Seite beim Programmbeispiel für den Kontaktplan des GX IEC Developers erläutert.

Kontaktplan (GX Developer)



MELSEC-Anweisungsliste

MELSEC

```
LD      X20
ANI     X0
AND     XF
MOVP    H0001   D1000
MOVP    H0001   D1001
MOVP    H2101   D1002
MOVP    H0080   D1003
MOVP    H0080   D1004
MOVP    H0600   D1005
LEDB    RLPA
SUB     K0
LEDC    D1000
LEDC    L1000
LEDR
```

11.5.2 RLPASET (System Q)

CPU

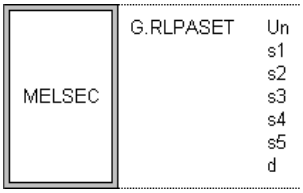
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					●

Operanden MELSEC Q

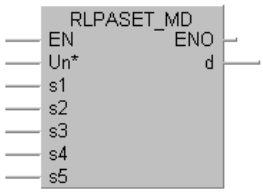
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□N□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	—	●	●	—	—	—	—	—	—	SM0	
s2	—	●	●	—	—	—	—	—	—		
s3	—	●	●	—	—	—	—	—	—		
s4	—	●	●	—	—	—	—	—	—		
s5	—	●	●	—	—	—	—	—	—		
d	●	●	●	—	—	—	—	—	—		

GX IEC Developer

MELSEC-Anweisungsliste



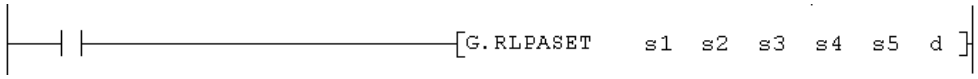
Kontaktplan



IEC-Anweisungsliste

```
RLPASET_MD  Un, s1, s2, s3, s4, d
```

GX Developer



Variablen

Operand	Bedeutung			Werte-bereich	Festlegung des Inhalts durch	Datentyp
Un	Kopfadresse des CC-Link-Moduls auf dem Baugruppenträger (Es werden nur die ersten beiden Stellen der 3-stelligen Adresse angegeben. Beispiel: Kopfadresse X/Y100 -> 10_H)			0 bis FE_H	Anwender	BIN-16-Bit
s1	Erster Operand des Bereiches mit Informationen zur Ausführung der Anweisung					Adresse
	Operand	**Bedeutung**	**Beschreibung**	**Werte-bereich**	**Festlegung durch**	
	(s1)+0	Ausführungs-status der Anweisung	Zeigt an, ob bei der Bearbeitung der Anweisung ein Fehler aufgetreten ist: 0000_H: Fehlerfreie Bearbeitung Jeder andere Wert als 0000_H: Bei der Bearbeitung ist ein Fehler aufgetreten.	—	System	
	(s1)+1	Umfang der Einstellungen	Mit den ersten vier Bits dieses Operanden wird angegeben, welche der in s2 bis s5 eingetragenen Einstellungen gültig ist: Bit 0 = 1: Einstellung zu Slave-Stationen (s2) Bit 1 = 1:Einstellung zu reservierten Stationen (s3) Bit 2 = 1:Einstellung zur Ignorierung gestörter Stationen (s4) Bit 3 = 1:Einstellungen zu Sende-, Empfangs- und autom. aktualisierten Puffern (s5) Für die Einstellungen, die nicht als gültig gekennzeichnet sind, werden die Standardwerte übernommen.	0 bis F	Anwender	
	(s1)+2	Anzahl der angeschlossenen Module	Angabe der angeschlossenen dezentralen/lokalen Module (inklusive reservierter Stationen)	1 bis 64		
	(s1)+3	Anzahl der Wiederholungsversuche	Angabe, wie oft versucht werden soll, die Kommunikation mit einer gestörten Station wieder aufzunehmen	1 bis 7		
	(s1)+4	Anzahl der Module mit autom.Wiedereingliederung	Angabe der Anzahl der angeschlossenen, dezentralen und lokalen Module, die nach einem Ausfall innerhalb eines Link-Zyklus wieder automatisch in den Datenaustausch einbezogen werden.	1 bis 10		
	(s1)+5	Verhalten bei einem Stopp der CPU der SPS	Angabe des Zustandes, den der Datenaustausch annehmen soll, wenn die SPS-CPU der Master-Station gestoppt wird. 0: Stoppen 1: Fortsetzen	0 oder 1		
	(s1)+6	Abtastsynchronisation	Auswahl zwischen Synchron- und Asynchronmodus 0: Der Datenaustausch läuft nicht synchron mit dem Ablaufprogramm. 1: Der Datenaustausch läuft synchron mit der Ausführung des Ablaufprogramms.	0 oder 1		
	(s1)+7	Verzögerungszeit	Intervall für die Abtastung des Links (Einheit: 50 µs)	0 bis 100		

Variablen

<table>
<tr><th>Operand</th><th colspan="3">Bedeutung</th><th>Werte-bereich</th><th>Festlegung des Inhalts durch</th><th>Datentyp</th></tr>
<tr><td rowspan="8">s2</td><td colspan="3">Erster Operand des Bereiches mit Einstellungen zu den Slave-Stationen</td><td></td><td></td><td rowspan="8">Adresse</td></tr>
<tr><td>Operand</td><td>Bedeutung</td><td>Beschreibung</td><td></td><td>Festlegung durch</td></tr>
<tr><td>(s2)+0</td><td>Einstellungen für Station 1</td><td rowspan="6">Siehe Tabelle auf Seite 96
Nehmen Sie die Einstellung für soviele Module vor, wie als Anzahl der angeschlossenen Module in (s1)+2 angegeben wurde.</td><td></td><td rowspan="6">Anwender</td></tr>
<tr><td>(s2)+1</td><td>Einstellungen für Station 2</td></tr>
<tr><td>•
•</td><td>•
•</td></tr>
<tr><td>(s2)+62</td><td>Einstellungen für Station 63</td></tr>
<tr><td>(s2)+63</td><td>Einstellungen für Station 64</td></tr>
<tr><td rowspan="6">s3</td><td colspan="3">Erster Operand des Bereiches mit Einstellungen zu reservierten Stationen
Nehmen Sie die Einstellung bis zu der größten Stationsnummer vor, die in s2 festgelegt wurde.</td><td></td><td></td><td rowspan="6">Adresse</td></tr>
<tr><td>Operand</td><td>Bedeutung</td><td>Beschreibung</td><td></td><td>Festlegung durch</td></tr>
<tr><td>(s3)+0</td><td>Einstellungen für Stationen 1 – 16</td><td rowspan="4">Für jede Station, die reserviert werden soll, wird das entsprechende Bit gesetzt (siehe Tabelle auf Seite 96). Bei Modulen, die mehr als eine Station belegen, wird das Bit gesetzt, das der ersten belegten Stationsnummer entspricht.

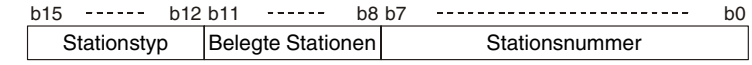
In der Voreinstellung ist keine Station reserviert.</td><td></td><td rowspan="4">Anwender</td></tr>
<tr><td>(s3)+1</td><td>Einstellungen für Stationen 17 – 32</td></tr>
<tr><td>(s3)+2</td><td>Einstellungen für Stationen 33 – 48</td></tr>
<tr><td>(s3)+3</td><td>Einstellungen für Stationen 49 – 64</td></tr>
<tr><td rowspan="6">s4</td><td colspan="3">Erster Operand des Bereiches mit Einstellungen zur Ignorierung gestörter Stationen
Nehmen Sie die Einstellung bis zu der größten Stationsnummer vor, die in s2 festgelegt wurde.</td><td></td><td></td><td rowspan="6">Adresse</td></tr>
<tr><td>Operand</td><td>Bedeutung</td><td>Beschreibung</td><td></td><td>Festlegung durch</td></tr>
<tr><td>(s4)+0</td><td>Einstellungen für Stationen 1 – 16</td><td rowspan="4">Für jede Station, deren Fehlermeldungen ignoriert werden soll, wird das entsprechende Bit gesetzt (siehe Tabelle auf Seite 96). Bei Modulen, die mehr als eine Station belegen, wird das Bit gesetzt, das der ersten belegten Stationsnummer entspricht.
Falls eine Station als reserviert gekennzeichnet ist und gleichzeitig die Fehlermeldungen ignoriert werden sollen, hat die Reservierung eine höhere Priorität. In der Voreinstellung ist keine Station angewählt.</td><td></td><td rowspan="4">Anwender</td></tr>
<tr><td>(s4)+1</td><td>Einstellungen für Stationen 17 – 32</td></tr>
<tr><td>(s4)+2</td><td>Einstellungen für Stationen 33 – 48</td></tr>
<tr><td>(s4)+3</td><td>Einstellungen für Stationen 49 – 64</td></tr>
</table>

Variablen

<table>
<tr><th>Operand</th><th colspan="4">Bedeutung</th><th>Werte-bereich</th><th>Festlegung des Inhalts durch</th><th>Datentyp</th></tr>
<tr><td rowspan="11">s5</td><td colspan="4">Erster Operand des Bereiches mit Einstellungen zur Puffergröße
Nehmen Sie die Einstellungen für die Stationen vor, die in s2 als lokale oder intelligente Stationen definiert wurden. Beginnen Sie bei der kleinsten Stationsnummer.</td><td></td><td></td><td rowspan="11">Adresse</td></tr>
<tr><td>Operand</td><td colspan="2">Bedeutung</td><td>Beschreibung</td><td>Werte-bereich</td><td>Festlegung des Inhalts durch</td></tr>
<tr><td>(s5)+0</td><td rowspan="3">1. Modul</td><td>Größe des Sendepuffers</td><td rowspan="2">Angabe der Puffergröße, die für die Kommunikation der Master-Station mit einer lokalen oder intelligenten Station genutzt wird.
Sende- und Empfangspuffer dürfen zusammen max. 4096 Worte (1000_H) groß sein.
Die Pufferspeicher für die Kommunikation sollten die Größe der zu sendenden/empfangenden Daten plus jeweils 7 Worte haben.</td><td>0_H: Kein Puffer
40_H bis 1000_H (64 bis 4096 Worte)
Voreinstellung: 40_H</td><td rowspan="8">Anwender</td></tr>
<tr><td>(s5)+1</td><td>Größe des Empfangspuffers</td><td>0_H: Kein Puffer
40_H bis 1000_H (64 bis 4096 Worte)
Voreinstellung: 40_H</td></tr>
<tr><td>(s5)+2</td><td>Größe des automatisch aktualisierten Puffers</td><td>Anzahl der Operanden des automatisch aktualisierten Puffers die für die Kommunikation zwischen Master- und lokaler/intelligenter Station verwendet werden.
Die Größe des automatisch aktualisierten Puffers muss der für den Datenaustausch mit den intelligenten Stationen vorgesehenen Datenmenge entsprechen.</td><td>0_H: Kein Puffer
80_H bis 1000_H (128 bis 4096 Worte)
Voreinstellung: 80_H</td></tr>
<tr><td>•
•</td><td>•
•</td><td>•
•</td><td>•
•</td><td>•
•</td></tr>
<tr><td>(s5)+75</td><td rowspan="3">26 Modul</td><td>Größe des Sendepuffers</td><td colspan="2" rowspan="3">Wie beim 1. Modul</td></tr>
<tr><td>(s5)+76</td><td>Größe des Empfangspuffers</td></tr>
<tr><td>(s5)+77</td><td>Größe des automatisch aktualisierten Puffers</td></tr>
<tr><td rowspan="3">d</td><td colspan="4">Bit-Operand, der nach der Ausführung der RLPASET-Anweisung für einen Zyklus gesetzt wird. Mit (d)+1 wird ein Fehler bei der Ausführung angezeigt.</td><td></td><td></td><td rowspan="3">Bit</td></tr>
<tr><td>Operand</td><td colspan="2">Bedeutung</td><td>Beschreibung</td><td>Werte-bereich</td><td>Festlegung des Inhalts durch</td></tr>
<tr><td>(d)+0</td><td colspan="2">Anweisung ausgeführt</td><td>Zeigt die Beendigung der RLPASET-Anweisung an.
EIN: Anweisung ausgeführt
AUS: Anweisung nicht ausgeführt</td><td>0 oder 1</td><td rowspan="2">System</td></tr>
<tr><td>(d)+1</td><td colspan="2">Anweisung mit Fehler ausgeführt</td><td>Zeigt an, ob bei der Ausführung der RLPASET-Anweisung ein Fehler aufgetreten ist.
EIN: Anweisung mit Fehler ausgeführt
AUS: Anweisung ohne Fehler ausgeführt</td><td>0 oder 1</td></tr>
</table>

Einstellungen zu Slave-Stationen

Für jede Station ist innerhalb der Parameter ein Wort ((s2)+0 bis (s2)+63) mit detaillierten Angaben zur Art der Station reserviert:

Bedeutung	Beschreibung	Werte-bereich
Einstellungen zu den einzelnen Stationen	 0: Dezentrale E/A-Station 1: Dezentrale Station 2: Intelligente Station (incl. lokale Stationen und Standby-Master-Station) Nummer der Station im Bereich von 1 bis 64 Angabe, wieviele Stationen durch das CC-Link-Modul belegt werden 1: 1 Station belegt 2: 2 Stationen belegt 3: 3 Stationen belegt 4: 4 Stationen belegt	b0 bis b7: 1 – 64 (01_H – 40_H) b8 bis b11: 1 – 4 b12 bis b15: 0 – 2

Die Voreinstellungen für (s2)+0 bis (s2)+63 sind „0101_H“ bis „0140_H“. (Stationsnummer 1 bis 64, eine Station belegt, dezentrale E/A-Station)

Angabe der Stationsnummer in s3 und s4

Jedes Bit der für s3 und s4 jeweils belegten vier Wortoperanden repräsentiert eine Station:

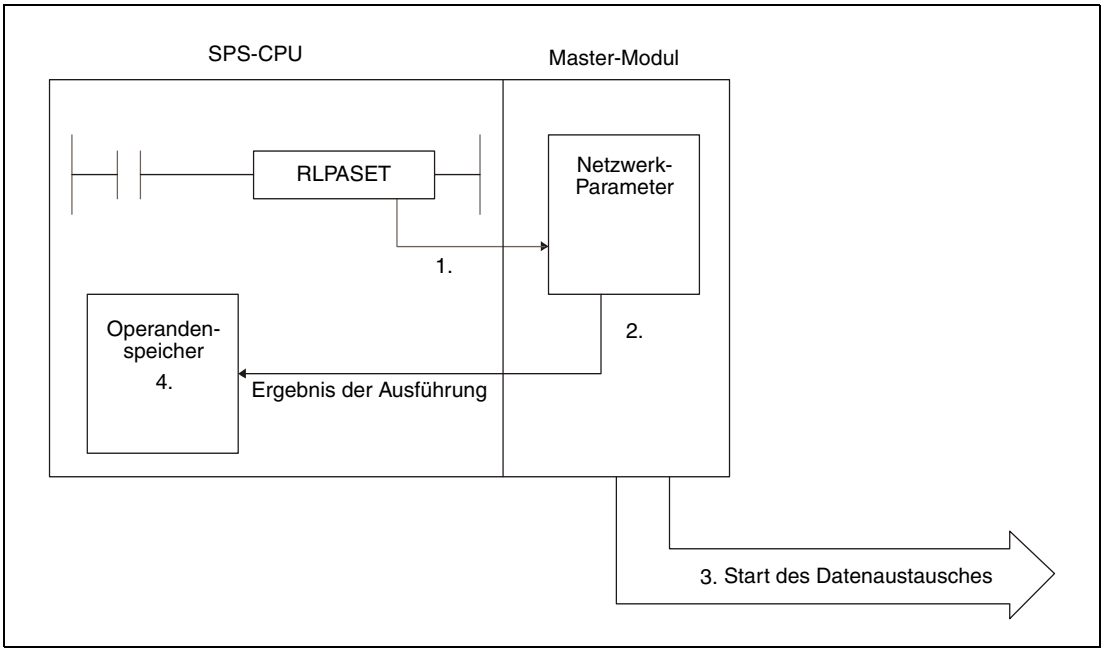
Operand	Bit															
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
(s3)+0 (s4)+0	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
(s3)+1 (s4)+1	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
(s3)+2 (s4)+2	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33
(s3)+3 (s4)+3	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49

Die Zahlen 1 bis 64 stehen für die Stationsnummern. Durch Setzen des entsprechenden Bits wird eine Station ausgewählt.

Funktionsweise

Parameter eines CC-Link-Netzwerks einstellen

RLPASET Parameter einstellen



1. Mit der RLPASET-Anweisung werden die Netzwerk-Parameter (s1 bis s5) in die Masterstation (Un) übertragen.
2. Das Master-Modul prüft die eingestellten Parameter.
3. Wenn diese fehlerfrei sind, wird der Datenaustausch gestartet.
4. Der Operand d wird gesetzt.

Es kann immer nur eine RLPASET-Anweisung zur selben Zeit ausgeführt werden.

Belegte Operanden

Die folgende Anzahl von Operanden wird durch eine RLPASET-Anweisung belegt:

- s1: 8 Wort-Operanden
- s2: 64 Wort-Operanden
- s3: 4 Wort-Operanden
- s4: 4 Wort-Operanden
- s5: 78 Wort-Operanden

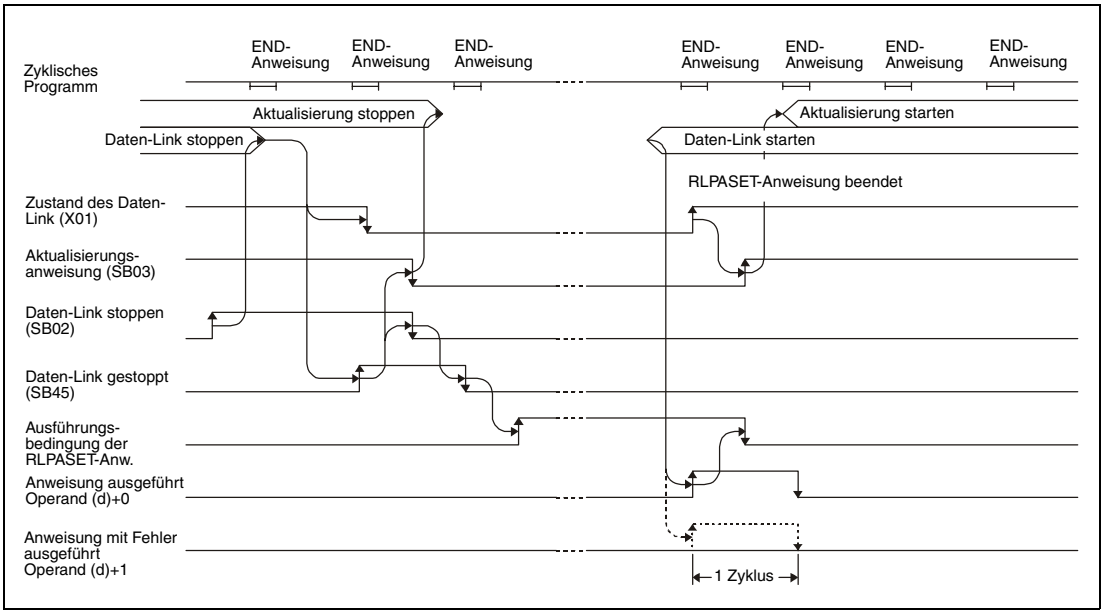
Beachten Sie bitte bei der Programmierung die benötigten Bereiche für s1 bis s5.

Beispiel:
An eine Master-Station sind vier Slave-Stationen angeschlossen. Bei der in der SPS eingesetzten Q02CPU stehen die Datenregister D0 bis D12287 zur Verfügung. Wird nun für s2 das Datenregister D12284 angegeben, weil nur 4 Slaves vorhanden sind, wird bei der Ausführung der RLPASET-Anweisung ein Fehler mit dem Code 4101 gemeldet, weil die CPU immer den Adressbereich für 64 Stationen prüft (in diesem Fall D12284 bis D12347) und der zur Verfügung stehende Adressbereich überschritten wird.

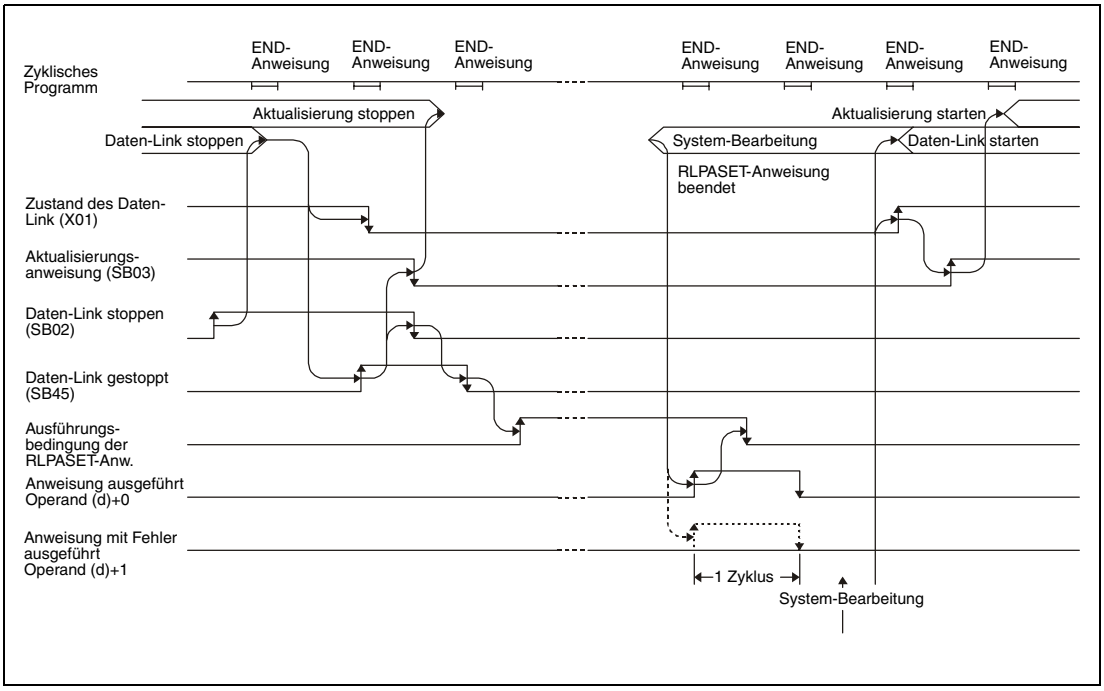
Durch die in (d)+0 und (d)+1 angegebenen Operanden wird angezeigt, ob die Anweisung ausgeführt wurde und ob dabei ein Fehler aufgetreten ist.

- Der Bit-Operand (d)+0 wird gesetzt, wenn die END-Anweisung des Zyklus ausgeführt wird, in dem die RLPASET-Anweisung beendet wurde. Bei der nächsten Bearbeitung der END-Anweisung wird dieser Bit-Operand wieder zurückgesetzt.
- Der Bit-Operand (d)+1 zeigt einen Fehler bei der Ausführung der RLPASET-Anweisung an. Bei fehlerfreier Ausführung bleibt dieser Bit-Operand zurückgesetzt. Bei einem Fehler jedoch wird (d)+1 gesetzt, wenn die END-Anweisung des Zyklus ausgeführt wird, in dem die RLPASET-Anweisung beendet wurde. Bei der nächsten Bearbeitung der END-Anweisung wird dieser Operand wieder zurückgesetzt.

Die folgende Abbildung zeigt den Signalverlauf bei Ausführung der RLPASET-Anweisung und fehlerfrei arbeitenden Stationen:



Das Verhalten bei Ausführung der RLPASET-Anweisung und gestörten Stationen ist in der folgenden Abbildung dargestellt:



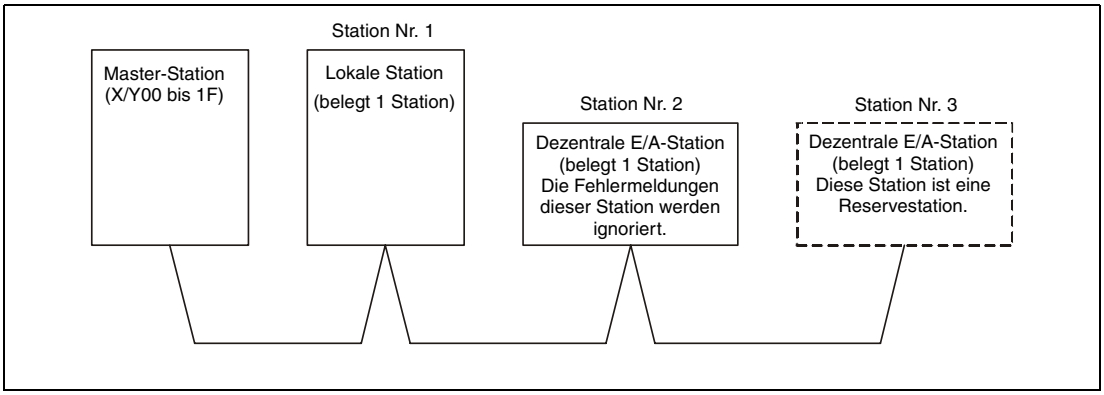
Fehlerquellen

Bei den folgenden Ereignissen bei der Ausführung der RLPASET-Anweisung wird das Error-Flag SM0 gesetzt und im Sonderregister SD0 ein Fehlercode eingetragen:

- Das in Un definierte Modul ist kein Sondermodul. (Fehlercode: 2112)
- Die auszuführende Anweisung wird nicht unterstützt. (Fehlercode: 4002)
- Die in der Anweisung angegebene Anzahl der Operanden ist ungültig. (Fehlercode: 4003)
- In der Anweisung ist ein unzulässiger Operand angegeben (Fehlercode: 4004)
- Die Anweisung enthält unzulässige Daten. (Fehlercode: 4100)
- Die angegebenen Operanden überschreiten den zulässigen Bereich. (Fehlercode: 4101)
- Die gespeicherten Daten oder Konstanten, die mit der Anweisung übertragen wurden, überschreiten den zulässigen Bereich. (Fehlercode: 4101)

Beispiel RLPASET

Das Beispiel-Programm überträgt die Netzwerk-Parameter für die Master-Station mit der E/A-Kopfadresse X/Y000. Das CC-Link-Netzwerk besteht aus drei Slave-Stationen:



Die Variablen s1 bis s5 der RLPASET-Anweisung enthalten die folgenden Werte:

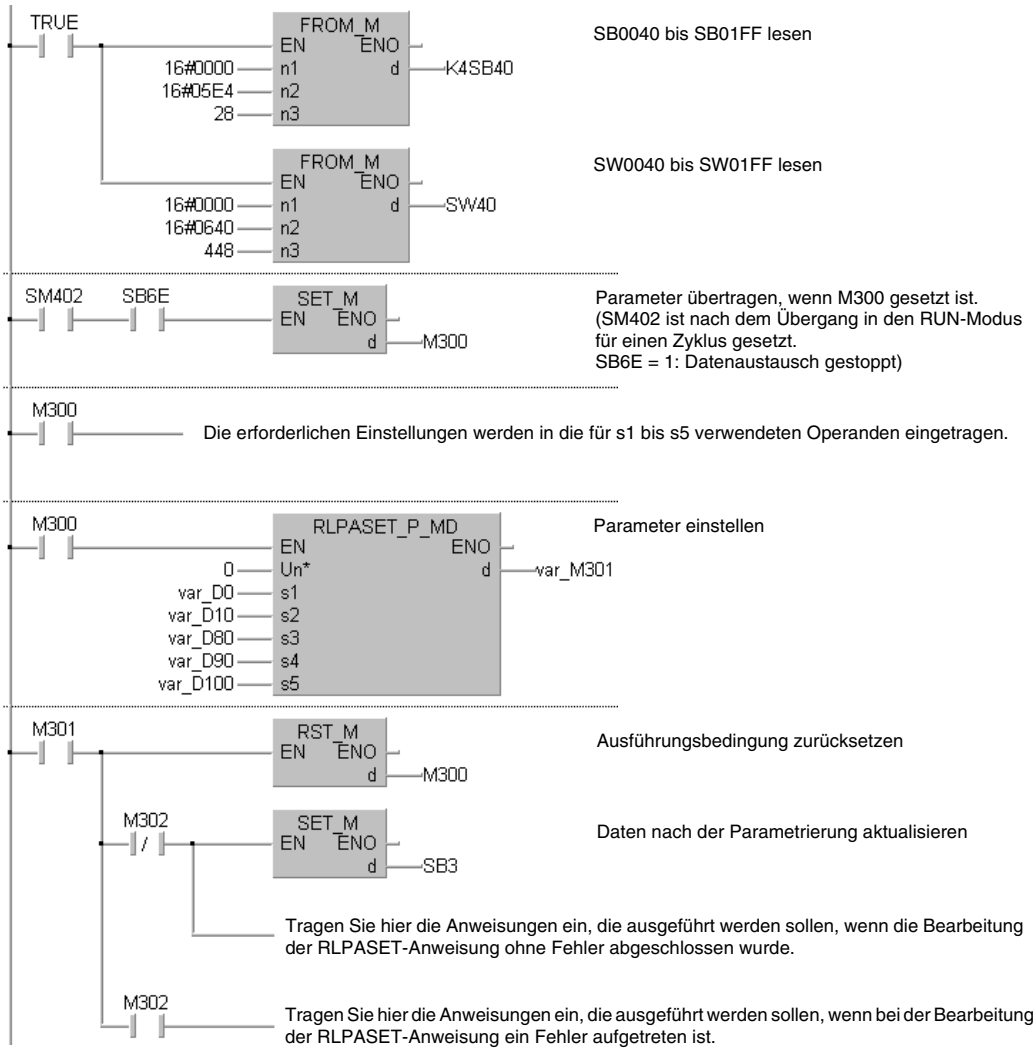
Parameter			Einstellung	Eingetragener Wert	Operand zur Speicherung
Daten zur Ausführung der Anweisung	(s1)+1	Umfang der Einstellungen	Alle Einstellungen sind gültig	15	D1
	(s1)+2	Anzahl der angeschlossenen Module	3 Slave-Module	3	D2
	(s1)+3	Anzahl der Wiederholungsversuche	3 Versuche	3	D3
	(s1)+4	Anzahl der Module mit autom.Wiedereingliederung	1 Modul	1	D4
	(s1)+5	Verhalten bei einem Stopp der CPU der SPS	Datenaustausch stoppen	0	D5
	(s1)+6	Abtastsynchronisation	Asynchon	0	D6
	(s1)+7	Verzögerungszeit	0 µs	0	D7
Einstellungen für Slave-Stationen	(s2)+0	Einstellungen für die erste Station	Lokale Station, belegt eine Station, Stations-Nr. 1	2101_H	D10
	(s2)+1	Einstellungen für die zweite Station	Dezentrale E/A-Station, eine Station belegt, Stations-Nr. 2	102_H	D11
	(s2)+2	Einstellungen für die dritte Station	Dezentrale E/A-Station, eine Station belegt, Stations-Nr. 3	103_H	D12
Reservierte Stationen	(s3)+0	Einstellungen zu reservierten Stationen	Station 3 reservieren (Bit 2 ist gesetzt)	4	D80
	(s3)+1			0	D81
	(s3)+2			0	D82
	(s3)+3			0	D83
Ignorierung gestörter Stationen	(s4)+0	Einstellungen zur Ignorierung der Fehlermeldungen gestörter Stationen	Fehlermeldungen von Station 2 ignorieren (Bit 1 gesetzt)	2	D90
	(s4)+1			0	D91
	(s4)+2			0	D92
	(s4)+3			0	D93

Parameter			Einstellung	Eingetragener Wert	Operand zur Speicherung
Einstellung der Puffergröße	(s5)+0	Sendepuffer der ersten lokalen Station (Station Nr. 1)	100 Worte	64_H	D100
	(s5)+1	Empfangspuffer der ersten lokalen Station (Station Nr. 1)	100 Worte	64_H	D101
	(s5)+2	Automatisch aktualisierter Puffer der ersten lokalen Station (Station Nr. 1)	Wird nicht verwendet	0_H	D102

Vor der Ausführung der RLPASET-Anweisung müssen die Werte der Parameter entsprechend der Tabelle in die Datenregister D1 bis D102 eingetragen werden.

- IEC-Editoren (Auf den folgenden Seiten ist dieses Beispiel in der MELSEC-Anweisungliste und dem Kontaktplan des GX Developers dargestellt.)

Kontaktplan (GX IEC Developer)



IEC-Anweisungsliste

```
LD              TRUE
FROM_M          16#0000,    16#05E4,   28,     K4SB40
FROM_M          16#0000,    16#0640,   448,    SW40
```

```
LD              SM402
AND             SB6E
SET_M           M300
```

Die Operanden und Anweisungen sind beim Programmbeispiel für den Kontaktplan erläutert.

```
LD              M300
```

Die erforderlichen Einstellungen werden mit MOV-Anweisungen
in die für s1 bis s5 verwendeten Operanden eingetragen.

```
LD              M300
RLPASET_P_MD    0, var_D0, var_D10, var_D80, var_D90, var_D100, var_M301
```

```
LD              M301
RST_M           M300
LD              M301
ANDN            M302
SET_M           SB3
```

Ausführung, wenn die Bearbeitung der RLPASET-Anweisung ohne Fehler abgeschlossen wurde

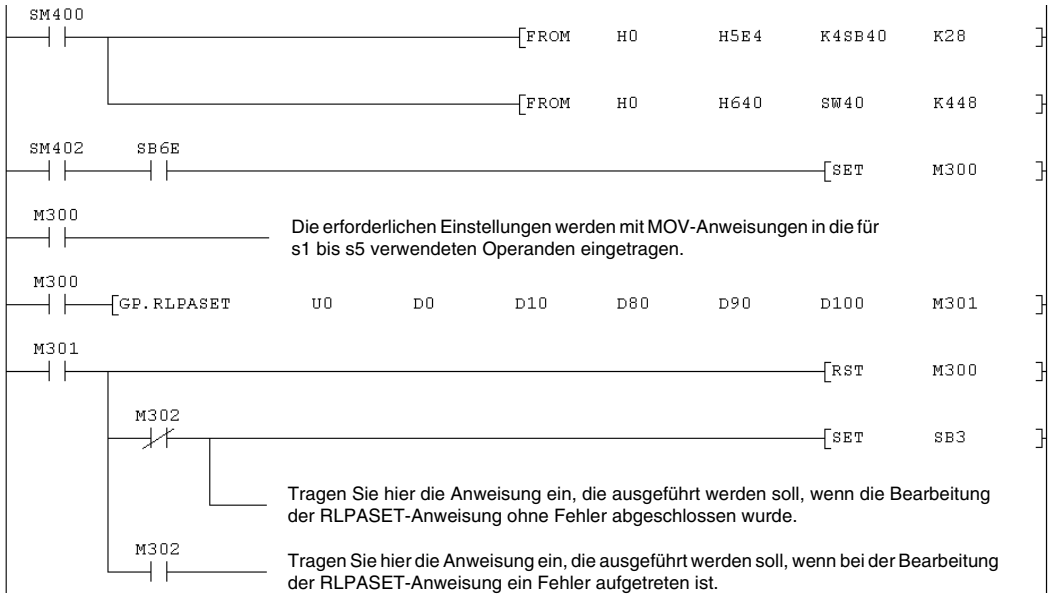
```
LD              M301
AND             M302
```

Ausführung, wenn ein Fehler bei der Bearbeitung der RLPASET-Anweisung aufgetreten ist

HINWEIS *Bei den IEC-Editoren des GX IEC Developers müssen die Variablen im Header der Programmorganisationseinheit (POE) definiert werden. Ohne Variablendefinition werden beim Überprüfen oder Kompilieren des Programms Fehler gemeldet.*
Weitere Informationen finden Sie im Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung.

- MELSEC-Anweisungliste und Kontaktplan des GX Developers
  Die Operanden und Anweisungen sind beim Programmbeispiel für den Kontaktplan des GX IEC Developers erläutert.

Kontaktplan (GX Developer)



MELSEC-Anweisungsliste

MELSEC	LD	SM400
	FROM	H0
		H5E4
		K4SB40
		K28
	FROM	H0
		H640
		SW40
		K448
MELSEC	LD	SM402
	AND	SB6E
	SET	M300
MELSEC	LD	M300
	Die erforderlichen Einstellungen werden mit MOV-Anweisungen in die für s1 bis s5 verwendeten Operanden eingetragen.	
MELSEC	LD	M300
	GP.RLPASET	U0
		D0
		D10
		D80
		D90
		D100
		M301
MELSEC	LD	M301
	RST	M300
	MPS	
	ANI	M302
	SET	SB3
	Ausführung, wenn die Bearbeitung der RLPASET-Anweisung ohne Fehler abgeschlossen wurde	
	MPP	
	AND	M302
	Ausführung, wenn ein Fehler bei der Bearbeitung der RLPASET-Anweisung aufgetreten ist	

11.5.3 RRPA (A-Serie)

CPU

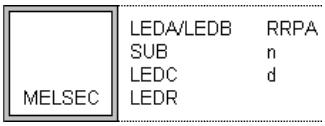
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●		

Operanden MELSEC A

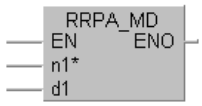
	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)								Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9011
n																	●	●					20			●
d								●	●	●	●	●														

GX IEC Developer

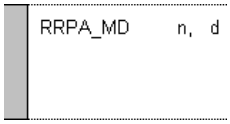
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



GX Developer

Hinweise zur Programmierung der erweiterten Anweisungen in den MELSEC-Editoren finden Sie im Kapitel 3.3 dieses Handbuchs.

Variablen

Operand	Bedeutung				Werte-bereich	Festlegung des Inhalts durch	Datentyp
n	Kopfadresse des CC-Link-Moduls auf dem Baugruppenträger (Es werden nur die ersten beiden Stellen der 3-stelligen Adresse angegeben. Beispiel: Kopfadresse X/Y100 -> 10_H)				0 bis FE_H	Anwender	BIN-16-Bit
d	Erster Operand des Bereiches mit Netzwerkparametern						BIN-16-Bit
	Operand	**Bedeutung**		**Beschreibung**		**Festlegung des Inhalts durch**	
	d+0	RX	Anfangs-adresse	Anfangsadresse der dezentralen Eingänge (RX) im Master- oder lokalem Modul		System	
	d+1		Operand in der CPU	Codierte Angabe, welche Operanden in der CPU automatisch aktualisiert werden (siehe unten)		Anwender	
	d+2		Anfangsadr. in der CPU	Erste Adresse des Operandenbereiches in der CPU*			
	d+3		Anzahl der Operanden	Anzahl der Operanden, die zwischen CC-LINK-Modul und CPU ausgetauscht werden*			
	d+4	RY	Anfangs-adresse	Anfangsadresse der dezentralen Ausgänge (RY) im Master- oder lokalem Modul		Anwender	
	d+5		Operand in der CPU	Codierte Angabe, welche Operanden in der CPU automatisch aktualisiert werden (siehe unten)			
	d+6		Anfangsadr. in der CPU	Erste Adresse des Operandenbereiches in der CPU*			
	d+7		Anzahl der Operanden	Anzahl der Operanden, die zwischen CC-LINK-Modul und CPU ausgetauscht werden*			
	d+8	RW	Anfangs-adresse	Anfangsadresse der dezentralen Register (RW) im Master- oder lokalem Modul		RWr: System	
						RWw: Anwender	
	d+9		Operand in der CPU	Codierte Angabe, welche Operanden in der CPU automatisch aktualisiert werden (siehe unten)		Anwender	
	d+10		Anfangsadr. in der CPU	Erste Adresse des Operandenbereiches in der CPU*			
	d+11		Anzahl der Operanden	Anzahl der Operanden, die zwischen CC-LINK-Modul und CPU ausgetauscht werden*			
	d+12	SB	Anfangs-adresse	Anfangsadresse der Link-Sondermerker (SB) im Master- oder lokalem Modul		System	
	d+13		Operand in der CPU	Codierte Angabe, welche Operanden in der CPU automatisch aktualisiert werden (siehe unten)		Anwender	
	d+14		Anfangsadr. in der CPU	Erste Adresse des Operandenbereiches in der CPU*			
	d+15		Anzahl der Operanden	Anzahl der Operanden, die zwischen CC-LINK-Modul und CPU ausgetauscht werden*			
	d+16	SW	Anfangs-adresse	Anfangsadresse der Link-Sonderregister (SW) im Master- oder lokalem Modul		System	
	d+17		Operand in der CPU	Codierte Angabe, welche Operanden in der CPU automatisch aktualisiert werden (siehe unten)		Anwender	
	d+18		Anfangsadr. in der CPU	Erste Adresse des Operandenbereiches in der CPU*			
	d+19		Anzahl der Operanden	Anzahl der Operanden, die zwischen CC-LINK-Modul und CPU ausgetauscht werden*			

* Bei Bit-Operanden (X, Y, M, B) muss als Anfangsadresse und als Anzahl entweder „0“ oder eine durch 16 teilbare Zahl angegeben werden. (Wird als Anzahl „0“ vorgegeben, wird der entsprechende Operand nicht automatisch aktualisiert.)
Falls das nicht beachtet wird, tritt ein Fehler auf.

In d+5, d+13 usw. wird festgelegt, welche Operanden in der CPU den Operanden des CC-Link-Moduls entsprechen. Beispielsweise können Merker (M) die Zustände der dezentralen Eingänge (RX) wiedergeben.

Code	Operand	Code	Operand
0	—	5	T
1	X	6	C
2	Y	7	D
3	M	8	W
4	B	9	R

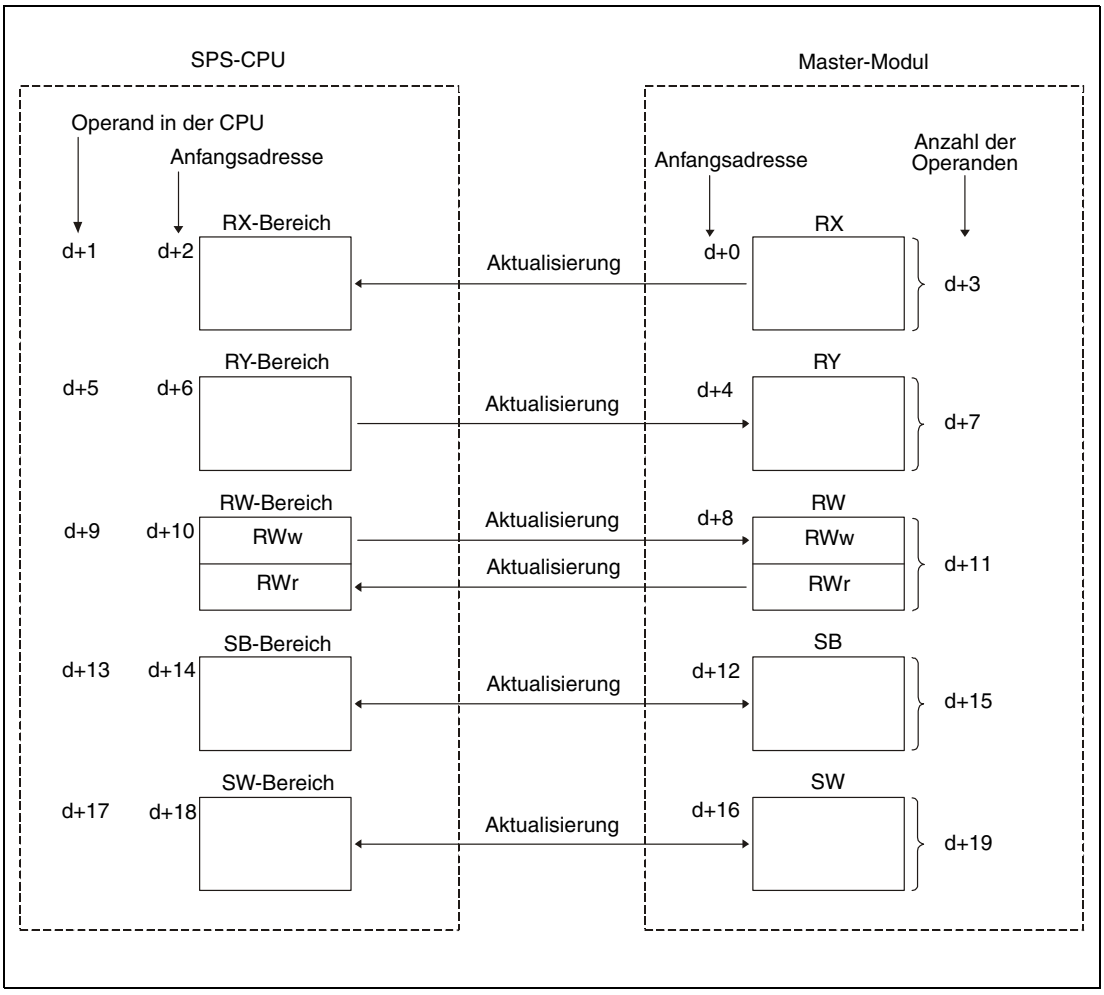
Funktions-weise

Parameter für automatische Aktualisierung einstellen

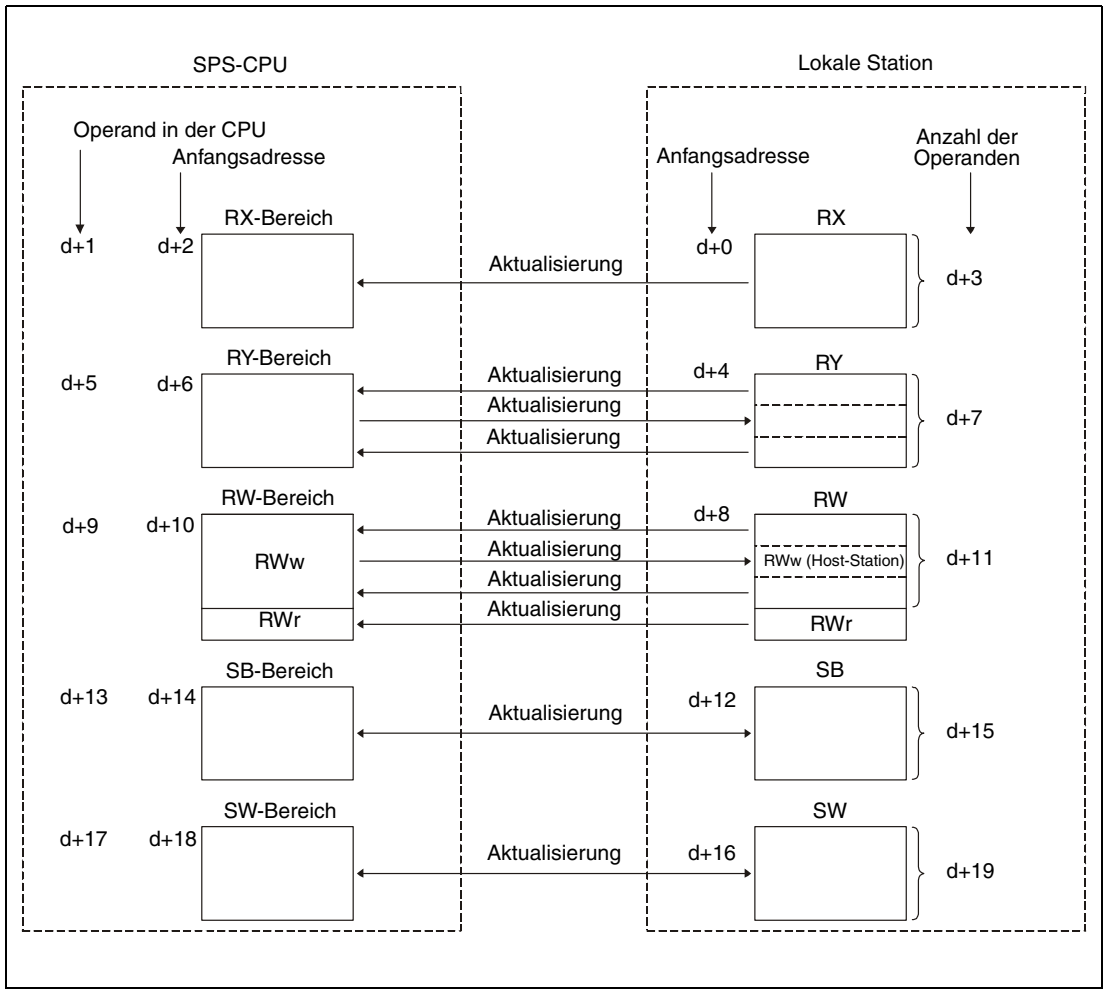
RRPA Parameter einstellen

Mit der RRPA-Anweisung werden die Operanden und deren Anzahl eingestellt, deren Zustände zwischen der SPS-CPU und dem Master- oder lokalen Modul automatisch ausgetauscht werden.
Falls zum Datenaustausch zwischen CPU und Master-/lokalem Modul FROM- und TO-Anweisungen verwendet werden, wird die RRPA-Anweisung nicht benötigt.

- Kommunikation zwischen der SPS-CPU und der Master-Station:



- Kommunikation zwischen der SPS-CPU und einer lokalen Station:

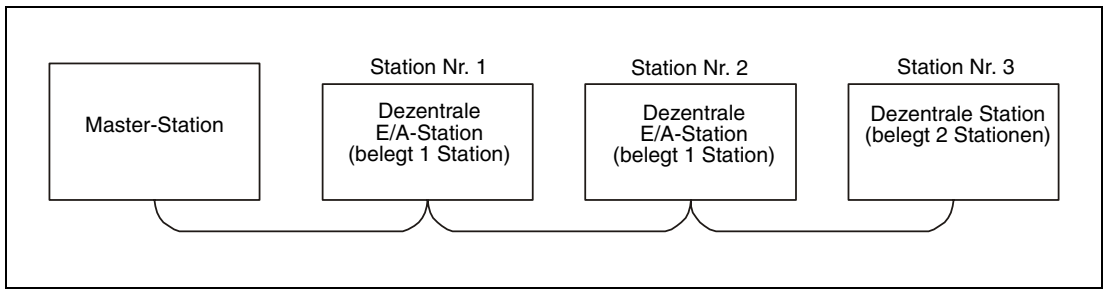


Bei der Ausführung der RRPA-Anweisung werden die Parameter für die automatische Aktualisierung in die CPU eingetragen und anschließend mit diesen Einstellungen der Datenaustausch zwischen der CPU und dem Master- oder lokalem Modul ausgeführt.

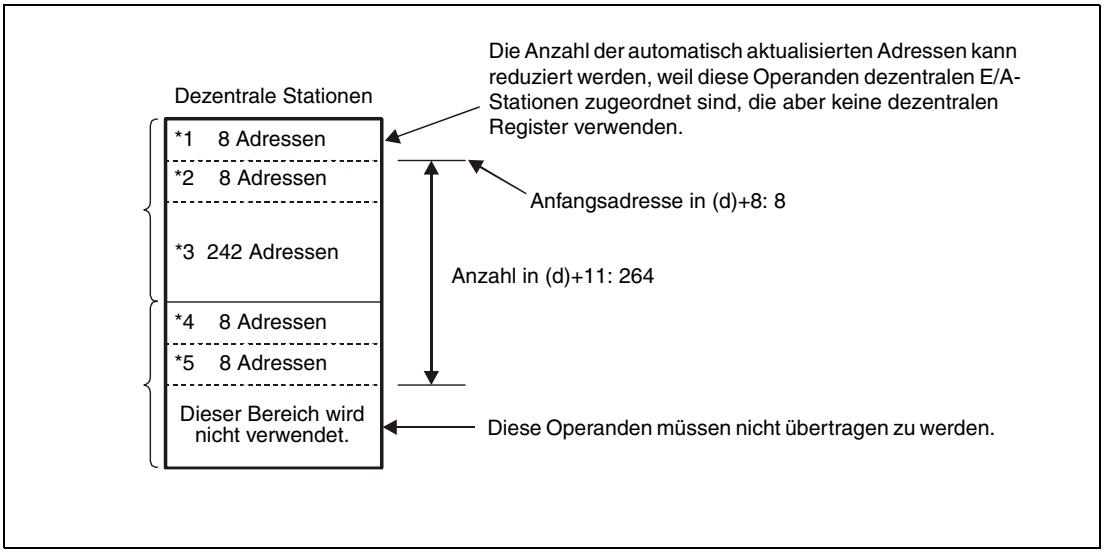
Die RRPA-Anweisung muss nur einmal ausgeführt werden. Werden mehrere RRPA-Anweisungen für ein Modul programmiert, sind nur die mit der ersten Anweisung festgelegten Einstellungen gültig. Zur Änderung der Parameter wird die RRPA-Anweisung mit neuen Einstellungen ausgeführt. Zur Übernahme der neuen Parameter muss die SPS-CPU dann in die Betriebsart STOP/PAUSE und anschließend in die Betriebsart RUN geschaltet werden.

Um alle dezentralen Register (RWw und RWr) zu aktualisieren, tragen Sie in d+8 als Anfangsadresse den Wert „0“ und in d+11 als Anzahl den Wert „512“ ein.

HINWEIS *Das folgende Beispiel erläutert die Vorgehensweise bei der Aktualisierung der dezentralen Register:*



Innerhalb der dezentralen Register werden alle 256 Worte (64 Stationen) für RWw belegt, auch wenn weniger als 64 Stationen angeschlossen sind. Der Bereich für RWr beginnt aus diesem Grund nach den 256 RWw-Adressen.



**1: RWw-Bereich (8 Operanden) der Stationen 1 und 2 (Dezentrale E/A-Stationen)*
**2: RWw-Bereich (8 Operanden) der Station 3 (Dezentrale Station)*
**3: 242 Operanden des RWw-Bereichs werden automatisch durch das System belegt*
**4: RWr-Bereich (8 Operanden) der Stationen 1 und 2 (Dezentrale E/A-Stationen)*
**5: RWr-Bereich (8 Operanden) der Station 3 (Dezentrale Station)*

Aktualisierung der Operanden SB und SW:

- Ordnen Sie den Link-Sondermerkern (SB) und -Sonderregistern (SW) Operanden in der SPS-CPU zu. Beachten Sie dabei die Anfangsadressen der Bereiche: Die Operanden SB0000 bis SB003F werden von der SPS-CPU zum Master-Modul und die Operanden SB0040 bis SB00FF werden vom Master-Modul zur CPU übertragen.
- File-Register können nicht als zu aktualisierende Operanden für SB und SW angegeben werden. Werden File-Register für SB oder SW angegeben und diese Einstellung in die CPU übertragen, wird ein Anweisungs-Code-Fehler erkannt und die CPU gestoppt.
- Die den Link-Sondermerkern (SB) und -Sonderregistern (SW) zugeordneten Operanden sollten nicht als Latch-Bereich definiert werden. Andernfalls wird beim Einschalten der Spannungsversorgung oder Zurücksetzen der CPU der normale Betrieb wegen undefinierter Daten nicht erreicht.
- Die mit der RRPA-Anweisung beim Einschalten der Spannung eingestellten Operanden für SB und SW können nicht verändert werden.

Ausführungsbedingungen

Wird die RRPA-Anweisung in Verbindung mit einer LEDA-Anweisung programmiert, wird die RRPA-Anweisung ausgeführt, solange die Ausführungsbedingung der LEDA-Anweisung eingeschaltet ist. Bei Anwendung einer LEDB-Anweisung dagegen wird der RRPA-Anweisung nur bei der steigenden Flanke der Ausführungsbedingung ausgeführt.

Fehlerquellen

Bei den folgenden Ereignissen wird ein Verarbeitungsfehler erkannt, das Error-Flag M9011 gesetzt und der Fehlercode „50“ in D908 eingetragen. (Bei einer AnU-CPU wird in D9001 und bei einer AnSH-CPU in D9092 der Code „503“ eingetragen.)

- Der Operanden-Code ist „0“ oder eine Zahl außerhalb des Bereichs von 1 bis 9.
- Die Anfangsadresse bei Bit-Operanden ist nicht „0“ oder ist nicht durch 16 teilbar.
- Die Anzahl der aktualisierten Adressen ist kein Mehrfaches von 16.

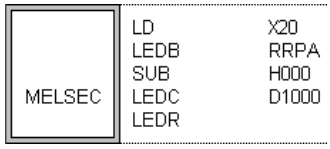
Beispiel RRPA

Das Beispiel-Programm überträgt die Parameter zur automatischen Aktualisierung, die ab dem File-Register D1000 gespeichert sind, in die Master-Station mit der E/A-Kopfadresse X/Y000.

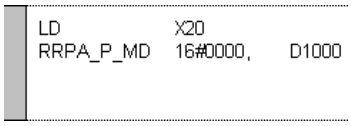
Parameter		Einstellung	Eingetragener Wert	Operand zur Speicherung
RX	Anfangsadresse	0	0_H	D1000
	Operand in der CPU	X (Code: 1)	1_H	D1001
	Anfangsadresse in der CPU	XA0	$A0_H$	D1002
	Anzahl der Operanden	32	32	D1003
RY	Anfangsadresse	0	0_H	D1004
	Operand in der CPU	Y (Code: 2)	2_H	D1005
	Anfangsadresse in der CPU	YA0	$A0_H$	D1006
	Anzahl der Operanden	48	48	D1007
RW	Anfangsadresse	0	0_H	D1008
	Operand in der CPU	D (Code: 7)	7_H	D1009
	Anfangsadresse in der CPU	D160	160	D1010
	Anzahl der Operanden	272	272	D1011
SB	Anfangsadresse	0	0_H	D1012
	Operand in der CPU	M (Code: 3)	3_H	D1013
	Anfangsadresse in der CPU	M160	160	D1014
	Anzahl der Operanden	256	256	D1015
SW	Anfangsadresse	0	0_H	D1016
	Operand in der CPU	W (Code: 8)	8_H	D1017
	Anfangsadresse in der CPU	WA0	$A0_H$	D1018
	Anzahl der Operanden	256	256	D1019

Vor der Ausführung der RRPA-Anweisung müssen die Werte der Parameter entsprechend der Tabelle in die Datenregister D1000 bis D1019 eingetragen werden.

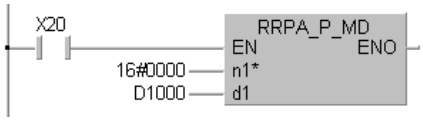
MELSEC-Anweisungsliste



IEC-Anweisungsliste



Kontaktplan (GX IEC Developer)



Kontaktplan (GX Developer)

11.5.4 RIRD (A-Serie)

CPU

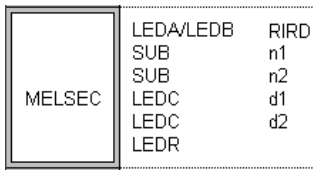
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●		

Operanden MELSEC A

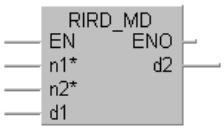
	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)									Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9011
n1																	●	●					26			●
n2																	●	●								
d1								●	●	●	●	●														
d2		●	●	●	●	●																				

GX IEC Developer

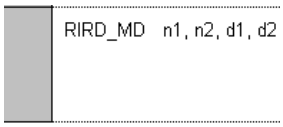
MELSEC-Anweisungsliste



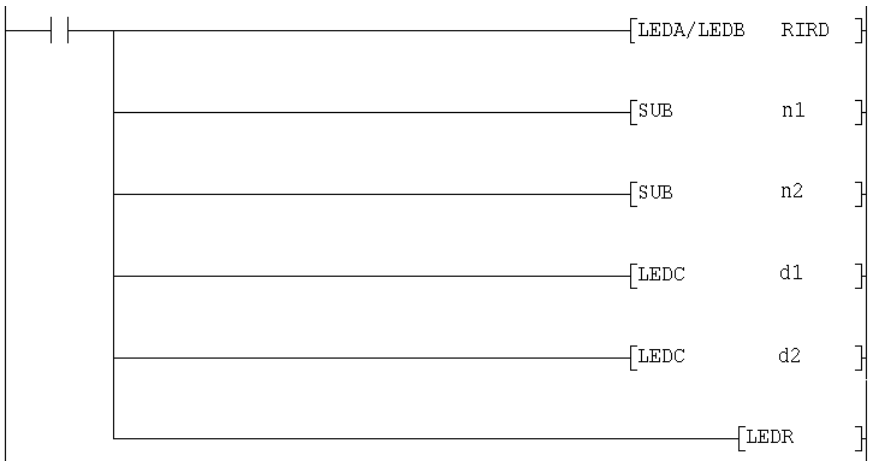
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

Operand	Bedeutung	Wertebereich	Festlegung des Inhalts durch	Datentyp
n1	Kopfadresse der CC-Link-Master- oder lokalen Station (Es werden nur die ersten beiden Stellen der 3-stelligen Adresse angegeben. Beispiel: Kopfadresse X/Y100 -> 10_H)	0 bis FE_H	Anwender	BIN-16-Bit
n2	Stationsnummer der Station, aus der Daten gelesen werden sollen Wertebereich: Bei Ausführung der RIRD-Anweisung in der Master-Station: 1 bis 64, bei Ausführung der RIRD-Anweisung in einer lokalen Station: 0 bis 64	siehe nebenstehende Erläuterung	Anwender	BIN-16-Bit

Variablen

<table>
<tr><th>Operand</th><th colspan="4">Bedeutung</th><th>Werte-bereich</th><th>Festlegung des Inhalts durch</th><th>Datentyp</th></tr>
<tr><td rowspan="10">d1</td><td colspan="4">Erster Operand des Bereiches mit Informationen zur Ausführung der Anweisung und zur Speicherung der gelesenen Daten</td><td></td><td></td><td rowspan="10">BIN-16-Bit</td></tr>
<tr><td>Operand</td><td>Bedeutung</td><td>Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(d1)+0</td><td>Ausführungs-status der Anweisung</td><td>Zeigt an, ob bei der Bearbeitung der Anweisung ein Fehler aufgetreten ist:
0000_H: Fehlerfreie Bearbeitung
Jeder andere Wert als 0000_H: Bei der Bearbeitung ist ein Fehler aufgetreten.
Nähere Hinweise zu Fehlercodes finden Sie in der Bedienungsanleitung des CC-Link-Moduls.</td><td>—</td><td>System</td></tr>
<tr><td>(d1)+1</td><td>Datenlänge</td><td>Geben Sie hier an, wieviele Daten (Worte) gelesen werden sollen. Die Datenlänge hängt davon ab, mit welcher CPU die Station ausgestattet ist, aus der gelesen wird:
AnU-, QnA-Serie, System Q: max. 480 Worte
Alle anderen CPUs: max. 32 Worte</td><td>1 bis 480
1 bis 32</td><td rowspan="4">Anwender</td></tr>
<tr><td rowspan="2">(d1)+2</td><td rowspan="2">Zugriffscode</td><td rowspan="2">• Bei einem Master-Modul mit der Software-Version A bis H
Bei der Angabe von 0004_H wird aus dem Pufferspeicher einer intelligenten Station gelesen. Tragen Sie den Wert 2004_H ein, wenn auf dem Pufferspeicher einer lokalen Station gelesen werden soll.</td><td rowspan="2">0004_H
oder
2004_H</td></tr>
<tr><td rowspan="2">Operanden- und Zugriffscode</td><td rowspan="2">• Bei einem Master-Modul ab der Software-Version J
In das höherwertige Byte dieses Operanden wird der Operandencode eingetragen.
Das niederwertige Byte dieses Operanden dient zur Festlegung, ob auf den Pufferspeicher eines CC-Link-Moduls (04_H) oder auf eine CPU (05_H) zugegriffen wird.</td></tr>
<tr><td>Höherwertiges Byte: siehe folgende Tabelle</td></tr>
<tr><td>(d1)+3</td><td>Anfangsadresse</td><td>• Bei einem Master-Modul mit der Software-Version A bis H
Anfangsadresse im Pufferspeicher
• Bei einem Master-Modul ab der Software-Version J
Anfangsadresse im Pufferspeicher oder erste Operandenadresse.</td><td>Niederwertiges Byte: 04_H oder 05_H
Abhängig von der Station, auf die zugegriffen wird</td></tr>
<tr><td>(d1)+4
bis
(d1)+n</td><td>Speicherbereich für die gelesenen Daten</td><td>Die Größe dieses Bereiches wird durch die in (d1)+1 angegebene Datenmenge bestimmt.</td><td>—</td><td>System</td></tr>
</table>

Variablen

<table>
<tr><th>Operand</th><th colspan="4">Bedeutung</th><th>Werte-bereich</th><th>Festlegung des Inhalts durch</th><th>Datentyp</th></tr>
<tr><td rowspan="4">d2</td><td colspan="4">Bit-Operand, der nach der Ausführung der RIRD-Anweisung für einen Zyklus gesetzt wird. Mit (d2)+1 wird ein Fehler bei der Ausführung angezeigt.</td><td></td><td></td><td rowspan="4">Bit</td></tr>
<tr><td>Operand</td><td>Bedeutung</td><td>Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(d2)+0</td><td>Anweisung ausgeführt</td><td>Zeigt die Beendigung der Anweisung an.
EIN: Anweisung ausgeführt
AUS: Anweisung nicht ausgeführt</td><td>—</td><td rowspan="2">System</td></tr>
<tr><td>(d2)+1</td><td>Anweisung mit Fehler ausgeführt</td><td>Zeigt an, ob bei der Ausführung der Anweisung ein Fehler aufgetreten ist.
EIN: Anweisung mit Fehler ausgeführt
AUS: Anweisung ohne Fehler ausgeführt</td><td>—</td></tr>
</table>

Ab der Software-Version J des Master-Moduls wird durch die Angabe eines Operanden- und Zugriffscodes in (d1)+2 festgelegt, aus welchem Teil des Pufferspeichers gelesen bzw. welche Operanden der CPU erfasst werden:

- Zugriff auf den Pufferspeicher eines CC-Link-Moduls (Zugriffscode 04_H)

Zugriff auf		Operandencode
Pufferspeicher in intelligenten Stationen		00_H
Pufferspeicher in Master- oder lokalen Station	Pufferspeicher mit freiem Zugriff	20_H
	Dezentrale Eingänge	21_H
	Dezentrale Ausgänge	22_H
	Dezentrale Register	24_H
	Link-Sondermerker	63_H
	Link-Sonderregister	64_H

- Zugriff auf Operanden in der CPU (Zugriffscode 05_H)
  Auf Operanden, die hier nicht aufgeführt sind, kann nicht zugegriffen werden. Geben Sie beim Zugriff auf Bit-Operanden eine Adresse an, die entweder 0 oder durch 16 teilbar ist.

Operand		Operandentyp		Operandencode
Bezeichnung	Symbol	Bit	Wort	
Eingänge	X	●		00_H
Ausgänge	Y	●		02_H
Merker	M	●		03_H
Latch-Merker	L	●		83_H
Link-Merker	B	●		23_H
Timer (Kontakt)	T	●		09_H
Timer (Spule)		●		$0A_H$
Timer (Aktueller Wert)			●	$0C_H$
Counter (Kontakt)	C	●		11_H
Counter (Spule)		●		12_H
Counter (Aktueller Wert)			●	14_H
Datenregister	D		●	04_H
Link-Register	W		●	24_H
File-Register	R		●	84_H

Funktionsweise

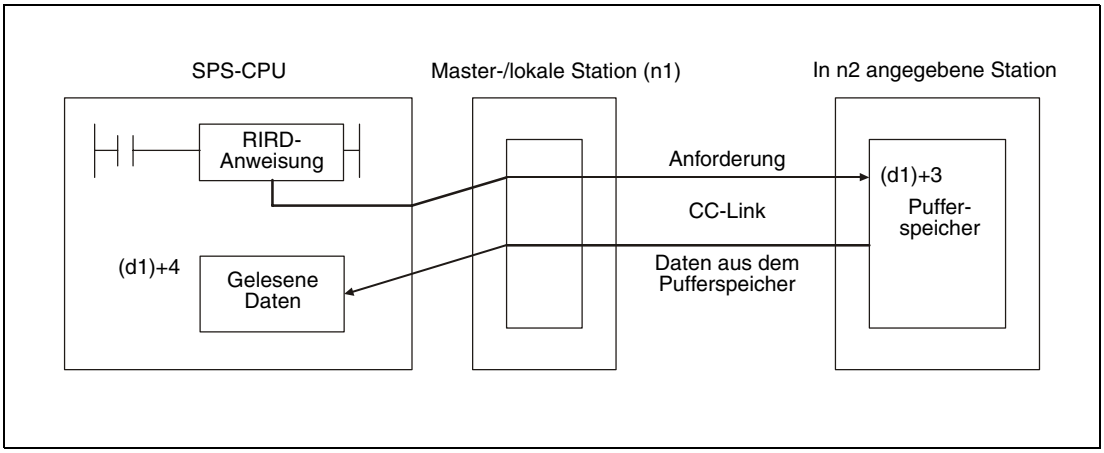
Daten aus dem Pufferspeicher einer andernen Station oder aus einer CPU lesen

RIRD Daten lesen

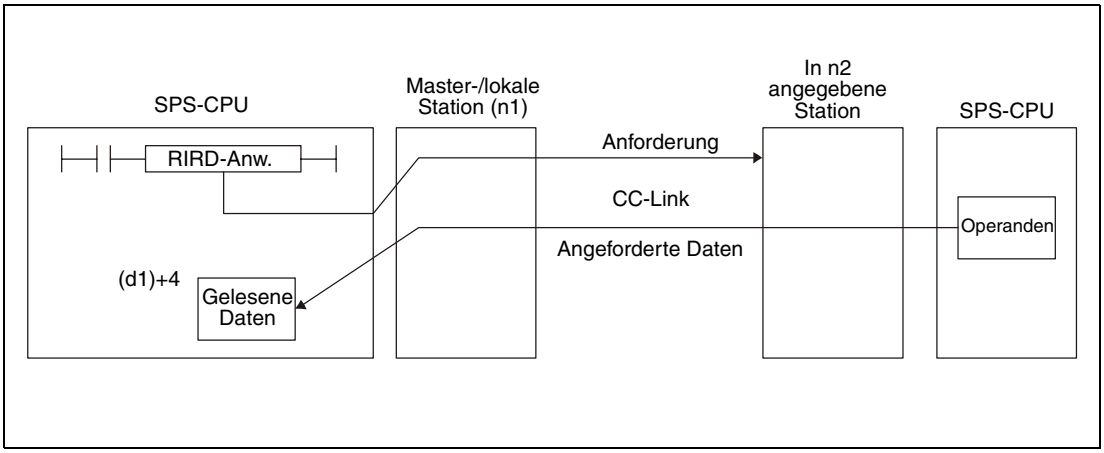
Mit der RIRD-Anweisung können Daten aus dem Pufferspeicher einer anderen Station am CC-Link gelesen werden. Bei Master-Modulen ab der Software-Version J ist auch der Zugriff auf Operanden in der SPS-CPU der anderen Station möglich.

In dem Operanden (d1)+3 wird die erste Pufferspeicheradresse oder der erste Operand angegeben, der gelesen werden soll. Der Operand n2 enthält die Stationsnummer der anderen Station. Diese Station ist an die Master/lokale Station aus n1 angeschlossen. Die ausgelesenen Daten werden in der CPU, in der die RIRD-Anweisung bearbeitet wird, ab (d1)+4 gespeichert. Der Operand (d1)+1 enthält die Angabe, wieviele Daten übertragen werden sollen.

- Funktionsschema bei den Software-Versionen A bis H:



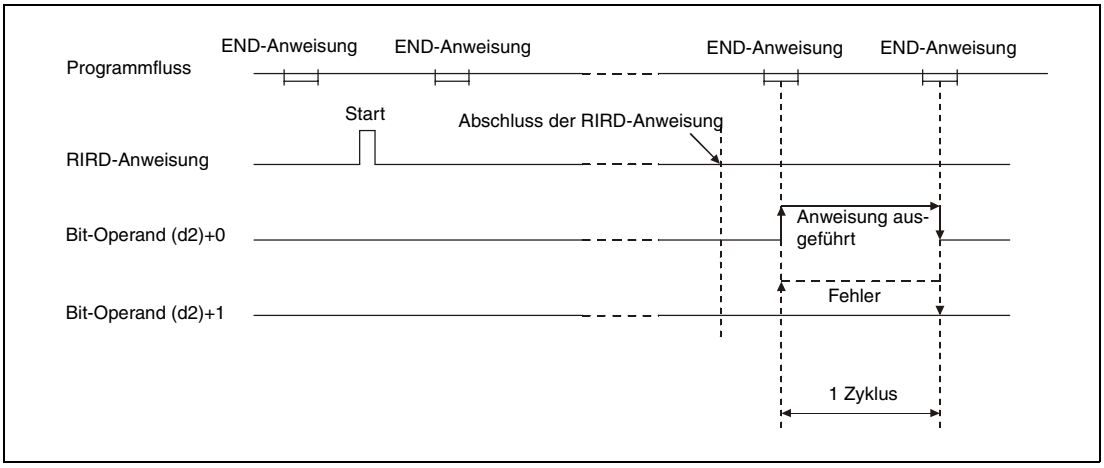
- Zusätzliche Funktion ab der Software-Version J:



Durch die in (d2)+0 und (d2)+1 angegebenen Operanden wird angezeigt, ob die Anweisung ausgeführt wurde und ob dabei ein Fehler aufgetreten ist.

- Der Bit-Operand (d2)+0 wird gesetzt, wenn die END-Anweisung des Zyklus ausgeführt wird, in dem die RIRD-Anweisung beendet wurde. Bei der nächsten Bearbeitung der END-Anweisung wird dieser Bit-Operand wieder zurückgesetzt.
- Der Bit-Operand (d2)+1 zeigt einen Fehler bei der Ausführung der RIRD-Anweisung an. Bei fehlerfreier Ausführung bleibt dieser Bit-Operand zurückgesetzt. Bei einem Fehler jedoch wird (d2)+1 gesetzt, wenn die END-Anweisung des Zyklus ausgeführt wird, in dem die RIRD-Anweisung beendet wurde. Bei der nächsten Bearbeitung der END-Anweisung wird dieser Operand wieder zurückgesetzt.

Die folgende Abbildung zeigt den Signalverlauf bei Ausführung der RIRD-Anweisung:



Für mehrere Stationen kann gleichzeitig eine RIRD-Anweisung ausgeführt werden. Auf dieselbe intelligente- oder lokale Station kann jedoch nicht gleichzeitig mit mehreren RIRD-Anweisungen von verschiedenen Stationen aus zugegriffen werden.

Die Netzwerkparameter müssen über die RLPA-Anweisung eingestellt sein, bevor die RIRD-Anweisung ausgeführt werden kann.

Wird als Anzahl der zu lesenden Daten in (d1)+1 eine 0 oder ein Wert außerhalb des Bereichs von 1 bis 480 angegeben, wird beim Abschluss der RIRD-Anweisung durch den Operanden (d2)+1 ein Fehler gemeldet.

Ausführungsbedingungen

Wird die RIRD-Anweisung in Verbindung mit einer LEDA-Anweisung programmiert, wird die RIRD-Anweisung ausgeführt, solange die Ausführungsbedingung der LEDA-Anweisung eingeschaltet ist. Bei Anwendung einer LEDB-Anweisung dagegen wird der Lesevorgang nur bei der steigenden Flanke der Ausführungsbedingung ausgeführt.

Beachten Sie, dass für die Bearbeitung der RIRD-Anweisung mehrere Zyklen benötigt werden. Starten Sie daher den nächsten Lesevorgang erst dann, nachdem durch den Operanden (d2)+0 angezeigt wurde, dass die Bearbeitung der RIRD-Anweisung abgeschlossen ist.

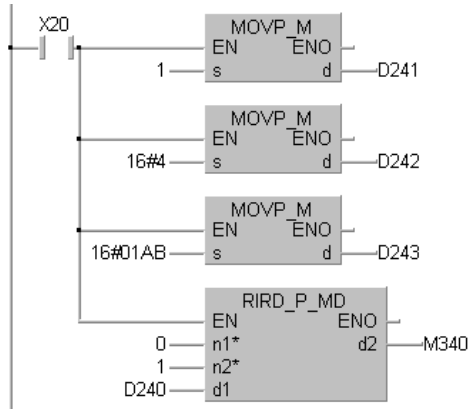
Beispiel RIRD

Das folgende Programm wird in der SPS-CPU der Master-Station bearbeitet und liest aus einer intelligenten Station mit der Stationsnummer 1 den Inhalt der Pufferspeicheradresse $1A8_H$ aus. Das CC-Link-Modul der Master-Station belegt den Adressbereich von X/Y000 bis X/Y01F.

MELSEC-Anweisungsliste

```
MELSEC  LD      X20
        MOVP    K1      D241
        MOVP    H4      D242
        MOVP    H01AB   D243
        LEDB    RIRD
        SUB     H0
        SUB     K1
        LEDC    D240
        LEDC    M340
        LEDR
```

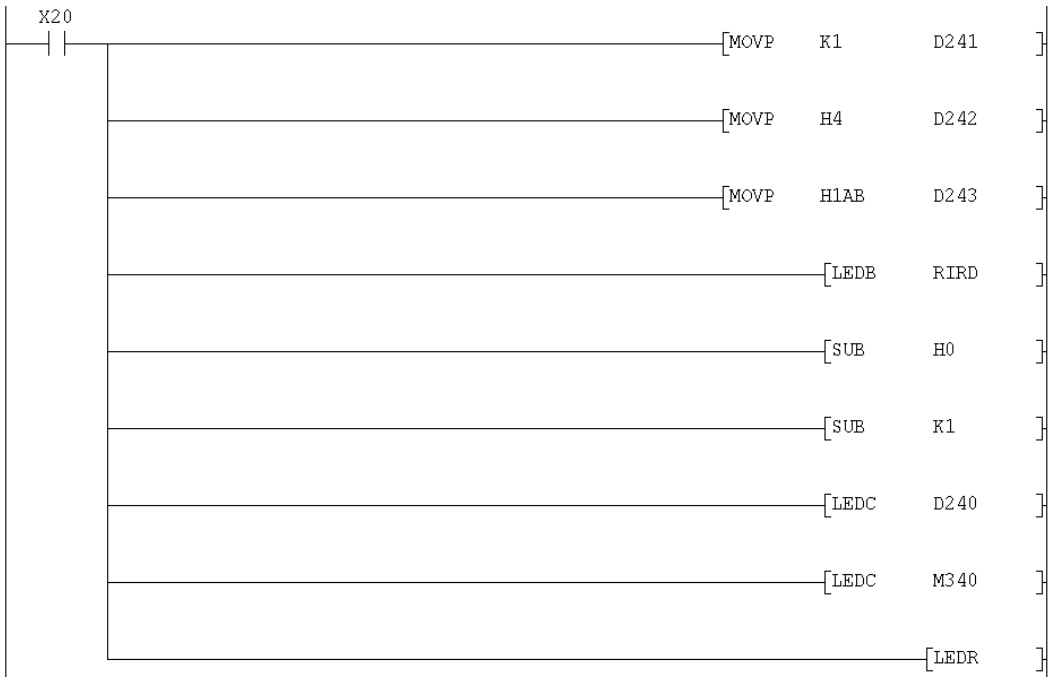
Kontaktplan (GX IEC Developer)



IEC-Anweisungsliste

```
LD          X20
MOVP_M      1,          D241
MOVP_M      16#4,       D242
MOVP_M      16#01AB,    D243
RIRD_P_MD   0,          1,      D240,   M340
```

Kontaktplan (GX Developer)



Hinweise zur Programmierung der erweiterten Anweisungen in den MELSEC-Editoren bei einer SPS der A-Serie finden Sie im Kapitel 3.3 dieses Handbuchs.

11.5.5 RIRD (QnA-Serie und System Q)

CPU

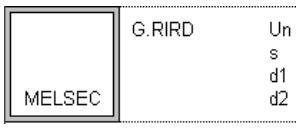
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Operanden MELSEC Q

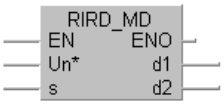
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□N□		Sondermodule U□NG□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	—	—	—	—	—	SM0	8
d1	—	●	●	—	—	—	—	—	—		
d2	●	●	●	—	—	—	—	—	—		

GX IEC Developer

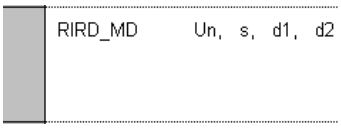
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

<table>
<tr><th>Operand</th><th colspan="4">Bedeutung</th><th>Werte-bereich</th><th>Festlegung des Inhalts durch</th><th>Datentyp</th></tr>
<tr><td>Un</td><td colspan="4">Kopfadresse des CC-Link-Moduls auf dem Baugruppenträger (Es werden nur die ersten beiden Stellen der 3-stelligen Adresse angegeben. Beispiel: Kopfadresse X/Y100 -> 10_H)</td><td>0 bis FE_H</td><td>Anwender</td><td>BIN-16-Bit</td></tr>
<tr><td rowspan="10">s</td><td colspan="4">Erster Operand des Bereiches mit Informationen zur Ausführung der Anweisung</td><td></td><td></td><td rowspan="10">BIN-16-Bit</td></tr>
<tr><td>Operand</td><td>Bedeutung</td><td colspan="2">Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(s)+0</td><td>Ausführungs-status der Anweisung</td><td colspan="2">Zeigt an, ob bei der Bearbeitung der Anweisung ein Fehler aufgetreten ist:
0000_H: Fehlerfreie Bearbeitung
Jeder andere Wert als 0000_H: Bei der Bearbeitung ist ein Fehler aufgetreten.(siehe Bedienungsanleitung des CC-Link-Moduls)</td><td>—</td><td>System</td></tr>
<tr><td>(s)+1</td><td>Stationsnummer</td><td colspan="2">Nummer der lokalen oder intelligenten Station, aus der Daten gelesen werden sollen</td><td>0 bis 64</td><td rowspan="4">Anwender</td></tr>
<tr><td rowspan="2">(s)+2</td><td>Zugriffscode</td><td colspan="2">• Bei einem Master-Modul der A/Q-Serie mit der Software-Version A bis H
Bei der Angabe von 0004_H wird aus dem Pufferspeicher einer intelligenten Station gelesen. Tragen Sie den Wert 2004_H ein, wenn auf dem Pufferspeicher einer lokalen Station gelesen werden soll.</td><td>0004_H oder 2004_H</td></tr>
<tr><td>Operanden- und Zugriffscode</td><td colspan="2">• Bei einem Master-Modul der A/Q-Serie ab der Software-Version J oder einem Modul des System Q
In das höherwertige Byte dieses Operanden wird der Operandencode eingetragen.
Das niederwertige Byte dieses Operanden dient zur Festlegung, ob auf den Pufferspeicher eines CC-Link-Moduls (04_H) oder auf eine CPU (05_H) zugegriffen wird.</td><td>Höherwertiges Byte: siehe folgende Tabelle
Niederwertiges Byte: 04_H oder 05_H</td></tr>
<tr><td rowspan="2">(s)+3</td><td rowspan="2">Anfangsadresse</td><td colspan="2">• Bei einem Master-Modul der A/Q-Serie mit der Software-Version A bis H
Anfangsadresse im Pufferspeicher</td><td rowspan="2">Abhängig von der Station, auf die zugegriffen wird</td></tr>
<tr><td colspan="2">• Bei einem Master-Modul der A/Q-Serie ab der Software-Version J oder einem Modul des System Q
Anfangsadresse im Pufferspeicher oder erste Operandenadresse.</td></tr>
<tr><td>(s)+4</td><td>Datenlänge</td><td colspan="2">Geben Sie hier an, wieviele Daten (Worte) gelesen werden sollen. Die Datenlänge hängt davon ab, mit welcher CPU die Station ausgestattet ist, aus der gelesen wird:
AnU-, QnA-Serie, System Q:
max. 480 Worte
Alle anderen CPUS:
max. 32 Worte</td><td>1 bis 480
1 bis 32</td></tr>
</table>

Variablen

<table>
<tr><th>Operand</th><th colspan="4">Bedeutung</th><th>Werte-bereich</th><th>Festlegung des Inhalts durch</th><th>Datentyp</th></tr>
<tr><td>d1</td><td colspan="5">Erste Adresse des Operandenbereichs, in dem die gelesenen Daten gespeichert werden</td><td>Anwender</td><td>BIN-16-Bit</td></tr>
<tr><td rowspan="4">d2</td><td colspan="5">Bit-Operand, der nach der Ausführung der RIRD-Anweisung für einen Zyklus gesetzt wird. Mit (d2)+1 wird ein Fehler bei der Ausführung angezeigt.</td><td rowspan="4">Bit</td></tr>
<tr><td>Operand</td><td colspan="2">Bedeutung</td><td>Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(d2)+0</td><td colspan="2">Anweisung ausgeführt</td><td>Zeigt die Beendigung der RIRD-Anweisung an.
EIN: Anweisung ausgeführt
AUS: Anweisung nicht ausgeführt</td><td>—</td><td rowspan="2">System</td></tr>
<tr><td>(d2)+1</td><td colspan="2">Anweisung mit Fehler ausgeführt</td><td>Zeigt an, ob bei der Ausführung der RIRD-Anweisung ein Fehler aufgetreten ist.
EIN: Anweisung mit Fehler ausgeführt
AUS: Anweisung ohne Fehler ausgeführt</td><td>—</td></tr>
</table>

Ab der Software-Version J eines Master-Moduls der A/Q-Serie und bei Modulen des System Q wird durch die Angabe eines Operanden- und Zugriffscodes in (s)+2 festgelegt, aus welchem Teil des Pufferspeichers gelesen bzw. welche Operanden der CPU erfasst werden:

- Zugriff auf den Pufferspeicher eines CC-Link-Moduls (Zugriffscode 04_H)

Zugriff auf		Operandencode
Pufferspeicher in intelligenten Stationen		00_H
Pufferspeicher in Master- oder lokalen Station	Pufferspeicher mit freiem Zugriff	20_H
	Dezentrale Eingänge	21_H
	Dezentrale Ausgänge	22_H
	Dezentrale Register	24_H
	Link-Sondermerker	63_H
	Link-Sonderregister	64_H

- Zugriff auf Operanden in der CPU (Zugriffscode 05_H)
  Auf Operanden, die hier nicht aufgeführt sind, kann nicht zugegriffen werden. Geben Sie beim Zugriff auf Bit-Operanden eine Adresse an, die entweder 0 oder durch 16 teilbar ist.

Operand		Operandentyp		Adressierung	Operandencode
Bezeichnung	**Symbol**	**Bit**	**Wort**		
Eingänge	X	●		Hexadezimal	00_H
Ausgänge	Y	●			02_H
Merker	M	●		Dezimal	03_H
Latch-Merker	L	●			83_H
Link-Merker	B	●		Hex.	23_H
Timer (Kontakt)	T	●		Dezimal	09_H
Timer (Spule)		●			$0A_H$
Timer (Aktueller Wert)			●		$0C_H$
Remanente Timer (Kontakt)	ST	●			89_H
Remanente Timer (Spule)		●			$8A_H$
Remanente Timer (Aktueller Wert)			●		$8C_H$

Operand		Operandentyp		Adressierung	Operandencode
Bezeichnung	**Symbol**	**Bit**	**Wort**		
Counter (Kontakt)	C	●		Dezimal	11_H
Counter (Spule)		●			12_H
Counter (Aktueller Wert)			●		14_H
Datenregister	D		●		04_H
Link-Register	W		●	Hex.	24_H
File-Register	R		●	Dezimal	84_H
Link-Sondermerker	SB	●		Hexadezimal	63_H
Link-Sonderregister	SW		●		64_H
Sondermerker	SM	●		Dezimal	43_H
Sonderregister	SD		●		44_H

Funktions-weise

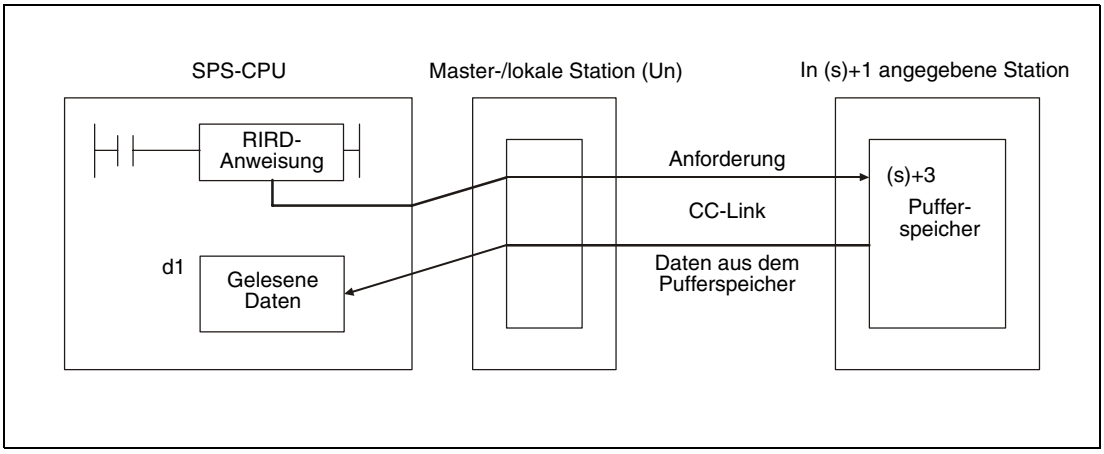
Daten aus dem Pufferspeicher einer andernen Station oder aus einer CPU lesen

RIRD Daten lesen

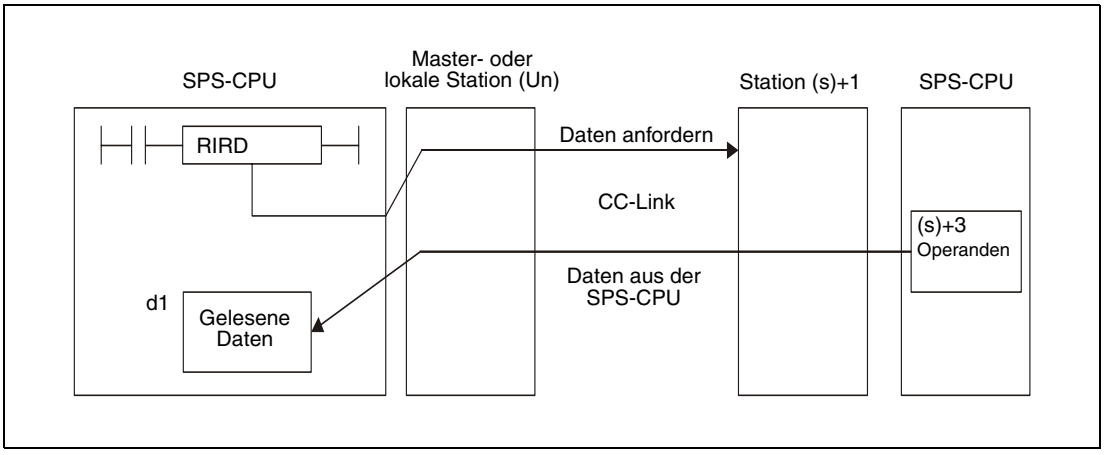
Mit der RIRD-Anweisung können Daten aus dem Pufferspeicher einer anderen Station am CC-Link gelesen werden. Bei Master-Modulen ab der Software-Version J und bei den CC-Link-Modulen des System Q ist auch der Zugriff auf Operanden in der SPS-CPU der anderen CC-Station möglich.

In dem Operanden (s)+3 wird die erste Pufferspeicheradresse oder der erste Operand angegeben, der gelesen werden soll. Der Operand (s)+1 enthält die Stationsnummer der anderen Station. Diese Station ist an die Master-Station angeschlossen, die mit Un spezifiziert ist. Die ausgelesenen Daten werden in der CPU, in der die RIRD-Anweisung bearbeitet wird, ab dem in d1 angegebenen Operanden gespeichert.
Der Operand (s)+4 enthält die Angabe, wieviele Daten übertragen werden sollen.

- Funktionsschema beim Lesen aus dem Pufferspeicher eines CC-Link-Moduls



- Funktionsschema beim Zugriff auf Operanden der SPS-CPU der anderen CC-Link-Station

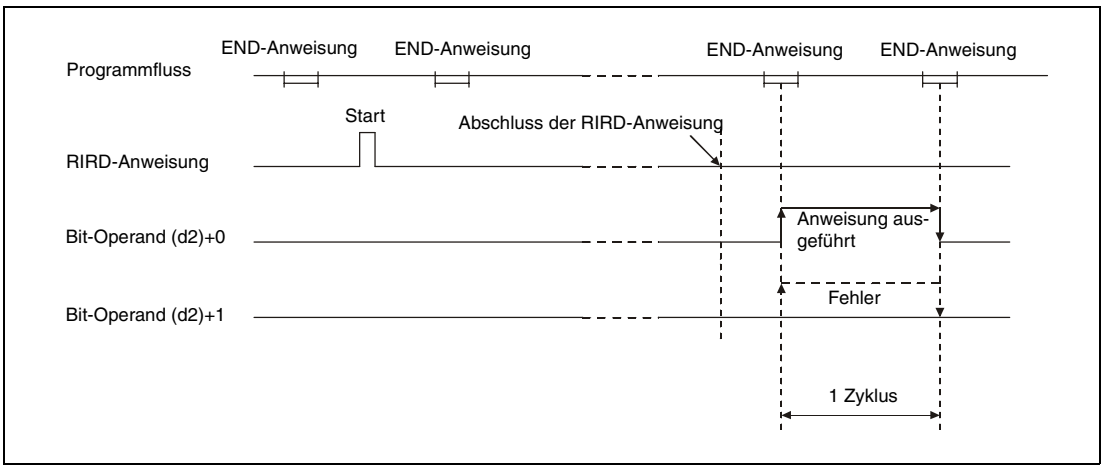


Durch die in (d2)+0 und (d2)+1 angegebenen Operanden wird angezeigt, ob die Anweisung ausgeführt wurde und ob dabei ein Fehler aufgetreten ist.

- Der Bit-Operand (d2)+0 wird gesetzt, wenn die END-Anweisung des Zyklus ausgeführt wird, in dem die RIRD-Anweisung beendet wurde. Bei der nächsten Bearbeitung der END-Anweisung wird dieser Bit-Operand wieder zurückgesetzt.
- Der Bit-Operand (d2)+1 zeigt einen Fehler bei der Ausführung der RIRD-Anweisung an. Bei fehlerfreier Ausführung bleibt dieser Bit-Operand zurückgesetzt. Bei einem Fehler jedoch wird (d2)+1 gesetzt, wenn die END-Anweisung des Zyklus ausgeführt wird, in dem die RIRD-

Anweisung beendet wurde. Bei der nächsten Bearbeitung der END-Anweisung wird dieser Operand wieder zurückgesetzt.

Die folgende Abbildung zeigt den Signalverlauf bei Ausführung der RIRD-Anweisung:



Für mehrere Stationen kann gleichzeitig eine RIRD-Anweisung ausgeführt werden. Auf eine intelligente- oder lokale Station kann jedoch nicht gleichzeitig mit einer RIRD-Anweisung von verschiedenen anderen Stationen aus zugegriffen werden.

Fehler-quellen

Wenn bei der Ausführung der RIRD-Anweisung ein Fehler auftritt, wird das Error-Flag SM0 gesetzt und im Sonderregister SD0 ein Fehlercode eingetragen.

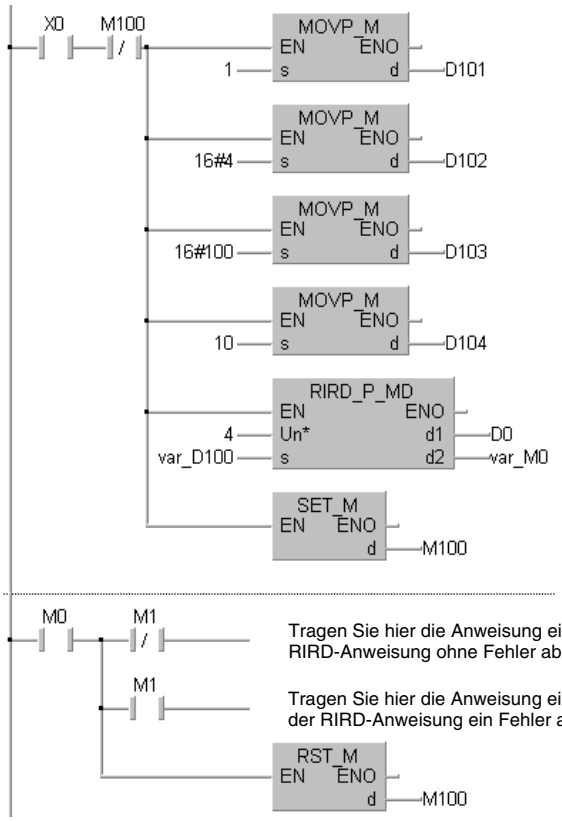
- Das in Un definierte Modul ist kein Sondermodul.
  (Fehlercodes: QnA 2110, System Q: 2112)
- Die auszuführende Anweisung wird nicht unterstützt. (Fehlercode: 4002)
- Die in der Anweisung angegebene Anzahl der Operanden ist ungültig. (Fehlercode: 4003)
- In der Anweisung ist ein unzulässiger Operand angegeben (Fehlercode: 4004)
- Der mit s definierte Operandenbereich enthält unzulässige Daten. (Fehlercode: 4100)
- Die Anzahl der Daten überschreitet den zulässigen Bereich. (Fehlercode: 4101)
- Die gespeicherten Daten oder Konstanten, die mit der Anweisung übertragen wurden, überschreiten den zulässigen Bereich. (Fehlercode: 4101)
- Bei QnA-Serie: Es werden zuviele erweiterte Anweisungen für CC-Link verwendet (Fehlercode: 4107).
- Bei QnA-Serie: Die Link-Parameter sind nicht eingestellt. (Fehlercode: 4108)

Beispiel RIRD

Das folgende Programm wird in der SPS der Master-Station bearbeitet. Wird der Eingang X0 gesetzt, wird aus einer intelligenten Station mit der Stationsnummer 1 der Inhalt von 10 Pufferspeicheradressen gelesen. Der Lesevorgang beginnt bei der Pufferspeicheradresse 100_H. Die erfassten Daten werden ab dem Register D0 gespeichert. Das CC-Link-Modul der Master-Station hat die Kopfadresse X/Y40.

- IEC-Editoren (Auf der nächsten Seite ist dieses Beispiel in der MELSEC-Anweisungliste und dem Kontaktplan des GX Developers dargestellt.)

Kontaktplan (GX IEC Developer)



Stationsnummer 1 in (s)+1 eintragen

Zugangscode 4_H für den Pufferspeicher einer intelligenten Station in (s)+2 eintragen

In (s)+3 wird die Anfangsadresse im Pufferspeicher eingetragen

10 Pufferspeicheradressen werden gelesen

Pufferspeicher auslesen

Merken, dass Daten gelesen werden

Tragen Sie hier die Anweisung ein, die ausgeführt werden soll, wenn die Bearbeitung der RIRD-Anweisung ohne Fehler abgeschlossen wurde.

Tragen Sie hier die Anweisung ein, die ausgeführt werden soll, wenn bei der Bearbeitung der RIRD-Anweisung ein Fehler aufgetreten ist.

Lesen der Daten abgeschlossen

IEC-Anweisungsliste

```
LD      X0
ANDN    M100
MOVP_M  1,      D101
MOVP_M  16#4,   D102
MOVP_M  16#100, D103
MOVP_M  10,     D104
RIRD_P_MD 4,    var_D100,  D0,  var_M0
SET_M   M100
```

Die Operanden und Anweisungen sind oben beim Programmbeispiel für den Kontaktplan erläutert.

```
LD      M0
ANDN    M1
```
Ausführung, wenn die Bearbeitung der RIRD-Anweisung ohne Fehler abgeschlossen wurde
```
LD      M0
AND     M1
```
Ausführung, wenn ein Fehler bei der Bearbeitung der RIRD-Anweisung aufgetreten ist
```
LD      M0
RST_M   M100
```

HINWEIS *Bei den IEC-Editoren des GX IEC Developers müssen die Variablen im Header der Programmorganisationseinheit (POE) definiert werden. Ohne Variablendefinition werden beim Überprüfen oder Kompilieren des Programms Fehler gemeldet.*
Weitere Informationen finden Sie im Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung.

- MELSEC-Anweisungliste und Kontaktplan des GX Developers
  Die Operanden und Anweisungen sind auf der vorherigen Seite beim Programmbeispiel für den Kontaktplan des GX IEC Developers erläutert.

Kontaktplan (GX Developer)

MELSEC-Anweisungsliste

```
MELSEC   LD        X0
         ANI       M100
         MOVP      K1        D101
         MOVP      H4        D102
         MOVP      H100      D103
         MOVP      K10       D104
         GP.RIRD   U4
                   D100
                   D0
                   M0
         SET       M100

MELSEC   LD        M0
         MPS
         ANI       M1
         Ausführung, wenn die Bearbeitung der RIRD-Anweisung ohne Fehler abgeschlossen wurde
         MRD
         AND M1
         Ausführung, wenn ein Fehler bei der Bearbeitung der RIRD-Anweisung aufgetreten ist
         MPP
         RST       M100
```

11.5.6 RIWT (A-Serie)

CPU

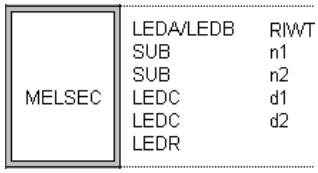
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●		

Operanden MELSEC A

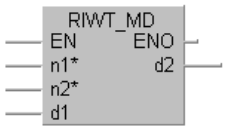
	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)								Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9011
n1																	●	●					26			●
n2																	●	●								
d1								●	●	●	●	●														
d2		●	●	●	●	●																				

GX IEC Developer

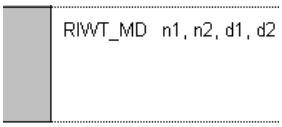
MELSEC-Anweisungsliste



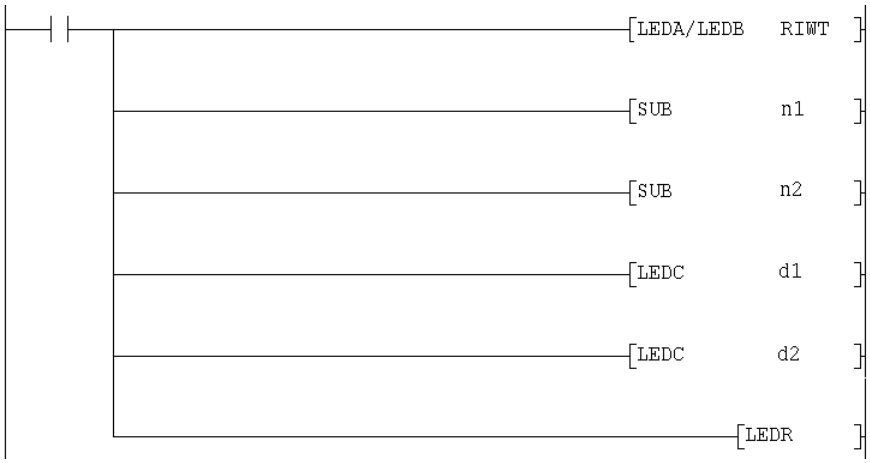
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

Operand	Bedeutung	Werte-bereich	Festlegung des Inhalts durch	Datentyp
n1	Kopfadresse des CC-Link-Moduls auf dem Baugruppenträger (Es werden nur die ersten beiden Stellen der 3-stelligen Adresse angegeben. Beispiel: Kopfadresse X/Y100 -> 10_H)	0 bis FE_H	Anwender	BIN-16-Bit
n2	Stationsnummer der Station, in die Daten übertragen werden Wertebereich: Bei Ausführung der RIWT-Anweisung in der Master-Station: 1 bis 64, bei Ausführung der RIWT-Anweisung in einer lokalen Station: 0 bis 64	siehe nebenstehende Erläuterung	Anwender	BIN-16-Bit

Variablen

<table>
<tr><th>Operand</th><th colspan="3">Bedeutung</th><th>Werte-bereich</th><th>Festlegung des Inhalts durch</th><th>Datentyp</th></tr>
<tr><td rowspan="9">d1</td><td colspan="5">Erster Operand des Bereiches mit Informationen zur Ausführung der Anweisung und zur Speicherung der zu übertragenen Daten.</td><td rowspan="9">BIN-16-Bit</td></tr>
<tr><td>Operand</td><td>Bedeutung</td><td>Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(d1)+0</td><td>Ausführungs-status der Anweisung</td><td>Zeigt an, ob bei der Bearbeitung der Anweisung ein Fehler aufgetreten ist:
0000_H: Fehlerfreie Bearbeitung
Jeder andere Wert als 0000_H: Bei der Bearbeitung ist ein Fehler aufgetreten.
Nähere Hinweise zu Fehlercodes finden Sie in der Bedienungsanleitung des CC-Link-Moduls</td><td>—</td><td>System</td></tr>
<tr><td>(d1)+1</td><td>Datenlänge</td><td>Geben Sie hier an, wieviele Daten (Worte) übertragen werden sollen. Die Datenlänge hängt davon ab, mit welcher CPU die Station ausgestattet ist, zu der Daten gesendet werden:
AnU-, QnA-Serie, System Q:
max. 480 Worte
Alle anderen CPUs:
max. 10 Worte</td><td>1 bis 480
1 bis 10</td><td rowspan="5">Anwender</td></tr>
<tr><td rowspan="3">(d1)+2</td><td>Zugriffscode</td><td>• Bei einem Master-Modul mit der Software-Version A bis H
Bei der Angabe von 0004_H wird in den Pufferspeicher einer intelligenten Station geschrieben.
Tragen Sie den Wert 2004_H ein, wenn in den Pufferspeicher einer lokalen Station geschrieben werden soll.</td><td>0004_H
oder
2004_H</td></tr>
<tr><td rowspan="2">Operanden- und Zugriffscode</td><td rowspan="2">• Bei einem Master-Modul ab der Software-Version J
In das höherwertige Byte dieses Operanden wird der Operandencode eingetragen.
Das niederwertige Byte dieses Operanden dient zur Festlegung, ob auf den Pufferspeicher eines CC-Link-Moduls (04_H) oder auf eine CPU (05_H) zugegriffen wird.</td><td>Höherwertiges Byte: siehe folgende Tabelle</td></tr>
<tr><td>Niederwertiges Byte:
04_H
oder
05_H</td></tr>
<tr><td rowspan="2">(d1)+3</td><td rowspan="2">Anfangsadresse</td><td>• Bei einem Master-Modul mit der Software-Version A bis H
Anfangsadresse im Pufferspeicher</td><td rowspan="2">Abhängig von der Station, auf die zugegriffen wird</td></tr>
<tr><td>• Bei einem Master-Modul ab der Software-Version J
Anfangsadresse im Pufferspeicher oder erste Operandenadresse.</td></tr>
<tr><td>(d1)+4
bis
(d1)+n</td><td>Speicherbereich für die Daten, die übertragen werden.</td><td>Geben Sie die Größe dieses Bereiches in (d1)+1 an.</td><td>—</td><td>Anwender</td></tr>
</table>

Variablen

<table>
<tr><th>Operand</th><th colspan="4">Bedeutung</th><th>Werte-bereich</th><th>Festlegung des Inhalts durch</th><th>Datentyp</th></tr>
<tr><td rowspan="3">d2</td><td colspan="4">Bit-Operand, der nach der Ausführung der RIWT-Anweisung für einen Zyklus gesetzt wird. Mit (d2)+1 wird ein Fehler bei der Ausführung angezeigt.</td><td></td><td></td><td rowspan="3">Bit</td></tr>
<tr><td>Operand</td><td>Bedeutung</td><td>Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(d2)+0</td><td>Anweisung ausgeführt</td><td>Zeigt die Beendigung der Anweisung an.
EIN: Anweisung ausgeführt
AUS: Anweisung nicht ausgeführt</td><td>—</td><td rowspan="2">System</td></tr>
<tr><td></td><td>(d2)+1</td><td>Anweisung mit Fehler ausgeführt</td><td>Zeigt an, ob bei der Ausführung der Anweisung ein Fehler aufgetreten ist.
EIN: Anweisung mit Fehler ausgeführt
AUS: Anweisung ohne Fehler ausgeführt</td><td>—</td><td></td></tr>
</table>

Ab der Software-Version J des Master-Moduls wird durch die Angabe eines Operanden- und Zugriffscodes in (d1)+2 festgelegt, in welchen Teil des Pufferspeichers geschrieben bzw. welche Operanden der CPU überschrieben werden:

- Zugriff auf den Pufferspeicher eines CC-Link-Moduls (Zugriffscode 04_H)

Zugriff auf		Operandencode
Pufferspeicher in intelligenten Stationen		00_H
Pufferspeicher in Master- oder lokalen Station	Pufferspeicher mit freiem Zugriff	20_H
	Dezentrale Eingänge	21_H
	Dezentrale Ausgänge	22_H
	Dezentrale Register	24_H
	Link-Sondermerker	63_H
	Link-Sonderregister	64_H

- Zugriff auf Operanden in der CPU (Zugriffscode 05_H)
  Auf Operanden, die hier nicht aufgeführt sind, kann nicht zugegriffen werden. Geben Sie beim Zugriff auf Bit-Operanden eine Adresse an, die entweder 0 oder durch 16 teilbar ist.

Operand		Operandentyp		Operandencode
Bezeichnung	**Symbol**	**Bit**	**Wort**	
Eingänge	X	●		01_H
Ausgänge	Y	●		02_H
Merker	M	●		03_H
Latch-Merker	L	●		83_H
Link-Merker	B	●		23_H
Timer (Kontakt)	T	●		09_H
Timer (Spule)		●		$0A_H$
Timer (Aktueller Wert)			●	$0C_H$
Counter (Kontakt)	C	●		11_H
Counter (Spule)		●		12_H
Counter (Aktueller Wert)			●	14_H
Datenregister	D		●	04_H
Link-Register	W		●	24_H
File-Register	R		●	84_H

Funktionsweise

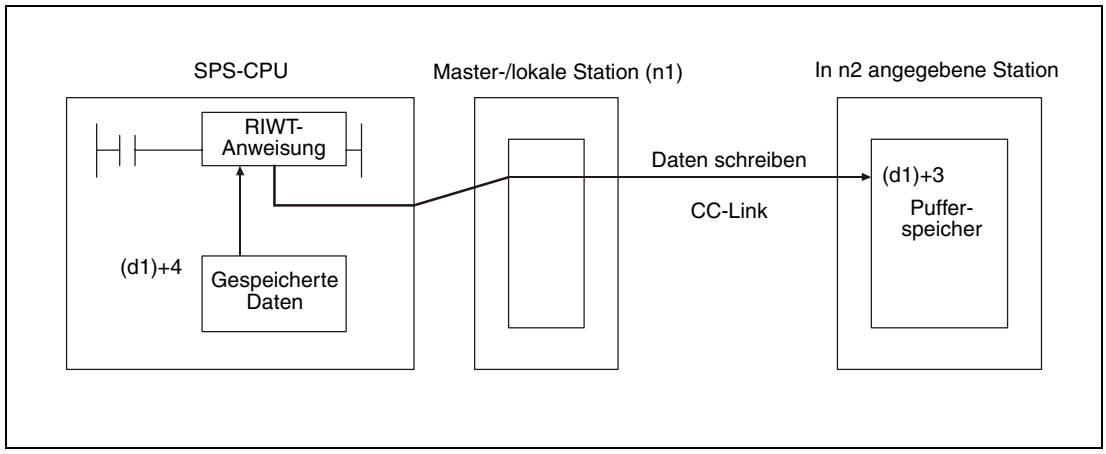
Daten in den Pufferspeicher einer andernen Station oder eine CPU schreiben

RIWT Daten schreiben

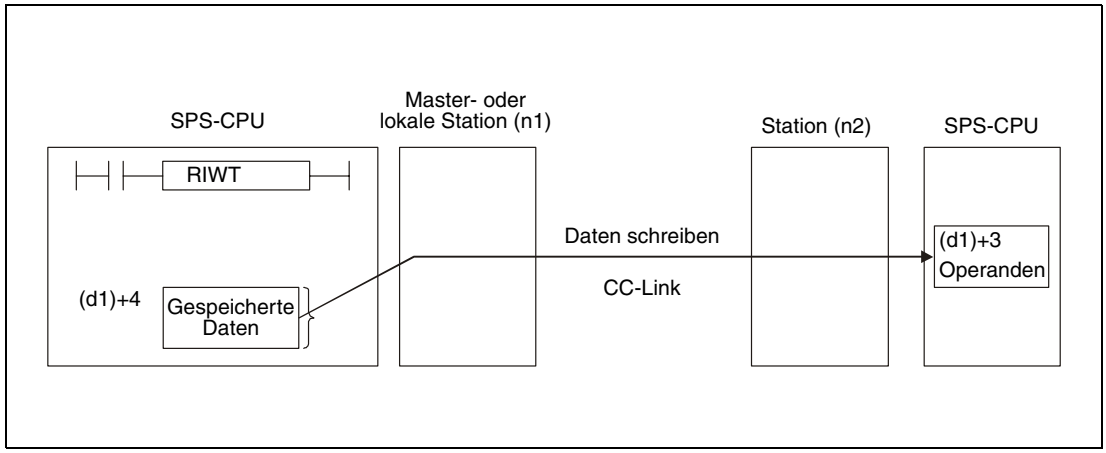
Mit der RIWT-Anweisung können Daten in den Pufferspeicher einer anderen Station am CC-Link übertragen werden. Bei Master-Modulen ab der Software-Version J ist auch der Zugriff auf Operanden in der SPS-CPU der anderen Station möglich.

Der Operand n2 enthält die Stationsnummer der anderen Station. Diese Station ist an die Master/lokale Station angeschlossen, die mit n1angegeben ist. Die zu übertragenen Daten sind vorab in der CPU, in der die RIWT-Anweisung bearbeitet wird, ab (d1)+4 eingetragen worden. Der Operand (d1)+1 enthält die Angabe, wieviele Daten übertragen werden sollen. In dem Operanden (d1)+3 wird die erste Pufferspeicheradresse oder der erste Operand angegeben, der überschrieben werden soll.

- Funktionsschema bei den Software-Versionen A bis H:



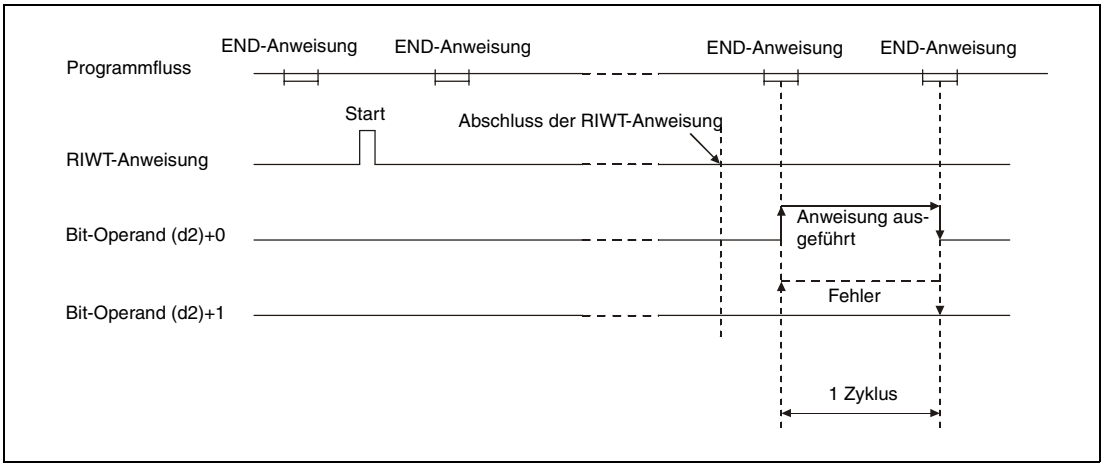
- Zusätzliche Funktion ab der Software-Version J:



Durch die in (d2)+0 und (d2)+1 angegebenen Operanden wird angezeigt, ob die Anweisung ausgeführt wurde und ob dabei ein Fehler aufgetreten ist.

- Der Bit-Operand (d2)+0 wird gesetzt, wenn die END-Anweisung des Zyklus ausgeführt wird, in dem die RIWT-Anweisung beendet wurde. Bei der nächsten Bearbeitung der END-Anweisung wird dieser Bit-Operand wieder zurückgesetzt.
- Der Bit-Operand (d2)+1 zeigt einen Fehler bei der Ausführung der RIWT-Anweisung an. Bei fehlerfreier Ausführung bleibt dieser Bit-Operand zurückgesetzt. Bei einem Fehler jedoch wird (d2)+1 gesetzt, wenn die END-Anweisung des Zyklus ausgeführt wird, in dem die RIWT-Anweisung beendet wurde. Bei der nächsten Bearbeitung der END-Anweisung wird dieser Operand wieder zurückgesetzt.

Die folgende Abbildung zeigt den Signalverlauf bei Ausführung der RIWT-Anweisung:



Für mehrere Stationen kann gleichzeitig eine RIWT-Anweisung ausgeführt werden. Auf dieselbe intelligente- oder lokale Station kann jedoch nicht gleichzeitig mit mehreren RIWT-Anweisungen von verschiedenen Stationen aus zugegriffen werden.

Die Netzwerkparameter müssen über die RLPA-Anweisung eingestellt sein, bevor die RIWT-Anweisung ausgeführt werden kann.

Wird als Anzahl der zu übertragenen Daten in (d1)+1 eine 0 oder ein Wert außerhalb des Bereichs von 1 bis 480 angegeben, wird beim Abschluss der RIWT-Anweisung ein Fehler durch den Operanden (d2)+1 gemeldet.

Ausführungsbedingungen

Wird die RIWT-Anweisung in Verbindung mit einer LEDA-Anweisung programmiert, wird die RIWT-Anweisung ausgeführt, solange die Ausführungsbedingung der LEDA-Anweisung eingeschaltet ist. Bei Anwendung einer LEDB-Anweisung dagegen wird die Datenübertragung nur bei der abfallenden Flanke der Ausführungsbedingung ausgeführt.

Beachten Sie, dass für die Bearbeitung der RIWT-Anweisung mehrere Zyklen benötigt werden. Starten Sie daher die nächste Datenübertragung erst dann, nachdem durch den Operanden (d2)+0 angezeigt wurde, dass die Bearbeitung der RIWT-Anweisung abgeschlossen ist.

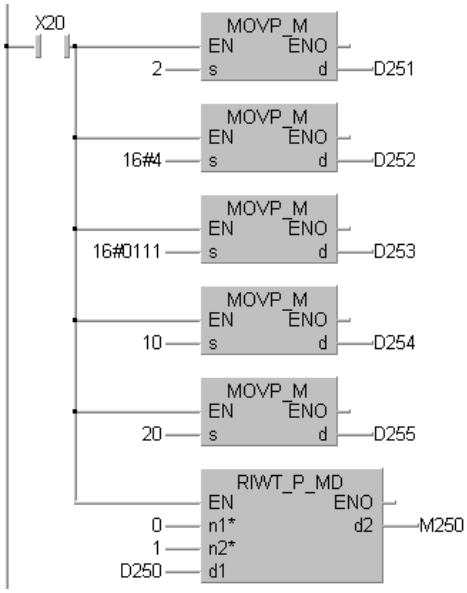
Beispiel RIWT

Das folgende Programm wird in der SPS der Master-Station bearbeitet und trägt den Wert 10 in die Pufferspeicheradresse 111_H und den Wert 20 in die Pufferspeicheradresse 112_H einer intelligenten Station mit der Stationsnummer 1 ein. Das CC-Link-Modul der Master-Station belegt den Adressbereich von X/Y000 bis X/Y01F.

MELSEC-Anweisungsliste

```
MELSEC  LD      X20
        MOVP    K2      D251
        MOVP    H4      D252
        MOVP    H0111   D253
        MOVP    K10     D254
        MOVP    K20     D255
        LEDB    RIWT
        SUB     H0
        SUB     K1
        LEDC    D250
        LEDC    M250
        LEDR
```

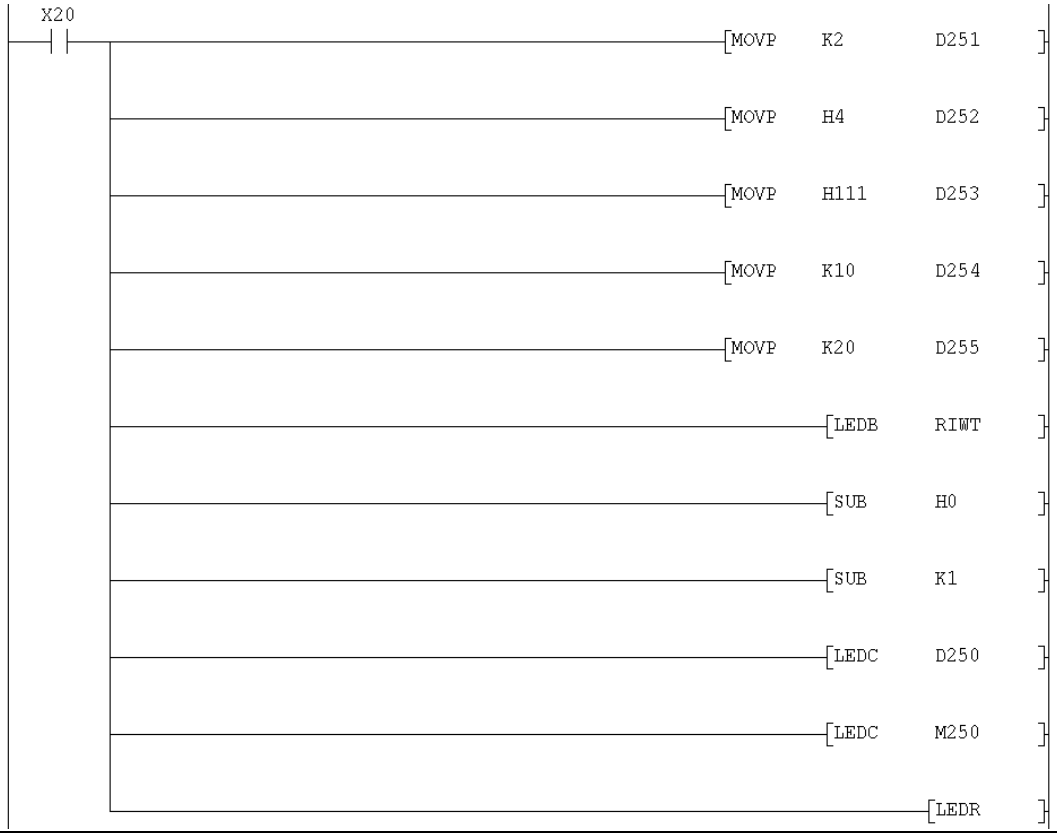
Kontaktplan (GX IEC Developer)



IEC-Anweisungsliste

```
LD          X20
MOVP_M      2,          D251
MOVP_M      16#0004,    D252
MOVP_M      16#0111,    D253
MOVP_M      10,         D254
MOVP_M      20,         D255
RIWT_P_MD   0,          1,      D250,   M250
```

Kontaktplan (GX Developer)



Hinweise zur Programmierung der erweiterten Anweisungen in den MELSEC-Editoren bei einer SPS der A-Serie finden Sie im Kapitel 3.3 dieses Handbuchs.

11.5.7 RIWT (QnA-Serie und System Q)

CPU

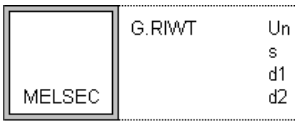
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Operanden MELSEC Q

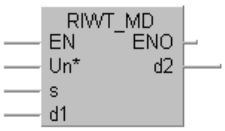
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□N□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s	—	●	●	—	—	—	—	—	—	SM0	8
d1	—	●	●	—	—	—	—	—	—		
d2	●	●	●	—	—	—	—	—	—		

GX IEC Developer

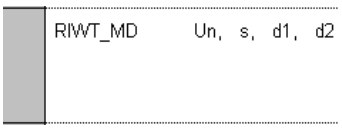
MELSEC-Anweisungsliste



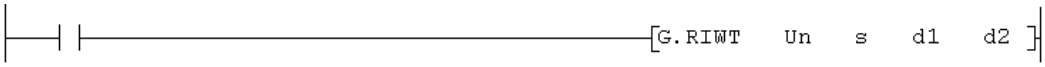
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

<table>
<tr><th>Operand</th><th colspan="4">Bedeutung</th><th>Werte-bereich</th><th>Festlegung des Inhalts durch</th><th>Datentyp</th></tr>
<tr><td>Un</td><td colspan="4">Kopfadresse des CC-Link-Moduls auf dem Baugruppenträger (Es werden nur die ersten beiden Stellen der 3-stelligen Adresse angegeben. Beispiel: Kopfadresse X/Y100 -> 10_H)</td><td>0 bis FE_H</td><td>Anwender</td><td>BIN-16-Bit</td></tr>
<tr><td rowspan="9">s</td><td colspan="4">Erster Operand des Bereiches mit Informationen zur Ausführung der Anweisung</td><td></td><td></td><td rowspan="9">BIN-16-Bit</td></tr>
<tr><td>Operand</td><td>Bedeutung</td><td>Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(s)+0</td><td>Ausführungs-status der Anweisung</td><td>Zeigt an, ob bei der Bearbeitung der Anweisung ein Fehler aufgetreten ist:
0000_H: Fehlerfreie Bearbeitung
Jeder andere Wert als 0000_H: Bei der Bearbeitung ist ein Fehler aufgetreten.(siehe Bedienungsanleitung des CC-Link-Moduls)</td><td>—</td><td>System</td></tr>
<tr><td>(s)+1</td><td>Stationsnummer</td><td>Nummer der lokalen oder intelligenten Station, zu der Daten übertragen werden.</td><td>0 bis 64</td><td rowspan="5">Anwender</td></tr>
<tr><td rowspan="2">(s)+2</td><td>Zugriffscode</td><td>• Bei einem Master-Modul der A/Q-Serie mit der Software-Version A bis H
Bei der Angabe von 0004_H wird in den Pufferspeicher einer intelligenten Station geschrieben. Tragen Sie den Wert 2004_H ein, wenn in den Pufferspeicher einer lokalen Station geschrieben werden soll.</td><td>0004_H oder 2004_H</td></tr>
<tr><td>Operanden- und Zugriffscode</td><td>• Bei einem Master-Modul der A/Q-Serie ab der Software-Version J oder einem Modul des System Q
In das höherwertige Byte dieses Operanden wird der Operandencode eingetragen.
Das niederwertige Byte dieses Operanden dient zur Festlegung, ob auf den Pufferspeicher eines CC-Link-Moduls (04_H) oder auf eine CPU (05_H) zugegriffen wird.</td><td>Höherwertiges Byte: siehe folgende Tabelle
Niederwertiges Byte: 04_H oder 05_H</td></tr>
<tr><td rowspan="2">(s)+3</td><td rowspan="2">Anfangsadresse</td><td>• Bei einem Master-Modul der A/Q-Serie mit der Software-Version A bis H
Anfangsadresse im Pufferspeicher</td><td rowspan="2">Abhängig von der Station, auf die zugegriffen wird</td></tr>
<tr><td>• Bei einem Master-Modul der A/Q-Serie ab der Software-Version J oder einem Modul des System Q
Anfangsadresse im Pufferspeicher oder erste Operandenadresse.</td></tr>
<tr><td>(s)+4</td><td>Number of points to write</td><td>Geben Sie hier an, wieviele Daten (Worte) übertragen werden sollen. Die Datenlänge hängt davon ab, mit welcher CPU die Station ausgestattet ist, zu der Daten gesendet werden:
AnU-, QnA-Serie, System Q:
max. 480 Worte
Alle anderen CPUS:
max. 10 Worte</td><td>1 bis 480
1 bis 10</td></tr>
</table>

Variablen

<table>
<tr><th>Operand</th><th colspan="4">Bedeutung</th><th>Werte-bereich</th><th>Festlegung des Inhalts durch</th><th>Datentyp</th></tr>
<tr><td>d1</td><td colspan="5">Erste Adresse des Operandenbereichs, in dem die zu übertragenden Daten gespeichert sind.</td><td>Anwender</td><td>BIN-16-Bit</td></tr>
<tr><td rowspan="4">d2</td><td colspan="5">Bit-Operand, der nach der Ausführung der RIWT-Anweisung für einen Zyklus gesetzt wird. Mit (d2)+1 wird ein Fehler bei der Ausführung angezeigt.</td><td rowspan="4">Bit</td></tr>
<tr><td>Operand</td><td colspan="2">Bedeutung</td><td>Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(d2)+0</td><td colspan="2">Anweisung ausgeführt</td><td>Zeigt die Beendigung der RIWT-Anweisung an.
EIN: Anweisung ausgeführt
AUS: Anweisung nicht ausgeführt</td><td>—</td><td rowspan="2">System</td></tr>
<tr><td>(d2)+1</td><td colspan="2">Anweisung mit Fehler ausgeführt</td><td>Zeigt an, ob bei der Ausführung der RIWT-Anweisung ein Fehler aufgetreten ist.
EIN: Anweisung mit Fehler ausgeführt
AUS: Anweisung ohne Fehler ausgeführt</td><td>—</td></tr>
</table>

Ab der Software-Version J eines Master-Moduls der A/Q-Serie und bei Modulen des System Q wird durch die Angabe eines Operanden- und Zugriffscodes in (s)+2 festgelegt, in welchen Teil des Pufferspeichers geschrieben bzw. welche Operanden der CPU überschrieben werden:

- Zugriff auf den Pufferspeicher eines CC-Link-Moduls (Zugriffscode 04_H)

Zugriff auf		Operandencode
Pufferspeicher in intelligenten Stationen		00_H
Pufferspeicher in Master- oder lokalen Station	Pufferspeicher mit freiem Zugriff	20_H
	Dezentrale Eingänge	21_H
	Dezentrale Ausgänge	22_H
	Dezentrale Register	24_H
	Link-Sondermerker	63_H
	Link-Sonderregister	64_H

- Zugriff auf Operanden in der CPU (Zugriffscode 05_H)
  Auf Operanden, die hier nicht aufgeführt sind, kann nicht zugegriffen werden. Geben Sie beim Zugriff auf Bit-Operanden eine Adresse an, die entweder 0 oder durch 16 teilbar ist.

Operand		Operandentyp		Adressierung	Operandencode
Bezeichnung	Symbol	Bit	Wort		
Eingänge	X	●		Hexadezimal	01_H
Ausgänge	Y	●			02_H
Merker	M	●		Dezimal	03_H
Latch-Merker	L	●			83_H
Link-Merker	B	●		Hex.	23_H
Timer (Kontakt)	T	●		Dezimal	09_H
Timer (Spule)		●			$0A_H$
Timer (Aktueller Wert)			●		$0C_H$
Remanente Timer (Kontakt)	ST	●			89_H
Remanente Timer (Spule)		●			$8A_H$
Remanente Timer (Aktueller Wert)			●		$8C_H$

Operand		Operandentyp		Adressierung	Operandencode
Bezeichnung	**Symbol**	**Bit**	**Wort**		
Counter (Kontakt)	C	●		Dezimal	11_H
Counter (Spule)		●			12_H
Counter (Aktueller Wert)			●		14_H
Datenregister	D		●		04_H
Link-Register	W		●	Hex.	24_H
File-Register	R		●	Dezimal	84_H
Link-Sondermerker	SB	●		Hexadezimal	63_H
Link-Sonderregister	SW		●		64_H
Sondermerker	SM	●		Dezimal	43_H
Sonderregister	SD		●		44_H

Funktionsweise

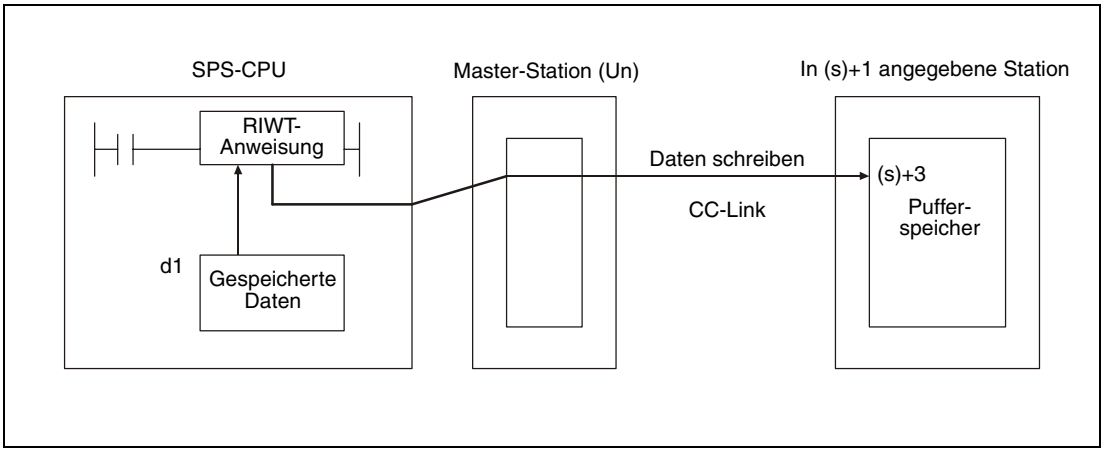
Daten in den Pufferspeicher einer andernen Station oder eine CPU schreiben

RIWT Daten schreiben

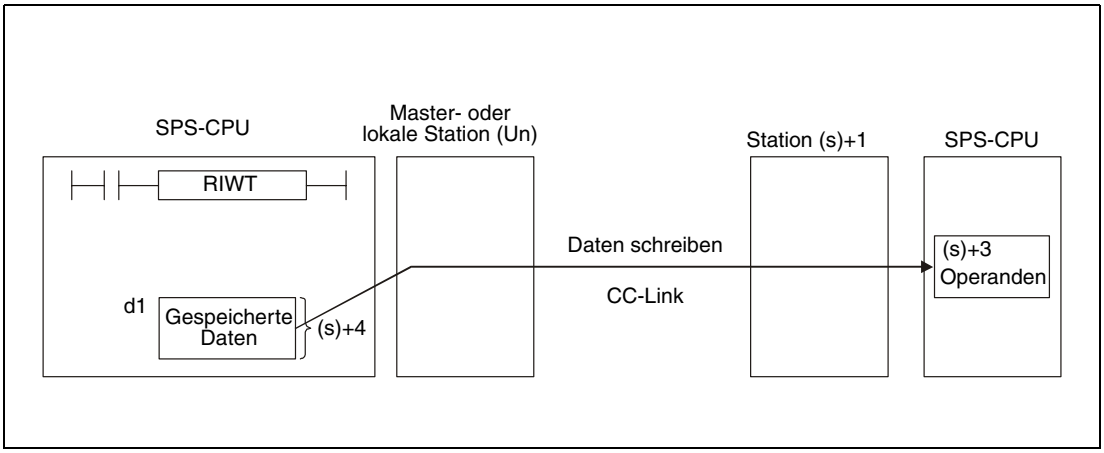
Mit der RIWT-Anweisung können Daten in den Pufferspeicher einer anderen Station am CC-Link übertragen werden. Bei Master-Modulen ab der Software-Version J und bei den CC-Link-Modulen des System Q ist auch der Zugriff auf Operanden in der SPS-CPU der anderen Station möglich.

Der Operand (s)+1 enthält die Stationsnummer der anderen Station. Diese Station ist an die Master-Station angeschlossen, die mit Un spezifiziert ist. Wo die Daten, die übertragen werden sollen, gespeichert sind, wird mit d1 festgelegt. In (s)+2 wird codiert angegeben, ob in einen Pufferspeicher oder der CPU geschrieben werden soll und welche Operanden dabei beeinflusst werden. Die Startadresse im Pufferspeicher oder die erste Operandenadresse wird in (s)+3 gespeichert. Im Operanden (s)+4 wird angegeben, wieviele Daten übertragen werden sollen.

- Funktionsschema beim Schreiben in den Pufferspeicher eines CC-Link-Moduls



- Funktionsschema beim Zugriff auf Operanden der SPS-CPU der anderen CC-Link-Station

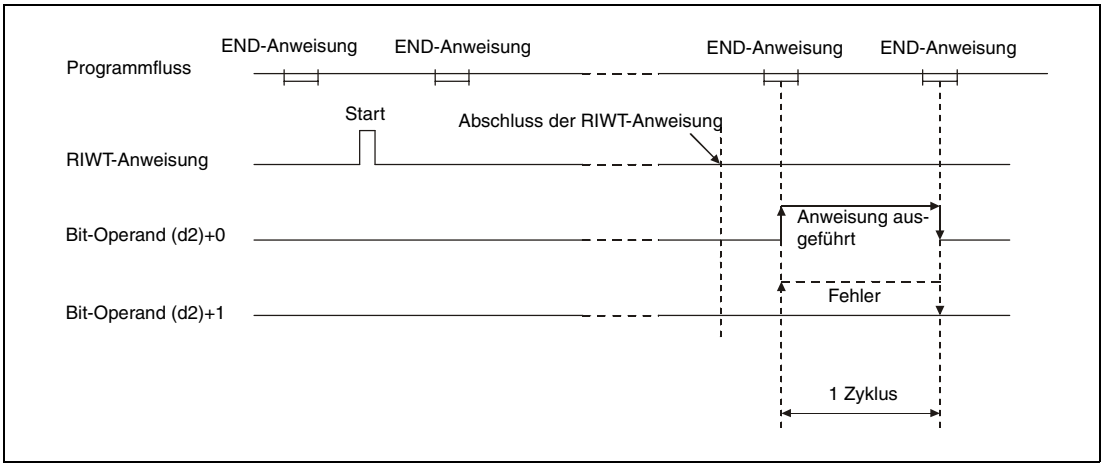


Durch die in (d2)+0 und (d2)+1 angegebenen Operanden wird angezeigt, ob die Anweisung ausgeführt wurde und ob dabei ein Fehler aufgetreten ist.

- Der Bit-Operand (d2)+0 wird gesetzt, wenn die END-Anweisung des Zyklus ausgeführt wird, in dem die RIWT-Anweisung beendet wurde. Bei der nächsten Bearbeitung der END-Anweisung wird dieser Bit-Operand wieder zurückgesetzt.
- Der Bit-Operand (d2)+1 zeigt einen Fehler bei der Ausführung der RIWT-Anweisung an. Bei fehlerfreier Ausführung bleibt dieser Bit-Operand zurückgesetzt. Bei einem Fehler

jedoch wird er gesetzt, wenn die END-Anweisung des Zyklus ausgeführt wird, in dem die RIWT-Anweisung beendet wurde. Bei der nächsten Bearbeitung der END-Anweisung wird der Bit-Operand (d2)+1 wieder zurückgesetzt.

Die folgende Abbildung zeigt den Signalverlauf bei Ausführung der RIWT-Anweisung:



Für mehrere Stationen kann gleichzeitig eine RIWT-Anweisung ausgeführt werden. Auf eine intelligente- oder lokale Station kann jedoch nicht gleichzeitig mit mehreren RIWT-Anweisungen von verschiedenen anderen Stationen aus zugegriffen werden.

Fehlerquellen

Wenn bei der Ausführung der RIWT-Anweisung ein Fehler auftritt, wird das Error-Flag SM0 gesetzt und im Sonderregister SD0 ein Fehlercode eingetragen.

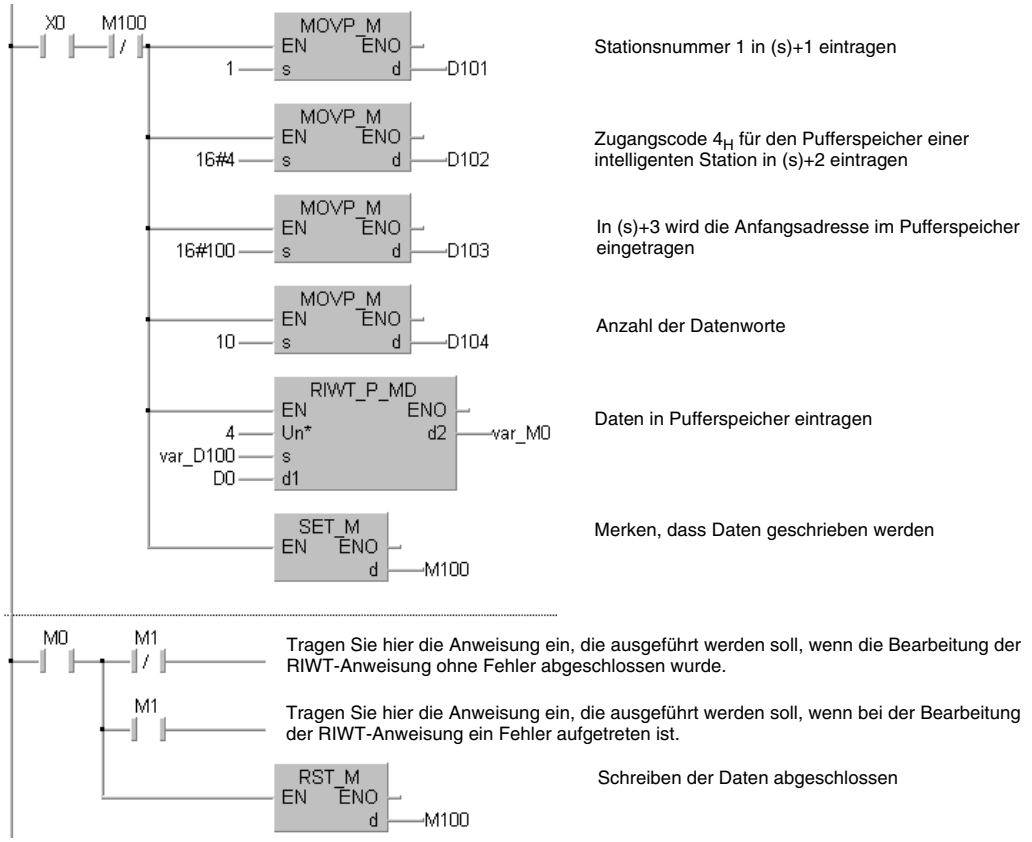
- Das in Un definierte Modul ist kein Sondermodul. (Fehlercode: 2112)
- Die auszuführende Anweisung wird nicht unterstützt. (Fehlercode: 4002)
- Die in der Anweisung angegebene Anzahl der Operanden ist ungültig. (Fehlercode: 4003)
- In der Anweisung ist ein unzulässiger Operand angegeben (Fehlercode: 4004)
- Der mit s definierte Operandenbereich enthält unzulässige Daten. (Fehlercode: 4100)
- Die Anzahl der Daten überschreitet den zulässigen Bereich. (Fehlercode: 4101)
- Die gespeicherten Daten oder Konstanten, die mit der Anweisung übertragen wurden, überschreiten den zulässigen Bereich. (Fehlercode: 4101)

Beispiel RIWT

Das folgende Programm wird in der SPS der Master-Station bearbeitet. Wenn der Eingang X0 gesetzt wird, werden in der intelligenten Station mit der Stationsnummer 1 ab der Pufferspeicheradresse 100_H zehn Pufferspeicheradressen beschrieben. Die Daten, die übertragen werden, sind in der CPU ab dem Register D0 gespeichert. Das CC-Link-Modul der Master-Station hat die Kopfadresse X/Y40.

- IEC-Editoren (Auf der nächsten Seite ist dieses Beispiel in der MELSEC-Anweisungliste und dem Kontaktplan des GX Developers dargestellt.)

Kontaktplan (GX IEC Developer)



IEC-Anweisungsliste

```
LD          X0
ANDN        M100
MOVP_M      1,          D101
MOVP_M      16#4,       D102
MOVP_M      16#100,     D103
MOVP_M      10,         D104
RIWT_P_MD   4,          var_D100,   D0,   var_M0
SET_M       M100
```

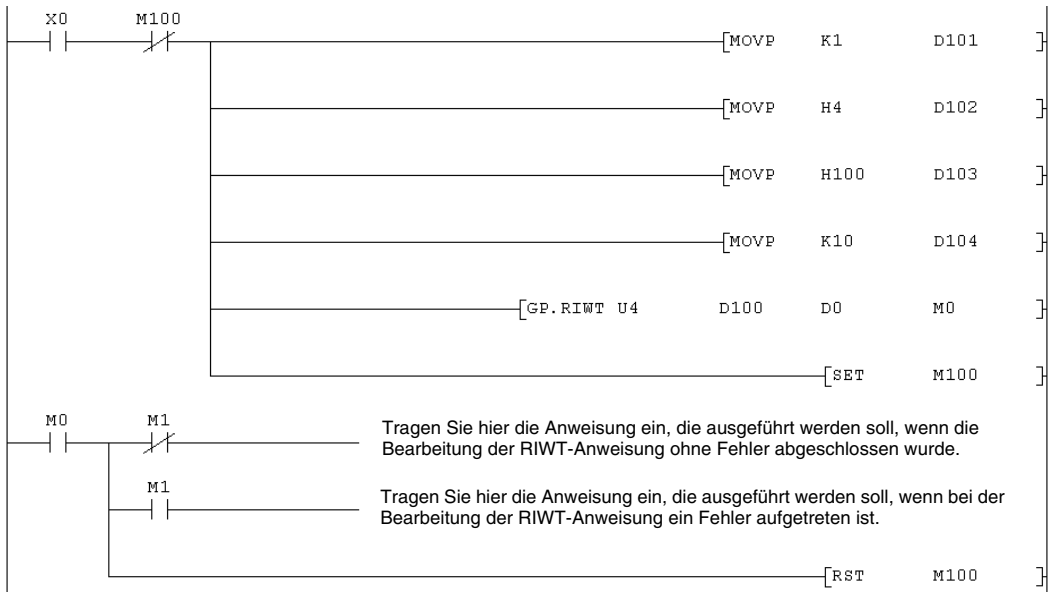
Die Operanden und Anweisungen sind oben beim Programmbeispiel für den Kontaktplan erläutert.

```
LD          M0
ANDN        M1
Ausführung, wenn die Bearbeitung der RIWT-Anweisung ohne Fehler abgeschlossen wurde
LD          M0
AND         M1
Ausführung, wenn ein Fehler bei der Bearbeitung der RIWT-Anweisung aufgetreten ist
LD          M0
RST_M       M100
```

HINWEIS *Bei den IEC-Editoren des GX IEC Developers müssen die Variablen im Header der Programmorganisationseinheit (POE) definiert werden. Ohne Variablendefinition werden beim Überprüfen oder Kompilieren des Programms Fehler gemeldet.*
Weitere Informationen finden Sie im Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung.

- MELSEC-Anweisungliste und Kontaktplan des GX Developers
  Die Operanden und Anweisungen sind auf der vorherigen Seite beim Programmbeispiel für den Kontaktplan des GX IEC Developers erläutert.

Kontaktplan (GX Developer)



MELSEC-Anweisungsliste

```
MELSEC
LD      X0
ANI     M100
MOVP    K1      D101
MOVP    H4      D102
MOVP    H100    D103
MOVP    K10     D104
GP.RIWT U4
        D100
        D0
        M0
SET     M100
```

```
MELSEC
LD      M0
MPS
ANI     M1
Ausführung, wenn die Bearbeitung der RIWT-Anweisung ohne Fehler abgeschlossen wurde
MRD
AND     M1
Ausführung, wenn ein Fehler bei der Bearbeitung der RIWT-Anweisung aufgetreten ist
MPP
RST     M100
```

11.5.8 RIRCV (A-Serie)

CPU

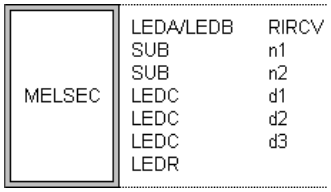
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●		

Operanden MELSEC A

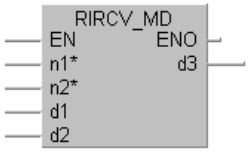
	Operanden																				Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)								Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9011
n1																	●	●								
n2																	●	●								
d1								●	●	●	●	●											29			●
d2								●	●	●	●	●														
d3		●	●	●	●	●																				

GX IEC Developer

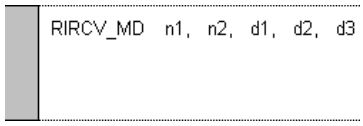
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

<table>
<tr><th>Operand</th><th colspan="4">Bedeutung</th><th>Werte-bereich</th><th>Festlegung des Inhalts durch</th><th>Datentyp</th></tr>
<tr><td>n1</td><td colspan="4">Kopfadresse des CC-Link-Master-Moduls auf dem Baugruppenträger
(Es werden nur die ersten beiden Stellen der 3-stelligen Adresse angegeben. Beispiel: Kopfadresse X/Y100 -> 10_H)</td><td>0 bis FE_H</td><td>Anwender</td><td>BIN-16-Bit</td></tr>
<tr><td>n2</td><td colspan="4">Stationsnummer der intelligenten Station, aus der Daten gelesen werden sollen</td><td>1 bis 64</td><td>Anwender</td><td>BIN-16-Bit</td></tr>
<tr><td rowspan="8">d1</td><td colspan="6">Erster Operand des Bereiches mit Informationen zur Ausführung der Anweisung und zur Speicherung der gelesenen Daten</td><td rowspan="8">BIN-16-Bit</td></tr>
<tr><td>Operand</td><td colspan="2">Bedeutung</td><td>Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(d1)+0</td><td colspan="2">Ausführungs-status der Anweisung</td><td>Zeigt an, ob bei der Bearbeitung der Anweisung ein Fehler aufgetreten ist:
0000_H: Fehlerfreie Bearbeitung
Jeder andere Wert als 0000_H: Bei der Bearbeitung ist ein Fehler aufgetreten.
Nähere Hinweise zu Fehlercodes finden Sie in der Bedienungsanleitung des CC-Link-Moduls</td><td>—</td><td>System</td></tr>
<tr><td>(d1)+1</td><td colspan="2">Datenlänge</td><td>Geben Sie hier an, wieviele Daten (Worte) gelesen werden sollen. Der angegebene Wert muss innerhalb der Pufferspeicherlänge der intell. Station und der Größe des Empfangspuffers der Master-Station liegen.</td><td>1 bis 480</td><td rowspan="4">Anwender</td></tr>
<tr><td>(d1)+2</td><td colspan="2">Zugriffscode</td><td>Geben Sie den Wert 0004_H an.
(Aus dem Pufferspeicher einer intelligenten Station lesen.)</td><td>0004_H</td></tr>
<tr><td>(d1)+3</td><td colspan="2">Fehlerprüfung</td><td>Geben Sie an, mit welchem Operanden ein Fehler bei der Ausführung der RIRCV-Anweisung angezeigt wird:
0: Operand d1
1: Operand RX+1</td><td>0 oder 1</td></tr>
<tr><td>(d1)+4</td><td colspan="2">Anfangsadresse</td><td>Anfangsadresse im Puffer-speicher der intelligenten Station</td><td>Abhängig von der Station, auf die zugegriffen wird</td></tr>
<tr><td>(d1)+5 bis (d1)+n</td><td colspan="2">Speicherbereich für die gelesenen Daten</td><td>Die Größe dieses Bereiches wird durch die in (d1)+1 angegebene Datenmenge bestimmt.</td><td>—</td><td>System</td></tr>
<tr><td rowspan="5">d2</td><td colspan="6">Angabe der Link-Operanden, die für das Handshake verwendet werden.</td><td rowspan="5">BIN-16-Bit</td></tr>
<tr><td>Operand</td><td colspan="2">Bedeutung</td><td>Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td rowspan="2">(d2)+0</td><td colspan="2" rowspan="2">Dezentraler Eingang (RX)
Dezentraler Ausgang (RY)</td><td>• Höherwertiges Byte
Adresse eines dezentralen Eingangs (RX) der intell. Station</td><td>0 bis 124</td><td rowspan="3">Anwender
(Die Adressen werden vom Anwender angegeben, gesetzt und zurückgesetzt bzw. verändert werden die Operanden aber vom System)</td></tr>
<tr><td>• Niederrwertiges Byte
Adresse eines dezentralen Ausgangs (RY) der intelligenten Station</td><td>0 bis 125</td></tr>
<tr><td>(d2)+1</td><td colspan="2">Dezentrales Register (RWr)</td><td>Adresse eines dezentralen Registers (RWr) der intelligenten Station</td><td>0 bis 15 oder FF (Bei der Angabe von FF ist kein Register angewählt)</td></tr>
</table>

Variablen

<table>
<tr><th>Operand</th><th colspan="4">Bedeutung</th><th>Werte-bereich</th><th>Festlegung des Inhalts durch</th><th>Datentyp</th></tr>
<tr><td rowspan="4">d3</td><td colspan="4">Bit-Operand, der nach der Ausführung der RIRCV-Anweisung für einen Zyklus gesetzt wird. Mit (d3)+1 wird ein Fehler bei der Ausführung angezeigt.</td><td></td><td></td><td rowspan="4">Bit</td></tr>
<tr><td>Operand</td><td>Bedeutung</td><td>Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(d3)+0</td><td>Anweisung ausgeführt</td><td>Zeigt die Beendigung der Anweisung an.
EIN: Anweisung ausgeführt
AUS: Anweisung nicht ausgeführt</td><td>—</td><td rowspan="2">System</td></tr>
<tr><td>(d3)+1</td><td>Anweisung mit Fehler ausgeführt</td><td>Zeigt an, ob bei der Ausführung der Anweisung ein Fehler aufgetreten ist.
EIN: Anweisung mit Fehler ausgeführt
AUS: Anweisung ohne Fehler ausgeführt</td><td>—</td></tr>
</table>

Funktions-weise

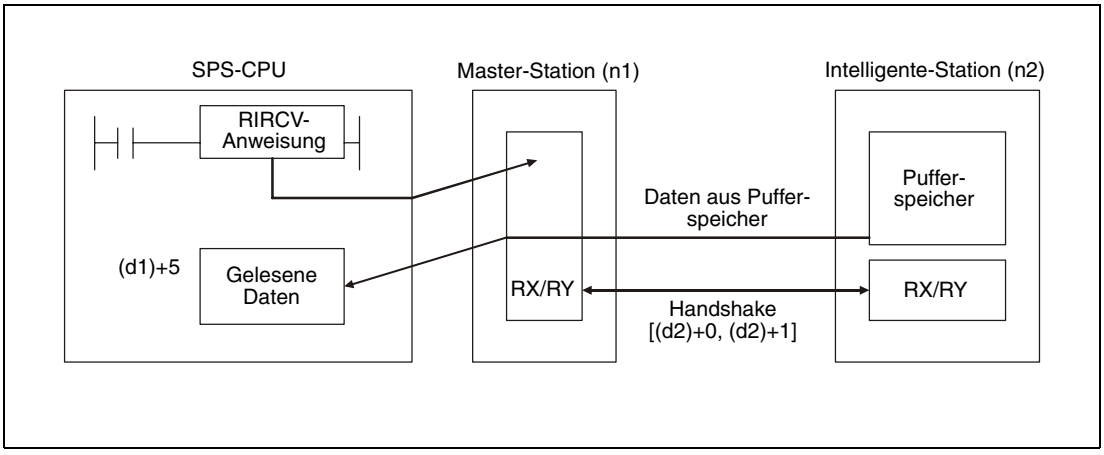
Daten aus dem Pufferspeicher einer intelligenten Station mit Handshake lesen

RIRCV Daten lesen (mit Handshake)

Die RIRCV-Anweisung kann nur in der SPS-CPU der Master-Station ausgeführt werden. Sie dient dazu, Daten aus dem Pufferspeicher einer intelligenten Station am CC-Link zu lesen. Der Datenaustausch wird über ein Handshake abgewickelt.

Der Operand (d1)+1 enthält die Angabe, wieviele Daten übertragen werden sollen. In dem Operanden (d1)+3 wird die erste Pufferspeicheradresse angegeben, die gelesen werden soll. Der Operand n2 enthält die Stationsnummer der anderen Station. Diese Station ist an die Master-Station angeschlossen, die in n1 spezifiziert ist. Die ausgelesenen Daten werden in der CPU, in der die RIRCV-Anweisung bearbeitet wird, ab (d1)+5 gespeichert.

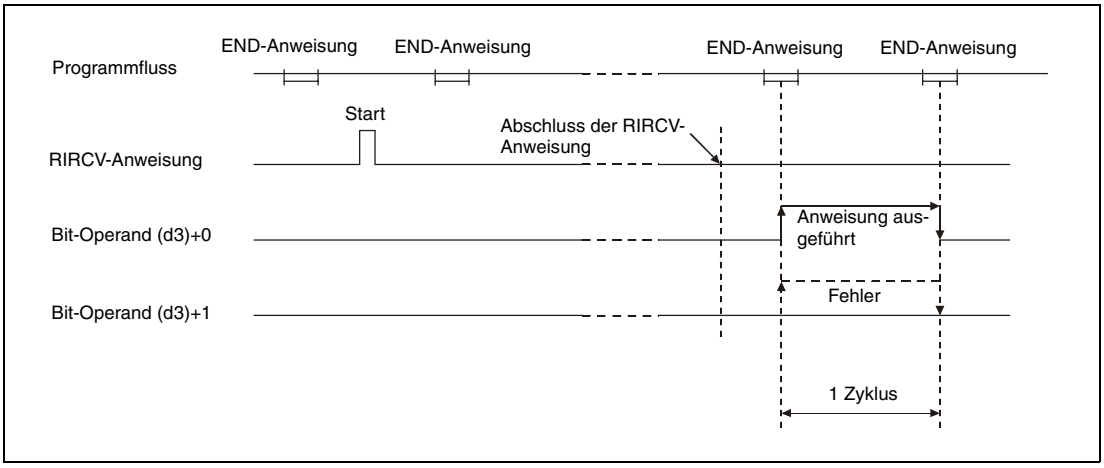
Funktionsschema der RIRCV-Anweisung:



Durch die in (d3)+0 und (d3)+1 angegebenen Operanden wird angezeigt, ob die Anweisung ausgeführt wurde und ob dabei ein Fehler aufgetreten ist.

- Der Bit-Operand (d3)+0 wird gesetzt, wenn die END-Anweisung des Zyklus ausgeführt wird, in dem die RIRCV-Anweisung beendet wurde. Bei der nächsten Bearbeitung der END-Anweisung wird dieser Bit-Operand wieder zurückgesetzt.
- Der Bit-Operand (d3)+1 zeigt einen Fehler bei der Ausführung der RIRCV-Anweisung an. Bei fehlerfreier Ausführung bleibt dieser Bit-Operand zurückgesetzt. Bei einem Fehler jedoch wird (d3)+1 gesetzt, wenn die END-Anweisung des Zyklus ausgeführt wird, in dem die RIRCV-Anweisung beendet wurde. Bei der nächsten Bearbeitung der END-Anweisung wird dieser Operand wieder zurückgesetzt.

Die folgende Abbildung zeigt den Signalverlauf bei Ausführung der RIRCV-Anweisung:



Für mehrere Stationen kann gleichzeitig eine RIRCV-Anweisung ausgeführt werden. Auf eine intelligente Station kann jedoch nicht gleichzeitig mit mehreren RIRCV-Anweisungen von verschiedenen Stationen aus zugegriffen werden.

Ausführungsbedingungen

Wird die RIRCV-Anweisung in Verbindung mit einer LEDA-Anweisung programmiert, wird die RIRCV-Anweisung ausgeführt, solange die Ausführungsbedingung der LEDA-Anweisung eingeschaltet ist.
Bei Anwendung einer LEDB-Anweisung dagegen wird der Lesevorgang nur bei der steigenden Flanke der Ausführungsbedingung ausgeführt.

Beachten Sie, dass für die Bearbeitung der RIRCV-Anweisung mehrere Zyklen benötigt werden. Starten Sie daher den nächsten Lesevorgang erst dann, nachdem durch den Operanden (d3)+0 angezeigt wurde, dass die Bearbeitung der RIRCV-Anweisung abgeschlossen ist. (Eine RIRCV-Anweisung wird nicht ausgeführt, wenn sie erneut gestartet wird, bevor die vorherige Bearbeitung abgeschlossen ist.)

Fehlerquellen

Die Netzwerkparameter müssen über die RLPA-Anweisung eingestellt sein, bevor die RIRCV-Anweisung ausgeführt werden kann. Wird dies nicht beachtet, wird nach der Ausführung der RIRCV-Anweisung der Fehlercode $4B00_H$ in den Operanden (d1)+0 eingetragen.

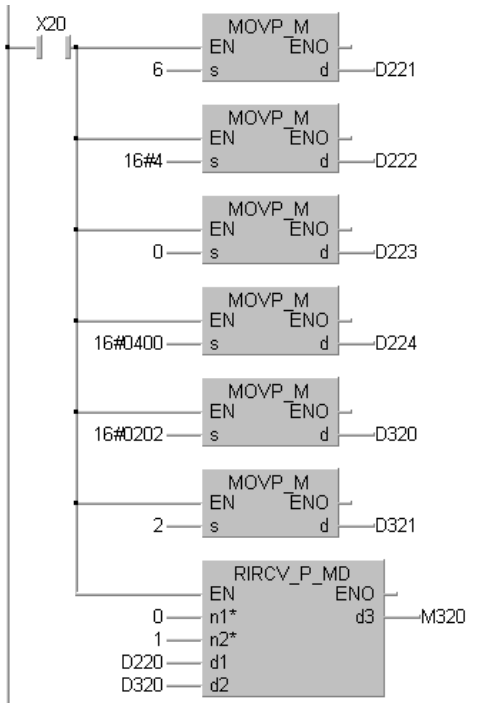
Wird als Anzahl der zu lesenden Daten in (d1)+1 eine 0 oder ein Wert außerhalb des Bereichs von 1 bis 480 angegeben, wird beim Abschluss der RIRCV-Anweisung der Fehlercode $BB42_H$ in den Operanden (d1)+0 eingetragen.

Beispiel RIRCV

Das folgende Programm, das in der SPS der Master-Station bearbeitet wird, liest die Inhalte der Pufferspeicheradressen 400_H bis 405_H aus der intelligenten Station mit der Stationsnummer 1. Als Operanden für den Handshake werden RX2, RY2 und RWr2 verwendet. Der in (d1)+0 angegebene Operand dient zur Anzeige eines Fehlers. Das CC-Link-Modul der Master-Station hat die Kopfadresse X/Y000.

- IEC-Editoren (Auf der nächsten Seite ist dieses Beispiel in der MELSEC-Anweisungliste und dem Kontaktplan des GX Developers dargestellt.)

Kontaktplan (GX IEC Developer)



Anzahl der Datenworte in (d1)+1 eintragen

Zugangscode 4_H für den Pufferspeicher einer intelligenten Station in (d1)+2 eintragen

In (d1)+3 wird festgelegt, dass ein Fehler durch den Operanden (d1)+0 angezeigt wird.

Anfangsadresse im Pufferspeicher in (d1)+4 eintragen

RX2 und RY2 in d2 eintragen

RWr2 in (d2)+1 eintragen

RIRCV-Anweisung ausführen

IEC-Anweisungsliste

```
LD          X20
MOVP_M      6,          D221
MOVP_M      16#0004,    D222
MOVP_M      0,          D223
MOVP_M      16#0400,    D224
MOVP_M      16#0202,    D320
MOVP_M      2,          D321
RIRCV_P_MD  0,          1,      D220,   D320,   M320
```

Die Operanden und Anweisungen sind oben beim Programmbeispiel für den Kontaktplan erläutert.

- MELSEC-Anweisungliste und Kontaktplan des GX Developers
  Die Operanden und Anweisungen sind auf der vorherigen Seite beim Programmbeispiel für den Kontaktplan des GX IEC Developers erläutert.

Kontaktplan (GX Developer)

MELSEC-Anweisungsliste

```
MELSEC
        LD      X20
        MOVP    K6      D221
        MOVP    H0004   D222
        MOVP    K0      D223
        MOVP    H0400   D224
        MOVP    H0202   D320
        MOVP    K2      D321
        LEDB    RIRCV
        SUB     H0
        SUB     K1
        LEDC    D220
        LEDC    D320
        LEDC    M320
        LEDR
```

Hinweise zur Programmierung der erweiterten Anweisungen bei einer SPS der A-Serie finden Sie im Kapitel 3.3 dieses Handbuchs.

11.5.9 RIRCV (QnA-Serie und System Q)

CPU

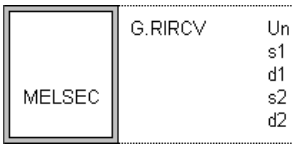
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Operanden MELSEC Q

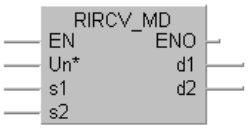
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□N□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	—	●	●	—	—	—	—	—	—	SM0	10
s1	—	●	●	—	—	—	—	—	—		
d1	—	●	●	—	—	—	—	—	—		
d2	●	●	●	—	—	—	—	—	—		

GX IEC Developer

MELSEC-Anweisungsliste



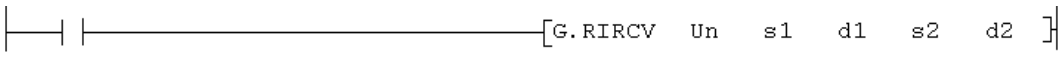
Kontaktplan



IEC-Anweisungsliste

```
RIRCV_MD   Un, s1, s2, d1, d2
```

GX Developer



Variablen

<table>
<tr><th>Operand</th><th colspan="4">Bedeutung</th><th>Werte-bereich</th><th>Festlegung des Inhalts durch</th><th>Datentyp</th></tr>
<tr><td>Un</td><td colspan="4">Kopfadresse des Master-Moduls auf dem Baugruppenträger (Es werden nur die ersten beiden Stellen der 3-stelligen Adresse angegeben. Beispiel: Kopfadresse X/Y100 -> 10_H)</td><td>0 bis FE_H</td><td>Anwender</td><td>BIN-16-Bit</td></tr>
<tr><td rowspan="7">s1</td><td colspan="4">Erster Operand des Bereiches mit Informationen zur Ausführung der Anweisung</td><td></td><td></td><td rowspan="7">BIN-16-Bit</td></tr>
<tr><td>Operand</td><td>Bedeutung</td><td colspan="2">Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(s1)+0</td><td>Ausführungs-status der Anweisung</td><td colspan="2">Zeigt an, ob bei der Bearbeitung der Anweisung ein Fehler aufgetreten ist:
0000_H: Fehlerfreie Bearbeitung
Jeder andere Wert als 0000_H: Bei der Bearbeitung ist ein Fehler aufgetreten.(siehe Bedienungsanleitung des CC-Link-Moduls)</td><td>—</td><td>System</td></tr>
<tr><td>(s1)+1</td><td>Stationsnummer</td><td colspan="2">Nummer der intelligenten Station, aus der Daten gelesen werden sollen</td><td>0 bis 64</td><td rowspan="4">Anwender</td></tr>
<tr><td>(s1)+2</td><td>Zugriffscode</td><td colspan="2">Geben Sie den Wert 0004_H an. (Aus dem Pufferspeicher einer intelligenten Station lesen.)</td><td>0004_H</td></tr>
<tr><td>(s1)+3</td><td>Anfangsadresse</td><td colspan="2">Anfangsadresse im Puffer-speicher</td><td>Abhängig von der Station, auf die zugegriffen wird</td></tr>
<tr><td>(s1)+4</td><td>Datenlänge</td><td colspan="2">Geben Sie hier an, wieviele Daten (Worte) gelesen werden sollen. Der angegebene Wert muss innerhalb der Pufferspeicherlänge der intell. Station und der Größe des Empfangspuffers der Master-Station liegen.</td><td>1 bis 480</td></tr>
<tr><td rowspan="7">s2</td><td colspan="4">Angabe der Link-Operanden, die für das Handshake verwendet werden.</td><td></td><td></td><td rowspan="7">BIN-16-Bit</td></tr>
<tr><td>Operand</td><td>Bedeutung</td><td colspan="2">Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td rowspan="2">(s2)+0</td><td rowspan="2">Dezentraler Ausgang (RY) für die Anforderung der Daten</td><td colspan="2">• Höherwertiges Byte
Wird auf „0“ gesetzt.</td><td>0</td><td rowspan="4">Anwender</td></tr>
<tr><td colspan="2">• Niederwertiges Byte
Adresse eines dezentralen Ausgangs (RY) der intelligenten Station</td><td>0 bis 127</td></tr>
<tr><td>(s2)+1</td><td>Dezentrales Register (RWr) für Fehlercode Dezentraler Eingang (RX) mit dem der Abschluss der Datenübertragung angezeigt wird.</td><td colspan="2">• Höherwertiges Byte
Adresse eines dezentralen Registers (RWr) der intelligenten Station, in dem derselbe Fehlercode eingetragen wird wie in dem mit (s1)+0 angegebenen Operanden
• Niederwertiges Byte
Adresse eines dezentralen Eingangs (RX) der intelligenten Station</td><td>0 bis 15 oder FF (Bei der Angabe von FF ist kein Register angewählt.)
0 bis 127</td></tr>
<tr><td>(s2)+2</td><td>Anzeige des Abschlusses der Datenübertragung</td><td colspan="2">Geben Sie an, wie der Abschluss der Datenübertragung angezeigt wird:
0: Mit einem Operanden (RXn)
1: Mit 2 Operanden (RXn, RXn+1) (RXn+1 wird bei einem Fehler gesetzt.)</td><td>0 oder 1</td></tr>
</table>

Variablen

<table>
<tr><th>Operand</th><th colspan="4">Bedeutung</th><th>Werte-bereich</th><th>Festlegung des Inhalts durch</th><th>Datentyp</th></tr>
<tr><td>d1</td><td colspan="5">Erste Adresse des Operandenbereichs, in dem die gelesenen Daten gespeichert werden</td><td>Anwender</td><td>BIN-16-Bit</td></tr>
<tr><td rowspan="4">d2</td><td colspan="5">Bit-Operand, der nach der Ausführung der RIRCV-Anweisung für einen Zyklus gesetzt wird. Mit (d2)+1 wird ein Fehler bei der Ausführung angezeigt.</td><td rowspan="4">Bit</td></tr>
<tr><td>Operand</td><td colspan="2">Bedeutung</td><td>Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(d2)+0</td><td colspan="2">Anweisung ausgeführt</td><td>Zeigt die Beendigung der RIRCV-Anweisung an.
EIN: Anweisung ausgeführt
AUS: Anweisung nicht ausgeführt</td><td>—</td><td rowspan="2">System</td></tr>
<tr><td>(d2)+1</td><td colspan="2">Anweisung mit Fehler ausgeführt</td><td>Zeigt an, ob bei der Ausführung der RIRCV-Anweisung ein Fehler aufgetreten ist.
EIN: Anweisung mit Fehler ausgeführt
AUS: Anweisung ohne Fehler ausgeführt</td><td>—</td></tr>
</table>

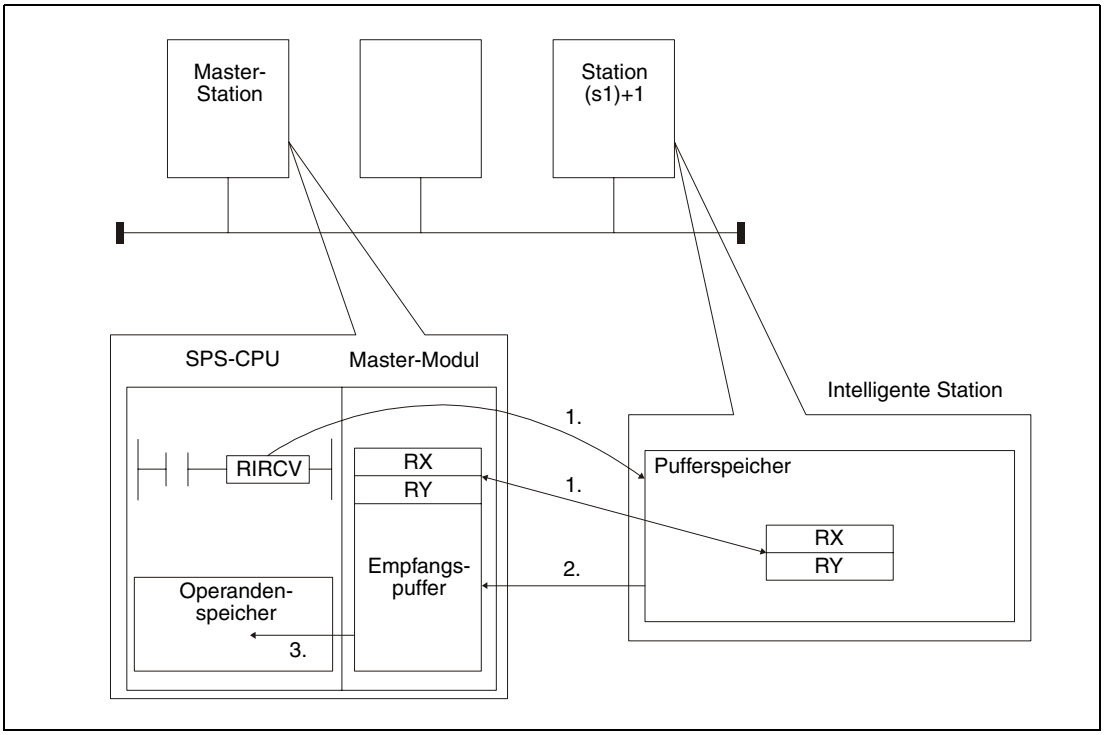
Funktions-weise

Daten aus dem Pufferspeicher einer intelligenten Station mit Handshake lesen

RIRCV Daten lesen (mit Handshake)

Die RIRCV-Anweisung kann nur in der SPS-CPU der Master-Station ausgeführt werden. Sie dient dazu, Daten aus dem Pufferspeicher einer intelligenten Station am CC-Link zu lesen. Der Datenaustausch wird über ein Handshake abgewickelt.

Funktionsschema der RIRCV-Anweisung:

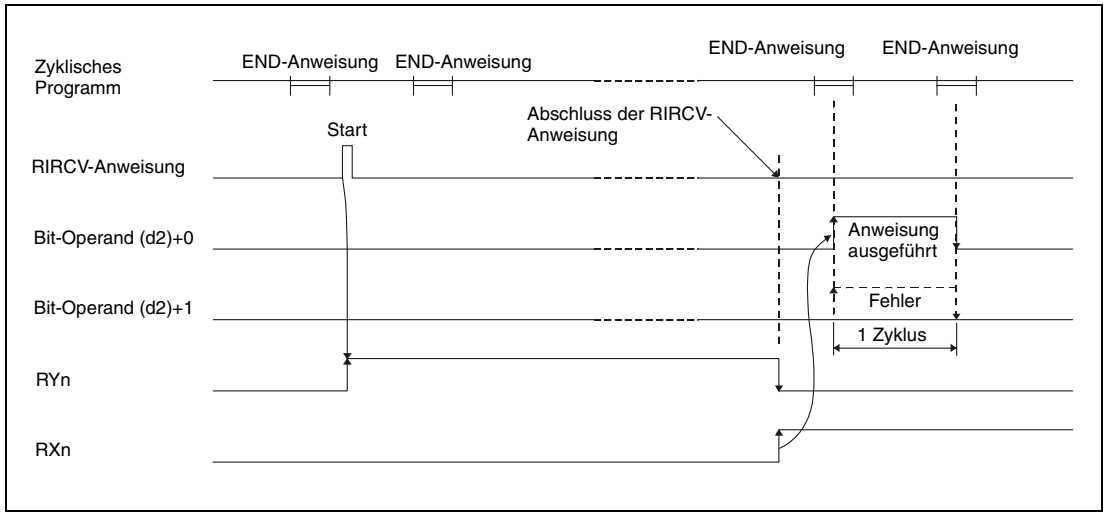


1. In der durch (s1)+1 angegebenen Station erfolgt der Zugriff auf dem Pufferspeicher ab der in (s1)+3 festgelegten Adresse. Die in s2 angegebenen Operanden werden für das Handshake verwendet.
2. Die Inhalte der Pufferspeicheradressen werden in den Empfangspuffer des Master-Moduls eingetragen.
3. In der SPS-CPU der Master-Station werden die Daten ab dem in d1 angegebenen Operanden gespeichert. Anschließend wird der in (d2)+0 angegebene Bit-Operand gesetzt.

Durch die in (d2)+0 und (d2)+1 angegebenen Operanden wird angezeigt, ob die Anweisung ausgeführt wurde und ob dabei ein Fehler aufgetreten ist.

- Der Bit-Operand (d2)+0 wird gesetzt, wenn die END-Anweisung des Zyklus ausgeführt wird, in dem die RIRCV-Anweisung beendet wurde. Bei der nächsten Bearbeitung der END-Anweisung wird dieser Bit-Operand wieder zurückgesetzt.
- Der Bit-Operand (d2)+1 zeigt einen Fehler bei der Ausführung der RIRCV-Anweisung an. Bei fehlerfreier Ausführung bleibt dieser Bit-Operand zurückgesetzt. Bei einem Fehler jedoch wird (d2)+1 gesetzt, wenn die END-Anweisung des Zyklus ausgeführt wird, in dem die RIRCV-Anweisung beendet wurde. Bei der nächsten Bearbeitung der END-Anweisung wird dieser Operand wieder zurückgesetzt.

Die folgende Abbildung zeigt den Signalverlauf bei Ausführung der RIRCV-Anweisung:



Für mehrere Stationen kann gleichzeitig eine RIRCVD-Anweisung ausgeführt werden. Auf eine intelligente Station kann jedoch nicht gleichzeitig mit mehreren RIRCV-Anweisungen von verschiedenen Stationen aus zugegriffen werden.

Fehlerquellen

Wenn bei der Ausführung der RIRCV-Anweisung ein Fehler auftritt, wird das Error-Flag SM0 gesetzt und im Sonderregister SD0 ein Fehlercode eingetragen.

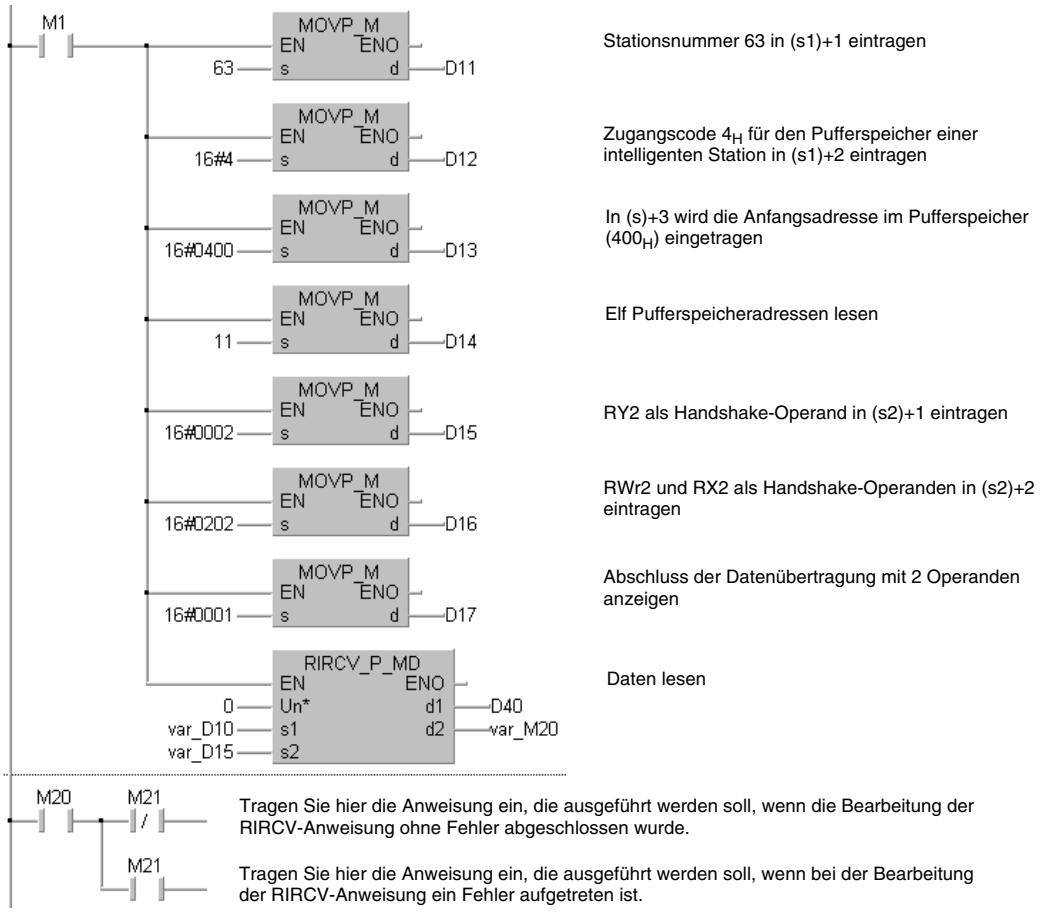
- Das in Un definierte Modul ist kein Sondermodul. (Fehlercode: 2112)
- Die auszuführende Anweisung wird nicht unterstützt. (Fehlercode: 4002)
- Die in der Anweisung angegebene Anzahl der Operanden ist ungültig. (Fehlercode: 4003)
- In der Anweisung ist ein unzulässiger Operand angegeben (Fehlercode: 4004)
- Der mit s definierte Operandenbereich enthält unzulässige Daten. (Fehlercode: 4100)
- Die Anzahl der Daten überschreitet den zulässigen Bereich. (Fehlercode: 4101)
- Die gespeicherten Daten oder Konstanten, die mit der Anweisung übertragen wurden, überschreiten den zulässigen Bereich. (Fehlercode: 4101)

Beispiel RIRCV

Das folgende Programm wird in der SPS der Master-Station bearbeitet. Wird der Merker M1 gesetzt, werden aus der intelligenten Station mit der Stationsnummer 63 ab der Pufferspeicheradresse 400_H die Inhalte von elf Pufferspeicheradressen gelesen. Die Daten werden in der CPU ab dem Register D40 gespeichert. Das CC-Link-Modul der Master-Station hat die Kopfadresse X/Y00. Als Operanden für den Handshake werden RX2, RY2 und RWr2 verwendet. Der Abschluss der Datenübertragung wird durch zwei Operanden angezeigt. ((s2)+2 = 1)

- IEC-Editoren (Auf der nächsten Seite ist dieses Beispiel in der MELSEC-Anweisungliste und dem Kontaktplan des GX Developers dargestellt.)

Kontaktplan (GX IEC Developer)



IEC-Anweisungsliste

```
LD              M1
MOVP_M          63,       D11
MOVP_M          16#0004,  D12
MOVP_M          16#0400,  D13
MOVP_M          11,       D14
MOVP_M          16#0002,  D15
MOVP_M          16#0202,  D16
MOVP_M          16#0001,  D17
RIRCV_P_MD      0,        var_D10,  var_d15,  D40,  var_M20
```

Die Operanden und Anweisungen sind oben beim Programmbeispiel für den Kontaktplan erläutert.

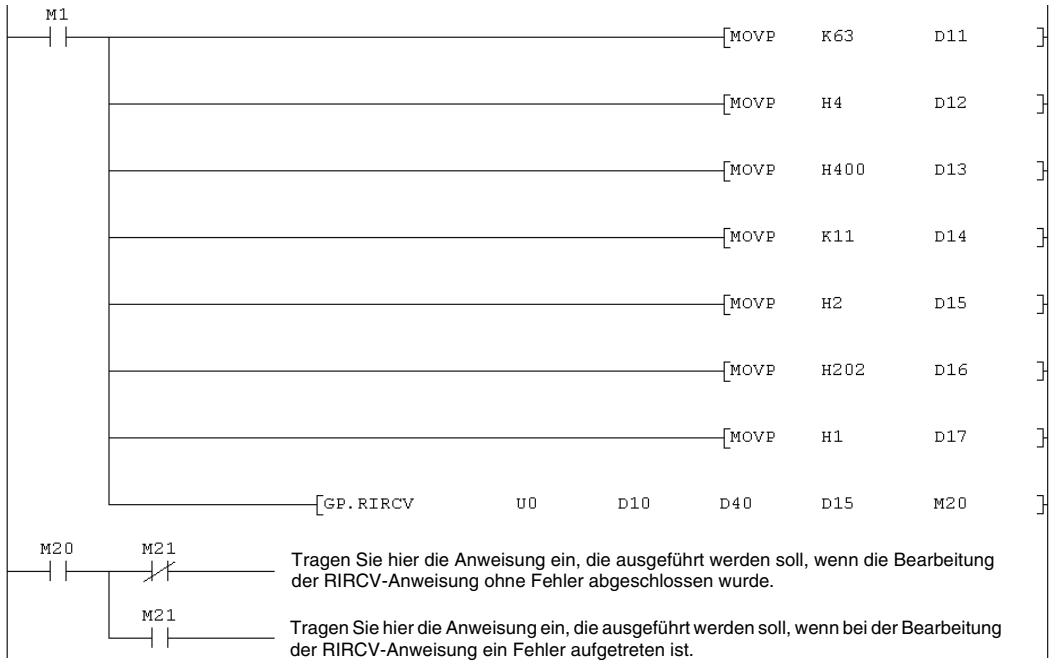
```
LD              M20
ANDN            M21
```
Ausführung, wenn die Bearbeitung der RIRCV-Anweisung ohne Fehler abgeschlossen wurde

```
LD              M20
AND             M21
```
Ausführung, wenn ein Fehler bei der Bearbeitung der RIRCV-Anweisung aufgetreten ist

HINWEIS *Bei den IEC-Editoren des GX IEC Developers müssen die Variablen im Header der Programmorganisationseinheit (POE) definiert werden. Ohne Variablendefinition werden beim Überprüfen oder Kompilieren des Programms Fehler gemeldet.*
Weitere Informationen finden Sie im Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung.

- MELSEC-Anweisungliste und Kontaktplan des GX Developers
  Die Operanden und Anweisungen sind auf der vorherigen Seite beim Programmbeispiel für den Kontaktplan des GX IEC Developers erläutert.

Kontaktplan (GX Developer)



MELSEC-Anweisungsliste

```
MELSEC
LD        M1
MOVP      K63     D11
MOVP      H0004   D12
MOVP      H0400   D13
MOVP      K11     D14
MOVP      H0002   D15
MOVP      H0202   D16
MOVP      H0001   D17
GP.RIRCV  U0
          D10
          D40
          D15
          M20

MELSEC
LD        M20
MPS
ANI       M21
Ausführung, wenn die Bearbeitung der RIRCV-Anweisung ohne Fehler abgeschlossen wurde
MRD
AND       M21
Ausführung, wenn ein Fehler bei der Bearbeitung der RIRCV-Anweisung aufgetreten ist
```

11.5.10 RISEND (A-Serie)

CPU

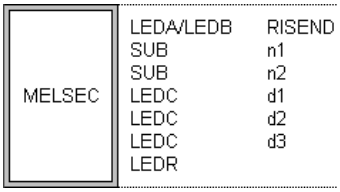
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●		

Operanden MELSEC A

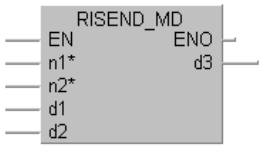
	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)								Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9011
n1																	●	●								
n2																	●	●								
d1								●	●	●	●	●											29			●
d2								●	●	●	●	●														
d3		●	●	●	●	●																				

GX IEC Developer

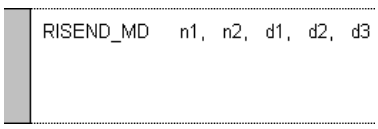
MELSEC-Anweisungsliste



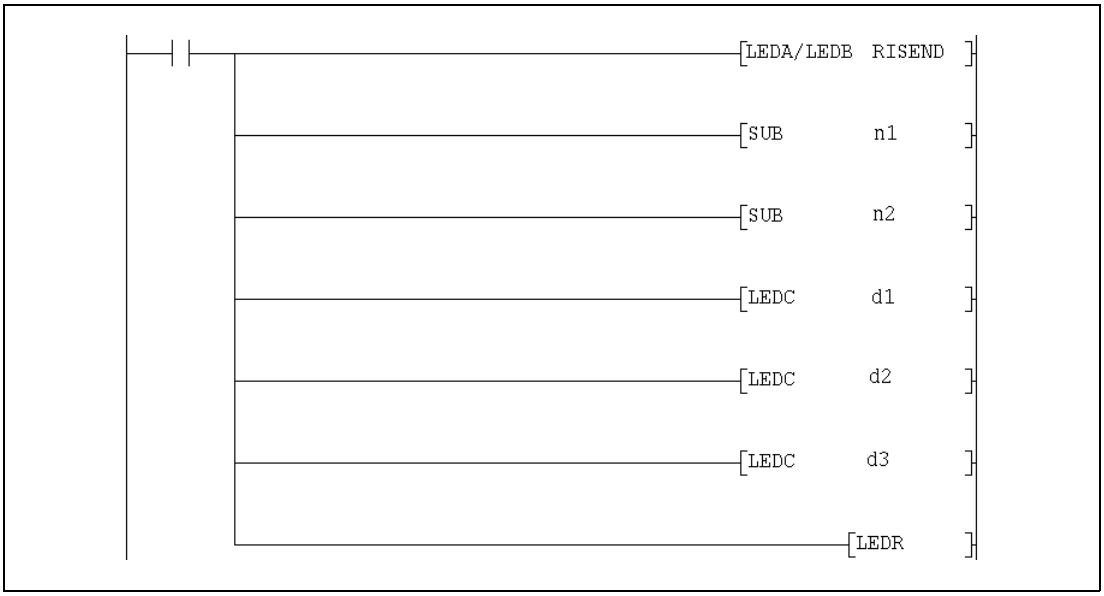
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

<table>
<tr><th>Operand</th><th colspan="4">Bedeutung</th><th>Werte-bereich</th><th>Festlegung des Inhalts durch</th><th>Datentyp</th></tr>
<tr><td>n1</td><td colspan="4">Kopfadresse des CC-Link-Master-Moduls auf dem Baugruppenträger
(Nur die ersten beiden Stellen der 3-stelligen Adresse werden angegeben. Beispiel: Kopfadresse X/Y100 -> 10_H)</td><td>0 bis FE_H</td><td>Anwender</td><td>BIN-16-Bit</td></tr>
<tr><td>n2</td><td colspan="4">Stationsnummer der intelligenten Station, zu der Daten übertragen werden sollen</td><td>1 bis 64</td><td>Anwender</td><td>BIN-16-Bit</td></tr>
<tr><td rowspan="8">d1</td><td colspan="5">Erster Operand des Bereiches mit Informationen zur Ausführung der Anweisung</td><td rowspan="8">BIN-16-Bit</td></tr>
<tr><td>Operand</td><td colspan="2">Bedeutung</td><td>Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(d1)+0</td><td colspan="2">Ausführungs-status der Anweisung</td><td>Zeigt an, ob bei der Bearbeitung der Anweisung ein Fehler aufgetreten ist:
0000_H: Fehlerfreie Bearbeitung
Jeder andere Wert als 0000_H: Bei der Bearbeitung ist ein Fehler aufgetreten.
Nähere Hinweise zu Fehlercodes finden Sie in der Bedienungsanleitung des CC-Link-Moduls</td><td>—</td><td>System</td></tr>
<tr><td>(d1)+1</td><td colspan="2">Datenlänge</td><td>Geben Sie hier an, wieviele Daten (Worte) geschrieben werden sollen.</td><td>1 bis 480</td><td rowspan="5">Anwender</td></tr>
<tr><td>(d1)+2</td><td colspan="2">Zugriffscode</td><td>Geben Sie den Wert 0004_H an.
(= In den Pufferspeicher einer intelligenten Station schreiben.)</td><td>0004_H</td></tr>
<tr><td>(d1)+3</td><td colspan="2">Fehlerprüfung</td><td>Geben Sie an, mit welchem Operanden ein Fehler bei der Ausführung der RISEND-Anweisung angezeigt wird:
0: Operand d1
1: Operand RX+1</td><td>0 oder 1</td></tr>
<tr><td>(d1)+4</td><td colspan="2">Anfangsadresse</td><td>Anfangsadresse im Puffer-speicher der intelligenten Station</td><td>Abhängig von der Station, auf die zugegriffen wird</td></tr>
<tr><td>(d1)+5 bis (d1)+n</td><td colspan="2">Speicherbereich für die Daten, die übertragen werden</td><td>Die Größe dieses Bereiches wird durch die in (d1)+1 angegebene Datenlänge bestimmt.</td><td>—</td></tr>
<tr><td rowspan="5">d2</td><td colspan="5">Angabe der Link-Operanden, die für das Handshake verwendet werden.</td><td rowspan="5">BIN-16-Bit</td></tr>
<tr><td>Operand</td><td colspan="2">Bedeutung</td><td>Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td rowspan="2">(d2)+0</td><td colspan="2" rowspan="2">Dezentraler Eingang (RX)
Dezentraler Ausgang (RY)</td><td>• Höherwertiges Byte
Adresse eines dezentralen Eingangs (RX) der intell. Station</td><td>0 bis 127</td><td rowspan="3">Anwender
(Die Adressen werden zwar vom Anwender angegeben, gesetzt und zurückgesetzt bzw. verändert werden die Operanden aber vom System)</td></tr>
<tr><td>• Niederwertiges Byte
Adresse eines dezentralen Ausgangs (RY) der intelligenten Station</td><td>0 bis 127</td></tr>
<tr><td>(d2)+1</td><td colspan="2">Dezentrales Register (RWr)</td><td>Adresse eines dezentralen Registers (RWr) der intelligenten Station</td><td>0 bis 15 oder FF (Bei der Angabe von FF ist kein Register angewählt.)</td></tr>
</table>

Variablen

<table>
<tr><th>Operand</th><th colspan="4">Bedeutung</th><th>Werte-bereich</th><th>Festlegung des Inhalts durch</th><th>Datentyp</th></tr>
<tr><td rowspan="3">d3</td><td colspan="4">Bit-Operand, der nach der Ausführung der RISEND-Anweisung für einen Zyklus gesetzt wird. Mit (d3)+1 wird ein Fehler bei der Ausführung angezeigt.</td><td></td><td></td><td rowspan="3">Bit</td></tr>
<tr><td>Operand</td><td>Bedeutung</td><td>Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(d3)+0</td><td>Anweisung ausgeführt</td><td>Zeigt die Beendigung der Anweisung an.
EIN: Anweisung ausgeführt
AUS: Anweisung nicht ausgeführt</td><td>—</td><td rowspan="2">System</td></tr>
<tr><td></td><td>(d3)+1</td><td>Anweisung mit Fehler ausgeführt</td><td>Zeigt an, ob bei der Ausführung der Anweisung ein Fehler aufgetreten ist.
EIN: Anweisung mit Fehler ausgeführt
AUS: Anweisung ohne Fehler ausgeführt</td><td>—</td><td></td></tr>
</table>

Funktions-weise

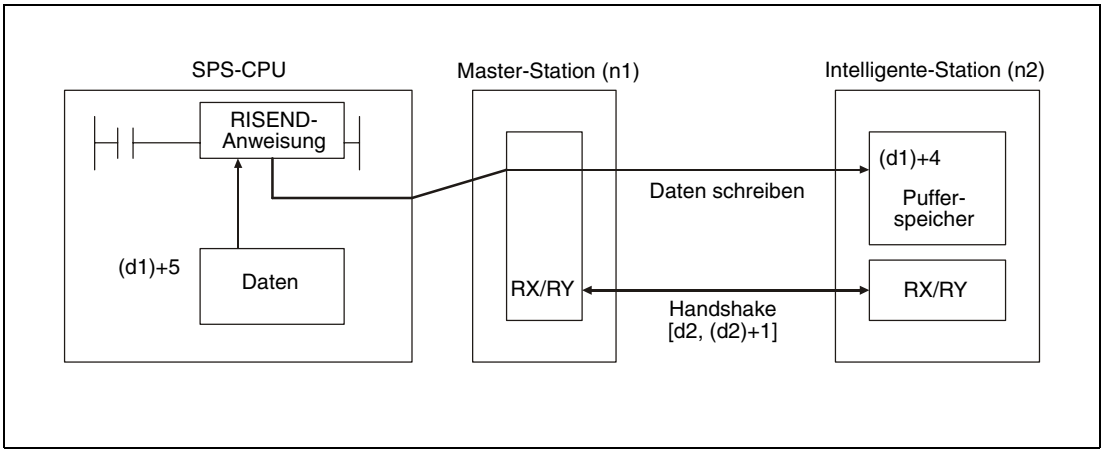
Daten mit Handshake in den Pufferspeicher einer intelligenten Station übertragen

RISEND Daten schreiben (mit Handshake)

Die RISEND-Anweisung kann nur in der SPS-CPU der Master-Station ausgeführt werden. Mit ihr werden Daten in den Pufferspeicher einer intelligenten CC-Link-Station eingetragen. Der Datenaustausch wird über ein Handshake abgewickelt.

Der Operand (d1)+1 enthält die Angabe, wieviele Daten übertragen werden sollen. In dem Operanden (d1)+3 wird die erste Pufferspeicheradresse angegeben, in die Daten eingetragen werden soll. Der Operand n2 enthält die Stationsnummer der anderen Station. Diese Station ist an die Master-Station angeschlossen, die in n1 spezifiziert ist. Die Daten, die zur intelligenten Station übertragen werden, sind ab (d1)+5 gespeichert.

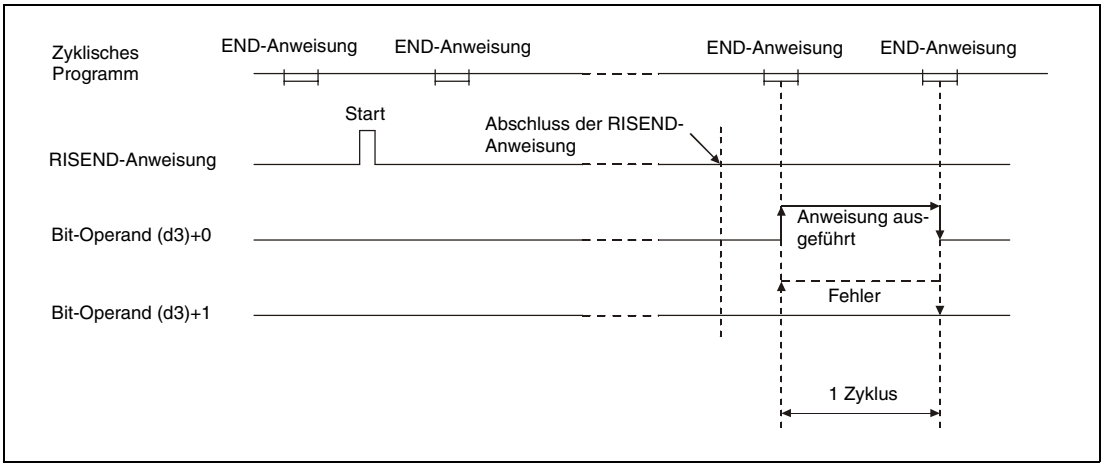
Funktionsschema der RISEND-Anweisung:



Durch die in (d3)+0 und (d3)+1 angegebenen Operanden wird angezeigt, ob die Anweisung ausgeführt wurde und ob dabei ein Fehler aufgetreten ist.

- Der Bit-Operand (d3)+0 wird gesetzt, wenn die END-Anweisung des Zyklus ausgeführt wird, in dem die RISEND-Anweisung beendet wurde. Bei der nächsten Bearbeitung der END-Anweisung wird dieser Bit-Operand wieder zurückgesetzt.
- Der Bit-Operand (d3)+1 zeigt einen Fehler bei der Ausführung der RISEND-Anweisung an. Bei fehlerfreier Ausführung bleibt dieser Bit-Operand zurückgesetzt. Bei einem Fehler jedoch wird (d3)+1 gesetzt, wenn die END-Anweisung des Zyklus ausgeführt wird, in dem die RISEND-Anweisung beendet wurde. Bei der nächsten Bearbeitung der END-Anweisung wird dieser Operand wieder zurückgesetzt.

Die folgende Abbildung zeigt den Signalverlauf bei Ausführung der RISEND-Anweisung:



Für mehrere Stationen kann gleichzeitig eine RISEND-Anweisung ausgeführt werden. Auf eine intelligente Station kann jedoch nicht gleichzeitig mit mehreren RISEND-Anweisungen von verschiedenen Stationen aus zugegriffen werden.

Ausführungsbedingungen

Wird die RISEND-Anweisung in Verbindung mit einer LEDA-Anweisung programmiert, wird die RISEND-Anweisung ausgeführt, solange die Ausführungsbedingung der LEDA-Anweisung eingeschaltet ist.
Bei Anwendung einer LEDB-Anweisung dagegen wird der Lesevorgang nur bei der steigenden Flanke der Ausführungsbedingung ausgeführt.

Beachten Sie, dass für die Bearbeitung der RISEND-Anweisung mehrere Zyklen benötigt werden. Starten Sie daher den nächsten Lesevorgang erst dann, nachdem durch den Operanden (d3)+0 angezeigt wurde, dass die Bearbeitung der RISEND-Anweisung abgeschlossen ist. (Eine RISEND-Anweisung wird nicht ausgeführt, wenn sie erneut gestartet wird, bevor die vorherige Bearbeitung abgeschlossen ist.)

Die Netzwerkparameter müssen über die RLPA-Anweisung eingestellt sein, bevor die RISEND-Anweisung ausgeführt werden kann.

Fehlerquellen

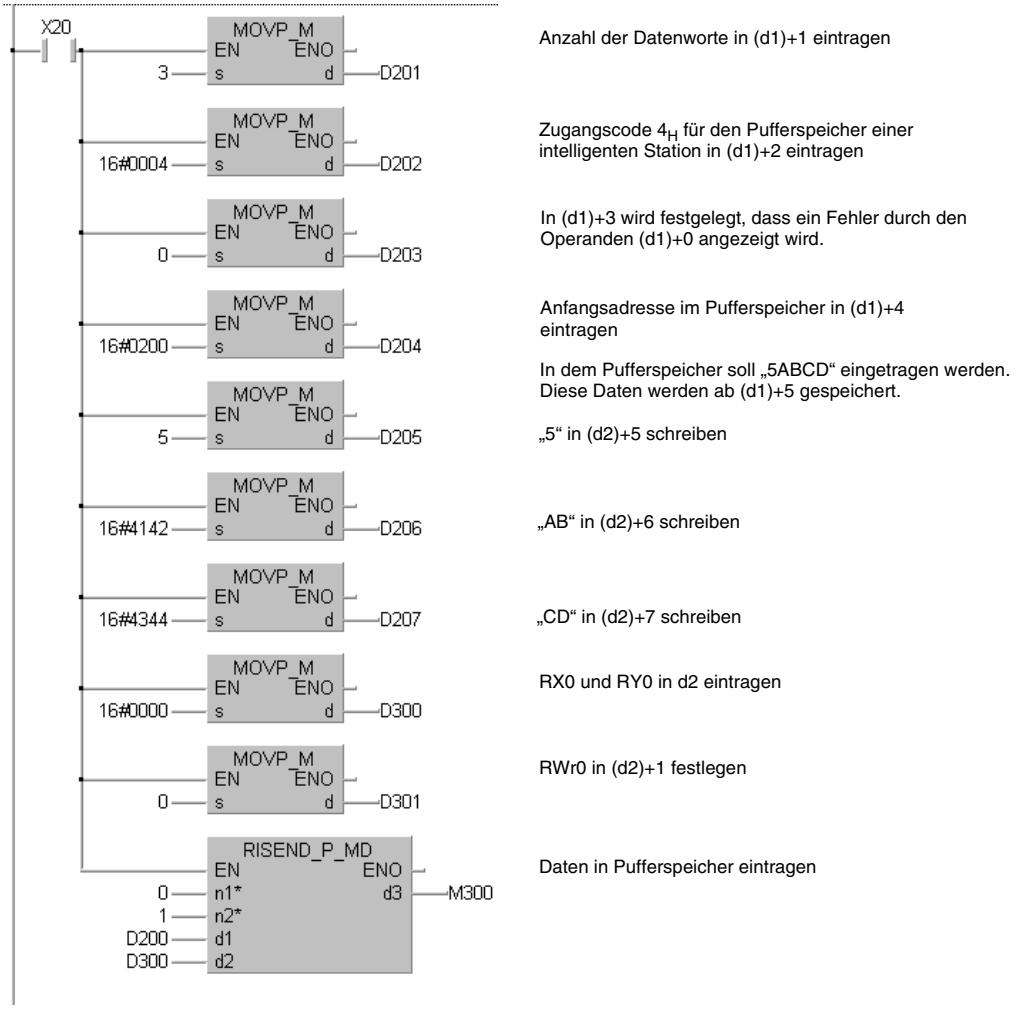
Wird als Anzahl der zu übertragenen Daten in (d1)+1 eine 0 oder ein Wert außerhalb des Bereichs von 1 bis 480 angegeben, wird beim Abschluss der RISEND-Anweisung der Fehlercode $BB42_H$ in den Operanden (d1)+0 eingetragen.

Beispiel RISEND

Das folgende Programm wird in der SPS der Master-Station bearbeitet und trägt Daten in die Pufferspeicheradressen 200_H bis 202_H der intelligenten Station mit der Stationsnummer 1 ein. Als Operanden für den Handshake werden RX0, RY0 und RWr0 verwendet. Der mit (d1)+0 angegebene Operand dient zur Anzeige eines Fehlers. Das CC-Link-Modul der Master-Station hat die Kopfadresse X/Y000.

- IEC-Editoren (Auf der nächsten Seite ist dieses Beispiel in der MELSEC-Anweisungliste und dem Kontaktplan des GX Developers dargestellt.)

Kontaktplan (GX IEC Developer)



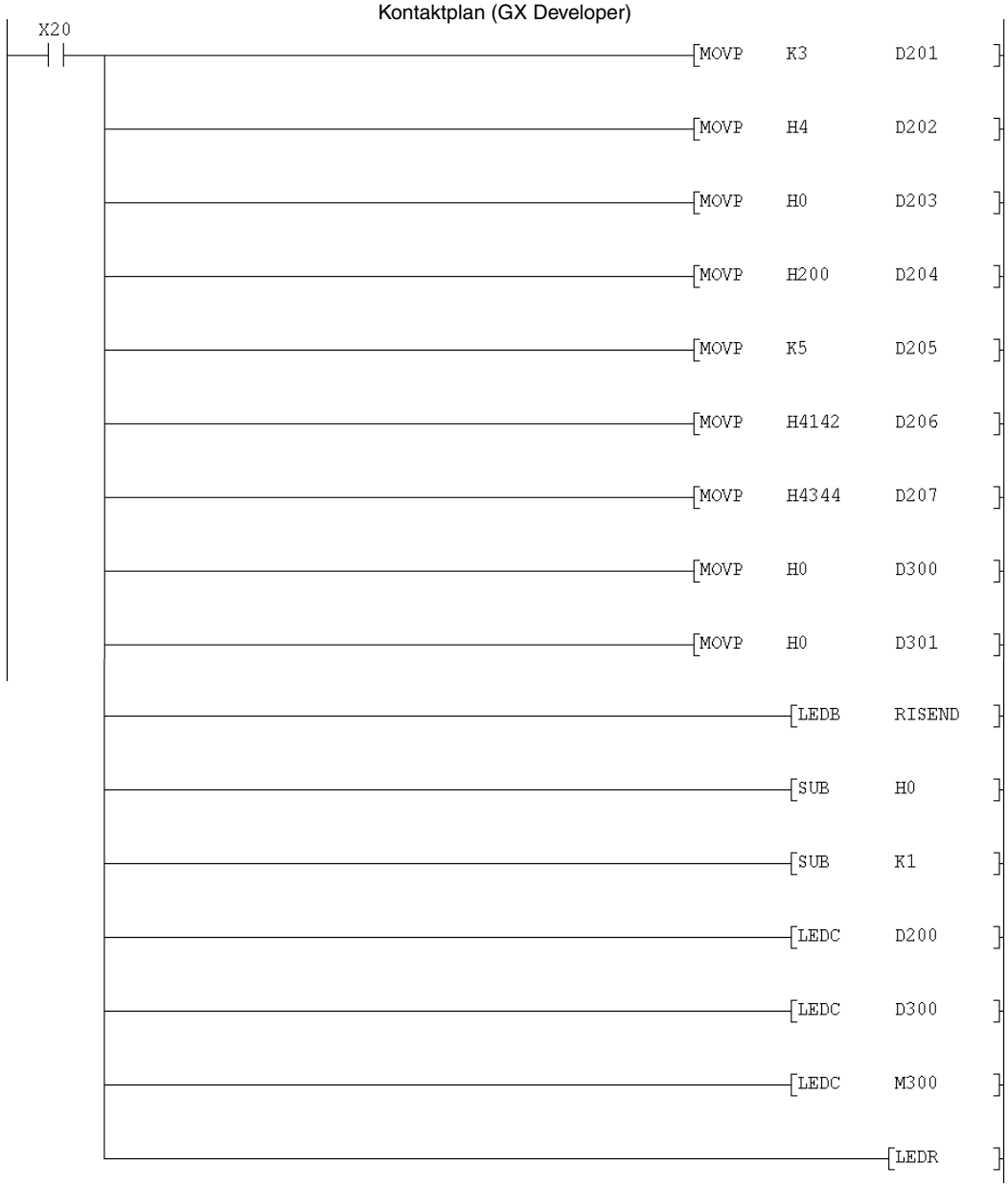
IEC-Anweisungsliste

```
LD          X20
MOVP_M      2,          D201
MOVP_M      16#0004,    D202
MOVP_M      16#0000,    D203
MOVP_M      16#0200,    D204
MOVP_M      5,          D205
MOVP_M      16#4142,    D206
MOVP_M      16#4344,    D207
MOVP_M      16#0000,    D300
MOVP_M      0,          D301
RISEND_P_MD             0,      1,      D200,   D300,   M300
```

Die Operanden und Anweisungen sind oben beim Programmbeispiel für den Kontaktplan erläutert.

- MELSEC-Anweisungliste und Kontaktplan des GX Developers
  Die Operanden und Anweisungen sind auf der vorherigen Seite beim Programmbeispiel für den Kontaktplan des GX IEC Developers erläutert.

Kontaktplan (GX Developer)



MELSEC-Anweisungsliste

```
MELSEC
        LD      X20
        MOVP    K3      D201
        MOVP    H0004   D202
        MOVP    H0      D203
        MOVP    H0200   D204
        MOVP    K5      D205
        MOVP    H4142   D206
        MOVP    H4344   D207
        MOVP    H0      D300
        MOVP    H0      D301
        LEDB    RISEND
        SUB     H0
        SUB     K1
        LEDC    D200
        LEDC    D300
        LEDC    M300
        LEDR
```

Hinweise zur Programmierung der erweiterten Anweisungen bei einer SPS der A-Serie finden Sie im Kapitel 3.3 dieses Handbuchs.

11.5.11 RISEND (QnA-Serie und System Q)

CPU

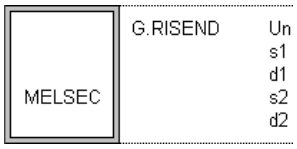
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Operanden MELSEC Q

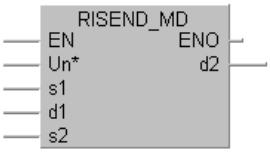
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□N□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
s1	—	●	●	—	—	—	—	—	—	SM0	10
s2	—	●	●	—	—	—	—	—	—		
d1	—	●	●	—	—	—	—	—	—		
d2	●	●	●	—	—	—	—	—	—		

GX IEC Developer

MELSEC-Anweisungsliste



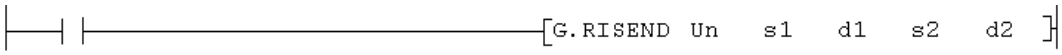
Kontaktplan



IEC-Anweisungsliste

```
RISEND_MD   Un, s1, d1, s2, d2
```

GX Developer



Variablen

<table>
<tr><th>Operand</th><th colspan="4">Bedeutung</th><th>Werte-bereich</th><th>Festlegung des Inhalts durch</th><th>Datentyp</th></tr>
<tr><td>Un</td><td colspan="4">Kopfadresse des Master-Moduls auf dem Baugruppenträger (Es werden nur die ersten beiden Stellen der 3-stelligen Adresse angegeben. Beispiel: Kopfadresse X/Y100 -> 10_H)</td><td>0 bis FE_H</td><td>Anwender</td><td>BIN-16-Bit</td></tr>
<tr><td rowspan="6">s1</td><td colspan="5">Erster Operand des Bereiches mit Informationen zur Ausführung der Anweisung</td><td rowspan="6">BIN-16-Bit</td></tr>
<tr><td>Operand</td><td>Bedeutung</td><td>Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(s1)+0</td><td>Ausführungs-status der Anweisung</td><td>Zeigt an, ob bei der Bearbeitung der Anweisung ein Fehler aufgetreten ist:
0000_H: Fehlerfreie Bearbeitung
Jeder andere Wert als 0000_H: Bei der Bearbeitung ist ein Fehler aufgetreten.(siehe Bedienungsanleitung des CC-Link-Moduls)</td><td>—</td><td>System</td></tr>
<tr><td>(s1)+1</td><td>Stationsnummer</td><td>Nummer der intelligenten Station, aus der Daten gelesen werden sollen</td><td>0 bis 64</td><td rowspan="4">Anwender</td></tr>
<tr><td>(s1)+2</td><td>Zugriffscode</td><td>Geben Sie den Wert 0004_H an.
(= in den Pufferspeicher einer intelligenten Station schreiben.)</td><td>0004_H</td></tr>
<tr><td>(s1)+3</td><td>Anfangsadresse</td><td>Anfangsadresse im Pufferspeicher</td><td>Abhängig von der Station, auf die zugegriffen wird</td></tr>
<tr><td>(s1)+4</td><td>Datenlänge</td><td>Geben Sie hier an, wieviele Daten (Worte) übertragen werden sollen.</td><td>1 bis 480</td></tr>
<tr><td rowspan="6">s2</td><td colspan="5">Angabe der Link-Operanden, die für das Handshake verwendet werden.</td><td rowspan="6">BIN-16-Bit</td></tr>
<tr><td>Operand</td><td>Bedeutung</td><td>Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td rowspan="2">(s2)+0</td><td rowspan="2">Dezentraler Ausgang (RY) für die Anforderung der Daten</td><td>• Höherwertiges Byte
Wird auf „0“ gesetzt.</td><td>0</td><td rowspan="4">Anwender</td></tr>
<tr><td>• Niederwertiges Byte
Adresse eines dezentralen Ausgangs (RY) der intelligenten Station</td><td>0 bis 127</td></tr>
<tr><td rowspan="2">(s2)+1</td><td rowspan="2">Dezentrales Register (RWr) für Fehlercode Dezentraler Eingang (RX), mit dem der Abschluss der Datenübertragung angezeigt wird.</td><td>• Höherwertiges Byte
Adresse eines dezentralen Registers (RWr) der intelligenten Station, in dem derselbe Fehlercode eingetragen wird wie in dem mit (s1)+0 angegebenen Operanden</td><td>0 bis 15 oder FF (Bei der Angabe von FF ist kein Register angewählt.)</td></tr>
<tr><td>• Niederwertiges Byte
Adresse eines dezentralen Eingangs (RX) der intelligenten Station</td><td>0 bis 127</td></tr>
<tr><td>(s2)+2</td><td>Anzeige des Abschlusses der Datenübertragung</td><td>Geben Sie an, wie der Abschluss der Datenübertragung angezeigt wird:
0: Mit einem Operanden (RXn)
1: Mit 2 Operanden (RXn, RXn+1) (RXn+1 wird bei einem Fehler gesetzt.)</td><td>0 oder 1</td></tr>
<tr><td>d1</td><td colspan="4">Erste Adresse des Operandenbereichs, in dem die Daten gespeichert sind, die zur intelligenten Station übertragen werden.</td><td>Anwender</td><td>BIN-16-Bit</td></tr>
</table>

Variablen

<table>
<tr><th>Operand</th><th colspan="3">Bedeutung</th><th>Werte-bereich</th><th>Festlegung des Inhalts durch</th><th>Datentyp</th></tr>
<tr><td rowspan="4">d2</td><td colspan="5">Bit-Operand, der nach der Ausführung der RISEND-Anweisung für einen Zyklus gesetzt wird. Mit (d2)+1 wird ein Fehler bei der Ausführung angezeigt.</td><td rowspan="4">Bit</td></tr>
<tr><td>Operand</td><td>Bedeutung</td><td>Beschreibung</td><td>Werte-bereich</td><td>Festlegung durch</td></tr>
<tr><td>(d2)+0</td><td>Anweisung ausgeführt</td><td>Zeigt die Beendigung der RISEND-Anweisung an.
EIN: Anweisung ausgeführt
AUS: Anweisung nicht ausgeführt</td><td>—</td><td rowspan="2">System</td></tr>
<tr><td>(d2)+1</td><td>Anweisung mit Fehler ausgeführt</td><td>Zeigt an, ob bei der Ausführung der RISEND-Anweisung ein Fehler aufgetreten ist.
EIN: Anweisung mit Fehler ausgeführt
AUS: Anweisung ohne Fehler ausgeführt</td><td>—</td></tr>
</table>

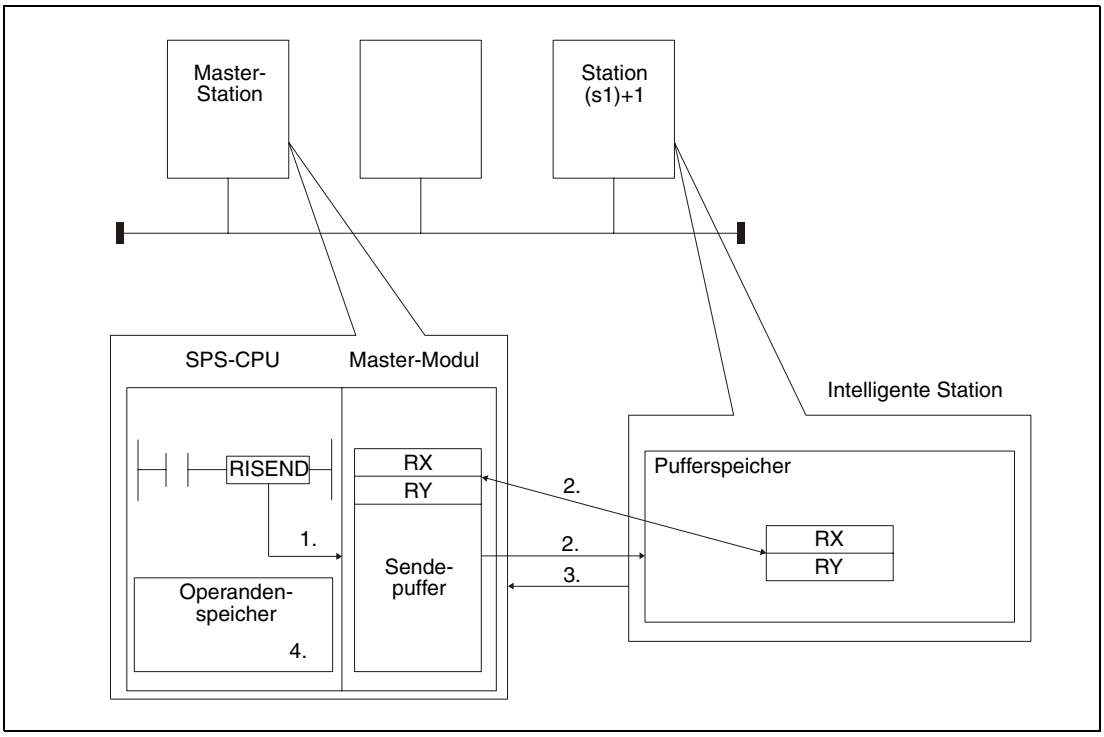
Funktions-weise

Daten mit Handshake in den Pufferspeicher einer intelligenten Station eintragen

RISEND Daten schreiben (mit Handshake)

Die RISEND-Anweisung kann nur in der SPS-CPU der Master-Station ausgeführt werden. Mit dieser Anweisung werden Daten in den Pufferspeicher einer intelligenten Station am CC-Link übertragen. Der Datenaustausch wird über ein Handshake abgewickelt.

Prinzipieller Ablauf der RISEND-Anweisung:

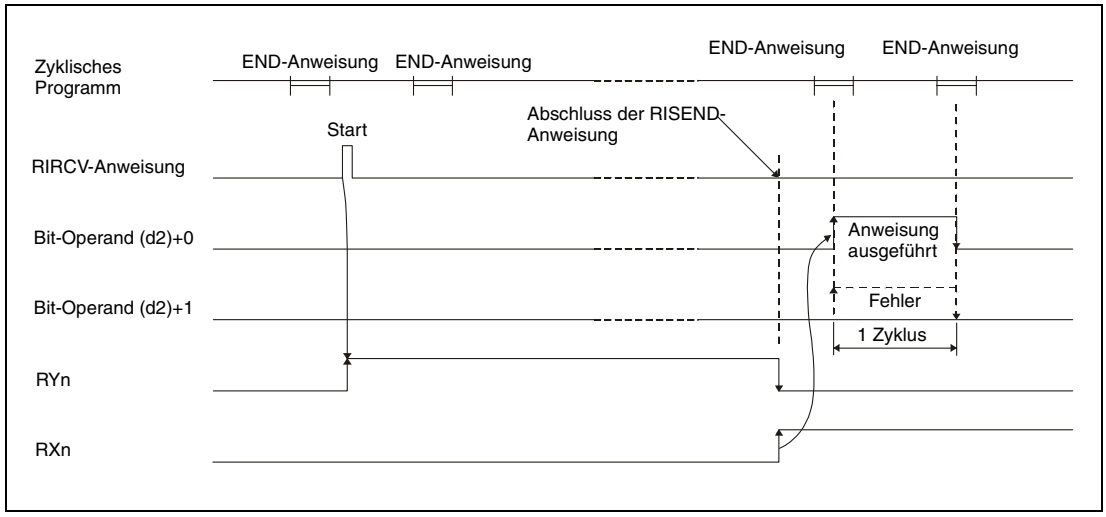


1. Die Daten für die intelligente Station werden in den Sendepuffer der Master-Station eingetragen.
2. In der durch (s1)+1 angegebenen Station werden die Daten in dem Pufferspeicher ab der in (s1)+3 festgelegten Adresse eingetragen. Die in s2 angegebenen Operanden werden für das Handshake verwendet.
3. Der Master-Station wird angezeigt, dass die Übertragung der Daten beendet ist.
4. Der in (d2)+0 angegebene Bit-Operand wird gesetzt.

Durch die in (d2)+0 und (d2)+1 angegebenen Operanden wird angezeigt, ob die Anweisung ausgeführt wurde und ob dabei ein Fehler aufgetreten ist.

- Der Bit-Operand (d2)+0 wird gesetzt, wenn die END-Anweisung des Zyklus ausgeführt wird, in dem die RISEND-Anweisung beendet wurde. Bei der nächsten Bearbeitung der END-Anweisung wird dieser Bit-Operand wieder zurückgesetzt.
- Der Bit-Operand (d2)+1 zeigt einen Fehler bei der Ausführung der RISEND-Anweisung an. Bei fehlerfreier Ausführung bleibt dieser Bit-Operand zurückgesetzt. Bei einem Fehler jedoch wird (d2)+1 gesetzt, wenn die END-Anweisung des Zyklus ausgeführt wird, in dem die RISEND-Anweisung beendet wurde. Bei der nächsten Bearbeitung der END-Anweisung wird dieser Operand wieder zurückgesetzt.

Die folgende Abbildung zeigt den Signalverlauf bei Ausführung der RISEND-Anweisung:



Für mehrere Stationen kann gleichzeitig eine RISEND-Anweisung ausgeführt werden. Auf eine intelligente Station kann jedoch nicht gleichzeitig mit mehreren RISEND-Anweisungen von verschiedenen Stationen aus zugegriffen werden.

Fehler-quellen

Wenn bei der Ausführung der RISEND-Anweisung ein Fehler auftritt, wird das Error-Flag SM0 gesetzt und im Sonderregister SD0 ein Fehlercode eingetragen.

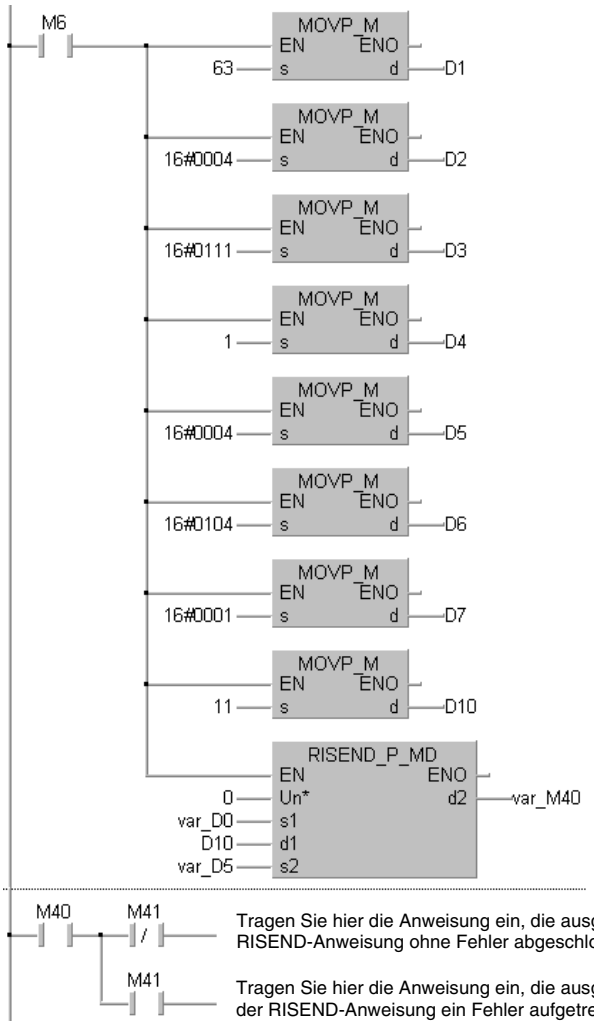
- Das in Un definierte Modul ist kein Sondermodul. (Fehlercode: 2112)
- Die auszuführende Anweisung wird nicht unterstützt. (Fehlercode: 4002)
- Die in der Anweisung angegebene Anzahl der Operanden ist ungültig. (Fehlercode: 4003)
- In der Anweisung ist ein unzulässiger Operand angegeben (Fehlercode: 4004)
- Der mit s definierte Operandenbereich enthält unzulässige Daten. (Fehlercode: 4100)
- Die Anzahl der Daten überschreitet den zulässigen Bereich. (Fehlercode: 4101)
- Die gespeicherten Daten oder Konstanten, die mit der Anweisung übertragen wurden, überschreiten den zulässigen Bereich. (Fehlercode: 4101)

Beispiel RISEND

Das folgende Programm wird in der SPS der Master-Station bearbeitet und überträgt ein Wort in die Pufferspeicheradresse 111_H der intelligenten Station mit der Stationsnummer 63. Das CC-Link-Modul der Master-Station hat die Kopfadresse X/Y00. Als Operanden für den Handshake werden RX4, RY4 und RWr1 verwendet. Der Abschluss der Datenübertragung wird durch zwei Operanden angezeigt. ((s2)+2 = 1)

- IEC-Editoren (Auf der nächsten Seite ist dieses Beispiel in der MELSEC-Anweisungliste und dem Kontaktplan des GX Developers dargestellt.)

Kontaktplan (GX IEC Developer)



Stationsnummer 63 in (s1)+1 eintragen

Zugangscode 4_H für den Pufferspeicher einer intelligenten Station in (s1)+2 eintragen

In (s)+3 wird die Anfangsadresse im Pufferspeicher (111_H) eingetragen

Eine Pufferspeicheradressen beschreiben

RY4 als Handshake-Operand in (s2)+1 eintragen

RWr1 und RX4 als Handshake-Operanden in (s2)+2 eintragen

Abschluss der Datenübertragung mit 2 Operanden anzeigen

In D10 wird der Wert „11“ eingetragen, der dann zur Pufferspeicher 111_H übertragen wird.

Daten übertragen

Tragen Sie hier die Anweisung ein, die ausgeführt werden soll, wenn die Bearbeitung der RISEND-Anweisung ohne Fehler abgeschlossen wurde.

Tragen Sie hier die Anweisung ein, die ausgeführt werden soll, wenn bei der Bearbeitung der RISEND-Anweisung ein Fehler aufgetreten ist.

IEC-Anweisungsliste

```
LD              M6
MOVP_M          63,       D1
MOVP_M          16#0004,  D2
MOVP_M          16#0111,  D3
MOVP_M          1,        D4
MOVP_M          16#0004,  D5
MOVP_M          16#0104,  D6
MOVP_M          16#0001,  D7
MOVP_M          11,       D10
RISEND_P_MD     0,        var_D0,  D10,  var_D5,  var_M40
```

Die Operanden und Anweisungen sind oben beim Programmbeispiel für den Kontaktplan erläutert.

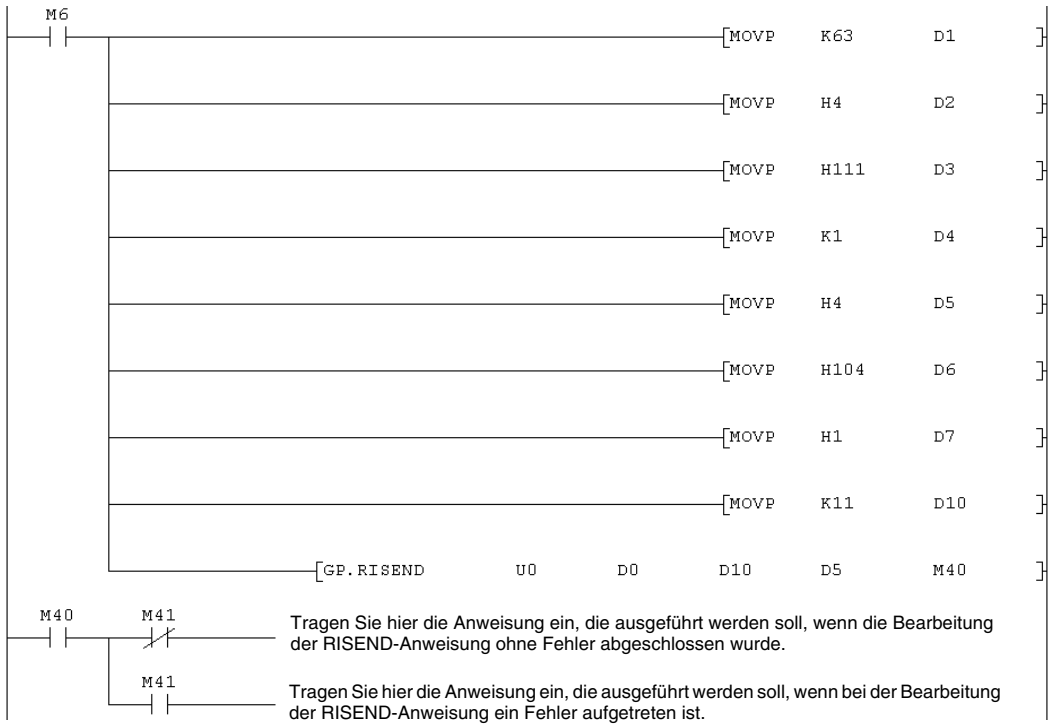
```
LD              M40
ANDN            M41
```
Ausführung, wenn die Bearbeitung der RISEND-Anweisung ohne Fehler abgeschlossen wurde

```
LD              M40
AND             M41
```
Ausführung, wenn ein Fehler bei der Bearbeitung der RISEND-Anweisung aufgetreten ist

HINWEIS *Bei den IEC-Editoren des GX IEC Developers müssen die Variablen im Header der Programmorganisationseinheit (POE) definiert werden. Ohne Variablendefinition werden beim Überprüfen oder Kompilieren des Programms Fehler gemeldet.*
Weitere Informationen finden Sie im Abs. 3.5.2 „Array- und Registeradressierung im GX IEC Developer“ dieser Programmieranleitung.

- MELSEC-Anweisungliste und Kontaktplan des GX Developers
  Die Operanden und Anweisungen sind auf der vorherigen Seite beim Programmbeispiel für den Kontaktplan des GX IEC Developers erläutert.

Kontaktplan (GX Developer)



MELSEC-Anweisungsliste

```
MELSEC  LD          M6
        MOVP        K63         D1
        MOVP        H0004       D2
        MOVP        H0111       D3
        MOVP        K1          D4
        MOVP        H0004       D5
        MOVP        H0104       D6
        MOVP        H0001       D7
        MOVP        K11         D10
        GP.RISEND   U0
                    D0
                    D10
                    D5
                    M40

MELSEC  LD          M40
        MPS
        ANI         M41
Ausführung, wenn die Bearbeitung der RISEND-Anweisung ohne Fehler abgeschlossen wurde
        MPP
        AND         M41
Ausführung, wenn ein Fehler bei der Bearbeitung der RISEND-Anweisung aufgetreten ist
```

11.5.12 RITO (A-Serie)

CPU

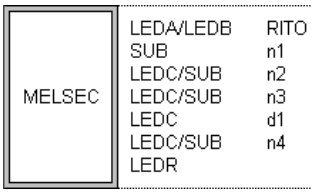
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●		

Operanden MELSEC A

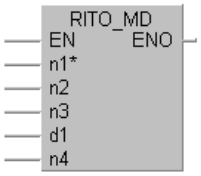
	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)								Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9011
n1																	●	●								
n2								●	●	●	●	●					●	●								
n3								●	●	●	●	●					●	●					29			●
d1								●	●	●	●	●														
n4								●	●	●	●	●					●	●								

GX IEC Developer

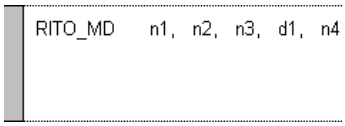
MELSEC-Anweisungsliste



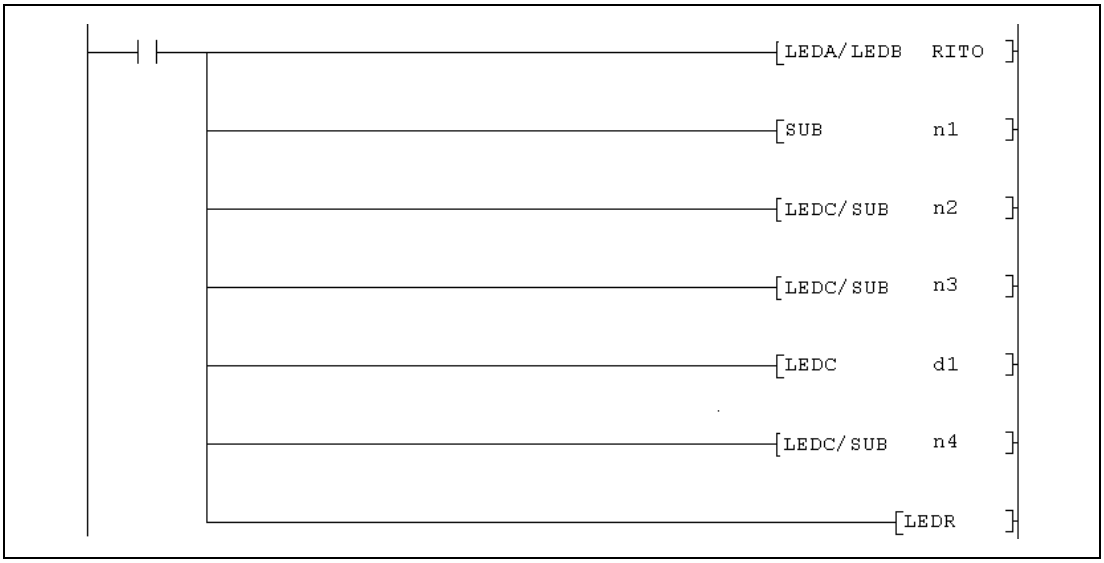
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

Operand	Bedeutung	Werte-bereich	Festlegung des Inhalts durch	Datentyp
n1	Kopfadresse des CC-Link-Master-Moduls auf dem Baugruppenträger (Nur die ersten beiden Stellen der 3-stelligen Adresse werden angegeben. Beispiel: Kopfadresse X/Y100 -> 10_H)	0 bis FE_H	Anwender	BIN-16-Bit
n2	Ziel der Datenübertragung Geben Sie die Stationsnummer der intelligenten Station an, zu der Daten übertragen werden sollen (Nur wenn die RITO-Anweisung in der Master-Station ausgeführt wird.) Geben Sie FF_H an, wenn Daten in den Puffer mit freiem Zugriff übertragen werden sollen.	1 bis 64 oder FF_H		
n3	Anfangsadresse des automatisch aktualisierten Speichers für die intelligente Station in der Master-Station oder Offset-Wert für den Puffer mit freiem Zugriff	Zwischen 0 und dem in den Parametern eingestellten max. Wert.		
d1	Erster Operand des Bereiches, in dem die zu übertragenen Daten gespeichert sind.	Gültige Operanden-adresse		Adresse
n4	Anzahl der Operanden (Worte), die übertragen werden	1 bis 4096		BIN-16-Bit

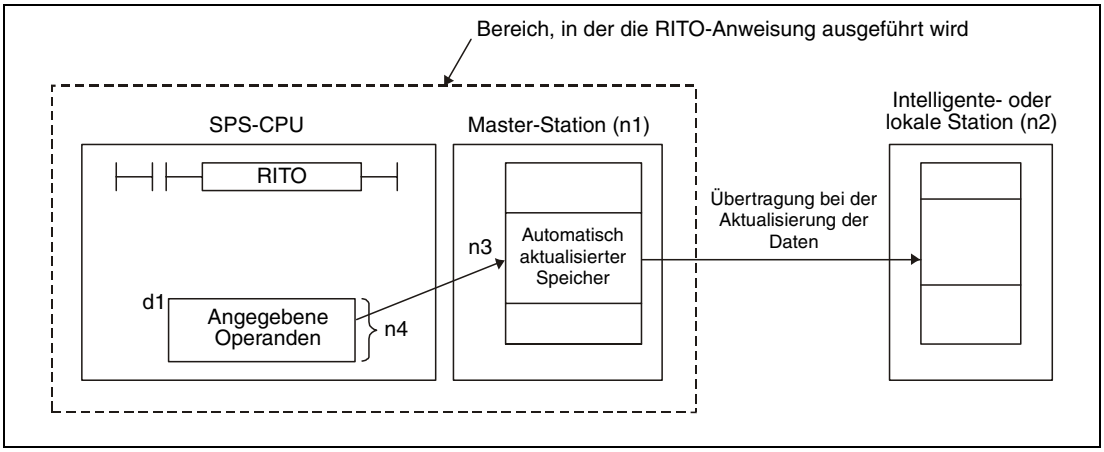
Funktions-weise

Daten in den automatisch aktualisierten Speicher eintragen

RITO Daten schreiben

Mit der RITO-Anweisung werden Daten aus der SPS-CPU in den automatisch aktualisierten Pufferspeicher der Master-Station übertragen. Die Daten werden durch ihre Anfangsadresse (d1) und ihre Länge (n4) spezifiziert. Das Datenziel in der Master-Station wird mit n2 (Nummer der Station, für die die Daten bestimmt sind) und n3 (Anfangsadresse des Speicherbereichs in der Master-Station) angegeben. Die E/A-Kopfadresse der Master-Station ist in n1 angegeben.

Die folgende Abbildung verdeutlicht die Funktion der RITO-Anweisung:



Maximal können mit einer RITO-Anweisung 4096 Worte übertragen werden.

Um die Anzahl der automatisch aktualisierten Adressen einzustellen, muss die Größe des automatisch aktualisierten Puffers über eine RLPA-Anweisung festgelegt werden.

Ausführungsbedingungen

Wird die RITO-Anweisung in Verbindung mit einer LEDA-Anweisung programmiert, wird die RITO-Anweisung ausgeführt, solange die Ausführungsbedingung der LEDA-Anweisung eingeschaltet ist.
Bei Anwendung einer LEDB-Anweisung dagegen werden die Daten nur jeweils bei der steigenden Flanke der Ausführungsbedingung übertragen.

Fehler-quellen

Bei den folgenden Ereignissen wird ein Verarbeitungsfehler erkannt, das Error-Flag M9011 gesetzt und ein Fehlercode ausgegeben.

- Die eingestellte Pufferspeicheradresse liegt außerhalb des zulässigen Bereichs.
  (Fehler-Code in D9008: 50,
  Fehler-Code in D9091 (AnU-CPU) oder D9092 (AnSH-CPU): 503)
- Die Anzahl der aktualisierten Adressen ist größer als 4096.
  (Fehler-Code in D9008: 50,
  Fehler-Code in D9091 (AnU-CPU) oder D9092 (AnSH-CPU): 503)

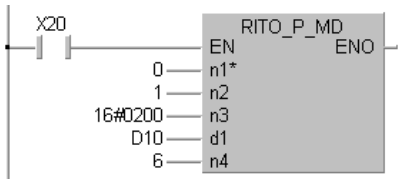
Beispiel RITO

Wenn der Eingang X20 gesetzt wird, überträgt das folgende Programm den Inhalt der sechs Daten-Register D10 bis D15 in den automatisch aktualisierten Speicher der Station mit der Nummer 1. Die Daten werden dort ab der Adresse 200_H eingetragen. Das CC-Link-Modul der Master-Station belegt den Adressbereich von X/Y000 bis X/Y01F.

MELSEC-Anweisungsliste

```
MELSEC  LD    X20
        LEDB  RITO
        SUB   K0
        SUB   K1
        SUB   H0200
        LEDC  D10
        SUB   K6
        LEDR
```

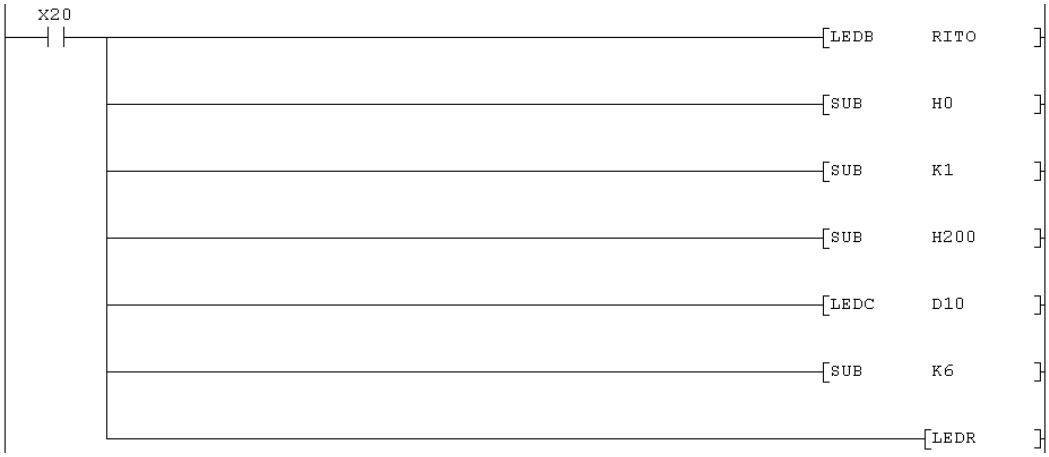
Kontaktplan (GX IEC Developer)



IEC-Anweisungsliste

```
LD           X20
RITO_P_MD    0, 1, 16#0200, D10, 6
```

Kontaktplan (GX Developer)



Hinweise zur Programmierung der erweiterten Anweisungen in den MELSEC-Editoren bei einer SPS der A-Serie finden Sie im Kapitel 3.3 dieses Handbuchs.

11.5.13 RITO (QnA-Serie und System Q)

CPU

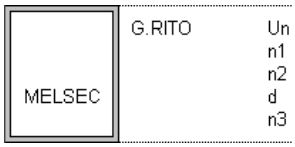
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Operanden MELSEC Q

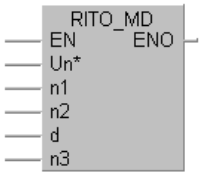
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□N□		Sondermodule U□\G□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
n1	●	●	●	—	—	—	—	●	—	SM0	9
n2	●	●	●	—	—	—	—	●	—		
d	—	●	●	—	—	—	—	—	—		
n3	●	●	●	—	—	—	—	●	—		

GX IEC Developer

MELSEC-Anweisungsliste



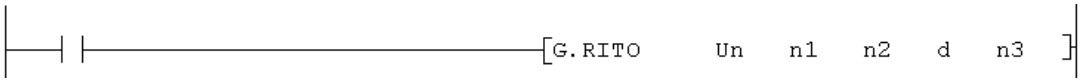
Kontaktplan



IEC-Anweisungsliste

```
RITO_MD    Un, n1, n2, d, n3
```

GX Developer



Variablen

Operand	Bedeutung	Werte-bereich	Festlegung des Inhalts durch	Datentyp
Un	Kopfadresse des CC-Link-Master-Moduls auf dem Baugruppenträger (Nur die ersten beiden Stellen der 3-stelligen Adresse werden angegeben. Beispiel: Kopfadresse X/Y100 -> 10_H)	0 bis FE_H	Anwender	BIN-16-Bit
n1	Ziel der Datenübertragung • Geben Sie die Stationsnummer der intelligenten Station an, zu der Daten übertragen werden sollen. • Geben Sie FF_H an, wenn Daten in den Puffer mit freiem Zugriff übertragen werden sollen.	1 bis 64 oder FF_H		
n2	Offset-Wert des automatisch aktualisierten Speichers für die intelligente Station in der Master-Station oder für den Puffer mit freiem Zugriff Die erste Adresse, die beschrieben wird, wird relativ zur Anfangsadresse des Speicherbereichs angegeben. Beispiel: Um Daten ab der Adresse 356_H des Speichers einzutragen, der bei 350_H beginnt, muss für n2 der Wert 6_H angegeben werden.	Zwischen 0 und dem in den Para-metern ein-gestellten max. Wert.		
d	Erster Operand des Bereiches, in dem die zu übertragenen Daten gespeichert sind.	Gültige Operanden-adresse		Adresse
n3	Anzahl der Operanden (Worte), die übertragen werden	1 bis 4096		BIN-16-Bit

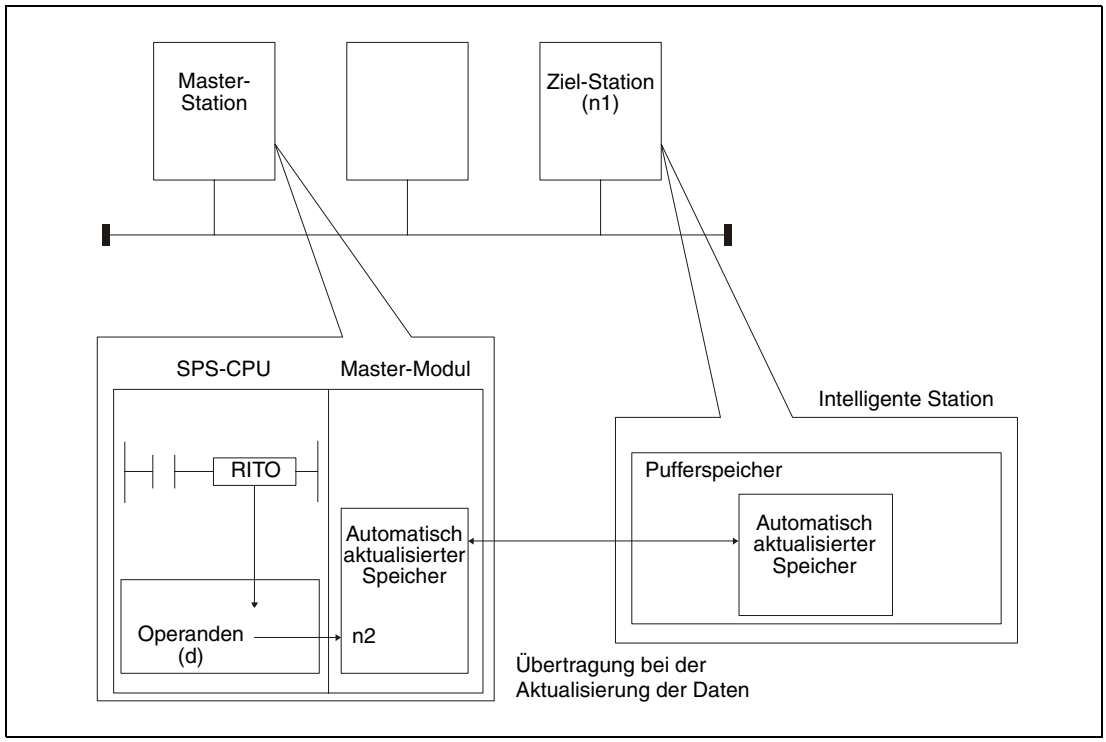
Funktionsweise

Daten in den automatisch aktualisierten Speicher eintragen

RITO Daten schreiben

Mit der RITO-Anweisung werden Daten aus der SPS-CPU in den automatisch aktualisierten Pufferspeicher der Master-Station übertragen. Die Daten werden durch ihre Anfangsadresse (d) und ihre Länge (n3) spezifiziert. Das Datenziel in der Master-Station wird mit n1 (Nummer der Station, für die die Daten bestimmt sind) und n2 (Anfangsadresse des Speicherbereichs in der Master-Station) angegeben. Die E/A-Kopfadresse der Master-Station ist in Un angegeben.

Die folgende Abbildung verdeutlicht die Funktion der RITO-Anweisung:



Auf dieselbe intelligente Station kann nicht gleichzeitig mit mehreren RITO-Anweisungen von verschiedenen Stationen aus zugegriffen werden.

Maximal können mit einer RITO-Anweisung 4096 Worte übertragen werden.

Die Zuweisung der automatisch aktualisierten Speicherbereiche können Sie mit Hilfe der Programmier-Software GX Developer oder GX IEC Developer innerhalb der Netzwerk-Parameter unter dem Punkt „Stationsinformationen“ vornehmen.

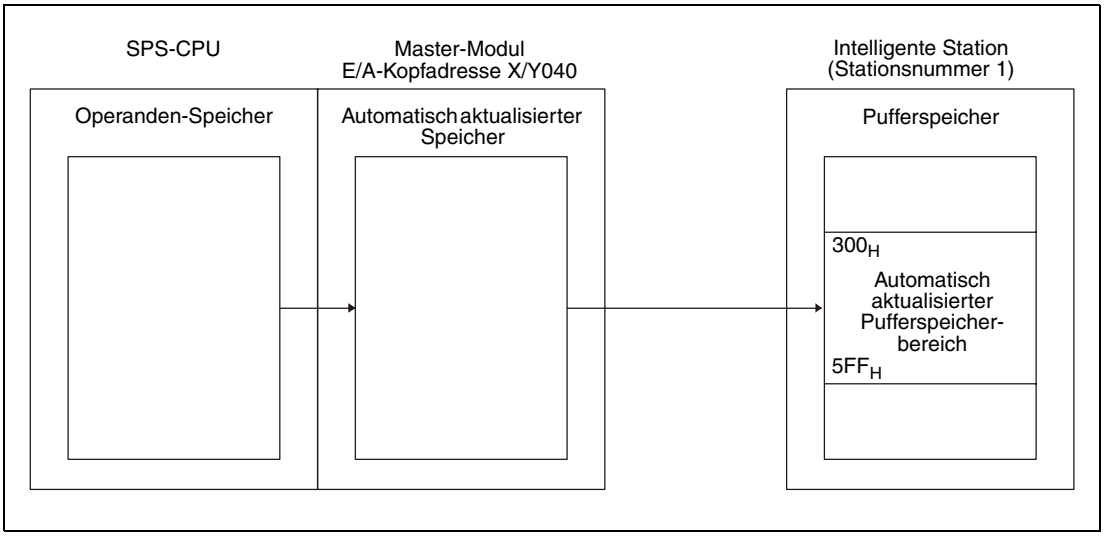
Fehlerquellen

Wenn bei der Ausführung der RITO-Anweisung einer der folgenden Fehler auftritt, wird das Error-Flag SM0 gesetzt und im Sonderregister SD0 ein Fehlercode eingetragen.

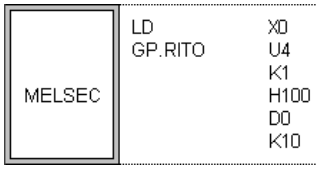
- Das in Un definierte Modul ist kein Sondermodul. (Fehlercode: 2112)
- Die auszuführende Anweisung wird nicht unterstützt. (Fehlercode: 4002)
- Die in der Anweisung angegebene Anzahl der Operanden ist ungültig. (Fehlercode: 4003)
- In der Anweisung ist ein unzulässiger Operand angegeben (Fehlercode: 4004)
- Die in n1 angegebene Stationsnummer existiert nicht. (Fehlercode: 4100)
- Die Anzahl der Daten in n3 überschreitet den zulässigen Bereich. (Fehlercode: 4100)

Beispiel RITO

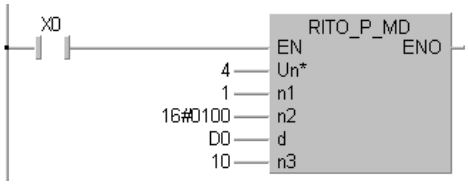
Wenn der Eingang X0 gesetzt wird, überträgt das folgende Programm den Inhalt der 10 Datenregister D0 bis D9 in den automatisch aktualisierten Speicher der Station mit der Nummer 1. Dieser Speicherbereich beginnt bei der Adresse 300_H. Die Daten werden ab der Adresse 400_H (Offset = 100) eingetragen.



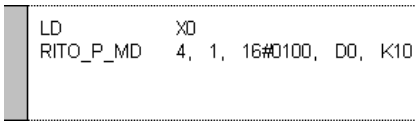
MELSEC-Anweisungsliste



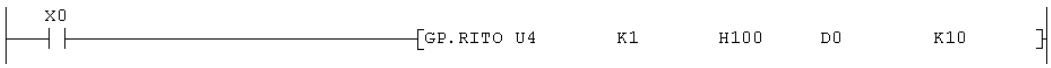
Kontaktplan (GX IEC Developer)



IEC-Anweisungsliste



Kontaktplan (GX Developer)



11.5.14 RIFR (A-Serie)

CPU

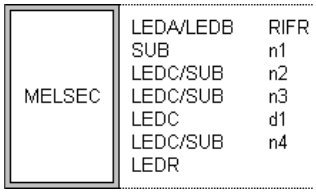
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●		

Operanden MELSEC A

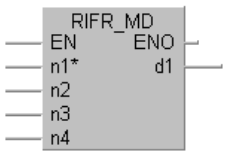
	Operanden																			Blocklänge	Schritte	Index	Carry Flag	Error Flag
	Bit-Operanden							Wortoperanden (16 Bit)								Konstante		Pointer		Ebene					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9011
n1																	●	●								
n2								●	●	●	●	●					●	●								
n3								●	●	●	●	●					●	●					29			●
n4								●	●	●	●	●					●	●								
d1								●	●	●	●	●														

GX IEC Developer

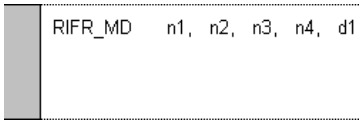
MELSEC-Anweisungsliste



Kontaktplan



IEC-Anweisungsliste



GX Developer

Variablen

Operand	Bedeutung	Wertebereich	Festlegung durch	Datentyp
n1	Kopfadresse des CC-Link-Master-Moduls auf dem Baugruppenträger (Nur die ersten beiden Stellen der 3-stelligen Adresse werden angegeben. Beispiel: Kopfadresse X/Y100 -> 10_H)	0 bis FE_H	Anwender	BIN-16-Bit
n2	Quelle der Daten Geben Sie die Stationsnummer der intelligenten Station an, von der Daten gelesen werden sollen. (Nur wenn die RIFR-Anweisung in der Master-Station ausgeführt wird.) Geben Sie FF_H an, wenn Daten aus den Puffer mit freiem Zugriff gelesen werden sollen.	1 bis 64 oder FF_H		
n3	Anfangsadresse des automatisch aktualisierten Speichers für die intelligente Station in der Master-Station oder Offset-Wert für den Puffer mit freiem Zugriff	Zwischen 0 und dem in den Parametern eingestellten max. Wert.		
n4	Anzahl der Operanden (Worte), die übertragen werden sollen	1 bis 4096		
d1	Erster Operand des Bereiches, in dem die zgelesenen Daten eingetragen werden.	Gültige Operandenadresse		Adresse

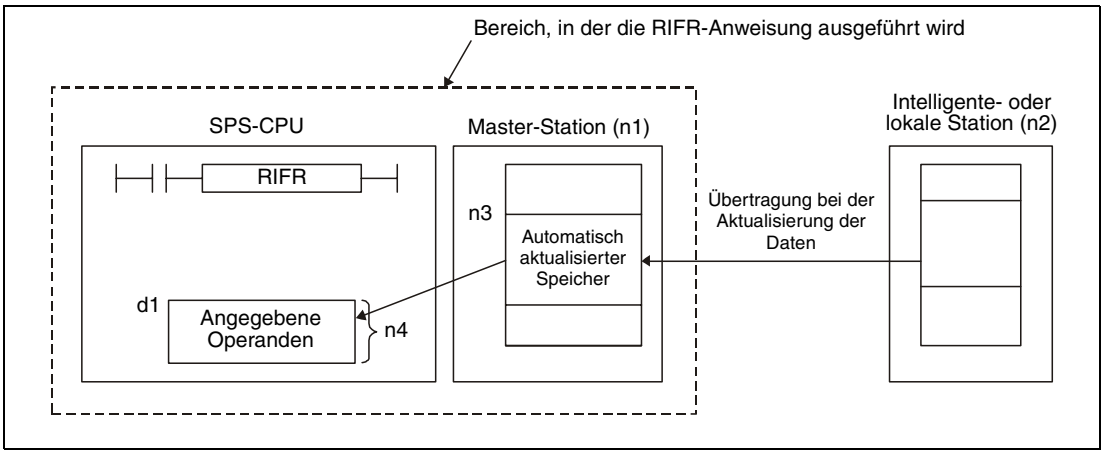
Funktions-weise

Daten aus dem automatisch aktualisierten Speicher lesen

RIFR Daten lesen

Mit der RIFR-Anweisung werden Daten aus dem automatisch aktualisierten Pufferspeicher der Master-Station in den Operandenspeicher der SPS-CPU übertragen. Wo die Daten dort gespeichert werden und welche Anzahl übertragen werden soll, wird durch d1und n4 festgelegt. Mit n2 wird die Nummer der Station, aus der Daten gelesen werden, angegeben und n3 enthält die erste Adresse, deren Inhalt gelesen wird. Die E/A-Kopfadresse der Master-Station wird durch n1 bestimmt.

Die folgende Abbildung verdeutlicht die Funktion der RIFR-Anweisung:



Maximal können mit einer RIFR-Anweisung 4096 Worte übertragen werden.

Um die Anzahl der automatisch aktualisierten Adressen einzustellen, muss die Größe des automatisch aktualisierten Puffers über eine RLPA-Anweisung festgelegt werden.

Ausführungsbedingungen

Wird die RIFR-Anweisung in Verbindung mit einer LEDA-Anweisung programmiert, wird die RIFR-Anweisung ausgeführt, solange die Ausführungsbedingung der LEDA-Anweisung eingeschaltet ist.
Bei Anwendung einer LEDB-Anweisung dagegen werden die Daten nur jeweils bei der steigenden Flanke der Ausführungsbedingung übertragen.

Fehler-quellen

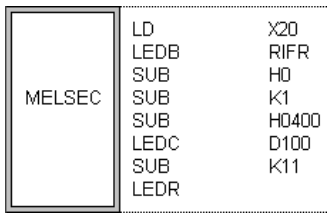
Bei den folgenden Ereignissen wird ein Verarbeitungsfehler erkannt, das Error-Flag M9011 gesetzt und ein Fehlercode ausgegeben.

- Die eingestellte Pufferspeicheradresse liegt außerhalb des zulässigen Bereichs.
  (Fehler-Code in D9008: 50,
  Fehler-Code in D9091 (AnU-CPU) oder D9092 (AnSH-CPU): 503)
- Die Anzahl der aktualisierten Adressen ist größer als 4096.
  (Fehler-Code in D9008: 50,
  Fehler-Code in D9091 (AnU-CPU) oder D9092 (AnSH-CPU): 503)

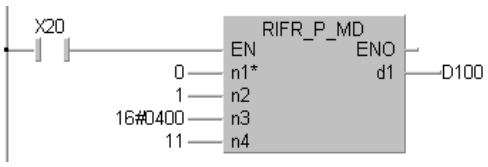
Beispiel RIFR

Wenn der Eingang X20 gesetzt wird, überträgt das folgende Programm den Inhalt von elf Adressen aus dem automatisch aktualisierten Puffer der Station 1 in die SPS-CPU und speichert die Daten dort ab dem Register D100. Im automatisch aktualisierten Puffer sind die Daten ab der Adresse 400_H gespeichert. Das CC-Link-Modul der Master-Station belegt den Adressbereich von X/Y000 bis X/Y01F.

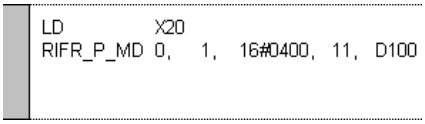
MELSEC-Anweisungsliste



Kontaktplan (GX IEC Developer)



IEC-Anweisungsliste



Kontaktplan (GX Developer)

Hinweise zur Programmierung der erweiterten Anweisungen in den MELSEC-Editoren bei einer SPS der A-Serie finden Sie im Kapitel 3.3 dieses Handbuchs.

11.5.15 RIFR (QnA-Serie und System Q)

CPU

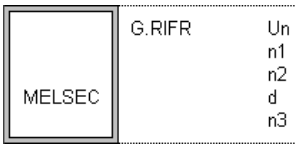
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Operanden MELSEC Q

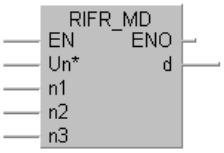
	Operanden									Error Flag	Schritte
	Interne Operanden (System, Anwender)		File-Register	MELSECNET/10 Direkt J□N□		Sondermodule U□NG□	Index-Register Zn	Konstanten K, H (16#)	Andere		
	Bit	Wort		Bit	Wort						
n1	●	●	●	—	—	—	—	●	—	SM0	9
n2	●	●	●	—	—	—	—	●	—		
n3	●	●	●	—	—	—	—	●	—		
d	—	●	●	—	—	—	—	—	—		

GX IEC Developer

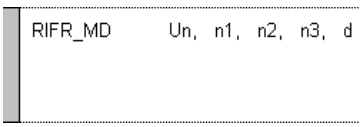
MELSEC-Anweisungsliste



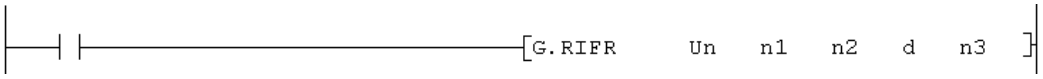
Kontaktplan



IEC-Anweisungsliste



GX Developer



Variablen

Operand	Bedeutung	Werte-bereich	Festlegung durch	Datentyp
Un	Kopfadresse des CC-Link-Master-Moduls auf dem Baugruppenträger (Nur die ersten beiden Stellen der 3-stelligen Adresse werden angegeben. Beispiel: Kopfadresse X/Y100 -> 10_H)	0 bis FE_H	Anwender	BIN-16-Bit
n1	Datenquelle • Geben Sie die Stationsnummer der intelligenten Station an, deren Daten gelesen werden sollen. • Geben Sie FF_H an, wenn Daten aus dem Puffer mit freiem Zugriff übertragen werden sollen.	1 bis 64 oder FF_H		
n2	Offset-Wert des automatisch aktualisierten Speichers für die intelligente Station in der Master-Station oder für den Puffer mit freiem Zugriff Die erste Adresse, die gelesen wird, wird relativ zur Anfangsadresse des Speicherbereichs angegeben. Beispiel: Um Daten ab der Adresse 356_H des Speichers zu lesen, der bei 350_H beginnt, muss für n2 der Wert 6_H angegeben werden.	Zwischen 0 und dem in den Para-metern ein-gestellten max. Wert.		
n3	Anzahl der Operanden (Worte), die übertragen werden	1 bis 4096		
d	Erster Operand des Bereiches, in dem die zu gelesenen Daten gespeichert werden.	Gültige Operanden-adresse		Adresse

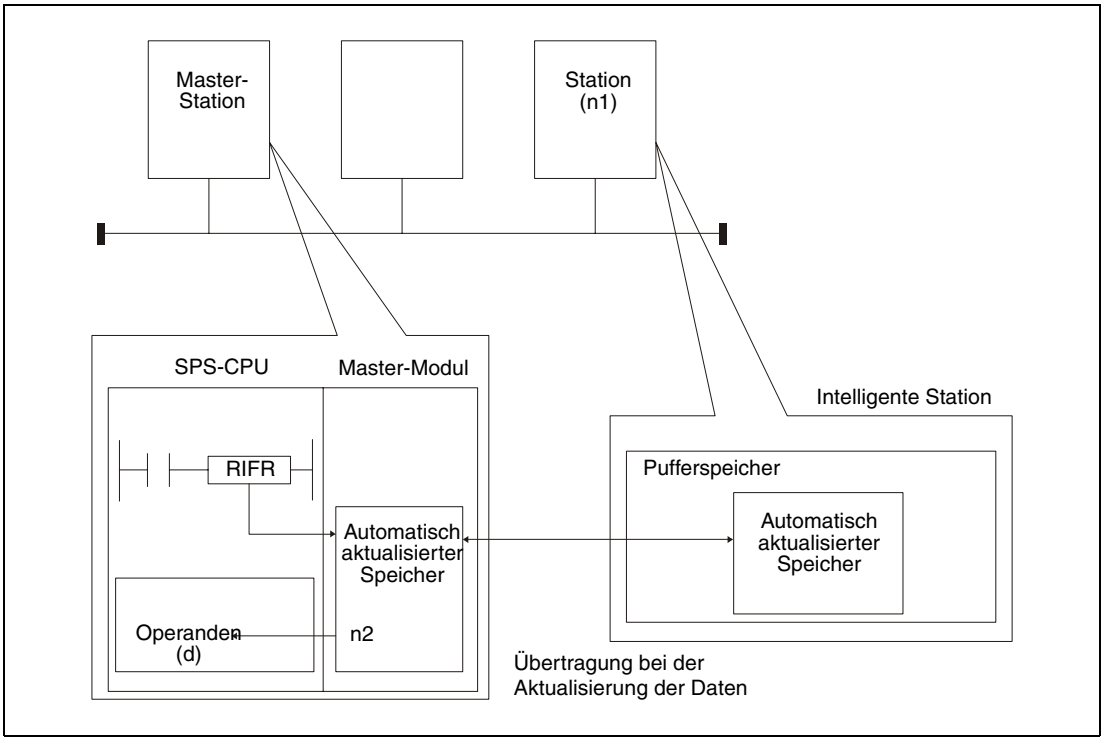
Funktions-weise

Daten aus dem automatisch aktualisierten Speicher lesen

RIFR Daten lesen

Mit der RIFR-Anweisung werden Daten aus dem automatisch aktualisierten Pufferspeicher der Master-Station in den Operandenspeicher der SPS-CPU übertragen. Wo die Daten dort gespeichert werden und welche Anzahl übertragen werden soll, wird durch d und n3 festgelegt. Mit n1 wird die Nummer der Station, aus der Daten gelesen werden, angegeben und n2 enthält die erste Adresse (als Offset-Wert), die gelesen wird. Die E/A-Kopfadresse der Master-Station wird durch Un bestimmt.

Die folgende Abbildung verdeutlicht die Funktion der RIFR-Anweisung:



Auf dieselbe intelligente Station kann nicht gleichzeitig mit mehreren RIFR-Anweisungen von verschiedenen Stationen aus zugegriffen werden.

Maximal können mit einer RIFR-Anweisung 4096 Worte übertragen werden.

Die Zuweisung der automatisch aktualisierten Speicherbereiche können Sie mit Hilfe der Programmier-Software GX Developer oder GX IEC Developer innerhalb der Netzwerk-Parameter unter dem Punkt „Stationsinformationen“ vornehmen.

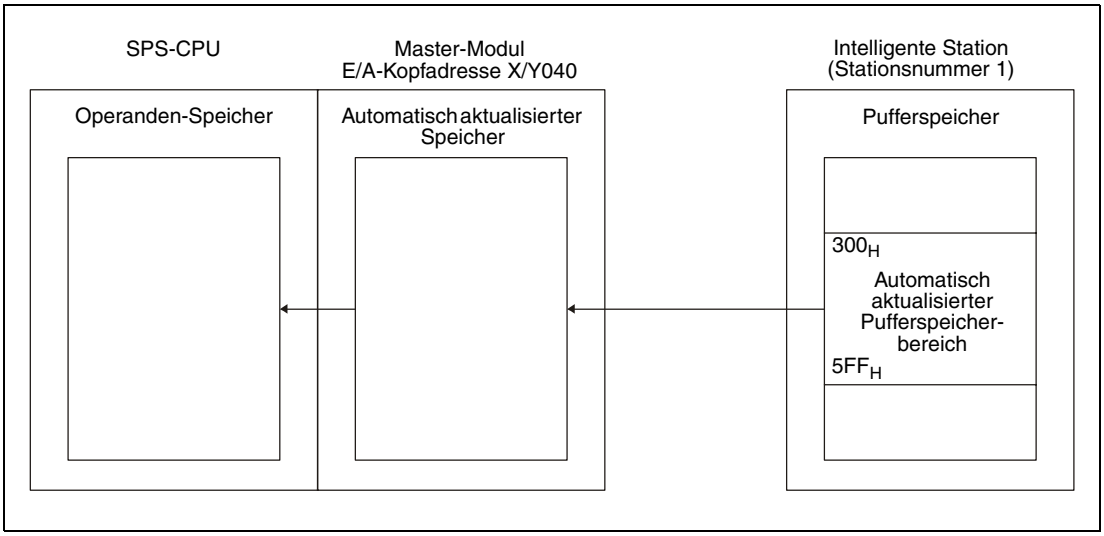
Fehler-quellen

Wenn bei der Ausführung der RITO-Anweisung einer der folgenden Fehler auftritt, wird das Error-Flag SM0 gesetzt und im Sonderregister SD0 ein Fehlercode eingetragen.

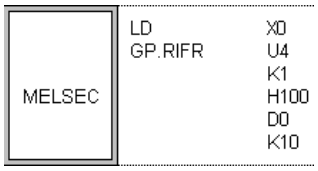
- Das in Un definierte Modul ist kein Sondermodul. (Fehlercode: 2112)
- Die auszuführende Anweisung wird nicht unterstützt. (Fehlercode: 4002)
- Die in der Anweisung angegebene Anzahl der Operanden ist ungültig. (Fehlercode: 4003)
- In der Anweisung ist ein unzulässiger Operand angegeben (Fehlercode: 4004)
- Die in n1 angegebene Stationsnummer existiert nicht. (Fehlercode: 4100)
- Die Anzahl der Daten in n3 überschreitet den zulässigen Bereich. (Fehlercode: 4100)

Beispiel RIFR

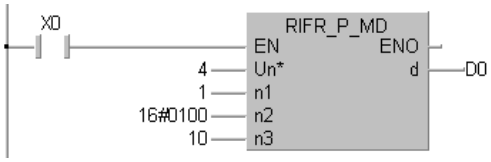
Wenn der Eingang X0 gesetzt wird, liest das folgende Programm 10 Worte aus dem automatisch aktualisierten Speicher der Station mit der Nummer 1 und speichert die Daten in der CPU ab D0. Der automatisch aktualisierte Speicherbereich beginnt bei der Adresse 300_H. Gelesen werden 10 Worte ab der Adresse 400_H (Offset = 100).



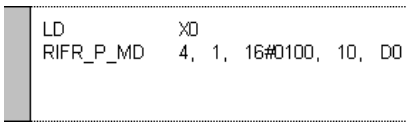
MELSEC-Anweisungsliste



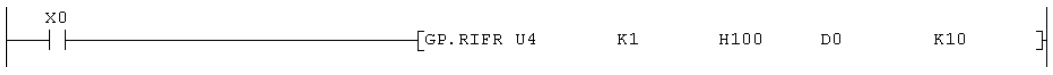
Kontaktplan (GX IEC Developer)



IEC-Anweisungsliste



Kontaktplan (GX Developer)



12 Mikrocomputer-Programm (AnN(S))

Die MELSEC A-Serie (ausgenommen AnA, AnAS und AnU) erlaubt eine kombinierte Verarbeitung von Ablaufprogramm und Mikrocomputer-Programm. Das Mikrocomputer-Programm gestattet es, Programmsequenzen außerhalb der Makroebene (Haupt- und Unterprogramm) auszuführen. Der Aufruf eines Mikrocomputer-Programms erfolgt mittels einer SUB(P)-Anweisung.

CPUs der MELSEC AnA-, AnAS-, AnUS-, QnA- und QnAS-Serie sowie CPUs des System Q können Mikrocomputer-Programme nicht verarbeiten.

12.1 Kapazitäten und Speicherbereiche

Die nachfolgende Tabelle enthält eine Übersicht der Kapazitäten und Speicherbereiche der Mikrocomputer-Programme bezogen auf die unterschiedlichen CPU-Typen.

CPU	Prozessor	Mikrocomputer-Programmbereich	Arbeitsbereich	Stapelbereich	Nicht anwendbare Anweisungen
A1	8086 (8 MHz)	0 – 10 kByte	A100H – A1FFH (256 Byte)	Nutzbarer Bereich: 128 Byte	INT, INTO, IRET, IN, OUT, HLT, WAIT, LOCK, ESC
A2		0 – 26 kByte			
A3		0 – 58 kByte (MAIN) 0 – 58 kByte (SUB)			
A1N	8086 (10 MHz)	0 – 10 kByte			
A2N-S1		0 – 26 kByte			
A2S		0 – 26 kByte			
A3N		0 – 58 kByte (MAIN) 0 – 58 kByte (SUB)			
A1S	8086 (8 MHz)	0 – 14 kByte			
A1S-S1		0 – 14 kByte			
A2C		0 – 26 kByte			
A3H	80286 (8 MHz)	0 – 58 kByte (MAIN) 0 – 58 kByte (SUB)			INT, INTO, IRET, IN, OUT, HLT, WAIT, LOCK, ESC, CLI, STI
A3M		0 – 58 kByte (MAIN) 0 – 58 kByte (SUB)			

HINWEISE Die Festlegung des Speicherbereiches für das Mikrocomputer-Programm erfolgt in Schritten von 2 kByte. Das Verhältnis zwischen den einzelnen Programmteilen muss so gewählt werden, dass Mikrocomputer- und Ablaufprogramm im MAIN- und SUB-Bereich nicht den gleichen Speicherplatz belegen.

In Verbindung mit einem Mikrocomputer-Programm dürfen ausschließlich die dafür vorgesehenen Anweisungen verwendet werden. Die Verwendung anderer Anweisungen führt bei Ausführung des Mikrocomputer-Programms zu einer Fehlfunktion der CPU.

12.2 Anwendung selbsterstellter Mikrocomputer-Programme

Das vom Anwender in 8086-Maschinensprache erstellte Quellprogramm muss mit Hilfe von Assemblern unter CP/M® oder MS-DOS® in eine für die SPS verständliche Maschinensprache kompiliert (übersetzt) werden. Das kompilierte Programm wird als "Objektprogramm" bezeichnet und im Mikrocomputer-Programmbereich der CPU gespeichert. Das OBJ-File wird vom C-Compiler mit dem Programmiergerät an die SPS übertragen.

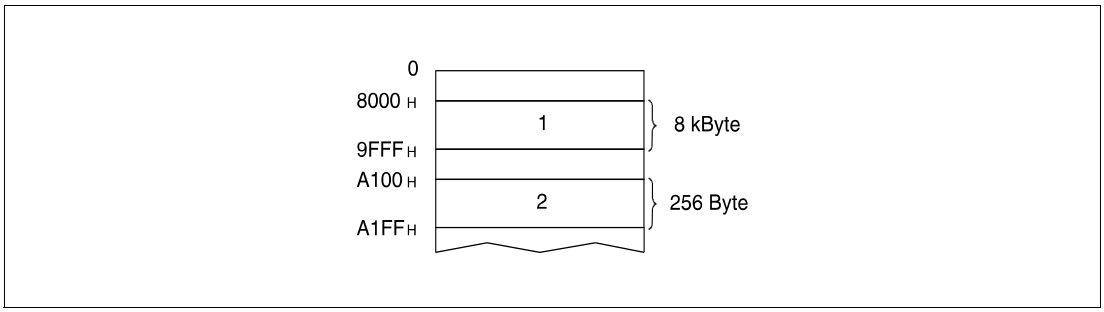
HINWEIS *„Bitte überprüfen Sie in Ihrer Software-Version des GX IEC Developers*, ob *diese Funktionalität unterstützt wird“.*

Wichtige Hinweise zur Erstellung eines Mikrocomputer-Programms

- Zu Beginn des Mikrocomputer-Programms muss die PUSH-Anweisung gesetzt werden, damit der Inhalt der verwendeten Datenregister in den Stapelspeicher übertragen wird. Am Ende des Programms muss eine POP-Anweisung stehen, damit die im Stapelspeicher gesicherten Daten der Datenregister zurückgespeichert werden.
- Alle in Verbindung mit dem Mikrocomputer-Programm genutzten Datenregister müssen zu Beginn der Programmausführung initialisiert, d. h. auf ihre Ausgangswerte zurückgesetzt werden. Der Inhalt der Datenregister nach Aufruf des Mikrocomputer-Programms aus dem Ablaufprogramm ist nicht definiert.
- Zur Ausführung eines Mikrocomputer-Programms muss dieses vom Ablaufprogramm aus über eine SUB(P)-Anweisung aufgerufen werden. Das Ablaufprogramm ist daher immer erforderlich.
- Die Rückkehr vom Mikrocomputer-Programm zum Ablaufprogramm erfolgt über eine RETF-Anweisung.

12.2.1 Speicheraufteilung

Das Mikrocomputer-Programm wird in zwei verschiedenen Speicherbereichen der CPU abgelegt. Der zwischen 8000H und 9FFFH liegende 8 kByte große Bereich wird zur Datenspeicherung und der Bereich zwischen A100H und A1FFH als Arbeitsbereich des Mikrocomputer-Programms genutzt.

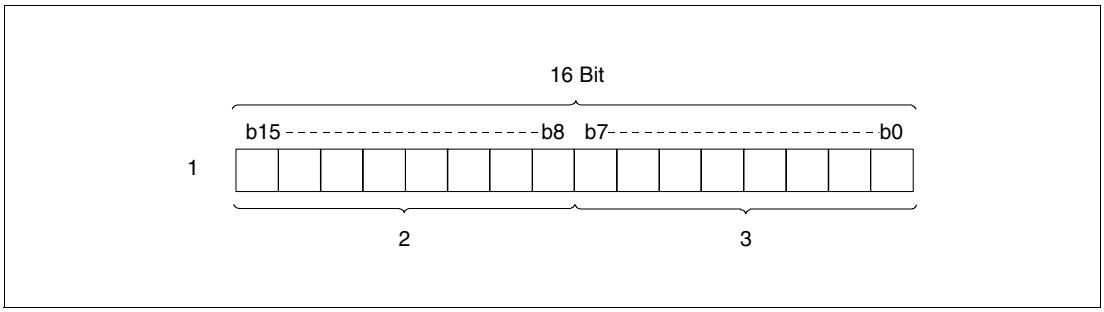


1 Datenspeicherbereich

2 Arbeitsbereich für Mikrocomputer-Programm

12.2.2 Adressenzuordnung des Datenspeicherbereiches

Eine Adresse des Datenspeicherbereichs besteht aus 16 Bit und wird in einen geraden und ungeraden 8 Bit großen Bereich aufgeteilt. Die folgende Abbildung zeigt die schematische Aufteilung einer Adresse.



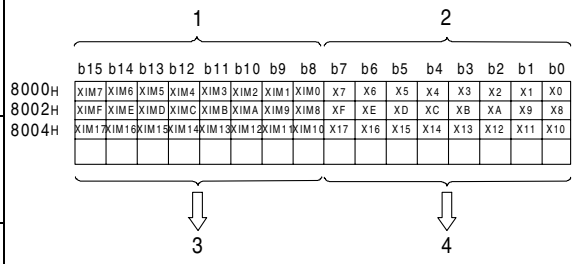
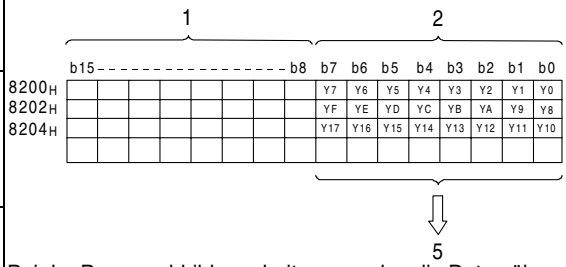
1 Erste Adresse 8000H

2 Ungerader 8-Bit-Bereich (8001H)

3 Gerader 8-Bit-Bereich (8000H)

12.2.3 Gliederung des Speicherbereiches

Der Speicherbereich zwischen 8000H und 9FFFH wird von der CPU zur Speicherung der Operandendaten genutzt. Eine Aufgliederung dieses Speicherbereiches nach Operandenadressen enthalten die folgenden Tabellen.

Operand	CPU-Typ	Adressen	Konfiguration
Eingang (X)	A1 A1N	8000H – 803FH: X0 - FF	
	A2 A2C A2N A2S	8000H – 803FH: X0 - 1FF	
	A2N-S1	8000H – 80FFH: X0 - 3FF	
	A3 A3N	8000H – 81FFH: X0 - 7FF	
Ausgang (Y)	A1 A1N	8200H – 823FH: Y0 - FF	 Bei der Prozessabbildverarbeitung werden die Daten über den Ausgangsspeicher in/aus dem Ausgangsmodul geschrieben oder gelesen. Bei der Direktverarbeitung erfolgt das Schreiben direkt zum Ausgangsmodul. Gelesen werden die Daten von dem Ausgangsmodul über den Ausgangsspeicher.
	A2 A2C A2N A2S	8000H – 827FH: Y0 -1FF	
	A2N-S1	8200H – 827FH: X0 - 3FF	
	A3 A3N	8200H – 83FFH: Y0 - 7FF	

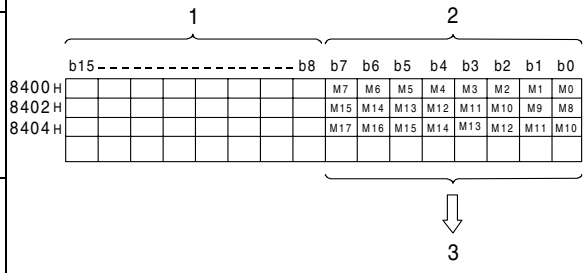
[1] Ungerade Adressen

[2] Gerade Adressen

[3] Bereich zur Speicherung der Datenzustände einer Remote-Station (Schreiben und Lesen), 0 = AUS, 1 = EIN. Die aktuelle Eingabe Eingang (X)=(XIM) $\vee$ ($\overline{X}$)

[4] Bereich zur Speicherung der Datenzustände eines Eingangsmoduls (nur Lesen), 0 = AUS, 1 = EIN.

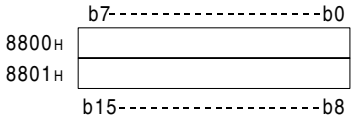
[5] Bereich zur Speicherung der Verarbeitungsergebnisse der SPS (Lesen und Schreiben).

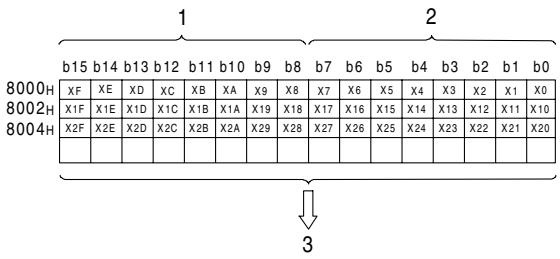
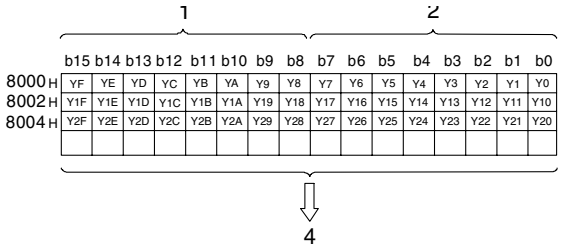
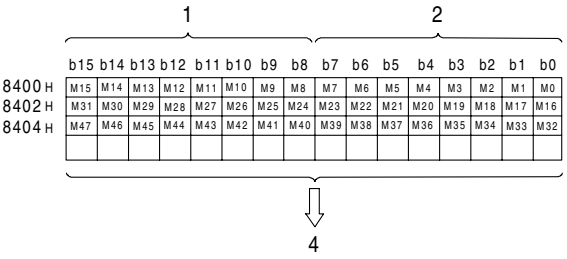
Operand	CPU-Typ	Adressen		Konfiguration
Merker (M) Latch-Merker (L) Schritt-merker (S)	A1 A2 A3 A1N A2N-S1 A3N A1S A1S-S1 A2C	8400H – 85FFH	M/L/S 0 - 2047	
Link-Merker (B)		8600H – 86FFH	F0 - 255	
Fehler-merker (F)		8700H – 873FH	F0 - 255	
Sonder-merker (M)		8740H – 877FH	M 9000-9255	
Timer-Eingangs-kontakt (T)		8780H – 87BFH	T0 - 255	
Counter-Eingangs-kontakt (C)		87C0H – 87FFH	C0 - 255	
Timer-Ausgangs-kontakt (T)		9C00H – 9C3FH	T0 - 255	
Counter-Ausgangs-kontakt (C)		9C40H – 9C7FH	C0 - 255	

[1] Ungerader Bereich

[2] Gerader Bereich

[3] Bereich zur Speicherung der Verarbeitungsergebnisse der SPS (Lesen und Schreiben)

Operand	CPU-Typ	Adressen	Konfiguration
Daten-Register (D)	A1 A2 A3 A1N A2N-S1 A3N A1S1 A1S-S1 A2C	8800H – 8FFFH: D0 - 1023	
Link-Register (W)		9000H – 97FFH: W0 - 3FF	
Istwert der Timer (T)		9800H – 99FFH: T0 - 255	
Istwert der Counter (C)		9A00H – 9BFFH: C0 - 255	
Sonder-register (D)		9D00H – 9EFFH: D9000-9255	
Akku-mulator (A0, A1)		9FF8H – 9FFAH: A0 / A1	
Index-Register (Z,V)		9FFCH – 9FFEH: Z / V	

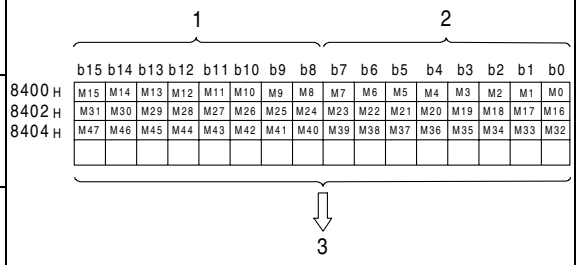
Operand	CPU-Typ	Adressen	Konfiguration
Eingang (X)	A3H A3M	8000H – 80FFH X0 - 7FF	
Ausgang (Y)		8200H – 82FFH Y0 - 7FF	 Bei der Prozessabbildverarbeitung werden die Daten über den Ausgangsspeicher in/aus dem Ausgangsmodul geschrieben oder gelesen. Bei der Direktverarbeitung erfolgt das Schreiben direkt zum Ausgangsmodul. Gelesen werden die Daten von dem Ausgangsmodul über den Ausgangsspeicher.
Merker (M) Latch-Merker (L) Schritt-merker (S)		8400H – 84FFH M/L/S 0 - 2047	
Link-Register (B)		8600H – 867FH B0 - 3FF	
Fehler-merker (F)		8700H – 871FH F0 - 255	

[1] Ungerade Adressen

[2] Gerade Adressen

[3] Bereich zur Speicherung der Datenzustände eines Eingangsmoduls (nur Lesen)

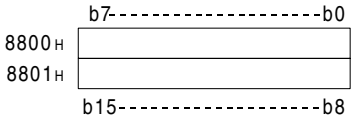
[4] Bereich zur Speicherung der Verarbeitungsergebnisse der SPS (Lesen und Schreiben)

Operand	CPU-Typ	Adressen		Konfiguration
Sondermerker (M)	A3H A3M	8740H – 875FH	M 9000-9255	
Timer-Eingangskontakt (T)		8780H – 879FH	T0 - 255	
Counter-Eingangskontakt (C)		87C0H – 87DFH	C0 - 255	
Timer-Ausgangskontakt (T)		9C00H – 9C1FH	T0 - 255	
Counter-Ausgangskontakt (C)		9C40H – 9C5FH	C0 - 255	

[1] Ungerade Adressen

[2] Gerade Adressen

[3] Bereich zur Speicherung der Verarbeitungsergebnisse der SPS (Lesen und Schreiben)

Operand	CPU-Typ	Adressen		Konfiguration
Datenregister (D)	A3H A3M	8800H – 8FFFH	D0 - 1023	
Linkregister (B)		9000H – 97FFH	W0 - 3FF	
Istwert der Timer (T)		9800H – 99FFH	T0 - 255	
Istwert der Counter (C)		9A00H – 9BFFH	C0 - 255	
Sonderregister (D)		9D00H – 9EFFH	D9000-9255	
Akkumulator (A0, A1)		9FF8H – 9FFAH	A0 / A1	
Indexregister (Z,V)		9FFCH – 9FFEH	Z / V	

Operand	CPU-Typ	Adressen
File Register (R) Block-Nr. 0	A2 A3 A2N A2N-S1 A2S A3N A3H A3M A1S A1S-S1 A2C	Startadresse der File-Register = 20000H + (Speicherkapazität der RAM-Kassette) - (Kapazität der File-Register) Speicherkapazität der RAM-Kassetten (kalkulatorischer Wert) A3(N)MCA-0 = 16 kByte A3(N)MCA-2 = 16 kByte A3(N)MCA-4 = 32 kByte A3(N)MCA-8 = 64 kByte A3MCA-12 = 96 kByte A3MCA-16 = 144 kByte (entspr. 128k verfügbarer Kapazität) A3MCA-18 = 144 kByte A3MCA-24 = 144 kByte (entspr. 192 k verfügbarer Kapazität) A3NMCA-40 = 144 kByte (entspr. 320 k verfügbarer Kapazität) A3NMCA-56 =144 kByte (entspr. 448 k verfügbarer Kapazität) kalkulatorischer Wert Kommentarkapazität: (Anzahl der Kommentare) x 16 Byte + 1 kByte Kapazität der File-Register: (Anzahl der File-Register) x 2 Byte Anmerkung: Bei Berechnung der Kapazitäten ist zu berücksichtigen, dass 1 kByte nicht 1000, sondern 1024 Byte entspricht !
Erweiterte-File-Register (R) Block-Nr. 1 – 9		Startadresse der File-Register für jede Blocknummer = 20000H + (Speicherkapazität der RAM-Kassette) - (Kommentarkapazität) -(Kapazität der File-Register) - (Kapazität des Status Latch) -(Kapazität des Sampling Trace) - 4000H x n Kommentarkapazität: (Anzahl der Kommentare) x 16 Byte + 1KByte Kapazität der File-Register: (Anzahl der File-Register) x 2 Byte Kapazität des Status Latch: Summe der gesetzten Bytes Kapazität des Sampling Trace: 8 kByte (soweit möglich) n: Blocknummer

Operand	CPU-Typ	Adressen
Erweiterte File-Register (R) Block-Nr. 10 - 28	A2 A3 A2N A2N-S1 A2S A3N A3H A3M A1S A1S-S1 A2C	Speicherkassetten

1

3	4
11	38000 H
10	3C000 H

2

3	4
28	A0000 H
27	A4000 H
26	A8000 H
25	AC000 H
24	B0000 H
23	B4000 H
22	B8000 H
21	BC000 H
20	C0000 H
19	C4000 H
18	C8000 H
17	CC000 H
16	D0000 H
15	D4000 H
14	D8000 H
13	DC000 H
12	E4000 H
11	E8000 H
10	EC000 H

[1] A3NMCA-16

[2] A3NMCA-24, 40 oder 56

[3] Blocknr.

[4] Startadresse

13 Fehlercodes

Tritt während der Inbetriebnahme, beim Setzen in den RUN-Modus, oder während des Betriebs der SPS ein Fehler auf, zeigt die Selbstdiagnose-Funktion der CPU einen Fehler an (LED-Anzeige zeigt Fehler) und speichert die Fehlerinformation in Sondermerkern (M) oder Diagnosemerkern (SM) und dem Sonderregister (D9008) oder den Diagnoseregistern (SD).

13.1 Liste der Fehlercodes (Q00J-, Q00- und Q01CPU)

Die folgende Tabelle enthält eine Übersicht der möglichen Fehler zusammen mit Fehlermeldungen, möglichen Ursachen und Hinweisen zur Behebung des Fehlers. In dieser Tabelle sind nur Fehlermeldungen der Q00J-, Q00- und Q01CPUs aufgeführt.

Fehlercode (SD0) [1]	Fehlermeldung	Allgemeine Information (SD5 bis 15) [1]	Spezifische Information (SD16 bis 26) [1]	LED-Status		CPU-Status	Diagnosezeitraum
				RUN	ERROR		
1000	MAIN CPU DOWN	—	—	AUS	Blinkt/EIN	Stopp	Kontinuierlich
1010	END NOT EXECUTE	—	—	AUS	Blinkt	Stopp	Während der Ausführung der END-Anweisung
1011							
1012							
1101	RAM ERROR	—	—	AUS	Blinkt	Stopp	Beim Einschalten/Beim Zurücksetzen
1102							
1103							
1104							
1200	OPE. CIRCUIT ERR.	—	—	AUS	Blinkt	Stopp	Beim Einschalten/Beim Zurücksetzen
1201							
1202							
1300	FUSE BREAK OFF	Stationsnr./Modulnr.	—	AUS/EIN	Blinkt/EIN	Stopp/ Fortsetzen [2]	Während der Ausführung der END-Anweisung.
1310	I/O INT ERROR	Stationsnr./Modulnr.	—	AUS	Blinkt	Stopp	Während eines Interruptes
1401	SP. UNIT DOWN	Stationsnr./Modulnr.	—	AUS	Blinkt	Stopp/ Fortsetzen [3]	Beim Einschalten/Beim Zurücksetzen Beim Zugriff auf ein Sondermodul
1402			Lokalisierung des Programmfehlers				Beim Zugriff auf ein Sondermodul
1403			—				Während der Ausführung der END-Anweisung

[1] Die Angaben in den runden Klammern kennzeichnen die Adressen der Sonderregister, in denen die einzelnen Fehlerinformationen gespeichert werden.

[2] Der Verarbeitungszustand der CPU beim Auftreten eines Fehlers kann in den Parametern gesetzt werden. Die LED-Anzeige ändert sich entsprechend.

[3] Für jedes Modul kann in den Parametern eingestellt werden, ob die CPU bei einem Fehler des Moduls gestoppt werden soll.

	Ursache	Abhilfe
	Abschaltung des RUN-Modus oder Fehler in der CPU 1.) Funktionsstörung aufgrund von Störspannungen (Rauschen) oder aus anderen Gründen 2.) Hardware-Fehler	1.) Messen Sie den Rauschpegel. 2.) Die CPU zurücksetzen und in den RUN-Modus schalten. Wenn der gleiche Fehler wieder angezeigt wird, deutet dieses auf einen Hardware-Fehler hin. Wenden Sie sich an den MITSUBISHI-Service.
	Das gesamte Programm wurde ohne Ausführung der END-Anweisung ausgeführt. 1.) Wenn die END-Anweisung ausgeführt wird, wird sie als ein anderer Anweisungscode gelesen. 2.) Die END-Anweisung wurde in einen anderen Anweisungscode geändert.	
	Fehler im internen RAM, in dem das CPU-Ablaufprogramm gespeichert ist.	Dies weist auf einen CPU-Hardware-Fehler hin. Wenden Sie sich an den MITSUBISHI-Service.
	Fehler im RAM, das als CPU-Arbeitsbereich benutzt wird.	
	Interner CPU-Fehler	
	Fehler bei der RAM-Adressierung in der CPU.	
	Der Schaltkreis, der für die Index-Verarbeitung in der CPU verantwortlich ist, arbeitet fehlerhaft.	
	Die CPU-Hardware (Logik) arbeitet fehlerhaft.	
	Der Schaltkreis, der für die Ablaufverarbeitung verantwortlich ist, arbeitet fehlerhaft.	
	Die Sicherung eines Ausgangsmoduls ist defekt.	1.) Überprüfen Sie die LED-Anzeigen der Sicherungen an den Ausgangsmodulen und wechseln Sie die Sicherung, deren LED leuchtet. 2.) Lesen Sie die allgemeinen Fehlerinformationen auf dem Display des Programmiergerätes, und wechseln Sie die Sicherung des Ausgangsmoduls, das angezeigt wird. Alternativ dazu überwachen Sie die Sonderregister SD130 bis SD137 auf dem Display des Programmiergerätes, und wechseln Sie die Sicherung des Ausgangsmoduls, bei dem das entsprechende Bit auf „1“ gesetzt ist.
	Ein Interrupt wurde ausgeführt, obwohl sich im System kein Interrupt-Modul befindet.	Eines der angeschlossenen Module weist einen Hardware-Fehler auf, überprüfen Sie die angeschlossenen Module. Informieren Sie den MITSUBISHI-Service, und beschreiben Sie die Probleme, die Sie mit den defekten Modulen haben.
	1.) Der Zugriff auf ein Sondermodul ist bei Kommunikationsbeginn nicht möglich. 2.) Die Größe des Pufferspeichers des Sondermoduls ist fehlerhaft.	Die CPU hat einen Hardware-Fehler. Wenden Sie sich an den MITSUBISHI-Service.
	Ein Zugriff auf das Sondermodul ist nicht möglich.	Dies deutet auf einen Hardware-Fehler hin. Wenden Sie sich an den MITSUBISHI-Service.
	1.) Während der Ausführung der END-Anweisung war ein Zugriff auf das Sondermodul nicht möglich. 2.) Bei dem Sondermodul wurde ein Fehler festgestellt.	Das Sondermodul, auf das zugegriffen wurde, hat ein Hardware-Problem. Wenden Sie sich an den MITSUBISHI-Service.

Fehlercodeliste Q00J-, Q00- und Q01CPU (Fortsetzung)

Fehlercode (SD0) [1]	Fehlermeldung	Allgemeine Informationen (SD5 bis 15) [1]	Spezifische Information (SD16 bis 26) [1]	LED-Status		CPU-Status	Diagnosezeitraum
				RUN	ERROR		
1411	CONTROL-BUS ERR.	Stationsnr./Modulnr.	Lokalisierung des Programmfehlers	AUS	Blinkt	Stopp	Beim Einschalten/Beim Zurücksetzen
1412							Während der Ausführung des FROM-/TO-Anweisungssatzes
1413		—	—				Kontinuierlich
1414		—	—				Während der Ausführung der END-Anweisung
1415		Nummer des Baugruppenträgers	—				
1500	AC DOWN	—	—	EIN	AUS	Fortsetzen	Kontinuierlich
1600	BATTERY ERROR	Laufwerksname	—	EIN	AUS	Fortsetzen	Kontinuierlich
2000	UNIT VERFIY ERR.	Stationsnr./Modulnr.	—	AUS/EIN	Blinkt/EIN	Stopp/Fortsetzen [2]	Während der Ausführung der END-Anweisung
2100	SP. UNIT LAY ERR.	Stationsnr./Modulnr.	—	AUS	Blinkt	Stopp	Beim Einschalten/Beim Zurücksetzen
2103							
2106							
2107							
2110	SP UNIT ERROR	Stationsnr./Modulnr.	Lokalisierung des Programmfehlers	AUS/EIN	Blinkt/EIN	Stopp/Fortsetzen [2]	Während der Ausführung der Anweisung
2111							
2112							STOP → RUN/Während der Ausführung der Anweisung

[1] Die Angaben in den runden Klammern kennzeichnen die Adressen der Sonderregister, in denen die spezifischen Fehlerinformationen gespeichert werden.

[2] Der Verarbeitungszustand der CPU beim Auftreten eines Fehlers kann in den Parametern gesetzt werden. Die LED-Anzeige ändert sich entsprechend.

	Ursache	Abhilfe
	Nach der Adressenzuordnung über Parameter ist der Zugriff auf ein Sondermodul bei Kommunikationsbeginn nicht möglich. Wenn dieser Fehler auftritt, wird die Initialisierungs-E/A-Adresse des Moduls gespeichert.	Es hat entweder ein Sondermodul, das CPU-Modul oder ein Baugruppenträger Hardware-Probleme. Wenden Sie sich an den MITSUBISHI-Service.
	FROM- und/oder TO-Anweisungen können wegen eines Steuerimpulsfehlers nicht ausgeführt werden. Wenn dieser Fehler auftritt, wird die Lokalisierung des Programmfehlers gespeichert.	
	Am Q-Bus wurde ein Fehler festgestellt. Die Wartezeit wurde überschritten.	Ein Sondermodul, die CPU oder ein Baugruppenträger können fehlerhaft sein. Wenden Sie sich an den MITSUBISHI-Service.
	Am Q-Bus wurde ein Fehler festgestellt.	
	Am Hauptbaugruppenträger oder an einem Erweiterungsbaugruppenträger wurde ein Fehler festgestellt	
	Kurzzeitige Unterbrechung der Spannungsversorgung.	Überprüfen Sie die Spannungsversorgung.
	1.) Die Spannung der Batterie in der CPU ist unter den zulässigen Wert gesunken. 2.) Die Batterie der CPU ist nicht mit der CPU verbunden.	1.) Wechseln Sie die Batterie. 2.) Ist die Batterie für das interne RAM oder für die Backup-Funktion vorgesehen, verbinden Sie die Batterieanschlussleitung mit der CPU.
	Beim Einschalten der Spannungsversorgung haben sich die Informationen des E/A-Moduls geändert. Während des Betriebes hat sich ein E/A-Modul (oder Sondermodul) vom Baugruppenträger gelöst oder ist nicht mit ihm verbunden.	Lesen Sie die allgemeinen Fehlerinformationen auf dem Display des Programmiergerätes, und überprüfen und/oder ändern Sie den Sitz der entsprechenden Module. Alternativ dazu können Sie die Sonderregister SD150 bis SD157 auf dem Programmiergerät überprüfen und den Sitz der Module, deren Bit auf „1“ gesetzt ist, überprüfen und/oder ändern.
	Die Adresszuordnung in den Parametern ist falsch: Einem Sondermodul wurde die Adresse eines E/A-Moduls zugeordnet (bzw. umgekehrt). Einem Modul, das keine CPU ist, wurde die Adresse einer CPU zugeordnet (bzw. umgekehrt). Ein Mehrzweckschalter wurde einem Modul zugeordnet, bei dem dies nicht möglich ist.	Setzen Sie die Parameter der Adresszuordnung, und passen Sie sie an die tatsächlichen Gegebenheiten an. Setzen Sie die Einstellung der Mehrzweckschalter zurück.
	Auf dem Baugruppenträger befinden sich mehr als ein QI60-Interrupt-Modul.	Installieren Sie nur ein QI60-Modul.
	1.) Es ist mehr als ein MELSECNET/H-Module im System installiert. 2.) Es ist mehr als ein Ethernet-Modul im System installiert. 3.) Es sind mehr als zwei CC-Link-Module im System installiert. 4.) Es existieren identische Netzwerk- oder Stationsnummer im MELSECNET/H Netzwerk.	1.) Betreiben Sie höchstens 1 Modul. 2.) Betreiben Sie höchstens 1 Modul. 3.) Betreiben Sie höchstens 2 Module. 4.) Überprüfen Sie die Netzwerk- und Stationsnummern.
	Die Kopfadresse, die für die Adresszuordnung in den Parametern gesetzt ist, ist die gleiche wie bei anderen Modulen.	Setzen Sie die Parameter der Adresszuordnung zurück, und passen Sie sie den tatsächlichen Gegebenheiten an.
	Das mittels FROM-/TO-Anweisung angesprochene Modul ist kein Sondermodul. Das angesprochene Sondermodul ist gestört.	Lesen Sie die spezifischen Fehlerinformationen, und überprüfen /bearbeiten Sie die Programmierung der FROM-/TO-Anweisungen. Wenden Sie sich im Falle eines defekten Sondermoduls an den MITSUBISHI-Service.
	Das mittels direkt adressierbarer Link-Operanden angesprochene Modul ist kein Netzwerkmodul.	
	Das angesprochene Sondermodul ist kein Sondermodul oder das falsche Sondermodul.	Lesen Sie die spezifischen Fehlerinformationen, und überprüfen/bearbeiten Sie die Programmierung der Anweisung.

Fehlercodeliste Q00J-, Q00- und Q01CPU (Fortsetzung)

Fehlercode (SD0) [1]	Fehlermeldung	Allgemeine Informationen (SD5 bis 15) [1]	Spezifische Information (SD16 bis 26) [1]	LED-Status		CPU-Status	Diagnosezeitraum
				RUN	ERROR		
2120	SP. UNIT LAY ERR.	—	—	AUS	Blinkt	Stopp	Beim Einschalten/Beim Zurücksetzen
2122							
2124							
2125							
2200	MISSING PARA.	Laufwerksname	—	AUS	Blinkt	Stopp	Beim Einschalten/Beim Zurücksetzen
2400	FILE SET ERROR	File-Name	Parameternummer	AUS	Blinkt	Stopp	Beim Einschalten/Beim Zurücksetzen
2401							
2500	CAN'T EXE. PRG.	File-Name	—	AUS	Blinkt	Stopp	Beim Einschalten/Beim Zurücksetzen
2501							
2502							
2503							
3000	PARAMETER ERROR	File-Name	Parameternummer	AUS	Blinkt	Stopp	Beim Einschalten/Zurücksetzen/ STOP → RUN
3001							
3003							
3004							

[1] Die Angaben in den runden Klammern kennzeichnen die Adressen der Sonderregister, in denen die spezifischen Fehlerinformationen gespeichert werden.

[2] Der Verarbeitungszustand der CPU beim Auftreten eines Fehlers kann in den Parametern gesetzt werden. Die LED-Anzeige ändert sich entsprechend.

	Ursache	Abhilfe
	Als Baugruppenträger wurde ein QA[]B oder QA1S[]B installiert.	Benutzen Sie ein Q[]B als Baugruppenträger
	Das QA1S[]B ist als Hauptbaugruppenträger installiert.	Benutzen Sie ein Q[]B als Hauptbaugruppenträger.
	Ein Modul ist auf dem 25. Steckplatz oder höher installiert (17. Steckplatz oder höher bei Q00JCPU). Ein Modul ist auf einen Steckplatz installiert, der in der Adresszuordnung nicht mehr vorgesehen ist. Ein Modul belegt E/A-Adressen, die ausserhalb der zugelassenen E/A-Adressen liegen. Ein Modul, das die letzte E/A-Adresse belegt, belegt noch weitere Adressen. Es sind mehr als 4 Erweiterungsbaugruppenträger angeschlossen (mehr als 2 Erweiterungsbaugruppenträger bei Q00JCPU).	Entfernen Sie alle Module, die ab dem 65. Steckplatz installiert sind. Entfernen Sie das Modul, das auf einen Steckplatz ausserhalb des zugeteilten Bereiches installiert ist. Entfernen Sie das Modul, das E/A-Adressen belegt, die außerhalb des zulässigen Bereiches liegen. Tauschen Sie das Modul gegen ein Modul, das die letzte E/A-Adresse nicht überschreitet. Schließen Sie maximal 4 bzw. 2 Erweiterungsbaugruppenträger an.
	Ein Modul, das von der Q-CPU nicht erkannt wird, wurde installiert. Es ist kein Zugriff auf ein Sondermodul möglich.	Benutzen Sie ein Modul, das zusammen mit der Q-CPU eingesetzt werden kann. Das Sondermodul, auf das zugegriffen wurde, hat ein Hardware-Problem. Wenden Sie sich an den MITSUBISHI-Service.
	Im Programmspeicher sind keine Parameter vorhanden.	Überprüfen Sie die Einstellung der Parameter auf Gültigkeit der möglichen Laufwerke. Speichern Sie die Parameter auf dem korrekten Laufwerk.
	Die Datei, die durch die SPS-File-Einstellungen in den Parametern festgelegt wurde, kann nicht gefunden werden.	Lesen Sie die spezifischen Fehlerinformationen auf dem Display des Programmiergerätes, und überprüfen Sie die Übereinstimmung des Laufwerks- und Dateinamens mit den Parametereinstellungen, und nehmen Sie ggf. Korrekturen vor. Erzeugen Sie das vorgegebene File.
	Das File, das durch die SPS-RAS-Einstellung in den Parametern festgelegt ist, konnte nicht im Fehlerprotokollbereich erzeugt werden.	Lesen Sie die spezifischen Fehlerinformationen auf dem Display des Programmiergerätes, und überprüfen Sie die Übereinstimmung des Programms mit den Parametereinstellungen, und korrigieren Sie sie gegebenenfalls.
	Es existiert ein Programm-File, das Operanden benutzt, die sich außerhalb des Bereichs befinden, der in den Operanden-Parametern festgelegt ist.	Lesen Sie die spezifischen Fehlerinformationen auf dem Display des Programmiergerätes, und überzeugen Sie sich davon, dass die Parameter-Operandeneinstellungen und Programm-File-Operanden den tatsächlichen Gegebenheiten entsprechen, und nehmen Sie ggf. Korrekturen vor.
	Es existieren Programm-Files, obwohl „Keine" in den Programm-Parametern angegeben ist.	Ändern Sie die Programm-Parameter auf "Ja". Löschen Sie nicht benötigte Programme.
	Das Programm-File ist fehlerhaft. Die File-Inhalte sind nicht in der Ablaufsprache verfasst.	Überprüfen Sie, ob es sich um das File-Format ***.QPG handelt und ob die File-Inhalte für ein Ablaufprogramm vorgesehen sind.
	Es existiert kein Programm-File.	Überprüfen Sie die Programmkonfiguration.
	Die Parametereinstellung für die Zeiteinstellung der Timer, den RUN-PAUSE-Kontakt, die allgemeine Pointer-Adresse, die Gesamtdatenverarbeitung, die Anzahl der freien Steckplätze oder die System-Interrupt-Einstellungen liegen ausserhalb des von der CPU nutzbaren Bereichs.	1.) Lesen Sie die detailierten Fehlerinformationen auf dem Display des Programmiergerätes, überprüfen Sie, ob die Eintragungen in den Parametern korrekt sind und nehmen Sie ggf. Korrekturen vor. 2.) Steht der Fehler noch an, folgen Sie den Verweisen der Parametereinstellungen. Wahrscheinlich liegt ein Fehler im internen CPU-RAM vor.
	Die Parameterinhalte wurden zerstört.	
	Die in den Operanden-Parametern gesetzte Anzahl von Operanden liegt außerhalb des von der CPU nutzbaren Bereiches.	
	Das Parameter-File ist nicht von der QnA CPU verwendbar, oder die Inhalte des Files sind keine Parameter.	Überprüfen Sie, ob das Parameter-File vom Format ***.QPA ist und ob deren Inhalte auch Parameter sind.

Fehlercodeliste Q00J-, Q00- und Q01CPU (Fortsetzung)

Fehlercode (SD0) [1]	Fehlermeldung	Allgemeine Informationen (SD5 bis 15) [1]	Spezifische Information (SD16 bis 26) [1]	LED-Status		CPU-Status	Diagnosezeitraum
				RUN	ERROR		
3100	LINK PARA. ERROR	File-Name	Parameternummer	AUS	Blinkt	Stopp	Beim Einschalten/Zurücksetzen/ STOP → RUN
3101							
3102							
3103							
3104							
3105							
3106							
3107							
3300	SP. PARA. ERROR	File-Name	Parameternummer	AUS	Blinkt	Stopp	Beim Einschalten/Zurücksetzen/ STOP → RUN
3301							
3302							
4000	INSTRCT CODE. ERR.	Lokalisierung des Programmfehlers	—	AUS	Blinkt	Stopp	Beim Einschalten/Zurücksetzen/ STOP → RUN
4002							
4003							
4004							
4010	MISSING END INS.	Lokalisierung des Programmfehlers	—	AUS	Blinkt	Stopp	Beim Einschalten/Zurücksetzen/ STOP → RUN
4021	CAN'T SET (P)						
4030	CAN'T SET (I)						

[1] Die Angaben in den runden Klammern kennzeichnen die Adressen der Sonderregister, in denen die spezifischen Fehlerinformationen gespeichert werden.

	Ursache	Abhilfe
	Die Anzahl der installierten Module weicht von der in den Parametern für MELSECNET/H vorgegebenen Anzahl ab. Die Anfangsadresse der installierten Module ist unterschiedlich zu der in den Parametern für MELSECNET10H vorgegebenen Anfangsadresse. Es können nicht alle Daten in den Parametern gelesen werden. Der Typ der Station am MELSECNET/H ist gewechselt worden, als die Spannung eingeschaltet war (ein Übergang von RESET nach RUN ist nötig, um den geänderten Typ zu erkennen).	1.) Schreiben Sie die Netzwerkparameter nach der Korrektur erneut. 2.) Steht der Fehler auch nach der Korrektur an, wenden Sie sich an den MITSUBISHI-Service.
	Die Netzwerknummer, die in einem Parameter vorgegeben wurde, stimmt nicht mit der des installierten Netzwerkes überein. Die Anfangsadresse der installierten E/A-Module ist unterschiedlich zu der in den Parametern vorgegebenen Anfangsadresse. Die Klasse des Netzwerkes, die in einem Parameter vorgegeben wurde, stimmt nicht mit der des tatsächlich installierten Netzwerkes überein. Es besteht ein Fehler bei den Refresh-Parametern des MELSECNET/H.	Bringen Sie die Parameter in Übereinstimmung mit dem installierten Netzwerk.
	Es ist bei der Überprüfung der Netzwerkparameter des Netzwerkmoduls ein Fehler aufgetreten.	1.) Schreiben Sie die Netzwerkparameter nach der Korrektur erneut. 2.) Steht der Fehler auch nach der Korrektur an, wenden Sie sich an den MITSUBISHI-Service.
	Obwohl in den Parametern mindestens ein Ethernet-Modul vorgegeben wurde, ist kein Modul installiert. Die Anfangsadresse der installierten E/A-Module ist unterschiedlich zu der in den Ethernet-Parametern vorgegebenen Anfangsadresse.	
	Ethernet und MELSECNET/10 sind die selben Netzwerknummern zugeteilt. Die Vorgaben in den Parametern für die Netzwerknummer, die Stationsnummer oder die Gruppennummer überschreiten den zulässigen Bereich. Die vorgegebene E/A-Adresse überschreitet den für die CPU zulässigen Bereich. Ein Parameter, der sich auf das Ethernet bezieht, ist fehlerhaft.	
	Obwohl in den Parametern mindestens ein CC-Link-Modul vorgegeben wurde, ist kein Modul installiert. Die in den allgemeinen Parametern vorgegebenen Anfangsadresse des E/A-Bereiches ist unterschiedlich zu der des installierten E/A-Moduls. Die Klassenzuordnung einer Station des CC-Link, die in einem Parameter vorgegeben wurde, stimmt nicht mit der der tatsächlich installierten Station überein.	1.) Schreiben Sie die Netzwerkparameter nach der Korrektur erneut. 2.) Steht der Fehler auch nach der Korrektur an, wenden Sie sich an den MITSUBISHI-Service.
	Es besteht ein Fehler bei den Refresh-Parametern des CC-Link.	Überprüfen Sie die Parametrierung.
	Der Inhalt der CC-Link-Parameter ist fehlerhaft.	
	Die durch den GX Configurator vergebene Kopfadresse für ein Sondermodul stimmt nicht mit der tatsächlichen E/A-Adresse überein.	
	Bei der Aktualisierung des Sondermoduls wird der Bereich für File-Register überschritten. Die Parameter zur Aktualisierung des Sondermoduls sind fehlerhaft.	
	Die Parametrierung des Sondermoduls ist fehlerhaft.	
	In dem Programm ist ein Anweisungscode enthalten, der nicht entschlüsselt werden kann Das Programm enthält eine Anweisung, die nicht ausgeführt werden kann.	Lesen Sie die allgemeinen Fehlerinformationen auf dem Display des Programmiergerätes, und überprüfen/korrigieren Sie den angegebenen Programmschritt.
	Die Anweisung hat einen falschen Namen.	
	Die Anweisung spricht die falsche Operandenadresse an.	
	Die Anweisung spricht einen nicht nutzbaren Operanden an.	
	Das Programm beinhaltet keine END-(FEND-)Anweisung	
	Die Adressen der allgemeinen Pointer, die von den entsprechenden Files genutzt werden, überlappen.	
	Die Adressen der zugeordneten Pointer, die von den entsprechenden Files genutzt werden, überlappen.	

Fehlercodeliste Q00J-, Q00- und Q01CPU (Fortsetzung)

Fehlercode (SD0) [1]	Fehlermeldung	Allgemeine Information (SD5 bis 15) [1]	Spezifische Information (SD16 bis 26) [1]	LED-Status		CPU-Status	Diagnosezeitraum
				RUN	ERROR		
4100	OPERATION ERROR	Lokalisierung des Programmfehlers	—	AUS/EIN	Blinkt/EIN	Stopp/ Fortsetzen [2]	Bei Ausführung einer Anweisung
4101							
4102							
4108							
4200	FOR NEXT ERROR	Lokalisierung des Programmfehlers	—	AUS	Blinkt	STOPP	Bei Ausführung einer Anweisung
4201							
4202							
4203							
4210	CAN'T EXECUTE (P)	Lokalisierung des Programmfehlers	—	AUS	Blinkt	STOPP	Bei Ausführung einer Anweisung
4211							
4212							
4213							
4220	CAN'T EXECUTE (I)	Lokalisierung des Programmfehlers	—	AUS	Blinkt	STOPP	Bei Ausführung einer Anweisung
4221							
4223							
4231	INST. FORMAT ERR	Lokalisierung des Programmfehlers	—	AUS	Blinkt	STOPP	Bei Ausführung einer Anweisung
5001	WDT ERROR	Zeit (Einstellwert)	Zeit (tatsächlich gemessener Wert)	AUS	Blinkt	STOPP	kontinuierlich
5010	PRG. TIME OVER	Zeit (Einstellwert)	Zeit (tatsächlich gemessener Wert)	EIN	EIN	Fortsetzen	kontinuierlich
9000	F**** [3]	Lokalisierung des Programmfehlers	Nr. des Fehlermerkers	EIN	AUS	Fortsetzen	Bei Ausführung einer Anweisung
				USER LED EIN			

[1] Die Angaben in den runden Klammern kennzeichnen die Adressen der Sonderregister, in denen die spezifischen Fehlerinformationen gespeichert werden.

[2] Der Verarbeitungszustand der CPU beim Auftreten eines Fehlers kann in den Parametern gesetzt werden. Die LED-Anzeige ändert sich entsprechend.

[3] **** weisen auf die erkannte Nummer des Fehlermerkers hin.

	Ursache	Abhilfe
	Die enthaltenen Daten können von der entsprechenden Anweisung nicht verarbeitet werden.	Lesen Sie die allgemeinen Fehlerinformationen auf dem Display des Programmiergerätes, und überprüfen/korrigieren Sie den angegebenen Programmschritt.
	Die angegebenen Adressen der Daten, die von dem Programm verarbeitet werden sollen, oder die gespeicherten Daten oder Konstanten der Operanden, die von den Anweisungen verwendet werden, liegen außerhalb des nutzbaren Adressbereichs.	
	Die Netzwerk- oder Stationsnummer, die durch eine erweiterte Netzwerkanweisung angesprochen wurde, ist nicht korrekt.	
	Bei Ausführung der CC-Link-Anweisung waren keine CC-Link-Parameter eingetragen.	Setzen Sie die Parameter vor Ausführung der CC-Link-Anweisung.
	Es wird keine NEXT-Anweisung nach der FOR-Anweisung ausgeführt, oder es existieren weniger NEXT- als FOR-Anweisungen.	Lesen Sie die allgemeinen Fehlerinformationen auf dem Display des Programmiergerätes und überprüfen/korrigieren Sie den angegebenen Programmschritt.
	Es wird eine NEXT-Anweisung ausgeführt, obwohl keine FOR-Anweisung ausgeführt wurde, oder es existieren mehr NEXT- als FOR-Anweisungen.	
	Es sind mehr als 16 Verschachtelungsebenen (Nesting) programmiert worden.	Reduzieren Sie die Anzahl der Verschachtelungsebenen auf weniger als 17.
	Es wird eine BREAK-Anweisung ausgeführt, obwohl keine FOR-Anweisung ausgeführt wurde.	Lesen Sie die allgemeinen Fehlerinformationen auf dem Display des Programmiergerätes, und überprüfen/korrigieren Sie den angegebenen Programmschritt.
	Die CALL-Anweisung wird ausgeführt, aber an dem angegebenen Pointer ist keine Unterprogrammroutine vorhanden.	Lesen Sie die allgemeinen Fehlerinformationen auf dem Display des Programmiergerätes, und überprüfen/korrigieren Sie den angegebenen Programmschritt.
	In dem ausgeführten Unterprogramm existiert keine RET-Anweisung.	
	Die RET-Anweisung steht vor der FEND-Anweisung im Hauptprogramm.	
	Es sind mehr als 16 Verschachtelungsebenen (Nesting) programmiert worden.	Reduzieren Sie die Anzahl der Verschachtelungsebenen auf weniger als 17.
	Es wurde eine Interrupt-Eingabe gemacht, aber kein entsprechender Interrupt-Pointer gefunden.	Lesen Sie die allgemeinen Fehlerinformationen auf dem Display des Programmiergerätes, und überprüfen/korrigieren Sie den angegebenen Programmschritt.
	In dem ausgeführten Interrupt-Programm existiert keine IRET-Anweisung.	
	Die IRET-Anweisung befindet sich im Hauptprogramm vor der FEND-Anweisung.	
	Die IX- und die IXEND-Anweisung sind nicht im Zusammenhang programmiert. Es existiert die gleiche Anzahl von IX- und IXEND-Anweisungen.	Lesen Sie die allgemeinen Fehlerinformationen auf dem Display des Programmiergerätes, und überprüfen/korrigieren Sie den angegebenen Programmschritt.
	Die Programmzykluszeit übersteigt die in dem PC RAS Parameter eingestellte Zeit der Fehlerüberwachung des „Watch Dog Timers“.	Lesen Sie die spezifischen Fehlerinformationen auf dem Display des Programmiergerätes, und überprüfen/korrigieren (verkürzen) Sie die eingestellte Zykluszeit.
	Die Programmzykluszeit eines Programmes mit dem Verarbeitungsmodus „Low speed scan“ übersteigt die in dem PC RAS-Parameter eingestellte konstante Zykluszeit.	Überprüfen und ändern Sie die konstante Zykluszeit oder die Ausführungszeit eines Programmes mit dem Verarbeitungsmodus „low speed scan“.
	Der Fehlermerker F wurde auf EIN gesetzt.	Lesen Sie die spezifischen Fehlerinformationen auf dem Display des Programmiergerätes, und überprüfen/korrigieren Sie das Programm anhand der eingetragenen Fehlermerkernummer.

13.2 Liste der Fehlercodes (QnA-Serie und System Q)

Die folgende Tabelle enthält eine Übersicht der möglichen Fehler zusammen mit Fehlermeldungen, möglichen Ursachen und Hinweisen zur Behebung des Fehlers. In dieser Tabelle sind nur Fehlermeldungen der Q02(H)-, Q06H-, Q12(P)H-, Q25(P)H-, QnA(S)- und Q4AR-CPUs aufgeführt. Das Zeichen „●“ in der letzten Spalte gibt an, dass der Fehlercode für alle genannten CPUs gilt. „Rem“ bedeutet Kompatibilität mit dezentralen E/A-Modulen. Ein in dieser Spalte eingetragener CPU-Typ weist darauf hin, dass dieser Fehlercode nur für diesen CPU-Typ gilt.

Fehlercode (SD0) [1]	Fehlermeldung	Allgemeine Information (SD5 bis 15) [1]	Spezifische Information (SD16 bis 26) [1]	LED-Status		CPU-Status	Diagnosezeitraum	
				RUN	ERROR			
1000	MAIN CPU DOWN	—	—	AUS	Blinkt/EIN	Stopp	Kontinuierlich	
1001				AUS	Blinkt	Stopp	Kontinuierlich	
1002								
1003								
1004								
1005								
1006								
1007								
1008								
1009								
1010	END NOT EXECUTE	—	—	AUS	Blinkt	Stopp	Während der Ausführung der END-Anweisung	
1011								
1012								
1101	RAM ERROR	—	—	AUS	Blinkt	Stopp	Beim Einschalten/Beim Zurücksetzen	
1102								
1103								
1104								
1105								
1200	OPE. CIRCUIT ERR.	—	—	AUS	Blinkt	Stopp	Beim Einschalten/Beim Zurücksetzen	
1201								
1202								
1203							Während der Ausführung der END-Anweisung	
1204								
1205								
1206							Bei Ausführung der Anweisung	

[1] Die Angaben in den runden Klammern kennzeichnen die Adressen der Sonderregister, in denen die einzelnen Fehlerinformationen gespeichert werden.

	Ursache	Abhilfe	Gültig für:
	Abschaltung des RUN-Modus oder Fehler in der CPU 1.) Funktionsstörung aufgrund von Störspannungen (Rauschen) oder aus anderen Gründen 2.) Hardware-Fehler	1.) Messen Sie den Rauschpegel. 2.) Die CPU zurücksetzen und in den RUN-Modus schalten. Wenn der gleiche Fehler wieder angezeigt wird, deutet dieses auf einen Hardware-Fehler hin. Wenden Sie sich an den MITSUBISHI-Service.	QnA-CPU
	Abschaltung des RUN-Modus oder Fehler in der CPU 1.) Funktionsstörung aufgrund von Störspannungen (Rauschen) oder aus anderen Gründen 2.) Hardware-Fehler	1.) Messen Sie den Rauschpegel. 2.) Die CPU zurücksetzen und in den RUN-Modus schalten. Wenn der gleiche Fehler wieder angezeigt wird, deutet dieses auf einen Hardware-Fehler hin. Wenden Sie sich an den MITSUBISHI-Service.	Q-CPU Q-CPU/Rem Q-CPU
	Das gesamte Programm wurde ohne Ausführung der END-Anweisung ausgeführt. 1.) Wenn die END-Anweisung ausgeführt wird, wird sie als ein anderer Anweisungscode gelesen. 2.) Die END-Anweisung wurde in einen anderen Anweisungscode geändert.	1.) Messen Sie den Rauschpegel. 2.) Die CPU zurücksetzen und in den RUN-Modus schalten. Wenn der gleiche Fehler wieder angezeigt wird, deutet dieses auf einen Hardware-Fehler hin. Wenden Sie sich an den MITSUBISHI-Service.	
	Fehler im internen RAM, in dem das CPU-Ablaufprogramm gespeichert ist.	Dies weist auf einen CPU-Hardware-Fehler hin. Wenden Sie sich an den MITSUBISHI-Service.	●
	Fehler im RAM, das als CPU-Arbeitsbereich benutzt wird.		
	Interner CPU-Fehler		
	Fehler bei der RAM-Adressierung in der CPU.		
	Fehler im gemeinsamen Speicherbereich für Multi-CPU-Betrieb	1.) Messen Sie den Rauschpegel. 2.) Die CPU zurücksetzen und in den RUN-Modus schalten. Wenn der gleiche Fehler wieder angezeigt wird, deutet dieses auf einen Hardware-Fehler hin. Wenden Sie sich an den MITSUBISHI-Service.	Q-CPU ab Version B
	Der CPU-Schaltkreis, der für die Index-Verarbeitung verantwortlich ist, arbeitet fehlerhaft.	Dies weist auf einen CPU-Hardware-Fehler hin. Wenden Sie sich an den MITSUBISHI-Service.	●
	Die CPU-Hardware (Logik) arbeitet fehlerhaft.		
	Der CPU-Schaltkreis, der für die Ablaufverarbeitung verantwortlich ist, arbeitet fehlerhaft.		
	Der CPU-Schaltkreis, der für die Index-Verarbeitung verantwortlich ist, arbeitet fehlerhaft	Dies weist auf einen CPU-Hardware-Fehler hin. Wenden Sie sich an den MITSUBISHI-Service.	Q4AR-CPU
	Die CPU-Hardware (Logik) arbeitet fehlerhaft.		
	Der Schaltkreis, der für die Ablaufverarbeitung verantwortlich ist, arbeitet fehlerhaft.		
	Der CPU-Schaltkreis, der für die DSP-Operationen verantwortlich ist, arbeitet fehlerhaft.		

Fehlercodeliste QnA-Serie und System Q (Fortsetzung)

Fehlercode (SD0) [1]	Fehlermeldung	Allgemeine Informationen (SD5 bis 15) [1]	Spezifische Information (SD16 bis 26) [1]	LED-Status		CPU-Status	Diagnosezeitraum
				RUN	ERROR		
1300	FUSE BREAK OFF	Stationsnr./Modulnr.	—	AUS/EIN	Blinkt/EIN	Stopp/ Fortsetzen [2]	Während der Ausführung der END-Anweisung.
1301	EX POWER OFF	Stationsnr./Modulnr	—	AUS/EIN	Blinkt/EIN	Stopp/ Fortsetzen [2]	Während der Ausführung der END-Anweisung.
1310	I/O INT ERROR	Stationsnr./Modulnr.	—	AUS	Blinkt/EIN	Stopp	Während eines Interruptes
1401	SP. UNIT DOWN	Stationsnr./Modulnr.	—	AUS/EIN	Blinkt/EIN	Stopp/ Fortsetzen [3]	Beim Einschalten/Beim Zurücksetzen Beim Zugriff auf ein Sondermodul
							Beim Einschalten/Beim Zurücksetzen
1402			Lokalisierung des Programmfehlers		Blinkt/EIN	Stopp/ Fortsetzen [4]	Beim Zugriff auf ein Sondermodul
							Während der Ausführung des FROM-/TO-Anweisungssatzes
1403			—				Während der Ausführung der END-Anweisung
1411	CONTROL-BUS ERR.	Stationsnr./Modulnr.	Lokalisierung des Programmfehlers	AUS	Blinkt	Stopp	Beim Einschalten/Beim Zurücksetzen
1412							Während der Ausführung des FROM-/TO-Anweisungssatzes
1413	CONTROL-BUS ERR.	—	—	AUS	Blinkt	Stopp	Kontinuierlich

[1] Die Angaben in den runden Klammern kennzeichnen die Adressen der Sonderregister, in denen die spezifischen Fehlerinformationen gespeichert werden.

[2] Der Verarbeitungszustand der CPU beim Auftreten eines Fehlers kann in den Parametern gesetzt werden. Die LED-Anzeige ändert sich entsprechend.

[3] Dieser Fehler kann nur in redudanten Systemen gemeldet werden. Die Fehlererkennung ist für das aktive System und das Reservesystem möglich.

[4] In den Parametern kann für jedes Modul eingestellt werden, wie sich das System bei einem Fehler verhalten soll.

	Ursache	Abhilfe	Gültig für:
	Die Sicherung eines Ausgangsmoduls ist defekt.	1.) Überprüfen Sie die ERR-LED an den Ausgangsmodulen und wechseln Sie das Modul, dessen LED leuchtet. 2.) Das Modul mit der defekten Sicherung kann auch durch Auswerten der Sonderregister SD1300 bis SD1331gefunden werden.	Q-CPU Rem
	Die Sicherung eines Ausgangsmoduls ist defekt.	1.) Überprüfen Sie die LED-Anzeigen der Sicherungen an den Ausgangsmodulen und wechseln Sie die Sicherung, deren LED leuchtet. 2.) Lesen Sie die allgemeinen Fehlerinformationen auf dem Display des Programmiergerätes, und wechseln Sie die Sicherung des Ausgangsmoduls, das angezeigt wird. Alternativ dazu prüfen Sie die Sonderregister SD1300 bis SD1331 mit dem Programmiergerät, und wechseln Sie die Sicherung des Ausgangsmoduls, bei dem das entsprechende Bit auf „1“ gesetzt ist	QnA-CPU, Q4AR-CPU
	Die Sicherung eines Ausgangsmoduls ist defekt. Die externe Spannungsversorgung der Ausgänge wurde abgeschaltet oder ist nicht angeschlossen.	1.) Überprüfen Sie die LED-Anzeigen der Sicherungen an den Ausgangsmodulen und wechseln Sie die Sicherung, deren LED leuchtet. 2.) Lesen Sie die allgemeinen Fehlerinformationen auf dem Display des Programmiergerätes, und wechseln Sie die Sicherung des Ausgangsmoduls, das angezeigt wird. Alternativ dazu prüfen Sie die Sonderregister SD1300 bis SD1331 mit dem Programmiergerät, und wechseln Sie die Sicherung des Ausgangsmoduls, bei dem das entsprechende Bit auf „1“ gesetzt ist. 3.) Überprüfen Sie die externe Versorgungsspannung der Ausgänge.	Q2AS-CPU
	Die externe Spannungsversorgung der Ausgänge wurde abgeschaltet oder ist nicht angeschlossen (in Vorbereitung).	Überprüfen Sie die externe Versorgungsspannung der Ausgänge.	Q-CPU/Rem
	Ein Interrupt wurde ausgeführt, obwohl sich im System kein Interrupt-Modul befindet.	Eines der angeschlossenen Module weist einen Hardware-Fehler auf. Überprüfen Sie die angeschlossenen Module. Wenden Sie sich an den MITSUBISHI-Service.	●
	1.) Der Zugriff auf ein Sondermodul ist bei Kommunikationsbeginn nicht möglich. 2.) Die Größe des Pufferspeichers des Sondermoduls ist fehlerhaft.	Die CPU hat einen Hardware-Fehler. Wenden Sie sich an den MITSUBISHI-Service.	Q-CPU Rem
	Nach der Adressenzuordnung über Parameter ist der Zugriff auf ein Sondermodul bei Kommunikationsbeginn nicht möglich. Wenn dieser Fehler auftritt, wird die Initialisierungs-E/A-Adresse des Moduls gespeichert	Das Sondermodul, auf das zugegriffen wurde, hat einen Hardware-Fehler. Wenden Sie sich an den MITSUBISHI-Service.	QnA-CPU
	Ein Zugriff auf das Sondermodul ist nicht möglich.	Die CPU hat einen Hardware-Fehler. Wenden Sie sich an den MITSUBISHI-Service.	Q-CPU Rem
	Der Zugriff auf ein Sondermodul nach Ausführung einer FROM- und/oder TO-Anweisung erfolgt ohne Reaktion des Sondermoduls. Wenn dieser Fehler auftritt, wird die Lokalisierung des Programmfehlers gespeichert.	Das Sondermodul, auf das zugegriffen wurde, hat ein Hardware-Problem. Wenden Sie sich an den MITSUBISHI-Service.	QnA-CPU
	1.) Während der Ausführung der END-Anweisung war ein Zugriff auf das Sondermodul nicht möglich. 2.) Bei dem Sondermodul wurde ein Fehler festgestellt.		Q-CPU Rem
	Nach der Adressenzuordnung über Parameter ist der Zugriff auf ein Sondermodul bei Kommunikationsbeginn nicht möglich. Wenn dieser Fehler auftritt, wird die Initialisierungs-E/A-Adresse des Moduls gespeichert.	Es hat entweder ein Sondermodul, das CPU-Modul oder ein Baugruppenträger Hardware-Probleme. Wenden Sie sich an den MITSUBISHI-Service.	● Rem
	FROM- und/oder TO-Anweisungen können wegen eines Steuerimpulsfehlers nicht ausgeführt werden. Wenn dieser Fehler auftritt, wird die Lokalisierung des Programmfehlers gespeichert.		●
	In einem Multi-CPU-System wurde eine CPU der Version A installiert.	Tauschen Sie die CPU gegen eine mit der Version B oder höher. Ein Sondermodul, die CPU oder der Baugruppenträger können fehlerhaft sein. Wenden Sie sich an den MITSUBISHI-Service.	Q-CPU ab Version B
	Am Q-Bus wurde ein Fehler festgestellt. Die Wartezeit wurde überschritten.	Ein Sondermodul, die CPU oder der Hauptbaugruppenträger können fehlerhaft sein. Wenden Sie sich an den MITSUBISHI-Service.	Q-CPU Rem

Fehlercodeliste QnA-Serie und System Q (Fortsetzung)

Fehlercode (SD0) [1]	Fehlermeldung	Allgemeine Informationen (SD5 bis 15) [1]	Spezifische Information (SD16 bis 26) [1]	LED-Status		CPU-Status	Diagnosezeitraum
				RUN	ERROR		
1414	CONTROL-BUS ERR.	Stationsnr./Modulnr.	—	AUS	Blinkt	Stopp	Während der Ausführung der END-Anweisung
		—					
1415		Nummer des Baugruppenträgers					
1416		Stationsnr./Modulnr.					Beim Einschalten/Beim Zurücksetzen
1421	SYS. UNIT DOWN [3]	—	—	AUS	Blinkt	Stopp	Kontinuierlich
1500	AC DOWN	—	—	EIN	AUS	Fortsetzen	Kontinuierlich
1510	DUAL DC DOWN 5V [4]	—	—	EIN	EIN	Fortsetzen	Kontinuierlich
1520	DC DOWN 5V [5]	—	—	AUS	Blinkt	Stopp	Kontinuierlich
1530	DC DOWN 24V [3]	—	—	EIN	EIN	Fortsetzen	Kontinuierlich
1600	BATTERY ERROR	Laufwerksname	—	EIN	AUS	Fortsetzen	Kontinuierlich
1601				BAT. ALM LED EIN			
1602							
2000	UNIT VERFIY ERR.	Stationsnr./Modulnr.	—	AUS/EIN	Blinkt/EIN	Stopp/ Fortsetzen [2]	Während der Ausführung der END-Anweisung
2100	SP. UNIT LAY ERR.	Stationsnr./Modulnr.	—	AUS	Blinkt	Stopp	Beim Einschalten/Beim Zurücksetzen
2101							

[1] Die Angaben in den runden Klammern kennzeichnen die Adressen der Sonderregister, in denen die spezifischen Fehlerinformationen gespeichert werden.

[2] Der Verarbeitungszustand der CPU beim Auftreten eines Fehlers kann in den Parametern gesetzt werden. Die LED-Anzeige ändert sich entsprechend.

[3] Dieser Fehler kann nur in redudanten Systemen gemeldet werden. Die Fehlererkennung ist für das aktive System und das Reservesystem möglich.

[4] Dieser Fehler kann nur in redudanten Systemen gemeldet werden.

[5] Dieser Fehler kann entweder in einem nichtredudanten System oder im aktiven Teil eines redudanten Systemes erkannt werden.

	Ursache	Abhilfe	Gültig für:
	Eines der installierten Module ist fehlerhaft. In einem Multi-CPU-System wurde eine CPU der Version A installiert.	Tauschen Sie die CPU gegen eine mit der Version B oder höher. Ein Sondermodul, die CPU oder der Baugruppenträger können fehlerhaft sein. Wenden Sie sich an den MITSUBISHI-Service.	Q-CPU ab Version B
	Am System-Bus wurde ein Fehler festgestellt.	Ein Sondermodul, die CPU oder der Baugruppenträger können fehlerhaft sein. Wenden Sie sich an den MITSUBISHI-Service.	Q-CPU Rem
	Beim Haupt- oder einem Erweiterungsbaugruppenträger wurde ein Fehler festgestellt.	Ein Sondermodul, die CPU oder der Baugruppenträger können fehlerhaft sein. Wenden Sie sich an den MITSUBISHI-Service	Q-CPU ab Version B
	Beim Einschalten wurde ein Busfehler festgestellt.		
	Das Systemmanagementmodul AS92R hat einen Hardware-Fehler	Wenden Sie sich an den MITSUBISHI-Service.	Q4AR-CPU
	Kurzzeitige Unterbrechung der Spannungsversorgung.	Überprüfen Sie die Spannungsversorgung.	●/Rem
	Eine der beiden 5 V Versorgungsspannungen im Erweiterungsbaugruppenträger des redundanten Systemes ist auf unter 85 % der Nennspannung gesunken.	Überprüfen Sie die das Ausgangsspannung des Netzteils. Wechseln Sie das Netzteil, wenn die Spannung nicht der Nennspannung entspricht.	Q4AR-CPU
	Die 5 Volt Versorgungsspannung im Erweiterungsbaugruppenträger ist auf unter 85 % der Nennspannung gesunken.		
	Die 24 V Versorgungsspannung des Management-Moduls AS92R ist auf unter 85 % der Nennspannung gesunken.	Überprüfen Sie die Spannungsversorgung.	
	1.) Die Spannung der Batterie in der CPU ist unter den zulässigen Wert gesunken. 2.) Die Batterie der CPU ist nicht mit der CPU verbunden.	1.) Wechseln Sie die Batterie. 2.) Ist die Batterie für das interne RAM oder für die Backup-Funktion vorgesehen, verbinden Sie die Batterieanschlussleitung mit der CPU.	●
	Die Spannung der Batterie in der Speicherkarte 1 ist unter dem zulässigen Wert gesunken.	Wechseln Sie die Batterie.	●
	Die Spannung der Batterie in der Speicherkarte 2 ist unter dem zulässigen Wert gesunken.		QnA-CPU
	Beim Einschalten der Spannungsversorgung haben sich die Informationen des E/A-Moduls geändert. Während des Betriebes hat sich ein E/A-Modul (oder Sondermodul) vom Baugruppenträger gelöst oder ist nicht mit ihm verbunden.	Lesen Sie die allgemeinen Fehlerinformationen auf dem Display des Programmiergerätes, und überprüfen und/oder ändern Sie den Sitz der entsprechenden Module. Alternativ dazu können Sie die Sonderregister SD1400 bis SD 1431 auf dem Programmiergerät überwachen und den Sitz der Module, deren Bit auf den Wert „1“ gesetzt ist, überprüfen und/oder ändern.	● Rem
	In einem Multi-CPU-System wurde eine CPU der Version A installiert.	Tauschen Sie die CPU gegen eine mit der Version B.	Q-CPU ab Version B
	Bei der E/A-Konfiguration wurde dem Steckplatz, auf dem ein QI60-Interrupt-Modul installiert ist, nicht „Intelligentes Sondermodul“ oder „Interrupt-Modul“ zugewiesen.	Passen Sie die E/A-Konfiguration dem tatsächlichen Systemaufbau an.	Q-CPU ab Version B
	Die Adresszuordnung in den Parametern ist falsch: Einem Sondermodul wurde die Adresse eines E/A-Moduls zugeordnet (bzw. umgekehrt). Einem Modul, das keine CPU ist, wurde die Adresse einer CPU zugeordnet (bzw. umgekehrt). Dem Steckplatz der CPU wurde keine CPU zugeordnet. Ein Mehrzweckschalter wurde einem Modul zugeordnet, bei dem dies nicht möglich ist.	Setzen Sie die Parameter der Adresszuordnung, und passen Sie sie an die tatsächlichen Gegebenheiten an. Setzen Sie die Einstellung der Mehrzweckschalter zurück.	Q-CPU Rem
	Die Adresszuordnung in den Parametern ist falsch. Einem Sondermodul wurde die Adresse eines E/A-Moduls zugeordnet (bzw. umgekehrt).	Setzen Sie die Parameter der Adresszuordnung, und passen Sie sie an die tatsächlichen Gegebenheiten an.	QnA-CPU
	Im System befinden sich mehr als 12 Sondermodule der A-Serie (ausgenommen QI60/ A1SI61), die einen Interrupt zur CPU ausführen können.	Reduzieren Sie die Anzahl der Sondermodule aus der A-Serie (ausgenommen Q/60 und A1SI61) auf 12 oder weniger.	Q-CPU
	Im System befinden sich mehr als 12 Sondermodule (ausgenommen AI61), die einen Interrupt zur CPU ausführen können.	Reduzieren Sie die Anzahl der Sondermodule (ausgenommen AI61) auf 12 oder weniger.	QnA-CPU

Fehlercodeliste QnA-Serie und System Q (Fortsetzung)

Fehlercode (SD0) [1]	Fehlermeldung	Allgemeine Informationen (SD5 bis 15) [1]	Spezifische Information (SD16 bis 26) [1]	LED-Status		CPU-Status	Diagnosezeitraum
				RUN	ERROR		
2102	SP. UNIT LAY ERR.	Stationsnr./Modulnr.	—	AUS	Blinkt	Stopp	Beim Einschalten/Beim Zurücksetzen
2103							
2104							
2105							
2106	SP. UNIT LAY ERR.	Stationsnr./Modulnr.	—	AUS	Blinkt	Stopp	Beim Einschalten/Beim Zurücksetzen
2107							
2108							
2109[2]							

[1] Die Angaben in den runden Klammern kennzeichnen die Adressen der Sonderregister, in denen die spezifischen Fehlerinformationen gespeichert werden.

[2] Dieser Fehler kann nur im Reservesystem eines redudanten Systems gemeldet werden.

Ursache	Abhilfe	Gültig für:
Im System befinden sich mehr als 6 Module A1SD51S.	Reduzieren Sie die Anzahl der A1SD51S auf 6 oder weniger.	Q-CPU
Im System befinden sich mehr als 6 serielle Kommunikationsmodule (ausgenommen A(1S)J71QC24).	Reduzieren Sie die Anzahl der seriellen Kommunikationsmodule (ausgenommen A(1S)J71QC24 auf 6 oder weniger.	QnA-CPU Rem
1.) In einem Single-CPU-System ist mehr als ein QI60- oder A1SI61-Interrupt-Modul installiert. 2.) In einem Multi-CPU-System ist einer CPU mehr als ein QI60/A1SI61-Interrupt-Modul zugeordnet. 3.) In einem Multi-CPU-System ist mehr als ein A1SI61-IModul installiert.	1.) Installieren Sie nur ein QI60 oder A1SI61. 2.) Jeder CPU darf nur ein QI60 oder A1SI61 zugeordnet werden. 3.) Installieren Sie nur ein A1SI61-Modul. Verwenden Sie das QI60, wenn in einem Multi-CPU-System jeder CPU ein Interrupt-Modul zugeordnet werden soll. Kombinieren sie ein A1S61 kann mit max. drei QI60 oder verwenden sie nur das QI60.	Q-CPU ab Version B
Auf dem Baugruppenträger befinden sich mehr als ein QI60/A(1)1SI61-Interrupt-Modul.	Installieren Sie nur ein OI60- oder A1SI61-Modul.	Q-CPU
Auf dem Baugruppenträger befinden sich mehr als ein AIS61-Interrupt-Modul.	Installieren Sie nur ein AIS61-Modul.	QnA-CPU
Die Parameterzuweisung eines automatischen MELSECNET/MINI-Refreshs für einige Module im Netzwerk stimmt nicht mit den tatsächlichen Gegebenheiten überein.	Setzen Sie die Parameter des automatischen MELSECNET/MINI-Refreshs zurück, und passen Sie sie den tatsächlichen Gegebenheiten an.	QnA-CPU
Die maximale Anzahl von Sondermodulen, die einem CPU-Modul zugeordnet sind und die die erweiteten Anweisungen verarbeiten können, ist überschritten (die maximale Anzahl darf 1344 nicht überschreiten). (Anzahl der installierten AD59 x 5) (Anzahl der installierten AD57(S1)/AD58 x 8) (Anzahl der installierten AJ71C24(S3/S6/S8 x 10) (Anzahl der installierten AJ71UC24 x 10) (Anzahl der installierten AJ71C21(S1) x 29) (Anzahl der installierten AJ71PT32(S3) x 125) (Anzahl der installierten AJ71QC24 (R2, R4) x 29) (Anzahl der installierten AJ71ID1 (2)-R4 x 18) (Anzahl der installierten AD75 x 12) Gesamt > 1344	Reduzieren Sie die Anzahl der installierten Sondermodule.	QnA-CPU
1.) In einem Multi-CPU-System sind mehr als vier MELSECNET/-Module installiert. 2.) In einem Multi-CPU-System sind mehr als vier System-Q-Ethernet-Module installiert.	Betreiben Sie höchstens 4 Module.	Q-CPU ab Version B
1.) Es sind mehr als vier MELSECNET/H-Module installiert. 2.) Es sind mehr als vier System-Q-Ethernet-Module installiert. 3.) Es existieren identische Netzwerk- oder Stationsnummer im MELSECNET/H-Netzwerk	1.) Betreiben Sie höchstens 4 Module. 2.) Betreiben Sie höchstens 4 Module. 3.) Überprüfen Sie die Netzwerk- und Stationsnummern.	Q-CPU Rem
1.) Es sind mehr als vier AJ71QLP21 oder AJ71QBR11 im System installiert. 2.) Es sind mehr als zwei AJ71AP21/R21 oder AJ71AT21B im System installiert. 3.) Es sind insgesamt mehr als vier AJ71QLP21, AJ71QBR11, AJ71AP21/R21, oder AJ71AT21im System installiert. 4.) Es existieren identische Netzwerk- oder Stationsnummer im MELSECNET/10-Netzwerk. 5.) Es sind mehr als eine Master-Station oder lokale Station zur gleichen Zeit im MELSECNET (II) oder MELSECNET/B Netzwerk vorhanden.	1.) Betreiben Sie höchstens 4 Module. 2.) Betreiben Sie höchstens 2 Module. 3.) Betreiben Sie insgesamt höchstens 4 Module. 4.) Überprüfen Sie die Netzwerk- und Stationsnummern. 5.) Überprüfen Sie die Stationsnummern.	QnA-CPU
Die Kopfadresse, die für die Adresszuordnung in den Parametern gesetzt ist, ist die gleiche wie bei anderen Modulen.	Setzen Sie die Parameter der Adresszuordnung zurück, und passen Sie sie den tatsächlichen Gegebenheiten an.	● Rem
Die Module A1SJ71LP21, A1SJ71BR11, A1SJ71AP21, A1SJ71AR21 oder A1SJ71AT2 sind für den Einsatz in einem A2USCPU-Netzwerk vorgesehen und sind in diesem Netzwerk installiert. Die Module A1SJ71QLP21oder A1SJ71QBR11, die für die Q2AS-CPU vorgesehen sind, sind in diesem Netzwerk installiert.	Wechseln Sie die Module gegen ein QJ71LP21 oder ein QJ71BR11 aus.	Q-CPU
Die Module A(1S)J71LP21 oder A(1S)J71BR11 sind für den Einsatz in einem AnUCPU-Netzwerk vorgesehen und sind in diesem Netzwerk installiert.	Tauschen Sie die Module gegen ein A(1S)J71QLP21 oder ein A(1S)J71QBR11.	QnA-CPU
Bei einem redundanten System im Backup-Modus sind aktives System und Reservesystem unterschiedlich konfiguriert.	Überprüfen Sie die Konfiguration des Reservesystems.	Q4AR-CPU

Fehlercode (SD0) [1]	Fehlermeldung	Allgemeine Informationen (SD5 bis 15) [1]	Spezifische Information (SD16 bis 26) [1]	LED-Status		CPU-Status	Diagnosezeitraum
				RUN	ERROR		
2110	SP UNIT ERROR	Stationsnr./Modulnr.	Lokalisierung des Programmfehlers	AUS/EIN	Blinkt/EIN	Stopp/ Fortsetzen [2]	Während der Ausführung des FROM/ TO-Anweisungsatzes
2111							
2112	SP UNIT ERROR	Stationsnr./Modulnr.	Lokalisierung des Programmfehlers	AUS/EIN	Blinkt/EIN	Stopp/ Fortsetzen [2]	Während der Ausführung des FROM-/ TO-Anweisungsatzes
2113		FFFFH (fest)					
2114	SP UNIT ERROR	Stationsnr./Modulnr.	Lokalisierung des Programmfehlers	Blinkt/ EIN	Blinkt/EIN	Stopp/ Fortsetzen	Bei der Ausführung der Anweisung
2115							
2116							
2117							
2120	SP. UNIT LAY ERR.	—	—	AUS	Blinkt	Stopp	Beim Einschalten/Beim Zurücksetzen
2121							
2122							
2124							
2125							
2126		Stationsnr./Modulnr.	—				
2150	SP.UNIT VER. ERR.	Stationsnr./Modulnr.	—	AUS	Blinkt	Stopp	Beim Einschalten/Beim Zurücksetzen

[1] Die Angaben in den runden Klammern kennzeichnen die Adressen der Sonderregister, in denen die spezifischen Fehlerinformationen gespeichert werden.

[2] Der Verarbeitungszustand der CPU beim Auftreten eines Fehlers kann in den Parametern gesetzt werden. Die LED-Anzeige ändert sich entsprechend.

	Ursache	Abhilfe	Gültig für:
	Bei einer Anweisung, die sich auf den gemeinsamen Speicherbereich im Multi-CPU-System bezieht, wurde eine CPU angegeben, die nicht existiert.	Lesen Sie die spezifischen Fehlerinformationen, und überprüfen/bearbeiten Sie die Programmierung der Anweisung.	Q-CPU ab Version B
	Das mittels FROM-/TO-Anweisung angesprochene Modul ist kein Sondermodul. Das angesprochene Sondermodul ist gestört.	Lesen Sie die spezifischen Fehlerinformationen, und überprüfen /bearbeiten Sie die Programmierung der FROM-/TO-Anweisungen. Wenden Sie sich im Falle eines defekten Sondermoduls an den MITSUBISHI-Service.	●
	Das mittels direkt adressierbarer Link-Operanden angesprochene Modul ist kein Netzwerkmodul.		
	Das angesprochene Sondermodul ist kein Sondermodul oder das falsche Sondermodul.	Lesen Sie die spezifischen Fehlerinformationen, und überprüfen/bearbeiten Sie die Programmierung der Anweisung.	● Rem
	Es sind keine Simulationsdaten des Sondermoduls gesetzt.	Lesen Sie die spezifischen Fehlerinformationen, und überprüfen/bearbeiten Sie die Simulationsdaten des Sondermoduls.	●
	Bei einer Anweisung, in der eine andere CPUs spezifiziert wird, ist die CPU angegeben worden, in der die Anweisung aufgerufen wird.	Lesen Sie die spezifischen Fehlerinformationen, und überprüfen /korrigieren Sie das Programm.	Q-CPU ab Version B
	Bei einer Anweisung, die sich auf die CPU bezieht, in der die Anweisung aufgerufen wird, ist eine andere CPU angegeben worden.		
	Es wurde eine Anweisung verwendet, bei der kein Sondermodul, das einer anderen CPU zugeordnet ist, spezifiziert werden darf.		
	Bei einer Multi-CPU-spezifischen Anweisung wurde eine unzulässige CPU angegeben.		
	Die Platzierung des Q[]B und des QA1S[]B ist nicht korrekt.	Überprüfen Sie die Positionierung.	Q-CPU Rem
	Die CPU ist nicht auf dem CPU-Steckplatz oder den Steckplätzen 0 bis 2 installiert.	Überprüfen Sie die Positionierung der CPU.	
	Das QA1S[] ist als Hauptbaugruppenträger installiert.	Benutzen Sie das Q[]B als Hauptbaugrupenträger.	
	Ein Modul ist auf dem 65. Steckplatz oder höher installiert. Ein Modul ist auf einen Steckplatz installiert, der in der Adresszuordnung nicht mehr vorgesehen ist. Ein Modul belegt E/A-Adressen, die ausserhalb der zugelassenen 4096 E/A-Adressen liegen. Ein Modul, das als 4096. E/A-Adresse installiert ist, belegt noch weitere Adressen.	Entfernen Sie alle Module, die ab dem 65. Steckplatz installiert sind. Entfernen Sie das Modul, das auf einen Steckplatz ausserhalb des zugeteilten Bereiches installiert ist. Entfernen Sie das Modul, das ausserhalb der 4096 E/A-Adressen installiert ist. Tauschen Sie das Modul gegen ein Modul, das die 4096 E/A-Adressen nicht überschreitet.	
	Ein Modul, das von der Q-CPU nicht erkannt wird, wurde installiert. Es ist kein Zugriff auf ein Sondermodul möglich.	Benutzen Sie ein Modul, das zusammen mit der Q-CPU eingesetzt werden kann. Das Sondermodul, auf das zugegriffen wurde, hat ein Hardware-Problem. Wenden Sie sich an den MITSUBISHI-Service.	
	1.) In einem Multi-CPU-System ist zwischen den CPUs ein leerer Steckplatz vorhanden. 2.) Zwischen zwei Q-CPUs ist ein anderes Modul (z.B. Motion-CPU, E/A-Modul) installiert.	1.) Zwischen den CPU-Modulen darf kein leerer Steckplatz sein. Rechts neben den CPUs dürfen Steckplätze frei bleiben. 2.) Entfernen Sie das Modul, das zwischen den CPUs installiert ist. Eine Motion-CPU muss rechts neben den Q-CPUs installiert werden.	Q-CPU ab Version B
	Ein Sondermodul, das nicht für den Multi-CPU-Betrieb geeignet ist, wurde der CPU 2, 3, oder 4 zugeordnet.	1.) Tauschen Sie das Sondermodul gegen ein Sondermodul. das kompatibel mit dem Multi-CPU-Betrieb ist. 2.) Ordnen Sie das Modul, das nicht für den Multi-CPU-Betrieb geeignet ist, der CPU 1 zu.	

Fehlercodeliste QnA-Serie und System Q (Fortsetzung)

Fehlercode (SD0) [1]	Fehlermeldung	Allgemeine Information (SD5 bis 15) [1]	Spezifische Information (SD16 bis 26) [1]	LED-Status		CPU-Status	Diagnosezeitraum
				RUN	ERROR		
2200	MISSING PARA.	Laufwerksname	—	AUS	Blinkt	Stopp	Beim Einschalten/Beim Zurücksetzen
2210	BOOT ERROR	Laufwerksname	—	AUS	Blinkt	Stopp	Beim Einschalten/Beim Zurücksetzen
2300 2301 2302	ICM. OPE. ERROR	Laufwerksname	—	AUS/EIN	Blinkt/EIN	Stopp/ Fortsetzen [2]	Wenn die Speicherkarte eingelegt oder entfernt wurde
2400 2401	FILE SET ERROR	File-Name	Parameternummer	AUS	Blinkt	Stopp	Beim Einschalten/Beim Zurücksetzen
2410 2411 2412 2413	FILE OPE. ERROR	File-Name	Lokalisierung des Programmfehlers	AUS/EIN	Blinkt/EIN	Stopp/ Fortsetzen [2]	Bei Ausführung einer Anweisung

[1] Die Angaben in den runden Klammern kennzeichnen die Adressen der Sonderregister, in denen die spezifischen Fehlerinformationen gespeichert werden.

[2] Der Verarbeitungszustand der CPU beim Auftreten eines Fehlers kann in den Parametern gesetzt werden. Die LED-Anzeige ändert sich entsprechend.

	Ursache	Abhilfe	Gültig für:
	Auf dem über DIP-Schalter festgelegten Laufwerk gibt es keine Parameterdatei.	Überprüfen Sie die Einstellung der Parameter auf Gültigkeit der möglichen Laufwerke. Speichern Sie auf dem durch die Parameter vorbestimmten Laufwerk ein Parameter-File.	●
	Der Inhalt der Boot-Datei ist fehlerhaft.	Überprüfen Sie die Einstellung der Parameter.	Q-CPU
	Auf dem über DIP-Schalter festgelegten Laufwerk gibt es keine Boot-Datei, obwohl der BOOT-DIP-Schalter auf EIN gesetzt ist.	Überprüfen Sie die Einstellung der Parameter auf Gültigkeit der möglichen Laufwerke. Speichern Sie auf dem durch die Parameter vorbestimmten Laufwerk ein Boot-File.	QnA-CPU
	Eine Speicherkarte wurde entfernt, ohne den Schalter zur Speicherkartenfreigabe auf EIN zuschalten.	Entfernen Sie die Speicherkarte erst, nachdem der Schalter zur Speicherkartenfreigabe auf EIN geschaltet wurde.	
	1.) Die Speicherkarte wurde nicht formatiert. 2.) Der Zustand des Formates der Speicherkarte ist nicht korrekt.	1.) Formatieren Sie die Speicherkarte. 2.) Formatieren Sie die Speicherkarte erneut.	●
	Es wurde eine Speicherkarte eingesetzt, die nicht für diese CPU geeignet ist.	Überprüfen Sie die Speicherkarte.	
	Es wurde versucht, bei einer CPU Daten automatisch in das Standard-ROM zu übertragen, bei der diese Funktion nicht möglich ist (Im Boot-File ist die automatische Übertragung von einer Speicherkarte in das Standard-ROM angewählt, und als gültige Parameterquelle ist die Speicherkarte angegeben).	1.) Stellen Sie das automatische Schreiben in das Standard-ROM nur bei den CPU-Typen ein, bei denen diese Funktion unterstützt wird. 2.) Übertragen Sie mit Hilfe der Programmier-Software Parameter und Programme in das Standard-ROM. 3.) Schalten Sie das automatische Schreiben in das Standard-ROM aus und starten Sie den Boot-Vorgang mit den Daten auf der Speicherkarte.	Q-CPU ab Version B
	Die Datei, die durch die SPS-File-Einstellungen in den Parametern festgelegt wurde, kann nicht gefunden werden.	Lesen Sie die spezifischen Fehlerinformationen auf dem Display des Programmiergerätes, und überprüfen Sie die Übereinstimmung des Laufwerks- und Dateinamens mit den Parametereinstellungen, und nehmen Sie ggf. Korrekturen vor. Erzeugen Sie das vorgegebene File.	●
	Ein Ethernet-Parameter, der für eine QnA-CPU, Version „B“, gültig ist, wurde für einer QnA-CPU benutzt, die nicht der Version „B“ entspricht.	Benutzen Sie eine QnA-CPU in der Version „B“. Löschen Sie den Ethernet-Parameter.	QnA-CPU
	Beim Boot-Vorgang oder beim automatischem Schreiben in das Standard-ROM wurde die Kapazität des Programmspeichers überschritten.	– Überprüfen und korrigieren Sie ggf. die Einstellungen zum Boot-Vorgang. – Löschen Sie nicht benötigte Dateien aus dem Programmspeicher. – Stellen Sie in den Parametern „Programmspeicher löschen“ ein, um vor dem Booten den Programmspeicher zu löschen.	Q-CPU ab Version B
	Das File, das durch die SPS-RAS-Einstellung in den Parametern festgelegt ist, konnte nicht im Fehlerprotokollbereich erzeugt werden.	Lesen Sie die spezifischen Fehlerinformationen auf dem Display des Programmiergerätes, und überprüfen Sie die Übereinstimmung des Programms mit den Parametereinstellungen, und korrigieren Sie sie gegebenenfalls. Überprüfen Sie den verbleibenden freien Speicher auf Ihrer Speicherkarte.	●
	Das File, das durch das Ablaufprogramm festgelegt ist, kann nicht gefunden werden.	Lesen Sie die spezifischen Fehlerinformationen auf dem Display des Programmiergerätes, und überzeugen Sie sich davon, dass sich das in den Parametern angegebene Programm auf dem angegebenen Laufwerk befindet, und nehmen Sie ggf. Korrekturen vor. Erzeugen Sie das angegebene File.	
	Das Ablaufprogramm kann diese Art von Files (Kommentar-Files, usw.) nicht ansprechen.	Lesen Sie die spezifischen Fehlerinformationen auf dem Display des Programmiergerätes, und überzeugen Sie sich davon, dass sich das in den Parametern angegebene Programm auf dem angegebenen Laufwerk befindet, und nehmen Sie ggf. Korrekturen vor.	●
	Das in der Ablaufsprache geschriebene Programm-File kann nicht vom Ablaufprogramm angesprochen werden.		
	Es wurden keine Daten in das von dem Ablauprogramm festgelegte File geschrieben.	Lesen Sie die spezifischen Fehlerinformationen auf dem Display des Programmiergerätes, und überzeugen Sie sich davon, dass sich das in den Parametern angegebene Programm auf dem angegebenen Laufwerk befindet, und nehmen Sie ggf. Korrekturen vor. Überprüfen Sie, ob das angegebene File schreibgeschüzt ist.	

Fehlercodeliste QnA-Serie und System Q (Fortsetzung)

Fehlercode (SD0) [1]	Fehlermeldung	Allgemeine Information (SD5 bis 15) [1]	Spezifische Information (SD16 bis 26) [1]	LED-Status		CPU-Status	Diagnosezeitraum
				RUN	ERROR		
2500	CAN'T EXE. PRG.	File-Name	—	AUS	Blinkt	Stopp	Beim Einschalten/Beim Zurücksetzen
2501							
2502							
2503							
2504							
3000	PARAMETER ERROR	File-Name	Parameternummer	AUS	Blinkt	Stopp	Beim Einschalten/Zurücksetzen/ STOP → RUN
3001							
3002							
3003		File-Name/ Laufwerksnummer					Während der Ausführung der END-Anweisung
		File-Name					Beim Einschalten/Zurücksetzen/ STOP → RUN
3004							
3009	PARAMETER ERROR	File-Name/ Laufwerksnummer	Parameternummer	AUS	Blinkt	Stopp	Beim Einschalten/Zurücksetzen/ STOP → RUN
3010							
3012							
3013							

[1] Die Angaben in den runden Klammern kennzeichnen die Adressen der Sonderregister, in denen die spezifischen Fehlerinformationen gespeichert werden.

	Ursache	Abhilfe	Gültig für:
	Es existiert ein Programm-File, das Operanden benuzt, die sich außerhalb des Bereichs befinden, der in den Operanden-Parametern festgelegt ist.	Lesen Sie die spezifischen Fehlerinformationen auf dem Display des Programmiergerätes, und überzeugen Sie sich davon, dass die Parameter-Operandeneinstellungen und Programm-File-Operanden den tatsächlichen Gegebenheiten entsprechen, und nehmen Sie ggf. Korrekturen vor.	
	Es existieren Programm-Files, obwohl „Keine" in den Programm-Parametern angegeben ist.	Ändern Sie die Programm-Parameter auf "Ja". Löschen Sie nicht benötigte Programme.	●
	Das Proramm-File ist nicht korrekt. Oder die File-Inhalte sind nicht in der Ablaufsprache verfasst.	Überprüfen Sie, ob es sich um das File-Format ***.QPG handelt und ob die File-Inhalte für ein Ablaufprogramm vorgesehen sind.	
	Es existiert kein Programm-File.	Überprüfen Sie die Programmkonfiguration.	
	Es ist mehr als ein Programm in der Ablaufsprache oder Steuerungsprogramm angegeben.	Überprüfen Sie die Parameter und die Programmkonfiguration.	
	Bei den Interrupt-Pointer-Einstellungen wurde ein Sondermodul angegeben, das im Multi-CPU-System einer anderen CPU zugeordnet ist.	1.) Geben Sie die Anfangsadresse des E/A-Bereiches für ein Modul ein, das nicht einer anderen CPU zugeordnet ist. 2.) Löschen Sie die Interrupt-Pointer-Einstellungen	Q-CPU ab Version B
	Die Parametereinstellung für die Zeiteinstellung der Timer, den RUN-PAUSE-Kontakt, die allgemeine Pointer-Adresse, die Gesamtdatenverarbeitung, die Anzahl der freien Steckplätze oder die System-Interrupt-Einstellungen liegen ausserhalb des von der CPU nutzbaren Bereichs.	1.) Lesen Sie die detailierten Fehlerinformationen auf dem Display des Programmiergerätes, überprüfen Sie, ob die Eintragungen in den Parametern korrekt sind und nehmen Sie ggf. Korrekturen vor. 2.) Steht der Fehler noch an, folgen SIe den Verweisen der Parametereinstellungen. Es ist wahrscheinlich, dass ein Speicherfehler im internen CPU-RAM oder auf der Speicherkarte vorliegt.	● Rem
	Die Parameterinhalte wurden zerstört.		
	Das Parameter-File, das bei der Parametrierung unter „use the following file" für die Q-CPU angegeben wurde, existiert nicht.		●
	Der Bereich zum automatischen Datenaustausch im Multi-CPU-System überschreitet den Bereich der zur Verfügung stehenden File-Register.	Verwenden Sie einen File-Registerbereich, mit dem der Datenaustausch möglich ist	Q-CPU ab Version B
	Die in den Operanden-Parametern gesetzte Anzahl von Operanden liegt außerhalb des von der CPU nutzbaren Bereiches.	1.) Lesen Sie die detailierten Fehlerinformationen auf dem Display des Programmiergerätes, überprüfen Sie, ob die Eintragungen in den Parametern korrekt sind und nehmen Sie ggf. Korrekturen vor. 2.) Steht der Fehler noch an, folgen SIe den Verweisen der Parametereinstellungen. Es ist wahrscheinlich, dass ein Speicherfehler im internen CPU-RAM oder auf der Speicherkarte vorliegt	●
	Das Parameter-File ist nicht von der CPU verwendbar, oder die Inhalte des Files sind keine Parameter.	Überprüfen Sie, ob das Parameter-File vom Format ***.QPA ist und ob deren Inhalte auch Parameter sind.	
	In einem Multi-CPU-System ist ein Modul mehreren CPUs zugeordnet worden.	Ein Modul kann nur einer CPU zugeordnet werden. Ändern Sie die E/A-Konfiguration in jeder CPU des Multi-CPU-Systems.	Q-CPU ab Version B
	Die eingestellte Anzahl von CPU-Modulen weicht von der tatsächlich vorhandenen Anzahl ab.	Passen Sie die Anzahl der CPU-Module der Systemkonfiguration an.	
	Die Parametrierung zum Multi-CPU-System weicht in den einzelnen CPU-Modulen von der Parametrierung in CPU 1 ab.	Bringen Sie die Parameter in den einzelnen CPU-Modulen in Übereinstimmung mit den Einstellungen in CPU 1.	
	Fehlerhafte Einstellungen zum automatischen Datenaustausch in einem Multi-CPU-System: 1.) Bei Bit-Operanden wurde eine Startadresse gewählt, die nicht 0 oder eine durch 16 teilbare Zahl ist. 2.) Es ist nicht der korrekte Operand angegeben worden. 3.) Die eingestellte Anzahl der Operanden ist eine ungerade Zahl.	1.) Geben Sie als Startadresse für Bit-Operanden entweder 0 oder eine durch 16 teilbare Zahl an. 2.) Geben Sie die korrekten Operanden an. 3.) Geben Sie eine gerade Anzahl von Operanden an.	

Fehlercodeliste QnA-Serie und System Q (Fortsetzung)

Fehlercode (SD0) [1]	Fehlermeldung	Allgemeine Information (SD5 bis 15) [1]	Spezifische Information (SD16 bis 26) [1]	LED-Status		CPU-Status	Diagnosezeitraum
				RUN	ERROR		
3100	LINK PARA. ERROR	File-Name	Parameternummer	AUS	AUS	Stopp	Beim Einschalten/Zurücksetzen/ STOP → RUN
3101							
3102							
3103							
3104							
3105							

[1] Die Angaben in den runden Klammern kennzeichnen die Adressen der Sonderregister, in denen die spezifischen Fehlerinformationen gespeichert werden.

	Ursache	Abhilfe	Gültig für:
	Das MELSECNET/H-Modul mit der in den Parametern angegebenen Anfangsadresse des E/A-Bereiches ist einer anderen CPU in dem Multi-CPU-System zugeordnet.	Löschen Sie die Parameter für das MELSECNET/H-Modul, das einer anderen CPU zugeordnet ist und geben Sie die Anfangsadr. für das richtige Modul an	Q-CPU ab Version B
	Die Netzwerk-Parameter einer Normal-Station wurden in die Kontrollstation übertragen oder umgekehrt.	Setzen Sie die CPU zurück.	
	Die Anzahl der installierten Module ist unterschiedlich zu der in den Parametern für MELSECNET/10(H) vorgegebenen Anzahl. Die Anfangsadresse der installierten Module ist unterschiedlich zu der in den Parametern für MELSECNET/10(H) vorgegebenen Anfangsadresse. Es können nicht alle Daten in den Parametern gelesen werden. Der Typ der Station am MELSECNET/10 (H) ist gewechselt worden, als die Spannung eingeschaltet war (ein Übergang von RESET nach RUN ist nötig, um den geänderten Typ zu erkennen).	1.) Schreiben Sie die Netzwerkparameter nach der Korrektur erneut. 2.) Steht der Fehler auch nach der Korrektur an, wenden Sie sich an den MITSUBISHI-Service.	Q-CPU
	Die Netzwerkparameter wurden nicht geschrieben, obwohl die QnA CPU die Kontrollstation bzw. die Master-Station ist.	1.) Schreiben Sie die Netzwerkparameter nach der Korrektur erneut. 2.) Steht der Fehler auch nach der Korrektur an, wenden Sie sich an den MITSUBISHI-Service.	QnA-CPU
	1.) Bei einem MELSECNET/H-Modul mit der Stationsnummer 0 wurden Einstellungen zur Kommunikation zwischen den Steuerungen gemacht. 2.) Bei einem MELSECNET/H-Modul, das nicht die Stationsnummer 0 hat, wurden Einstellungen für einen Remote-Master gemacht.	Ändern Sie den Typ der Station oder die Stationsnummer.	Q-CPU ab Version B
	Die Netzwerknummer, die in einem Parameter vorgegeben wurde, stimmt nicht mit der des installierten Netzwerkes überein. Die Anfangsadresse der installierten E/A-Module ist unterschiedlich zu der in den Parametern vorgegebenen Anfangsadresse. Die Klasse des Netzwerkes, die in einem Parameter vorgegeben wurde, stimmt nicht mit der des tatsächlich installierten Netzwerkes überein. Es besteht ein Fehler bei den Refresh-Parametern des MELSECNET/10(H).	Bringen Sie die Parameter in Übereinstimmung mit dem installierten Netzwerk.	●
	Es ist bei der Überprüfung der Netzwerkparameter des Netzwerkmoduls ein Fehler aufgetreten.	1.) Schreiben Sie die Netzwerkparameter nach der Korrektur erneut. 2.) Steht der Fehler auch nach der Korrektur an, wenden Sie sich an den MITSUBISHI-Service.	●
	Das Ethernet-Modul mit der in den Parametern angegebenen Anfangsadresse des E/A-Bereiches ist einer anderen CPU in dem Multi-CPU-System zugeordnet.	Löschen Sie die Parameter für das Ethernet-Modul, das einer anderen CPU zugeordnet ist und geben Sie die Anfangsadr. für das richtige Modul an.	Q-CPU ab Version B
	Obwohl in den Parametern mindestens ein Ethernet-Modul vorgegeben wurde, ist kein Modul installiert. Die Anfangsadresse der installierten E/A-Module ist unterschiedlich zu der in den Ethernet-Parametern vorgegebenen Anfangsadresse.	1.) Schreiben Sie die Netzwerkparameter nach der Korrektur erneut. 2.) Steht der Fehler auch nach der Korrektur an, wenden Sie sich an den MITSUBISHI-Service	● Rem
	Das AJ71QE71 belegt nicht den in den Parametern vorgegebenen E/A-Bereich. Die E/A-Adressen sind überlappend vergeben worden. Die Vorgaben in den Parametern und die aus dem AJ71QE71 geladenen Parameter sind unterschiedlich. Die Summe der Ethernet-Parameter und die zugehörigen Anweisungen ist auf mehr als 5 eingestellt.		QnA-CPU
	Ethernet und MELSECNET/10 sind die selben Netzwerknummern zugeteilt. Die Vorgaben in den Parametern für die Netzwerknummer, die Stationsnummer oder die Gruppennummer überschreiten den zulässigen Bereich. Die vorgegebene E/A-Adresse überschreitet den für die CPU zulässigen Bereich.		● Rem
	Das CC-Link-Modul mit der in den Parametern angegebenen Anfangsadresse des E/A-Bereiches ist einer anderen CPU in dem Multi-CPU-System zugeordnet.	Löschen Sie die Parameter für das CC-Link-Modul, das einer anderen CPU zugeordnet ist und sprechen Sie das richtige Modul an.	Q-CPU ab Version B
	Obwohl in den Parametern mindestens ein CC-Link-Modul vorgegeben wurde, ist kein Modul installiert. Die in den allgemeinen Parametern vorgegebenen Anfangsadresse des E/A-Bereiches ist unterschiedlich zu der des installierten E/A-Moduls. Die Klassenzuordnung einer Station des CC-Link, die in einem Parameter vorgegeben wurde, stimmt nicht mit der der tatsächlich installierten Station überein.	1.) Schreiben Sie die Netzwerkparameter nach der Korrektur erneut. 2.) Steht der Fehler auch nach der Korrektur an, wenden Sie sich an den MITSUBISHI-Service.	● Rem
	Der Inhalt der Ethernet-Parameter ist fehlerhaft.	Schreiben Sie die Netzwerkparameter nach der Korrektur erneut.	QnA-CPU

Fehlercodeliste QnA-Serie und System Q (Fortsetzung)

Fehlercode (SD0) [1]	Fehlermeldung	Allgemeine Information (SD5 bis 15) [1]	Spezifische Information (SD16 bis 26) [1]	LED-Status		CPU-Status	Diagnosezeitraum
				RUN	ERROR		
3106	LINK PARA. ERROR	File-Name/ Laufwerksname	Parameternummer	AUS	Blinkt	Stopp	Während der Ausführung der END-Anweisung
		File-Name					Beim Einschalten/Zurücksetzen/ STOP → RUN
3107		File-Name					
3200	SFC PARA. ERROR	File-Name	Parameternummer	AUS	Blinkt	Stopp	STOP → RUN
3201							
3202							
3203							
3300	SP. PARA. ERROR	File-Name	Parameternummer	AUS	Blinkt	Stopp	Beim Einschalten/Zurücksetzen/ STOP → RUN
3301		File-Name	Parameternummer				Während der Ausführung der END-Anweisung
3302							Beim Einschalten/Zurücksetzen/ STOP → RUN
3303		File-Name/ Laufwerksname	Parameternummer				Beim Einschalten/Zurücksetzen/ STOP → RUN
3400	REMOTE PASS. ERROR	—	—	AUS	Blinkt	Stopp	Beim Einschalten/Zurücksetzen/ STOP → RUN
3401							
4000	INSTRCT CODE. ERR.	Lokalisierung des Programmfehlers	—	AUS	Blinkt	Stopp	Beim Einschalten/Zurücksetzen/ STOP → RUN
4001							
4002							
4003							
4004							

[1] Die Angaben in den runden Klammern kennzeichnen die Adressen der Sonderregister, in denen die spezifischen Fehlerinformationen gespeichert werden.

	Ursache	Abhilfe	Gültig für:
	Der Bereich zur Aktualisierung des CC-Link überschreitet den Bereich der zur Verfügung stehenden File-Register	Verwenden Sie zur Aktualisierung einen File-Registerbereich, mit dem die Aktualisierung gemäß den Einstellungen möglich ist.	Q-CPU ab Version B
	Es besteht ein Fehler bei den Refresh-Parametern des CC-Link.	Überprüfen Sie die Parametrierung.	Q-CPU/Rem
	Der Inhalt der CC-Link-Parameter ist fehlerhaft.	Überprüfen Sie die Parametrierung.	●
	Die Parameterinhalte sind nicht korrekt.	Scheiben Sie die Parameter nach der Korrektur erneut.	●
	Die Attribut-Informationen der Programmblöcke der Ablaufsprache sind falsch.		
	Die in den Parametern festgelegte Anzahl der Schrittmerker ist kleiner als die Anzahl der vom Programm verwendeten Schrittmerker.		
	Die Einstellung in den Parametern für den Verarbeitungsmodus (Execution type) ist für ein Programm in der Ablaufsprache eine andere als der Verarbeitungsmodus "Scan execution type".		
	Die durch den GX Configurator vergebene Kopfadresse für ein Sondermodul stimmt nicht mit der tatsächlichen E/A-Adresse überein.	Überprüfen Sie die Parametrierung	Q-CPU/Rem
	Die Aktualisierungseinstellung für ein Sondermodul überschreitet den Bereich der zur Verfügung stehenden File-Register.	Verwenden Sie zur Aktualisierung einen File-Registerbereich, mit dem die Aktualisierung gemäß den Einstellungen möglich ist.	Q-CPU ab Version B
	Die Einstellungen zur Aktualisierung eines Sondermoduls liegen außerhalb des zulässigen Bereiches.	Überpüfen Sie die Parametrierung	
	Die Einstellungen zur Aktualisierung eines Sondermoduls sind nicht korrekt.		● Rem
	In einem Multi-CPU-System wurden Einstellungen für ein Sondermodul gemacht, das einer anderen CPU zugeordnet ist.	Löschen Sie die Einstellungen für das Modul, das einer anderen CPU zugeordnet ist, und parametrieren Sie das Modul in dieser CPU.	●
	In der Remote-Passwortdatei ist die Kopfadresse des angesprochenen Moduls nicht 0_H oder $0FF0_H$.	Ändern die Kopfadresse des angesprochenen Moduls in 0_H oder $0FF0_H$.	Q-CPU ab Version B
	Der durch die Kopfadresse in der Remote-Passwortdatei angegebene Steckplatz ist nicht korrekt. Mögliche Ursachen: – Das Modul ist nicht installiert. – Es ist kein QJ71C24(-R2) oder kein Ethernet-Modul des System Q installiert. – Es ist ein QJ71C24(-R2) oder ein Ethernet-Modul des System Q der Version A installiert .	Installieren Sie auf dem durch die Kopfadresse in der Remote-Passwortdatei angegebenen Steckplatz ein QJ71C24(-R2), Version B oder ein Ethernet-Modul des System Q, Version B	
	Es wird ein QJ71C24(-R2), Version B oder ein Ethernet-Modul des System Q, Version B angesprochen, das einer anderen CPU zugeordnet ist.	Sprechen Sie das korrekte Modul an und löschen Sie die Einstellungen für das Remote-Password.	
	In dem Programm ist ein Anweisungscode enthalten, der nicht entschlüsselt werden kann.	Lesen Sie die allgemeinen Fehlerinformationen auf dem Display des Programmiergerätes, und überprüfen/korrigieren Sie den angegebenen Programmschritt.	●
	Das Programm enthält eine erweiterte Anweisung für Ablaufprogramm, obwohl es kein Ablaufprogramm ist.		
	Die Anweisung hat einen falschen Namen.		● Rem
	Die Anweisung spricht die falsche Operandenadresse an.		
	Die Anweisung spricht einen nicht nutzbaren Operanden an.		●

Fehlercodeliste QnA-Serie und System Q (Fortsetzung)

Fehlercode (SD0) [1]	Fehlermeldung	Allgemeine Information (SD5 bis 15) [1]	Spezifische Information (SD16 bis 26) [1]	LED-Status		CPU-Status	Diagnosezeitraum
				RUN	ERROR		
4010	MISSING END INS.	Lokalisierung des Programmfehlers	—	AUS	Blinkt	Stopp	Beim Einschalten/Zurücksetzen/ STOP → RUN
4020	CAN'T SET (P)	Lokalisierung des Programmfehlers	—	AUS	Blinkt	Stopp	Beim Einschalten/Zurücksetzen/ STOP → RUN
4021							
4030	CAN'T SET (I)	Lokalisierung des Programmfehlers	—	AUS	Blinkt	Stopp	Beim Einschalten/Zurücksetzen/ STOP → RUN
4100	OPERATION ERROR	Lokalisierung des Programmfehlers	—	AUS/EIN	Blinkt/EIN	Stopp/ Fortsetzen [2]	Bei Ausführung einer Anweisung
4101							
4102		Programm	Lokalisierung des Programmfehlers				
		Lokalisierung des Programmfehlers	—				
4103		Lokalisierung des Programmfehlers	—				
4107		Programm	Lokalisierung des Programmfehlers				
		Lokalisierung des Programmfehlers	—				
4108		Lokalisierung des Programmfehlers	—				
4200	FOR NEXT ERROR	Lokalisierung des Programmfehlers	—	AUS	Blinkt	STOPP	Bei Ausführung einer Anweisung
4201							
4202							
4203							
4210	CAN'T EXECUTE (P)	Lokalisierung des Programmfehlers	—	AUS	Blinkt	STOPP	Bei Ausführung einer Anweisung
4211							
4212							
4213							

[1] Die Angaben in den runden Klammern kennzeichnen die Adressen der Sonderregister, in denen die spezifischen Fehlerinformationen gespeichert werden.

[2] Der Verarbeitungszustand der CPU beim Auftreten eines Fehlers kann in den Parametern gesetzt werden. Die LED-Anzeige ändert sich entsprechend.

	Ursache	Abhilfe	Gültig für:
	Das Programm beinhaltet keine END-(FEND-)Anweisung	Lesen Sie die allgemeinen Fehlerinformationen auf dem Display des Programmiergerätes, und überprüfen/korrigieren Sie die angegebenen Files.	●
	Die gesamte Anzahl der internen Dateizeiger, die von dem Programm benutzt werden, überschreitet die in den Parametern gesetzte Anzahl.	Lesen Sie die allgemeinen Fehlerinformationen auf dem Display des Programmiergerätes, und überprüfen/korrigieren Sie den angegebenen Programmschritt.	●
	Die Adressen der allgemeinen Pointer, die von den entsprechenden Files genutzt werden, überlappen.		
	Die Adressen der zugeordneten Pointer, die von den entsprechenden Files genutzt werden, überlappen.		
	Die enthaltenen Daten können von der entsprechenden Anweisung nicht verarbeitet werden.	Lesen Sie die allgemeinen Fehlerinformationen auf dem Display des Programmiergerätes, und überprüfen/korrigieren Sie den angegebenen Programmschritt.	●
	Die angegebenen Adressen der Daten, die von dem Programm verarbeitet werden sollen, oder die gespeicherten Daten oder Konstanten der Operanden, die von den Anweisungen verwendet werden, liegen außerhalb des nutzbaren Adressbereichs.		
	In einem Multi-CPU-System wurde mit einer erweiterten Netzwerkanweisung ein Modul angesprochen, das einer anderen CPU zugeordnet ist.	Löschen Sie aus dem Programm die Anweisung, die den Fehler verursacht hat und greifen Sie mit der CPU auf das Modul zu, der das Modul zugeordnet ist.	Q-CPU ab Version B
	Die Netzwerk- oder Stationsnummer, die durch eine erweiterte Netzwerkanweisung angesprochen wurde, ist nicht korrekt.	Lesen Sie die allgemeinen Fehlerinformationen auf dem Display des Programmiergerätes, und überprüfen/korrigieren Sie den angegebenen Programmschritt.	● Rem
	Die Konfiguration der erweiterten PID-Anweisung ist falsch.		●
	Von einer CPU in einem Multi-CPU-System wurden mehr als 32 Multi-CPU-spezifische Anweisungen ausgeführt	Verwenden Sie zur Verriegelung den Bit-Operanden, der die Ausführung einer Anweisung anzeigt, um die gleichzeitige Ausführung von mehr als 32 Anweisungen für Multi-CPU-Systeme zu verhindern.	Q-CPU ab Version B
	Die CC-Link-Anweisung wird mehr als 64 mal ausgeführt.	Begrenzen Sie die Ausführungen der CC-Link-Anweisung auf max. 64.	QnA-CPU
	Bei Ausführung der CC-Link-Anweisung waren keine CC-Link-Parameter eingetragen.	Setzen Sie die Parameter vor Ausführung der CC-Link-Anweisung.	
	Es wird keine NEXT-Anweisung nach der FOR-Anweisung ausgeführt, oder es existieren weniger NEXT- als FOR-Anweisungen.	Lesen Sie die allgemeinen Fehlerinformationen auf dem Display des Programmiergerätes und überprüfen/korrigieren Sie den angegebenen Programmschritt.	●
	Es wird eine NEXT-Anweisung ausgeführt, obwohl keine FOR-Anweisung ausgeführt wurde, oder es existieren mehr NEXT- als FOR-Anweisungen.		
	Es sind mehr als 16 Verschachtelungsebenen (Nesting) programmiert worden.	Reduzieren Sie die Anzahl der Verschachtelungsebenen auf weniger als 17.	●
	Es wird eine BREAK-Anweisung ausgeführt, obwohl keine FOR-Anweisung ausgeführt wurde.	Lesen Sie die allgemeinen Fehlerinformationen auf dem Display des Programmiergerätes, und überprüfen/korrigieren Sie den angegebenen Programmschritt.	
	Die CALL-Anweisung wird ausgeführt, aber an dem angegebenen Pointer ist keine Unterprogrammroutine vorhanden.	Lesen Sie die allgemeinen Fehlerinformationen auf dem Display des Programmiergerätes, und überprüfen/korrigieren Sie den angegebenen Programmschritt.	●
	In dem ausgeführten Unterprogramm existiert keine RET-Anweisung.		
	Die RET-Anweisung steht vor der FEND-Anweisung im Hauptprogramm.		
	Es sind mehr als 16 Verschachtelungsebenen (Nesting) programmiert worden.	Reduzieren Sie die Anzahl der Verschachtelungsebenen auf weniger als 17.	

Fehlercodeliste QnA-Serie und System Q (Fortsetzung)

Fehlercode (SD0) [1]	Fehlermeldung	Allgemeine Information (SD5 bis 15) [1]	Spezifische Information (SD16 bis 26) [1]	LED-Status		CPU-Status	Diagnosezeitraum
				RUN	ERROR		
4220	CAN'T EXECUTE (I)	Lokalisierung des Programmfehlers	—	AUS	Blinkt	STOPP	Bei Ausführung einer Anweisung
4221							
4223							
4230	INST. FORMAT ERR	Lokalisierung des Programmfehlers	—	AUS	Blinkt	STOPP	Bei Ausführung einer Anweisung
4231							
4235							
4300	EXTEND INST. ERR.	Lokalisierung des Programmfehlers	—	AUS/EIN	Blinkt/EIN	STOPP/ Fortsetzen [2]	Bei Ausführung einer Anweisung
4301							
4400	SFCP. CODE ERROR	Lokalisierung des Programmfehlers	—	AUS	Blinkt	STOPP	STOP → RUN
4410	CAN'T SET (BL)	Lokalisierung des Programmfehlers	—	AUS	Blinkt	STOPP	STOP → RUN
4411							
4420	CAN'T SET (S)	Lokalisierung des Programmfehlers	—	AUS	Blinkt	STOPP	STOP → RUN
4421							
4422							
4500	SFCP. FORMAT ERR.	Lokalisierung des Programmfehlers	—	AUS	Blinkt	STOPP	STOP → RUN
4501							
4502							
4503							
4504							
4600	SFCP. OPE. ERROR	Lokalisierung des Programmfehlers	—	AUS/EIN	Blinkt/EIN	STOPP/ Fortsetzen [2]	Bei Ausführung einer Anweisung
4601							
4602							
4610	SFCP. EXE. ERROR	Lokalisierung des Programmfehlers	—	EIN	EIN	Fortsetzen	STOP → RUN
4611							

[1] Die Angaben in den runden Klammern kennzeichnen die Adressen der Sonderregister, in denen die spezifischen Fehlerinformationen gespeichert werden.

[2] Der Verarbeitungszustand der CPU beim Auftreten eines Fehlers kann in den Parametern gesetzt werden. Die LED-Anzeige ändert sich entsprechend.

	Ursache	Abhilfe	Gültig für:
	Es wurde eine Interrupt-Eingabe gemacht, aber kein entsprechender Interrupt-Pointer gefunden.	Lesen Sie die allgemeinen Fehlerinformationen auf dem Display des Programmiergerätes, und überprüfen/korrigieren Sie den angegebenen Programmschritt.	●
	In dem ausgeführten Interrupt-Programm existiert keine RET-Anweisung.		
	Die IRET-Anweisung befindet sich im Hauptprogramm vor der FEND-Anweisung.		
	Die CHKEND-Anweisung wird nicht nach der CHKCIR-Anweisung ausgeführt. Es existiert die gleiche Anzahl von CHK- und CHKEND-Anweisungen.	Lesen Sie die allgemeinen Fehlerinformationen auf dem Display des Programmiergerätes, und überprüfen/korrigieren Sie den angegebenen Programmschritt.	●
	Die IX- und die IXEND-Anweisung sind nicht im Zusammenhang programmiert. Es existiert die gleiche Anzahl von IX- und IXEND-Anweisungen.		
	Die Kontrollbedingungen der CHK-Anweisung sind ungültig, oder die CHK-Anweisung wird in einem Low-Speed-Programm verwendet.		
	Die Angabe der MELSECNET/MINI-S3-Master-Modul-Steueranweisung ist falsch.	Lesen Sie die allgemeinen Fehlerinformationen auf dem Display des Programmiergerätes, und überprüfen/korrigieren Sie den angegebenen Programmschritt.	QnA-CPU
	Die Angabe der AD57/AD58-Steueranweisung ist falsch.		
	Es existiert keine SFCP- oder SFCPEND-Anweisung in dem Programm der Ablaufsprache.	Lesen Sie die allgemeinen Fehlerinformationen auf dem Display des Programmiergerätes, und überprüfen/korrigieren Sie den angegebenen Programmschritt.	●
	Die von dem Programm der Ablaufsprache verwendeten Blockadressen liegen außerhalb des Adressbereichs.		●
	Die Adressen der Blöcke innerhalb des Programms der Ablaufsprache überlappen.		
	Die Schrittnummer innerhalb des Programms der Ablaufsprache übersteigt die Schrittnummer 255.		
	Die Anzahl aller Schritte innerhalb aller Programme der Ablaufsprache übersteigt den zulässigen Wert.	Reduzieren Sie die Anzahl der Schritte.	●
	Die Numerierung der Schritte innerhalb des Programms der Ablaufsprache überlappt.	Lesen Sie die allgemeinen Fehlerinformationen auf dem Display des Programmiergerätes, und überprüfen/korrigieren Sie den angegebenen Programmschritt.	
	Die Anzahl der BLOCK- und BEND-Anweisungen innerhalb des Programms der Ablaufsprache sind nicht im Verhältnis 1 zu 1.		●
	Der Aufbau der STEP*- zu TRAN*- zu TSET- zu SEND-Anweisungen innerhalb des Programms der Ablaufsprache ist fehlerhaft.		
	Es existiert keine STEPI*-Anweisung innerhalb des Programmblocks der Ablaufsprache.		
	Der Schritt, der durch die TSET-Anweisung innerhalb des Programms der Ablaufsprache angesprochen wird, existiert nicht.	Lesen Sie die allgemeinen Fehlerinformationen auf dem Display des Programmiergerätes, und überprüfen/korrigieren Sie den angegebenen Programmschritt.	
	Der Schritt, der durch die TAND-Anweisung innerhalb des Programms der Ablaufsprache angesprochen wird, existiert nicht.		
	Das Programm der Ablaufsprache enthält Daten, die nicht verarbeitet werden können.	Lesen Sie die allgemeinen Fehlerinformationen auf dem Display des Programmiergerätes, und überprüfen/korrigieren Sie den angegebenen Programmschritt. Das Programm startet beim Initialisierungsschritt.	●
	Der im Programm der Ablaufsprache festgelegte Operandenbereich wird überschritten.		
	In der Schrittfolge des Programms der Ablaufsprache geht die END-Anweisung der START-Anweisung voraus.		
	Die Information des aktiven Schrittes zur Wiederaufnahme der Verarbeitung des Programms der Ablaufsprache sind falsch.	Lesen Sie die allgemeinen Fehlerinformationen auf dem Display des Programmiergerätes, und überprüfen/korrigieren Sie den angegebenen Programmschritt.	●
	Der Schlüsselschalter wurde während der Wiederaufnahme der Verarbeitung der Programme der Ablaufsprache von RUN auf RESET geschaltet.		

Fehlercodeliste QnA-Serie und System Q (Fortsetzung)

Fehlercode (SD0) [1]	Fehlermeldung	Allgemeine Information (SD5 bis 15) [1]	Spezifische Information (SD16 bis 26) [1]	LED-Status		CPU-Status	Diagnosezeitraum
				RUN	ERROR		
4620	BLOCK EXE. ERROR	Lokalisierung des Programmfehlers	—	AUS	Blinkt	STOPP	Bei Ausführung einer Anweisung
4621							
4630	STEP EXE. ERROR	Lokalisierung des Programmfehlers	—	AUS	Blinkt	STOPP	Bei Ausführung einer Anweisung
4631							
4632							
4633							
5000	WDT ERROR	Zeit (Einstellwert)	Zeit (tatsächlich gemessener Wert)	AUS	Blinkt	STOPP	kontinuierlich
5001							
5010	PRG. TIME OVER	Zeit (Einstellwert)	Zeit (tatsächlich gemessener Wert)	EIN	EIN	Fortsetzen	kontinuierlich
5011							
6000	PRG. VERIFY ERR. [2]	File-Name	—	AUS	Blinkt	Stopp	kontinuierlich
6010	MODF VERIFY ERR. [2]	—	—	EIN	EIN	Fortsetzen	kontinuierlich
6100	TRK. MEMORY ERR. [3]	—	—	EIN	EIN	Fortsetzen	Beim Einschalten/Zurücksetzen/ STOP → RUN
6101							Bei Ausführung der END-Anweisung
6200	CONTROL EXE. [4]	Grund der Umschaltung	—	EIN	AUS	Fortsetzen	kontinuierlich
6210	CONTROL WAIT. [2]	Grund der Umschaltung	—	EIN	AUS	Fortsetzen	kontinuierlich
6220	CAN'T EXE CHANGE [4]	Grund der Umschaltung	—	EIN	EIN	Fortsetzen	kontinuierlich
6221							
6222							

[1] Die Angaben in den runden Klammern kennzeichnen die Adressen der Sonderregister, in denen die spezifischen Fehlerinformationen gespeichert werden.

[2] Dieser Fehler kann nur im Reservesystem eines redudanten Systemen erkannt werden.

[3] Dieser Fehler kann nur in redudanten Systemen gemeldet werden. Die Fehlererkennung ist für das aktive System und das Reservesystem möglich.

[4] Dieser Fehler kann nur im aktiven Teil eines redudanten Systemen erkannt werden..

	Ursache	Abhilfe	Gültig für:
	Es wurde versucht, einen bereits gestarteten Programmblock der Ablaufsprache erneut zu starten.	Lesen Sie die allgemeinen Fehlerinformationen auf dem Display des Programmiergerätes, und überprüfen/korrigieren Sie den angegebenen Programmschritt.	●
	Es wurde versucht, einen Programmblock der Ablaufsprache zu starten, der nicht existiert.		
	Es wurde versucht, einen bereits gestarteten Programmblock der Ablaufsprache erneut zu starten.		●
	Es wurde versucht einen Programmblock der Ablaufsprache zu starten, der nicht existiert.		
	In den Programmblöcken eines Programms der Ablaufsprache sind zu viele Schritte gleichzeitig aktiv .		
	In den Programmblöcken aller Programme der Ablaufsprache sind zu viele Schritte gleichzeitig aktiv .		
	Die Programmzykluszeit eines Programms mit dem Verarbeitungsmodus „Initial execution type“ übersteigt die in dem PC RAS-Parameter eingestellte Zeit des „Watch Dog Timers“ zur Überwachung von Programmen dieses Typs.	Lesen Sie die spezifischen Fehlerinformationen auf dem Display des Programmiergerätes, und überprüfen/korrigieren (verkürzen) Sie die eingestellte Zykluszeit.	●
	Die Programmzykluszeit übersteigt die in dem PC RAS Parameter eingestellte Zeit der Fehlerüberwachung des „Watch Dog Timers“.		
	Die Programmzykluszeit eines Programmes mit dem Verarbeitungsmodus „Low speed scan“ übersteigt die in dem PC RAS-Parameter eingestellte konstante Zykluszeit.	Überprüfen und ändern Sie die konstante Zykluszeit oder die Ausführungszeit eines Programmes mit dem Verarbeitungsmodus „low speed scan“.	●
	Die Programmzykluszeit eines Programms mit dem Verarbeitungsmodus „Low speed scan type“ übersteigt die in dem PC RAS-Parameter eingestellte Zeit des „Watch Dog Timers“ zur Überwachung von Programmen dieses Typs.	Lesen Sie die spezifischen Fehlerinformationen auf dem Display des Programmiergerätes, und überprüfen/korrigieren (verkürzen) Sie die eingestellte Zykluszeit.	
	Das aktive System und das Reservesystem eines redundanten Aufbaues haben nicht dieselben Programme und Parameter.	Passen Sie sie Programme und Parameter beider Systeme an.	Q4AR-CPU
	Das aktive System und das Reservesystem eines redundanten Systemes haben nicht dieselben Betriebsarten.	Betreiben Sie beide Systeme in der selben Betriebsart.	
	Während des Anlaufes der Steuerung wurde ein Fehler im Tracking-Speicher der CPU entdeckt.	Dies ist ein Hardware-Fehler der CPU. Wenden Sie sich in diesem Fall an den MITSUBISHI-Service. Tauschen Sie erst die CPU des Reservesystems, dann die CPU des aktiven Systemes.	
	Während des Handshake beim Tracking hat die CPU einen Fehler festgestellt.	Prüfen Sie den Zustand der anderen Stationen.	
	Das Reservesystem einer redundanten Steuerung wurde als aktives System geschaltet.	Überprüfen SIe den Zustand des aktiven Systemes.	
	Das aktive System einer redundanten Steuerung wurde als Reservesystem geschaltet.		
	Das Reservesystem konnte nicht vom Zustand des aktiven Systemes in den Zustand des Reservesystemes geschaltet werden.	Überprüfen SIe den Zustand des Reservesystemes.	
	Die Umschaltung ist wegen eines fehlerhaften Bus-Umschaltmoduls gesperrt	Dies ist ein Hardware-Fehler des Umschaltmoduls. Wenden Sie sich in diesem Fall an den MITSUBISHI-Service.	
	Die Umschaltung ist gesperrt, weil die Master-Station einer dezentralen Peripherie, die im Multiplexbetrieb arbeitet, im Reservesystem installiert wurde.	Überprüfen Sie die Einstellungen der dezentralen Peripherie.	

Fehlercodeliste QnA-Serie und System Q (Fortsetzung)

Fehlercode (SD0) [1]	Fehlermeldung	Allgemeine Information (SD5 bis 15) [1]	Spezifische Information (SD16 bis 26) [1]	LED-Status		CPU-Status	Diagnosezeitraum
				RUN	ERROR		
7000	MULT CPU DOWN	Stationsnummer Modulnummer	—	AUS	Blinkt	Stopp	kontinuierlich
7002							Beim Einschalten/Beim Zurücksetzen
7003							Beim Einschalten/Beim Zurücksetzen
7010	MULTI EXE. ERROR	Stationsnummer Modulnummer	—	AUS	Blinkt	Stopp	Beim Einschalten/Beim Zurücksetzen
7020	MULTI CPU ERROR		—	EIN	EIN	Fortsetzen	kontinuierlich
9000	F**** [2]	Lokalisierung des Programmfehlers	Nr. des Fehlermerkers	EIN	AUS	Fortsetzen	Bei Ausführung einer Anweisung
				USER LED EIN			
9010	<CHK> ERR ***-*** [3]	Lokalisierung des Programmfehlers	Fehlernr.	EIN	AUS	Fortsetzen	Bei Ausführung einer Anweisung
				USER LED EIN			
9020	BOOT OK	—	—	AUS	Blinkt	Stopp	Beim Einschalten/Zurücksetzen
10000	CONT.UNIT ERROR	—	—	—	—	—	—

[1] Die Angaben in den runden Klammern kennzeichnen die Adressen der Sonderregister, in denen die spezifischen Fehlerinformationen gespeichert werden.

[2] **** weisen auf die erkannte Nummer des Fehlermerkers hin.

[3] *** weisen auf Kontakt und Prüfnetzwerknummer hin.

	Ursache	Abhilfe	Gültig für:
	1.) In einem CPU-Modul, bei dem eingestellt ist, dass alle CPUs des Multi-CPU-Systems bei einem Fehler dieser CPU Fehler gestoppt werden, ist ein Fehler aufgetreten. 2.) In einem Multi-CPU-System wurde eine Q-CPU der Version A installiert.	1.) Setzen Sie die Fehlersuche bei dem entsprechenden CPU-Modul fort. 2.) In einem Multi-CPU-System können nur Q-CPUs ab Version B eingesetzt werden.	Q-CPU ab Version B
	Die CPU 1 in einem Multi-CPU-System wurde beim Einschalten der Spannung durch einen Fehler gestoppt. Dadurch können die anderen CPUs nicht anlaufen. Diese Fehlermeldung wird bei CPU 2, 3 und 4 gemeldet.	Setzen Sie die Fehlersuche bei CPU 1 fort.	
	1.) In einem Multi-CPU-System kam beim Aufbau einer Kommunikationsverbindung keine Reaktion von der Ziel-CPU. 2.) In einem Multi-CPU-System wurde eine Q-CPU der Version A installiert.	1.) Setzen Sie die CPU zurück. Wenn danach der Fehler wieder auftritt, liegt wahrscheinlich ein Hardware-Fehler bei einer CPU vor. Wenden Sie sich in diesem Fall an den MITSUBISHI-Service. 2.) In einem Multi-CPU-System können nur Q-CPUs ab Version B eingesetzt werden.	
	In einem Multi-CPU-System kam beim Aufbau einer Kommunikationsverbindung keine Reaktion von der Ziel-CPU.	Setzen Sie die CPU zurück. Wenn danach der Fehler wieder auftritt, liegt wahrscheinlich ein Hardware-Fehler bei einer CPU vor. Wenden Sie sich in diesem Fall an den MITSUBISHI-Service.	
	1.) In einem Multi-CPU-System ist eine CPU defekt. 2.) In einem Multi-CPU-System wurde eine Q-CPU der Version A installiert (Dieser Fehler wird bei den anderen CPUs (Q-CPU ab Version B) gemeldet). 3.)CPU2, 3 oder 4 wurde zurückgesetzt (Diese Fehlermeldung erscheint bei der CPU, die zurückgesetzt wurde).	1.) Werten Sie die Fehlerinformation aus. Tauschen Sie die defekte CPU. 2.) In einem Multi-CPU-System können nur Q-CPUs ab Version B eingesetzt werden. Tauschen Sie die CPU (Ver. A) gegen eine CPU (ab Ver. B). 3.) Setzen Sie CPU 1 zurück, um das gesamte Multi-CPU-System zurückzusetzen.	
	In einem CPU-Modul, bei dem eingestellt ist, dass alle CPUs des Multi-CPU-Systems bei einem Fehler in dieser CPU gestoppt werden, ist ein Fehler aufgetreten (Diese Fehlermeldung erscheint bei den CPUs, die durch diese CPU gestoppt wurden).	Setzen Sie die Fehlersuche bei dem entsprechenden CPU-Modul fort.	
	Der Fehlermerker F wurde auf EIN gesetzt.	Lesen Sie die spezifischen Fehlerinformationen auf dem Display des Programmiergerätes, und überprüfen/korrigieren Sie das Programm mit Hilfe der eingetragenen Fehlermerkernummer.	●
	Ein Fehler wurde mittels der CHK-Anweisung festgestellt.		
	Die Speicherung von Daten auf das Standard-ROM wurde fehlerfrei abgeschlossen. Die BOOT-LED blinkt ebenfalls.	Wählen Sie in den Parametern das Standard-ROM als Speicherort für die Boot-Daten, und schalten Sie die Spannungsversorgung aus und wieder ein, um die Boot-Daten vom Standard-ROM zu laden.	Q-CPU ab Version B
	Im Multi-CPU-System ist bei einer anderen als einer Q-CPU (z. B. Motion-CPU) ein Fehler aufgetreten	Setzen Sie die Fehlersuche bei dem entsprechenden CPU-Modul fort.	

13.2 Liste der Fehlercodes A-Serie (außer AnA und AnAS)

Die folgende Tabelle enthält eine Übersicht der möglichen Fehlercodes zusammen mit Fehlermeldungen, möglichen Ursachen und Hinweise zur Behebung des Fehlers. Fehlercodes werden in das Sonderregister D9008 und die zugehörige Schrittnummern, an denen der Fehler aufgetreten ist, in die Sonderregister D9010 und D9011 geschrieben. In dieser Tabelle sind nur Fehlermeldungen der AnN-, AnU-, AnS-, A3M- und A2C-CPUs aufgeführt.

Fehlermeldung	Inhalt von D9008	CPU-Status	Ursache	Abhilfe
INSTRCT. CODE ERR Anweisung fehlerhaft (die Überprüfung erfolgt während der Programmausführung)	10	STOP	Im Programm ist eine Anweisung enthalten, die nicht von der CPU verarbeitet werden kann. Ein EPROM-Chip mit fehlerhaftem Programm wurde eingesetzt. Der Speicherinhalt wurde verändert. Eine PR- oder IRET-Anweisung wurde programmiert.	Fehlerhaften Programmschritt mit Hilfe eines Programmiergerätes lesen und Programmzeile korrigieren. Bei fehlerhaftem Programm im EPROM ist das EPROM-Programm entsprechend zu ändern oder der Chip auszutauschen.
PARAMETER ERROR Parameter fehlerhaft (die Überprüfung erfolgt beim Einschalten, einem Reset, dem Umschalten von STOP auf RUN oder von PAUSE auf RUN)	11	STOP	Für die Speicherkapazität der CPU wurde ein zu großer Wert festgelegt. Parameter der CPU wurden falsch gesetzt oder haben sich aufgrund von Störeinflüssen geändert (Parameterspeicher ist gelöscht). Der RAM-Speicher ist nicht eingesetzt (bei A1 oder A1N-CPUs).	Überprüfen, ob der Speicher-Chip richtig im Sockel sitzt. Parameterbereich mit Hilfe eines Programmiergerätes überprüfen, korrigieren und ggf. erneut in die CPU schreiben.
MISSING END INS. END-Anweisung fehlt (die Überprüfung erfolgt nach dem Setzen von M9056 oder M9057 oder dem Umschalten von STOP auf RUN bzw. von PAUSE auf RUN)	12	STOP	Das Hauptprogramm enthält keine END-Anweisung. Ein Unterprogramm enthält keine END-Anweisung (wenn ein Unterprogramm über Parameter definiert ist).	END-Anweisung an das Ende des Programms/ Unterprogramms setzen.
CAN'T EXECUTE (P) Programmsprung ist nicht ausführbar (die Überprüfung erfolgt bei Ausführung einer der folgenden Anweisungen: CJ, SCJ, JMP, CALLP, FOR/NEXT sowie nach dem Umschalten von STOP auf RUN, bzw. von PAUSE auf RUN)	13	STOP	Das Sprungziel, das über eine der Anweisungen CJ, SCJ, CALL, CALLP oder JMP angesprochen wurde, ist fehlerhaft oder nicht vorhanden. Das Programm enthält eine CHG-Anweisung, aber kein Unterprogramm. Eine RET-Anweisung wurde programmiert (und ausgeführt), obwohl das Programm keine CALL-Anweisung enthält. Das Sprungziel, das über eine der Anweisungen CJ, SCJ, CALL, CALLP oder JMP angesprochen wurde, befindet sich hinter einer END-Anweisung. Die Anzahl von FOR-Anweisungen entspricht nicht der Anzahl von NEXT-Anweisungen. Das Sprungziel einer zwischen FOR und NEXT programmierten JMP-Anweisung liegt außerhalb der FOR-NEXT-Schleife. Das in einer JMP-Anweisung vorgegebene Sprungziel liegt nicht innerhalb der Unterroutine vor Ausführung einer RET-Anweisung. Das Sprungziel einer JMP-Anweisung liegt an einem Programmschritt oder in einem Unterprogramm, der bzw. das zwischen einer FOR-NEXT-Schleife liegt. Eine STOP-Anweisung wurde innerhalb einer Interrupt- oder Unterprogramm-Routine oder innerhalb einer FOR-NEXT-Schleife ausgeführt.	Fehlerhaften Programmschritt mit Hilfe eines Programmiergerätes lesen und Programmzeile korrigieren (als korrigierende Maßnahme muss beispielsweise eine Sprunganweisung eingefügt oder das Sprungziel geändert werden).

Fehlercodeliste der A-Serie (außer AnA und AnAS (Fortsetzung))

Fehlermeldung	Inhalt von D9008	CPU-Status	Ursache	Abhilfe
CHK FORMAT ERR Fehler im CHK-Format (die Überprüfung erfolgt nach dem Umschalten von STOP auf RUN, bzw. von PAUSE auf RUN)	14	STOP	In einem CHK-Anweisungsblock befinden sich andere Anweisungen (NOP eingeschlossen) als LDX, LDIX, ANDX und ANIX. Das Programm enthält mehr als eine CHK-Anweisung. Der CHK-Block enthält mehr als 150 Eingangskontakte (Kontrollbedingungen). Die Adresse einer Eingangsanweisung X im CHK-Block liegt über dem Maximalwert. Oberhalb des CHK-Anweisungsblocks befindet sich keine CJ-Anweisung mit Eingangsbedingung. Die Operandenadresse von D1 in der Anweisung CHK D1 D2 entspricht nicht der Operandenadresse oberhalb der CJ-Anweisung. Der Pointer P254 ist nicht dem Beginn des CHK-Anweisungsblocks zugeordnet.	CHK-Anweisungsblock im Programm auf die möglichen Fehlerursachen entsprechend der nebenstehenden Punkte überprüfen und korrigieren.
CAN'T EXECUTE (1) Ausführung nicht möglich (die Überprüfung erfolgt bei Ausführung eines Interrupts)	15	STOP	Die Adresse des Interrupt-Pointers (I), mit dem das Interrupt-Modul angesprochen werden soll, ist fehlerhaft oder mehrfach im Programm vorhanden. Im Interrupt-Programm befindet sich keine IRET-Anweisung. Die IRET-Anweisung befindet sich in einem anderen Programmteil als dem Interrupt-Programm.	Vorkommen und Zuordnung von Interrupt-Programm und Interrupt-Modul überprüfen und doppelt programmierte Interrupt-Pointer ggf. löschen. Interrupt-Programmteil auf fehlende IRET-Anweisung untersuchen und Anweisung ggf. einfügen. Überprüfen, ob sich eine IRET-Anweisung außerhalb des Interrupt-Programmteils befindet, und entsprechende Programmstelle löschen.
CASSETTE ERROR Zugriff auf die Speicherkassette ist nicht möglich	16	STOP	Die Speicherkassette befindet sich nicht in der CPU oder ist defekt.	Speicherkassette ersetzen bzw. bei abgeschalteter Versorgungsspannung einsetzen.
ROM-ERROR EPROM-Fehler (die Überprüfung erfolgt nach dem Einschalten und einem Reset)	17	STOP	Parameter und Ablaufprogramm sind nicht korrekt im eingesetzten EPROM gespeichert. Das EPROM ist defekt.	Programm und Parameter erneut in das EPROM übertragen oder EPROM bei Defekt austauschen.
MEMORY PROTECT ERROR Schreibschutzfehler (die Überprüfung erfolgt nach dem Einschalten und einem Reset)	18	STOP	Der Schreibschutz ist während des Zugriffs der CPU auf das im EPROM gespeicherte Programm eingeschaltet (DIP-Schalter auf ON-Position).	Schreibschutzschalter (DIP-Schalter bei einigen CPUs) auf die OFF-Position schalten.
RAM ERROR Fehler beim RAM-Speicherzugriff (die Überprüfung erfolgt nach dem Einschalten, einem Reset und dem Setzen von M9084 während STOP)	20	STOP	Die CPU überprüft selbständig die Schreib- und Lesevorgänge zum und vom Datenspeicher auf korrekte Ausführung. Beim Zugriff ist ein Fehler aufgetreten.	Mitsubishi-Serviceabteilung informieren.
OPE. CIRCUIT ERR Fehler im Betriebsschaltkreis (die Überprüfung erfolgt nach dem Einschalten und einem Reset)	21	STOP	Der Betriebsschaltkreis, der für die Ablaufverarbeitung in der CPU verantwortlich ist, arbeitet fehlerhaft.	Mitsubishi-Serviceabteilung informieren.

Fehlercodeliste der A-Serie (außer AnA und AnAS (Fortsetzung))

Fehlermeldung	Inhalt von D9008	CPU-Status	Ursache	Abhilfe
WDT ERROR (1) WDT-Fehler 1 (die Überprüfung erfolgt nach Ausführung einer END-Anweisung)	22	STOP	Die Programmzykluszeit übersteigt die Zeit der Fehlerüberwachung des Watch Dog Timers. Die Zykluszeit des Programms hat sich stark vergrößert. Die Zykluszeit hat aufgrund eines kurzzeitigen Spannungsabfalls, der während eines Programmdurchlaufs aufgetreten ist, zugenommen.	Zykluszeit des Programms überprüfen und neu berechnen. Die Programmzykluszeit kann mit Hilfe von CJ-Anweisungen usw. entsprechend reduziert werden. Liegt der Fehler nicht in der Programmstruktur, ist der Inhalt des Sonderregisters D9005 mit Hilfe eines Programmiergerätes anzuzeigen. Beträgt der Wert nicht 0, liegt eine ungenügende Versorgungsspannung vor. In diesem Fall ist die Spannungsversorgung zu überprüfen und die Ursache für den Spannungsverlust zu beheben.
SUB-CPU ERROR Fehler einer untergeordneten CPU	23	STOP	Die untergeordnete CPU ist defekt oder verriegelt.	Mitsubishi-Serviceabteilung informieren.
END NOT EXECUTE END-Anweisung konnte nicht ausgeführt werden (die Überprüfung erfolgt während der Ausführung der END-Anweisung)	24	STOP	Während der Ausführung einer END-Anweisung wird ein anderer Anweisungscode aufgrund von Störeinflüssen (induzierte Spannungen etc.) gelesen. Die END-Anweisung hat sich aufgrund von Störeinflüssen in eine andere Anweisung geändert.	Reset ausführen und CPU erneut in den RUN-Betrieb setzen. Tritt der gleiche Fehler erneut auf, liegt möglicherweise ein Hardware-Fehler vor. In diesem Fall ist der Mitsubishi-Service zu informieren.
WDT ERROR (2) WDT-Fehler 2 (ständige Überwachung)	25	STOP	Die Programmverarbeitung der CPU befindet sich in einer Endlosschleife (A3-, A1N-, A2N(S1)-, A3N-, A3H-).	CPU auf STOP setzen und mit Hilfe des Schlüsselschalters einen Reset durchführen. Anschließend das Programm auf die korrekte Programmierung der JMP-, CJ- und SCJ-Anweisung sowie des Pointers (P) überprüfen.
MAIN CPU DOWN (1) Fehlfunktion der Master-CPU (ständige Überwachung)	26	STOP	Fehlfunktion oder Defekt der Haupt-CPU.	Mitsubishi-Serviceabteilung informieren.
UNIT VERIFY ERR. Kommunikationsfehler (ständige Überwachung)	31	RUN (STOP)	Die Daten eines E-/A-Moduls haben sich nach dem Einschalten verändert. Das E-/A-Modul (einschließlich Sondermodul) befindet sich nicht in seinem ursprünglichen Steckplatz oder ein anderes Modul wurde anstelle dessen eingesetzt.	Sonderregister D9116 bis D9123 zur Eingrenzung des Moduls mit Hilfe eines Programmiergerätes lesen. Das zugehörige Bit des Moduls, bei dem der Vergleichsfehler aufgetreten ist, ist auf „1““ gesetzt. Ist die ursprüngliche Modulanordnung wieder hergestellt, muss das System über den RESET-Schalter zurückgesetzt werden.
FUSE BREAK OFF Sicherung defekt (ständige Überwachung)	32	RUN (STOP)	Die Sicherung eines Ausgangsmoduls ist defekt.	Sonderrregister D9100 bis D9107 zur Eingrenzung des Moduls mit Hilfe eines Programmiergerätes lesen. Das zugehörige Bit des Moduls, bei dem die Sicherung defekt ist, ist auf „1““ gesetzt.
CONTROL BUS ERR. Störung am Systembus (die Überprüfung erfolgt bei Ausführung einer FROM-/TO-Anweisung)	40	STOP	FROM- und/oder TO-Anweisungen können nicht ausgeführt werden. Während der Übertragung zu einem Sondermodul ist am Systembus eine Störung aufgetreten.	Der Fehler kann durch ein defektes Sondermodul, CPU-Modul oder den Systembus des Baugruppenträgers hervorgerufen worden sein. Baugruppenträger und Module sollten daher eingehend geprüft und ggf. der Mitsubishi-Service verständigt werden.
SP.UNIT DOWN Fehlfunktion eines Sondermoduls (die Überprüfung erfolgt bei Ausführung einer FROM-/TO-Anweisung)	41	STOP	Der Zugriff auf ein Sondermodul nach Ausführung einer FROM- und/oder TO-Anweisung erfolgte ohne Reaktion des Sondermoduls. Das angesprochene Sondermodul weist eine Fehlfunktion auf oder ist defekt.	Mitsubishi-Serviceabteilung informieren.

Fehlercodeliste der A-Serie (außer AnA und AnAS (Fortsetzung))

Fehlermeldung	Inhalt von D9008	CPU-Status	Ursache	Abhilfe
LINK UNIT ERROR Interface-Modul falsch plaziert	42	STOP	Ein Interface-Modul für das Data-Link-Netzwerk (z.B. AJ71R22, AJ71P22) befindet sich im Baugruppenträger der Master-Station.	Modul aus dem Baugruppenträger der Master-Station entfernen, System über den RESET-Schalter zurücksetzen und Initialisierung wiederholen.
I/O INT. ERROR Interrupt-Fehler	43	STOP	Ein Interrupt wurde ausgeführt, obwohl sich im System kein Interrupt-Modul befindet.	Mitsubishi-Serviceabteilung informieren.
SP.UNIT LAY ERROR Fehler in Verbindung mit einem Compter-Link-Modul	44	STOP	Im System befinden sich bezogen auf eine CPU – mehr als 2 Computer-Link-Module (z.B. AJ71C21 oder AJ71C24). Auf dem Baugruppenträger befindet sich mehr als ein Interface-Modul für das Data-Link-Netzwerk (z.B. AJ71R22, AJ71P22). Auf dem Baugruppenträger befindet sich mehr als ein Interrupt-Modul. Ein Sondermodul befindet sich auf dem Steckplatz, der in den Parametern einem E-/A-Modul zugeordnet ist (bzw. umgekehrt).	Entsprechende Module, die sich doppelt im System oder auf einem Baugruppenträger befinden, entfernen. Bei falscher Adressenzuordnung eines Steckplatzes ist dieser über ein Programmiergerät neu zu konfigurieren.
SP. UNIT ERROR Einstellwerte fehlerhaft (die Überprüfung erfolgt bei Ausführung einer FROM-/TO-Anweisung)	46	STOP	Über eine FROM-/TO-Anweisung wird eine Station im MELSECNET/MINI angesprochen, die in der Initialisierung nicht definiert ist. Über eine FROM-TO-Anweisung wird ein Sondermodul im MELSECNET angesprochen, das an der angegeben Adersse nicht vorhanden ist.	Fehlerhaften Programmschritt mit Hilfe eines Programmiergerätes lesen und Programmzeile mit der FROM-/TO-Anweisung entsprechend korrigieren.
LINK PARA. ERROR Fehler in den Parametern zur Datenkommunikation	47	Betrieb wird fortgesetzt	Die Parameter zur Festlegung des Übertragungsbereiches für die Datenkommunikation im Data-Link-Netzwerk sind fehlerhaft oder nicht vorhanden. Die Einstellung der Summe der im Netzwerk befindlichen lokalen Stationen lautet „0"".	Parameter überprüfen und korrigieren oder ggf. neu eingeben. Zusätzlich ist die Einstellung der Stationsnummern zu überprüfen. Tritt der Fehler wiederholt auf, liegt ein Hardware-Fehler vor. In diesem Fall ist der Mitsubishi-Service zu informieren.
OPERATION ERROR Ausführungsfehler (die Überprüfung erfolgt bei Ausführung einer Anweisung)	50	Betrieb wird fortgesetzt	Das Ergebnis einer BCD-Umrechnung überschreitet den erlaubten Bereich (9999 oder 99999999). Die programmierte Operandenadresse liegt außerhalb des erlaubten Bereichs und konnte nicht ausgeführt werden. Das Programm enthält File-Register, für die keine Bereichsvorgabe erfolgt ist. Während der Verarbeitung einer RTOP-, RFRP-, LWTP- oder LRDP-Anweisung ist ein Verarbeitungsfehler aufgetreten. Bei Einsatz des MELSECNET/MINI ist die angesprochene Stationsnummer in einer FROM-TO-Anweisung gleich 0 oder > 62.	Programmschritte mit Hilfe eines Programmiergerätes anzeigen und auf mögliche Fehlerquellen (Adressenbereich der Operanden, Datenwerte für die BCD-Wandlung, Parametereinstellung der File-Register etc.) überprüfen.
MAIN CPU DOWN (2) Fehlfunktion der Master-CPU (Fehler beim Interrupt)	60	STOP	Eine INT-Anweisung wurde im Mikrocomputer-Programmbereich verarbeitet. Fehlfunktion der CPU aufgrund von Störspannungen (Rauschen etc.). Hardware-Fehler	INT-Anweisung löschen. Bei starken Störeinflüssen ist die Störquelle ausfindig zu machen und zu entfernen. Kann keine Abhilfe geschaffen werden, ist das CPU-Modul auszutauschen.
BATTERY ERROR Batteriefehler (ständige Überwachung)	70	RUN	Die Spannung der Batterie ist unter ihren Minimalwert gesunken. Die Batterie ist nicht oder nicht korrekt angeschlossen.	Batterieanschluss überprüfen und Batterie ggf. ersetzen. Der Einsatz eines RAM-Speichers oder die Anwendung der Netzausfallsicherung setzt die Verwendung einer Batterie voraus.

13.3 Liste der Fehlercodes (für die AnA und AnAS)

Die folgende Tabelle enthält eine Übersicht der möglichen Fehlercodes zusammen mit Fehlermeldungen, möglichen Ursachen und Hinweisen zur Behebung des Fehlers. Die Fehlercodes werden in das Sonderregister D9008, die genaue Aufschlüsselung des Codes in D9091 und die zugehörigen Schrittnummern, in die Sonderregister D9010 und D9011 geschrieben. In dieser Tabelle sind nur Fehlermeldungen der AnA- und AnAS-CPU aufgeführt.

Fehlermeldung	Inhalt von D9008	Inhalt von D9091	Ursache	Abhilfe
INSTRCT CODE ERR Anweisung fehlerhaft (die Überprüfung erfolgt nach dem Umschalten von STOP auf RUN oder der Ausführung einer Anweisung)	10	101	Das Programm enthält Anweisungen, die nicht zum Anweisungsumfang der CPU gehören und nicht verarbeitet werden können.	Programmschritt, an dem der Fehler aufgetreten ist, mit Hilfe eines Programmiergerätes auslesen und entsprechende Programmstelle korrigieren. Erfolgt die Programmverarbeitung über einen ROM-Speicherbaustein, ist zu prüfen, inwieweit der eingesetzte ROM-Baustein für die CPU geeignet ist.
		102	Die Index-Vergabe wurde für eine 32-Bit-Konstante durchgeführt.	Programmschritt, an dem der Fehler aufgetreten ist, mit Hilfe eines Programmiergerätes auslesen und entsprechende Programmstelle korrigieren.
		103	Ein Operand, der über eine Erweiterte Applikationsanweisung angesprochen wurde, ist fehlerhaft bestimmt.	
		104	Eine Erweiterte Applikationsanweisung weist eine fehlerhafte Programmstruktur auf.	
		105	Eine Erweiterte Applikationsanweisung weist einen fehlerhaften Anweisungsteil auf.	
		106	Die Index-Vergabe über die Operanden Z oder V liegt im Programm zwischen den Anweisungen LEDA/B IX und LEDA/B IXEND.	
		107	Die Index-Vergabe erfolgte für Operandenadressen und Sollwerte in einem Ausgangskontakt eines Timers oder Counters. Die Index-Vergabe erfolgte an der Pointermarkierung, die zu dem Sprungziel einer CJ-, SCJ-, CALL(P)-, JMP-, LEDA/B FCALL- oder LEDA/B BREAK-Anweisung gehört.	
		108	Der Fehler hat eine andere Ursache als in den Codes 101 bis 107 beschrieben.	

Fehlercodeliste der AnA- und AnAS-CPUs (Fortsetzung)

Fehlermeldung	Inhalt von D9008	Inhalt von D9091	Ursache	Abhilfe
PARAMETER ERROR Parameter fehlerhaft (die Überprüfung erfolgt beim Einschalten, einem Reset, dem Umschalten von STOP auf RUN oder von PAUSE auf RUN)	11	111	Die Festlegung der Speicherbereiche in den Parametern liegt für eine oder mehrere der folgenden Positionen außerhalb des nutzbaren Bereiches der CPU: Haupt- und Unterprogramm, Mikrocomputer-Programm, Kommentare für File-Register, Status Latch, Sampling Trace und erweiterte File-Register.	Parameter aus dem Speicher der CPU auslesen und den Inhalt kontrollieren. Fehlerhafte Parameterwerte ändern und korrigierte Werte wieder in den Speicher schreiben.
		112	Die Gesamtkapazität von Haupt- und Unterprogramm, Mikrocomputer-Programm, Kommentaren für File-Register, Status Latch, Sampling Trace und erweiterte File-Register übersteigt die Maximalkapaziät der Speicherkassette.	
		113	Der über Parameter oder die Operanden M, L oder S festgelegte Zwischenspeicherbereich (Status-Latch) ist fehlerhaft definiert.	
		114	Verarbeitungsfehler (Sum check).	
		115	Eine der folgenden Positionen ist in den Parametern falsch definiert: Remote-RUN-/PAUSE-Kontakt, Betriebsart bei Auftreten eines Fehlers, Anzeigeart der Fehlermelder, Anzeigeart von STOP/RUN.	
		116	Die Einstellung der automatischen Aktualisierung (Refresh) der Netzwerkdaten in den Parametern ist fehlerhaft.	
		117	Die Festlegung der Timer in den Parametern ist fehlerhaft.	
		118	Die Festlegung der Counter in den Parametern ist fehlerhaft.	
MISSING END INS. END-Anweisung fehlt (die Überprüfung erfolgt nach dem Umschalten von STOP auf RUN)	12	121	Die END-/FEND-Anweisung fehlt im Hauptprogramm.	END-Anweisung an das Ende des Programms/Unterprogramms setzen.
		122	Die END-/FEND-Anweisung fehlt im Unterprogramm (soweit ein Unterprogramm über Parameter festgelegt wurde).	
CAN'T EXECUTE (P) Programmsprung ist nicht ausführbar (die Überprüfung erfolgt bei Ausführung einer Anweisung)	13	131	An zwei oder mehr Stellen im Programm wurden identische Adressen für Pointer (P) oder Interrupt-Pointer (I) zur Markierung eines Sprungziels vergeben.	Identische Pointer-Adressen an den Sprunzielen entfernen.
		132	Das Sprungziel, das über eine der Anweisungen CJ, SCJ, CALL(P), JMP, LEDA/B FCALL oder LEDA/B BREAK angesprochen wurde, befindet sich nicht im Programm oder hinter einer END-Anweisung.	Programmschritt, an dem der Fehler aufgetreten ist, auslesen und Sprungzieladresse (P) an der richtigen Programmstelle einfügen.

Fehlercodeliste der AnA- und AnAS-CPUs (Fortsetzung)

Fehlermeldung	Inhalt von D9008	Inhalt von D9091	Ursache	Abhilfe
CAN'T EXECUTE (P) Programmsprung ist nicht ausführbar (die Überprüfung erfolgt bei Ausführung einer Anweisung)	13	133	Eine RET-Anweisung wurde ausgeführt, obwohl das Programm keine CALL-Anweisung enthält. Eine NEXT- und LEDA/B BREAK-Anweisung ist im Programm enthalten, obwohl die zugehörige FOR-Anweisung fehlt. CALL(P)- oder FOR-Anweisungen sind über mehr als 5 Ebenen verschachtelt. Nach einer CALL(P)- oder FOR-Anweisung fehlt die zugehörige RET- oder NEXT-Anweisung.	Programmschritt, an dem der Fehler aufgetreten ist, mit Hilfe eines Programmiergerätes auslesen und entsprechende Programmstelle korrigieren. Die Anzahl der Ebenen bei CALL(P)-RET- und FOR- NEXT-Verschachtelungen ist auf maximal 5 zu reduzieren.
		134	Das Programm enthält eine CHG-Anweisung, aber kein Unterprogramm.	Fehlerhafte Programmschrittnummer auslesen und CHG-Anweisung löschen.
		135	Die LEDA/B IX- und LEDA IXEND-Anweisungen sind nicht paarweise vorhanden. Das Programm enthält mehr als 32 LEDA/B IX- und LEDA IXEND-Anweisungspaare.	Fehlerhafte Programmschrittnummer auslesen und entsprechende Programmzeile korrigieren.
CHK FORMAT ERR Fehler im CHK-Format (die Überprüfung erfolgt nach dem Umschalten von STOP auf RUN bzw. von PAUSE auf RUN)	14	141	In einem CHK-Anweisungsblock befinden sich andere Anweisungen (NOP eingeschlossen) als LDX, LDIX, ANDX und ANIX.	CHK-Anweisungsblock im Programm auf die möglichen Fehlerursachen entsprechend der nebenstehenden detaillierten Fehlercodes überprüfen und korrigieren.
		142	Die CHK-Anweisung wurde mehr als einmal im Programm gesetzt.	
		143	Der CHK-Block enthält mehr als 150 Eingangskontakte (Kontrollbedingungen).	
		144	Die LEDA/CHK-Anweisung ist nicht in Verbindung mit einer LEDA/CHKEND-Anweisung programmiert oder es wurde mehr als 1 Paar dieser Anweisungen programmiert.	
		145	Oberhalb des CHK-Anweisungsblocks befindet sich keine CJ-Anweisung mit Eingangsbedingung.	
		146	Die Operandenadresse von D1 in der Anweisung CHK D1 D2 entspricht nicht der Operandenadresse oberhalb der CJ-Anweisung.	
		147	Im CHK-Anweisungsblock befindet sich eine Index-Vergabe.	
		148	Das Programm enthält mehr als ein LEDA/CHK- und LEDA/CHKEND-Anweisungspaar. Im LEDA/CHK- und LEDA/CHKEND-Anweisungsblock sind mehr als 6 Eingangskontakte (Kontrollbedingungen) vorhanden. Die Kontrollbedingungen innerhalb des LEDA/-CHK und LEDA/CHKEND-Anweisungsblocks enthalten keine Ein- oder Ausgangsoperanden (X und Y). Der LEDA/CHK- und LEDA/CHKEND-Anweisungsblock umfasst mehr als 256 Programmschritte.	CHK-Anweisungsblock im Programm auf die möglichen Fehlerursachen entsprechend der nebenstehenden detaillierten Fehlercodes überprüfen und korrigieren.

Fehlercodeliste der AnA- und AnAS-CPUs (Fortsetzung)

Fehlermeldung	Inhalt von D9008	Inhalt von D9091	Ursache	Abhilfe
CAN'T EXECUTE (1) Ausführung nicht möglich (die Überprüfung erfolgt bei Ausführung eines Interrupts)	15	151	Die IRET-Anweisung befindet sich in einem anderen Programmteil als dem Interrupt-Programm.	Überprüfen, ob sich eine IRET-Anweisung außerhalb des Interrupt-Programmteils befindet und entsprechende Programmstelle löschen.
		152	Im Interrupt-Programm befindet sich keine IRET-Anweisung.	Interrupt-Programmteil auf fehlende IRET-Anweisung untersuchen und Anweisung einfügen.
		153	Die Adresse des Interrupt-Pointers (I), mit dem das Interrupt-Modul angesprochen werden soll, ist fehlerhaft oder nicht vorhanden. Die fehlerhafte Pointer-Adresse wird in Sonderregister D9011 abgelegt.	Inhalt des Sonderregisters D9011 lesen und das Vorkommen und die Zuordnung von Interrupt-Programm und Interrupt-Modul überprüfen. Doppelt programmierte Interrupt-Pointer sind ggf. zu löschen.
CASSETTE ERROR Zugriff auf die Speicher-kassette ist nicht möglich	16		Die Speicherkassette befindet sich nicht in der CPU oder ist defekt.	Speicherkassette ersetzen bzw. bei abgeschalteter Versorgungsspannung einsetzen.
RAM ERROR Fehler beim RAM-Speicherzugriff (die Überprüfung erfolgt nach dem Einschalten)	20	201	Beim Zugriff auf das gespeicherte Ablaufprogramm ist ein Fehler aufgetreten.	Mitsubishi-Serviceabteilung informieren.
		202	Beim Zugriff auf den Arbeitsspeicher der CPU ist ein Fehler aufgetreten.	
		203	Beim Zugriff auf den Operandenspeicher der CPU ist ein Fehler aufgetreten.	
		204	Beim Zugriff auf den Adressenspeicher der CPU ist ein Fehler aufgetreten.	
OPE. CIRCUIT ERR Fehler im Betriebsschaltkreis (die Überprüfung erfolgt nach dem Einschalten und einem Reset)	21	211	Der Betriebsschaltkreis, der für die Index-Verarbeitung in der CPU verantwortlich ist, arbeitet fehlerhaft.	Mitsubishi-Serviceabteilung informieren.
		212	Die Hardware-Logik der CPU arbeitet fehlerhaft.	
		213	Der Betriebsschaltkreis, der für die Ablaufverarbeitung in der CPU verantwortlich ist, arbeitet fehlerhaft.	
WDT ERROR WDT-Fehler (die Überprüfung erfolgt nach Ausführung einer END-Anweisung)	22		Die Zykluszeit des Programms hat sich stark vergrößert. Die Zykluszeit hat aufgrund eines kurzzeitigen Spannungsabfalls, der während eines Programmdurchlaufs aufgetreten ist, zugenommen.	Zykluszeit des Programms überprüfen und neu berechnen. Die Programmzykluszeit kann mit Hilfe von CJ-Anweisungen usw. entsprechend reduziert werden. Liegt der Fehler nicht in der Programmstruktur, ist der Inhalt des Sonderregisters D9005 auszulesen. Beträgt der Wert nicht „0“, liegt eine ungenügende Versorgungsspannung vor. In diesem Fall ist die Spannungversorgung zu überprüfen und die Ursache für den Spannungsverlust zu beheben.
END NOT EXECUTE END-Anweisung konnte nicht ausgeführt werden (die Überprüfung erfolgt während der Ausführung der END-Anweisung)	24	241	Das Programm wurde ohne Ausführung der END-Anweisung verarbeitet. Während der Ausführung einer END-Anweisung wird aufgrund von Störeinflüssen (induzierte Spannungen etc.) ein anderer Anweisungscode gelesen. Die END-Anweisung hat sich aufgrund von Störeinflüssen in eine andere Anweisung geändert.	Reset ausführen und CPU erneut in den RUN-Betrieb setzen. Tritt der gleiche Fehler erneut auf, liegt möglicherweise ein Hardware-Fehler vor. In diesem Fall ist der Mitsubishi-Service zu informieren.

Fehlercodeliste der AnA- und AnAS-CPUs (Fortsetzung)

Fehlermeldung	Inhalt von D9008	Inhalt von D9091	Ursache	Abhilfe
MAIN CPU DOWN (1) Fehlfunktion der Master-CPU (ständige Überwachung)	26		Fehlfunktion oder Defekt der Haupt-CPU.	Mitsubishi-Serviceabteilung informieren.
UNIT VERIFY ERR. Kommunikationsfehler (ständige Überwachung)	31		Die Daten eines E-/A-Moduls haben sich nach dem Einschalten der Spannungsversorgung verändert. Das E-/A-Modul (einschließlich Sondermodul) befindet sich nicht in seinem ursprünglichen Steckplatz, oder ein anderes Modul wurde anstelle dessen eingesetzt.	Sonderrregister D9116 bis D9123 zur Eingrenzung des fehlerhaften Moduls mit Hilfe eines Programmiergerätes lesen. Das zugehörige Bit des Moduls, bei dem der Vergleichsfehler aufgetreten ist, ist auf „1““gesetzt. Ist die ursprüngliche Modulanordnung wieder hergestellt, muss das System über den RESET-Schalter zurückgesetzt werden.
'FUSE BREAK OFF Sicherung defekt (ständige Überwachung)	32		Die Sicherung eines Ausgangsmoduls ist defekt.	Sonderrregister D9100 bis D9107 zur Eingrenzung des Moduls mit Hilfe eines Programmiergerätes lesen. Das zugehörige Bit des Moduls, bei dem die Sicherung defekt ist, ist auf „1““gesetzt.
CONTROL BUS ERR. Störung am Systembus	40	401	FROM- und/oder TO-Anweisungen können nicht ausgeführt werden. Während der Übertragung zu einem Sondermodul ist am Systembus eine Störung aufgetreten.	Der Fehler kann durch ein defektes Sondermodul, CPU-Modul oder den Systembus des Baugruppenträgers hervorgerufen worden sein. Baugruppenträger und Module sollten daher eingehend geprüft und ggf. der Mitsubishi-Service verständigt werden.
		402	Bei Ausführung der Adressenzuordnung über Parameter ist der Zugriff auf ein Sondermodul bei Kommunikationsbeginn nicht möglich. Nach Auftreten des Fehlers wird die Startadresse des Sondermoduls in Sonderregister D9011 gespeichert.	
SP.UNIT DOWN Fehlfunktion eines Sondermoduls	41	411	Der Zugriff auf ein Sondermodul nach Ausführung einer FROM- und/oder TO-Anweisung erfolgte ohne Reaktion des Sondermoduls.	Mitsubishi-Serviceabteilung informieren.
		412	Bei Ausführung der Adressenzuordnung über Parameter ist der Zugriff auf ein Sondermodul bei Kommunikationsbeginn nicht möglich. Nach Auftreten des Fehlers wird die Startadresse des Sondermoduls in Sonderregister D9011 gespeichert.	
LINK UNIT ERROR Interface-Modul falsch plaziert	42		Ein Interface-Modul für das Data-Link-Netzwerk (z.B. AJ71R22, AJ71P22) ist im Baugruppenträger der Master-Station eingesetzt.	Modul aus dem Baugruppenträger der Master-Station entfernen, System über den RESET-Schalter zurücksetzen und Initialisierung wiederholen.
I/O INT. ERROR Interrupt-Fehler	43		Ein Interrupt wurde ausgeführt, obwohl sich im System kein Interrupt-Modul befindet.	Mitsubishi-Serviceabteilung informieren.

Fehlercodeliste der AnA- und AnAS-CPUs (Fortsetzung)

Fehlermeldung	Inhalt von D9008	Inhalt von D9091	Ursache	Abhilfe
SP.UNIT LAY ERROR Fehler in Verbindung mit einem Compter-Link-Modul	44	441	Die Adressenzuordnung in den Parametern ist falsch. Einem Sondermodul wurde die Adresse eines E-/A-Moduls zugeordnet (bzw. umgekehrt).	Bei falscher Adressenzuordnung eines Sondermoduls ist diese über ein Programmiergerät neu zu konfigurieren.
		442	Im System befinden sich mehr als 8 Sondermodule (ausgenommen AI61-S1), die einen Interrupt zur CPU ausführen können.	Anzahl der Module, die einen Interrupt ausführen können, auf 8 reduzieren.
		443	Auf dem Baugruppenträger befindet sich mehr als ein Interface-Modul für das Data-Link-Netzwerk (z.B. AJ71R22, AJ71P22).	Anzahl der AJ71R22-, AJ71P22-Module auf „1“ reduzieren.
		444	Auf dem Baugruppenträger eines CPU-Moduls befinden sich mehr als 6 Computer-Link-Module.	Anzahl der Computer-Link-Module auf „6“ beschränken.
		445	Auf einem Baugruppenträger befindet sich mehr als ein AI61-S1-Modul.	Anzahl der AI61-S1-Module auf „1“ reduzieren.
		446	Die Parameterzuweisung eines automatischen MNTMINI-Refreshs für einige Module im Netzwerk stimmt nicht mit den tatsächlichen Gegebenheiten überein.	Parameterzuweisung für den automatischen MNTMINI-Refresh neu definieren.
		447	Die maximale Anzahl von Sondermodulen, die einem CPU-Modul zugeordnet sind und die den erweiterten Anweisungsumfang verarbeiten können, ist überschritten (die maximale Anzahl darf 1344 nicht überschreiten).	Gesamtzahl der Sondermodule im System reduzieren.
SP. UNIT ERROR Einstellwerte fehlerhaft (die Überprüfung erfolgt bei Ausführung einer FROM-/TO-Anweisung)	46	461	Das mittels FROM-/TO-Anweisung angesprochene Modul ist kein Sondermodul.	Programmschritt, an dem der Fehler aufgetreten ist, mit Hilfe eines Programmiergerätes auslesen und den Inhalt der FROM-/TO-Anweisung korrigieren.
		462	Das mittels einer erweiterten Applikationsanweisung der AnA-/AnU-CPUs angesprochene Modul ist kein oder nicht das richtige Sondermodul.	Programmschritt, an dem der Fehler aufgetreten ist, mit Hilfe eines Programmiergerätes auslesen und den Inhalt der entsprechenden Anweisung korrigieren.
LINK PARA. ERROR Fehler in den Parametern zur Datenkommunikation	47		Die Parameter zur Festlegung des Übertragungsbereiches für die Datenkommunikation im Data-Link-Netzwerk sind fehlerhaft oder nicht vorhanden. Die Einstellung der Summe der im Netzwerk befindlichen lokalen Stationen lautet „0““.	Parameter überprüfen und korrigieren oder ggf. neu eingeben. Zusätzlich ist die Einstellung der Stationsnummern zu überprüfen. Tritt der Fehler wiederholt auf, liegt ein Hardware-Fehler vor. In diesem Fall ist der Mitsubishi-Service zu informieren.
BATTERY ERROR Batteriefehler (ständige Überwachung)	70		Die Spannung der Batterie ist unter ihren Minimalwert gesunken. Die Batterie ist nicht oder nicht korrekt angeschlossen.	Batterieanschluss überprüfen und Batterie ggf. ersetzen. Der Einsatz eines RAM-Speichers oder die Anwendung der Netzausfallsicherung setzt die Verwendung einer Batterie voraus.

Fehlercodeliste der AnA und AnAS CPUs (Fortsetzung)

Fehlermeldung	Inhalt von D9008	Inhalt von D9091	Ursache	Abhilfe
OPERATION ERROR Ausführungsfehler (die Überprüfung erfolgt bei Ausführung einer Anweisung)	50	501	Die Verarbeitung von File-Registern (R) soll außerhalb der vorgegebenen Bereiche der Operandenadressen und der Blockadressen der File-Register erfolgen. Im Programm sollen File-Register (R) verarbeitet werden, obwohl die Kapazität der Register nicht vorgegeben wurde.	Programmschritt, an dem der Fehler aufgetreten ist, mit Hilfe eines Programmiergerätes auslesen und den Inhalt der entsprechenden fehlerhaften Anweisung korrigieren.
		502	Die in einer Anweisung programmierte Kombination von Operanden ist nicht zulässig.	
		503	Ein gespeicherter Datenwert oder die Konstante eines vorgegebenen Operanden liegt außerhalb des maximal zulässigen Bereiches.	
		504	Die vorgegebene Anzahl der Daten, die verarbeitet werden sollen, liegt über dem maximal möglichen Bereich.	
		505	Die in einer LEDA/B LRDP-, LCDA/B LWTP-, LRDP- oder LWTP-Anweisung vorgegebene Stationsadresse spricht keine lokale Station an. An der E-/A-Adresse, die in einer LEDA/B RFRP-, LEDA/B RTOP-, RFRP- oder RTOP-Anweisung vorgegeben ist, befindet sich keine Remote-Station.	
		506	Über die E-/A-Adresse, die in einer LEDA/B RFRP-, LEDA/B RTOP-, RFRP- oder RTOP-Anweisung vorgegeben ist, wird kein Sondermodul angesprochen.	
		507	Auf ein AD57(S1-) oder AD58-Modul wurde während der Verarbeitung von Anweisungen im „divided processing mode““ über das Netzwerk zugegriffen. Während der Verarbeitung von Anweisungen im „divided processing mode“ eines AD57(S1)- oder AD58-Moduls wurde gleichzeitig auf ein weiteres AD57(S1)- oder AD58-Modul zugegriffen.	Programmschritt, an dem der Fehler aufgetreten ist, mit Hilfe eines Programmiergerätes auslesen und den Inhalt der entsprechenden fehlerhaften Anweisung korrigieren.
		509	Über das Netzwerk wurde eine Anweisung ausgegeben, die von einem Remote-Terminal-Modul nicht verarbeitet werden kann. Der Speicherbereich zur Registrierung der Kommunikationsanforderung ist belegt, wenn eine PRC-Anweisung an ein Remote-Terminal-Modul übertragen wird. Eine PIDCONT-Anweisung wurde ohne zugehörige PIDINIT-Anweisung ausgeführt. Eine PID57-Anweisung wurde ohne zugehörige PIDINIT- oder PIDCONT-Anweisung ausgeführt.	Fehlerhafte Programmschrittnummer auslesen und betreffende Programmzeile unter Berücksichtigung der Moduldaten korrigieren. Es ist darauf zu achten, dass der Zugriff auf ein Remote-Terminal-Modul mittels PRC-Anweisung über Sondermerker M9081 oder Datenregister D9081 verriegelt ist. Die PIDCONT-Anweisung muss nach Ausführung der PIDINIT-Anweisung programmiert sein. Die Ausführung der PID57-Anweisung darf erst nach Ausführung der PIDINIT- und PIDCONT-Anweisung erfolgen.
MAIN CPU DOWN (2) Fehlfunktion der Master CPU (Fehler beim Interrupt)	60		Eine INT-Anweisung wurde im Mikrocomputer-Programmbereich verarbeitet. Fehlfunktion der CPU aufgrund von Störspannungen (Rauschen). Hardware-Fehler	INT-Anweisung löschen. Bei starken Störeinflüssen ist die Störquelle ausfindig zu machen und zu entfernen. Kann keine Abhilfe geschaffen werden, ist das CPU-Modul auszutauschen.

A Anhang A

A.1 Definition der Verarbeitungszeit

Die Verarbeitungszeit ist die Summe aus:

- Der Summe der Verarbeitungszeiten für jede Anweisung.
- Der Verarbeitungszeit der END-Anweisung. Diese besteht aus der Zeit, die für die Ausführung der END-Anweisung benötigt wird, der Auffrischzeit eines eventuell angeschlossenen MELSECNET, der Zeit, die für den Datenaustausch mit peripheren Geräten benötigt wird und der Zeit, die für serielle Kommunikation aufgewendet wird.
- Der Auffrischzeit für die Ein- und Ausgänge, die folgendermaßen berechnet wird:

$$\text{E/A-Auffrischzeit} = \frac{\text{Summe der Eingänge}}{16} \times N1 + \frac{\text{Summe der Ausgänge}}{16} \times N2$$

Nachfolgende Tabelle gibt die Zeiten N1 und N2 für QnA-CPUs und CPUs des MELSEC System Q an.

Typ der CPU	N1 (µs)			N2 (µs)		
	Q-Hauptbau-gruppenträger	Q-Erweiterungs-baugruppenträger	QnA-Erweiterungs-baugruppenträger	Q-Haupt-baugruppenträger	Q-Erweiterungs-baugruppenträger	QnA-Erweiterungs-baugruppenträger
Q00JCPU	2,5	3,3	—	1,3	2,3	—
Q00CPU	2,4	3,2	—		2,3	—
Q01CPU	2,3	3,1	—		2,3	—
Q02CPU)	2,2	2,9	4,3		2,1	3,5
Q02HCPU Q06HCPU Q12HCPU Q12PHCPU Q25HCPU Q25PHCPU	1,7	2,4	3,7		2,1	3,5
Q2ASCPU (S1) Q2ACPU	5,2			5,0		
Q3ACPU	4,8			4,65		
Q2ASHCPU (S1) Q4ACPU Q4ARCPU	4,34			4,26		

A.2 Verarbeitungszeiten

Auf den folgenden Seiten werden alle Verarbeitungszeiten der Anweisungen in tabellarischer Form aufgeführt. Die jeweiligen Verarbeitungszeiten sind von den Werten in den Quell- und Zieldaten abhängig. Die in den Tabellen aufgeführten Werte dienen als Richtlinie für eine Kalkulation der Gesamtverarbeitungszeit.

Die Verarbeitungszeit einer Anweisung beinhaltet nicht die Zeit für die Index-Vergabe. Die Verarbeitungszeit für den Fall, dass die Anweisung nicht bearbeitet wird, kann berechnet werden:

Typ der CPU	Verarbeitungszeit bei nicht ausgeführter Anweisung (µs)
Q00JCPU	0,20 x (Anzahl der Schritte pro Anweisung +1)
Q00CPU	0,16 x (Anzahl der Schritte pro Anweisung +1)
Q01CPU	0,10 x (Anzahl der Schritte pro Anweisung +1)
Q02CPU)	0,079 x (Anzahl der Schritte pro Anweisung +1)
Q02HCPU Q06HCPU Q12HCPU Q12PHCPU Q25HCPU Q25PHCPU	0,034 x (Anzahl der Schritte pro Anweisung +1)
Q2ASCPU (S1) Q2ACPU	0,20 x (Anzahl der Schritte pro Anweisung +1)
Q3ACPU	0,15 x (Anzahl der Schritte pro Anweisung +1)
Q2ASHCPU (S1) Q4ACPU Q4ARCPU	0,075 x (Anzahl der Schritte pro Anweisung +1)

A.2.1 Liste der Verarbeitungszeiten (QnA-Serie und System Q)

Anweisung	Verarbeitung (Operand)			Verarbeitungszeiten (µs)								
				QnA-Serie				System Q				
				Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H
LD LDI AND ANI OR ORI		z. B. X0		0,20	0,15	0,075	0,075	0,20	0,16	0,10	0,079	0,034
		z.B. D0.0						0,30	0,24	0,15		
LDP LDF ANDP ANDF ORP ORF				6,6	5,0	2,5	2,5	0,30	0,24	0,15	0,158	0,068
ANB ORB MPS MRD MPP				0,20	0,15	0,075	0,075	0,20	0,16	0,10	0,079	0,034
INV	nicht ausgeführt			2,4	1,8	0,9	0,9	0,20	0,16	0,10	0,079	0,034
	ausgeführt											
MEP MEF	nicht ausgeführt			2,0	1,5	0,75	0,75	0,30	0,24	0,15	0,173	0,073
	ausgeführt											
EGP	gleichbleibender Zustand			0,6	0,3	0,15	0,15	0,20	0,16	0,10	0,158	0,068
	wechselnder Zustand (AUS/EIN oder EIN/AUS)											
EGF	gleichbleibender Zustand							17	9,5	9,4		
	wechselnder Zustand (AUS/EIN oder EIN/AUS)							18	14	14		
OUT	außer F, T,C	gleichbleibender Zustand		0,40	0,30	0,15	0,15	0,20	0,16	0,10	0,158	0,068
		wechselnder Zustand (AUS/EIN oder EIN/AUS)										
	D0.0	gleichbleibender Zustand						0,40	0,32	0,20		
		wechselnder Zustand (AUS/EIN oder EIN/AUS)										
	F	nicht ausgeführt		7,0	5,3	2,7	2,7	24	20	19	2,8	1,2
		ausgeführt	angezeigt	167	126	63	63	260	210	200	162	69,7
			Anzeige abgeschlossen	166	125	62	62	205	165	155	126	54
	T	nicht ausgeführt		1,6	1,2	0,6	0,6	1,1	0,88	0,55	0,63	0,27
		ausgeführt	nach Ablauf									
			zusätzlich mit K									
			zusätzlich mit D					1,2	0,96	0,60		
	C	nicht ausgeführt		1,6	1,2	0,6	0,6	1,1	0,88	0,55	0,63	0,27
		ausgeführt	nach Ablauf									
			zusätzlich mit K									
			zusätzlich mit D					1,2	0,96	0,6		
OUTH	T	nicht ausgeführt		1,6	1,2	0,6	0,6	1,1	0,88	0,55	0,63	0,27
		ausgeführt	nach Ablauf									
			zusätzlich mit K									
			zusätzlich mit D					1,2	0,96	0,6		
SET	Alle außer F und D0.0	nicht ausgeführt		0,40	0,30	0,15	0,15	0,20	0,16	0,10	0,158	0,068
		ausgeführt	gleichbleibender Zustand									
			wechselnder Zustand									
	D0.0	nicht ausgeführt						0,40	0,32	0,20		
		ausgeführt	gleichbleibender Zustand									
			wechselnder Zustand									
	F	nicht ausgeführt		1,2	0,90	0,45	0,45	0,50	0,44	0,25	0,47	0,20
		ausgeführt	angezeigt	277	208	104	104	255	205	195	161	69
			Anzeige abgeschlossen	1,2	0,90	0,45	0,45	195	160	150	0,47	0,20

Anweisung	Verarbeitung (Operand)			Verarbeitungszeiten (µs)								
				QnA-Serie				System Q				
				Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H
RST	Alle außer den unten genannten Operanden	nicht ausgeführt		0,40	0,30	0,15	0,15	0,20	0,16	0,10	0,158	0,068
		ausgeführt	gleichbleibender Zustand									
			wechselnder Zustand									
	D0.0	nicht ausgeführt		0,40	0.30	0,15	0,15	0,40	0,32	0,20	0,158	0,068
		ausgeführt	gleichbleibender Zustand									
			wechselnder Zustand									
	SM	nicht ausgeführt		0,40	0,30	0,15	0,15	0,20	0,16	0,10	0,158	0,068
		ausgeführt										
	F	nicht ausgeführt		1,2	0,90	0,45	0,45	0,48	0,44	0,25	0,47	0,20
		ausgeführt	angezeigt	148	112	56	56	75	69	65	90	38
			Anzeige abgeschlossen	1,2	0,90	0,45	0,45	43	35	33	0,47	0,20
	T, C	nicht ausgeführt		1,4	1,1	0,6	0,6	0,80	0,64	0,40	0,63	0,27
		ausgeführt						1,0	0,80	0,50		
	D	nicht ausgeführt		0,60	0,45	0,23	0,23	0,40	0,32	0,20	0,24	0,10
		ausgeführt						0,60	0,48	0,30		
	Z	nicht ausgeführt		1,2	0,90	0,45	0,45	0,50	0,40	0,25	0,47	0,20
		ausgeführt		10,8	8,1	4,1	4,1	9,4	7,9	7,4	4,3	1,9
	R	nicht ausgeführt		1,0	0,75	0,38	0,38	-	0,32	0,20	0,40	0,17
		ausgeführt						-	0,48	0,30		
PLS				2,6	2,0	0,98	0,98	12	9,5	9,2	1,0	0,44
PLF				2,6	2,0	0,98	0,98	11	9,5	8,9	1,0	0,44
FF	Y	nicht ausgeführt		1,2	0,90	0,45	0,45	0,68	0,40	0,25	0,47	0,20
		ausgeführt						7,5	6,2	5,7		
DELTA	DY0	nicht ausgeführt		1,2	0,90	0,45	0,45	0,50	0,40	0,25	0,47	0,20
		ausgeführt		16,8	14,1	11,1	11,1	26	21	21	5,9	2,6
DELTAP	DY0	nicht ausgeführt		1,2	0,90	0,45	0,45	0,48	0,40	0,25	0,47	0,20
		ausgeführt		16,8	14,1	11,1	11,1	58	45	43	5,9	2,6
SFT SFTP	nicht ausgeführt			1,2	0,90	0,45	0,45	0,50	0,34	0,25	0,47	0,20
	ausgeführt			4,2	3,2	1,6	1,6	12	8,7	8,3	1,66	0,71
MC	M0.0			0,60	0,45	0,23	0,23	0,40	0,32	0,20	0,24	0,10
	D.0							3,3	2,9	2,8		
MCR				0,20	0,15	0,075	0,075	0,20	0,16	0,10	0,079	0,034
FEND END	Fehlerkontrolle ausgeführt			1643	1236	618	618	660	530	480	348	150
	ohne Fehlerkontrolle: -Batterietest -Schmelzsicherungstest -Überprüfung der E/A-Modul			1106	832	416	416	660	530	480	359	150
NOP				0,2	0,15	0,075	0,075	0,20	0,16	0,10	0,079	0,034
NOPLF PAGE				0,2	0,15	0,075	0,075	0,20	0,16	0,10	0,79	0,034
LD=	mit Kontinuität			3,8	2,9	1,5	1,5	0,80	0,64	0,40	0,24	0,10
	ohne Kontinuität			3,6	2,7	1,4	1,4					
AND=	nicht ausgeführt			1,4	1,1	0,55	0,55	0,70	0,56	0,35	0,24	0,10
	ausgeführt	mit Kontinuität		2,8	2,1	1,1	1,1	0,80	0,64	0,40		
		ohne Kontinuität		3,2	2,4	1,2	1,2					
OR=	nicht ausgeführt			1,4	1,1	0,55	0,55	0,70	0,56	0,35	0,24	0,10
	ausgeführt	mit Kontinuität		3,8	2,9	1,5	1,5	0,80	0,64	0,40		
		ohne Kontinuität		2,8	2,1	1,1	1,1					
LD<>	mit Kontinuität			4,4	3,3	1,7	1,7	0,80	0,64	0,40	0,24	0,10
	ohne Kontinuität			3,6	2,7	1,4	1,4					
AND<>	nicht ausgeführt			1,4	1,1	0,55	0,55	0,70	0,56	0,35	0,24	0,10
	ausgeführt	mit Kontinuität		2,8	2,1	1,1	1,1	0,80	0,64	0,40		
		ohne Kontinuität		3,2	2,4	1,2	1,2					

Anweisung	Verarbeitung (Operand)		Verarbeitungszeiten (µs)								
			QnA-Serie				System Q				
			Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H
OR<>	nicht ausgeführt		1,4	1,1	0,55	0,55	0,70	0,56	0,35	0,24	0,10
	ausgeführt	mit Kontinuität	3,8	2,9	1,5	1,5	0,80	0,64	0,40		
		ohne Kontinuität	2,8	2,1	1,1	1,1					
LD>	mit Kontinuität		4,4	3,3	1,7	1,7	0,80	0,64	0,40	0,24	0,10
	ohne Kontinuität		3,6	2,7	1,4	1,4					
AND>	nicht ausgeführt		1,4	1,1	0,55	0,55	0,70	0,56	0,35	0,24	0,10
	ausgeführt	mit Kontinuität	2,8	2,1	1,1	1,1	0,80	0,64	0,40		
		ohne Kontinuität	3,2	2,4	1,2	1,2					
OR>	nicht ausgeführt		1,4	1,1	0,55	0,55	0,70	0,56	0,35	0,24	0,10
	ausgeführt	mit Kontinuität	3,8	2,9	1,5	1,5	0,80	0,64	0,40		
		ohne Kontinuität	2,8	2,1	1,1	1,1					
LD<=	mit Kontinuität		4,4	3,3	1,7	1,7	0,80	0,64	0,40	0,24	0,10
	ohne Kontinuität		3,6	2,7	1,4	1,4					
AND<=	nicht ausgeführt		1,4	1,1	0,55	0,55	0,70	0,56	0,35	0,24	0,10
	ausgeführt	mit Kontinuität	2,8	2,1	1,1	1,1	0,80	0,64	0,40		
		ohne Kontinuität	3,2	2,4	1,2	1,2					
OR<=	nicht ausgeführt		1,4	1,1	0,55	0,55	0,70	0,56	0,35	0,24	0,10
	ausgeführt	mit Kontinuität	3,8	2,9	1,5	1,5	0,80	0,64	0,40		
		ohne Kontinuität	2,8	2,1	1,1	1,1					
LD<	mit Kontinuität		4,4	3,3	1,7	1,7	0,80	0,64	0,40	0,24	0,10
	ohne Kontinuität		3,6	2,7	1,4	1,4					
AND<	nicht ausgeführt		1,4	1,1	0,55	0,55	0,70	0,56	0,35	0,24	0,10
	ausgeführt	mit Kontinuität	2,8	2,1	1,1	1,1	0,80	0,64	0,40		
		ohne Kontinuität	3,2	2,4	1,2	1,2					
OR<	nicht ausgeführt		1,4	1,1	0,55	0,55	0,70	0,56	0,35	0,24	0,10
	ausgeführt	mit Kontinuität	3,8	2,9	1,5	1,5	0,80	0,64	0,40		
		ohne Kontinuität	2,8	2,1	1,1	1,1					
LD>=	mit Kontinuität		4,4	3,3	1,7	1,7	0,80	0,64	0,40	0,24	0,10
	ohne Kontinuität		3,6	2,7	1,4	1,4					
AND>=	nicht ausgeführt		1,4	1,1	0,55	0,55	0,70	0,56	0,35	0,24	0,10
	ausgeführt	mit Kontinuität	2,8	2,1	1,1	1,1	0,80	0,64	0,40		
		ohne Kontinuität	3,2	2,4	1,2	1,2					
OR>=	nicht ausgeführt		1,4	1,1	0,55	0,55	0,70	0,56	0,35	0,24	0,10
	ausgeführt	mit Kontinuität	3,8	2,9	1,5	1,5	0,80	0,64	0,40		
		ohne Kontinuität	2,8	2,1	1,1	1,1					
LDD=	mit Kontinuität		5,0	3,8	1,9	1,9	1,0	0,80	0,50	0,55	0,24
	ohne Kontinuität		4,2	3,2	1,6	1,6				0,39	0,17
ANDD=	nicht ausgeführt		1,4	1,1	0,55	0,55	0,80	0,64	0,40	0,39	0,17
	ausgeführt	mit Kontinuität	3,4	2,6	1,3	1,3	1,0	0,80	0,50	0,55	0,24
		ohne Kontinuität	3,8	2,9	1,5	1,5				0,39	0,17
ORD=	nicht ausgeführt		1,4	1,1	0,55	0,55	0,80	0,64	0,40	0,39	0,17
	c	mit Kontinuität	4,4	3,3	1,7	1,7	1,0	0,80	0,50	0,55	0,24
		ohne Kontinuität	3,4	2,6	1,3	1,3					
LDD<>	mit Kontinuität		5,0	3,8	1,9	1,9	1,0	0,80	0,50	0,55	0,24
	ohne Kontinuität		4,2	3,2	1,6	1,6					
ANDD<>	nicht ausgeführt		1,4	1,1	0,55	0,55	0,80	0,64	0,40	0,39	0,17
	ausgeführt	mit Kontinuität	3,4	2,6	1,3	1,3	1,0	0,80	0,50	0,55	0,24
		ohne Kontinuität	3,8	2,9	1,5	1,5					
ORD<>	nicht ausgeführt		1,4	1,1	0,55	0,55	0,80	0,64	0,40	0,39	0,17
	ausgeführt	mit Kontinuität	4,4	3,3	1,7	1,7	1,0	0,80	0,50	0,55	0,24
		ohne Kontinuität	3,4	2,6	1,3	1,3					
LDD>	mit Kontinuität		3,8	2,9	1,5	1,5	1,0	0,80	0,50	0,55	0,24
	ohne Kontinuität		4,2	3,2	1,6	1,6					

Anweisung	Verarbeitung (Operand)			Verarbeitungszeiten (µs)								
				QnA-Serie				System Q				
				Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H
ANDD>	nicht ausgeführt			1,4	1,1	0,55	0,55	0,80	0,64	0,40	0,39	0,17
	ausgeführt	mit Kontinuität		2,8	2,1	1,1	1,1	1,0	0,80	0,50	0,55	0,24
		ohne Kontinuität		3,8	2,9	1,5	1,5					
ORD>	nicht ausgeführt			1,4	1,1	0,55	0,55	0,80	0,64	0,40	0,39	0,17
	ausgeführt	mit Kontinuität		3,8	2,9	1,5	1,5	1,0	0,80	0,50	0,55	0,24
		ohne Kontinuität		3,4	2,6	1,3	1,3					
LDD<=	mit Kontinuität			4,4	3,3	1,7	1,7	1,0	0,80	0,50	0,55	0,24
	ohne Kontinuität			3,6	2,7	1,4	1,4					
ANDD<=	nicht ausgeführt			1,4	1,1	0,55	0,55	0,80	0,64	0,40	0,39	0,17
	ausgeführt	mit Kontinuität		3,4	2,6	1,3	1,3	1,0	0,80	0,50	0,55	0,24
		ohne Kontinuität		3,2	2,4	1,2	1,2					
ORD<=	nicht ausgeführt			1,4	1,1	0,55	0,55	0,80	0,64	0,40	0,39	0,17
	ausgeführt	mit Kontinuität		4,4	3,3	1,7	1,7	1,0	0,80	0,50	0,55	0,24
		ohne Kontinuität		2,8	2,1	1,1	1,1					
LDD<	mit Kontinuität			3,8	2,9	1,5	1,5	1,0	0,80	0,50	0,55	0,24
	ohne Kontinuität			4,2	3,2	1,6	1,6					
ANDD<	nicht ausgeführt			1,4	1,1	0,55	0,55	0,80	0,64	0,40	0,39	0,17
	ausgeführt	mit Kontinuität		2,8	2,1	1,1	1,1	1,0	0,80	0,50	0,55	0,24
		ohne Kontinuität		3,8	2,9	1,5	1,5					
ORD<	nicht ausgeführt			1,4	1,1	0,55	0,55	0,80	0,64	0,40	0,39	0,17
	ausgeführt	mit Kontinuität		3,8	2,9	1,5	1,5	1,0	0,80	0,50	0,55	0,24
		ohne Kontinuität		3,4	2,6	1,3	1,3					
LDD>=	mit Kontinuität			4,4	3,3	1,7	1,7	1,0	0,80	0,50	0,55	0,24
	ohne Kontinuität			3,6	2,7	1,4	1,4					
ANDD>=	nicht ausgeführt			1,4	1,1	0,55	0,55	0,80	0,64	0,40	0,39	0,17
	ausgeführt	mit Kontinuität		3,4	2,6	1,3	1,3	1,0	0,80	0,50	0,55	0,24
		ohne Kontinuität		3,2	2.4	1,2	1,2					
ORD>=	nicht ausgeführt			1,4	1,1	0,55	0,55	0,80	0,64	0,40	0,39	0,17
	ausgeführt	mit Kontinuität		4,4	3,3	1,7	1,7	1,0	0,80	0,50	0,55	0,24
		ohne Kontinuität		2,8	2,1	1,1	1,1					
LDE=	Mit einfacher Genauigkeit	mit Kontinuität		235	177	89	35	—	—	—	93	40
		ohne Kontinuität		231	174	87	87	—	—	—	92	
	Mit doppelter Genauigkeit	mit Kontinuität		—	—	—	—	—	—	—	93	40
		ohne Kontinuität		—	—	—	—	—	—	—	92	
ANDE=	Mit einfacher Genauigkeit	nicht ausgeführt		1,4	1,1	0,55	0,55	—	—	—	0,55	0,24
		ausgeführt	mit Kontinuität	234	176	88	35	—	—	—	93	40
			ohne Kontinuität	230	172	86	86	—	—	—	92	
	Mit doppelter Genauigkeit	nicht ausgeführt		—	—	—	—	—	—	—	—	—
		ausgeführt	mit Kontinuität	—	—	—	—	—	—	—	93	40
			ohne Kontinuität	—	—	—	—	—	—	—	92	
ORE=	Mit einfacher Genauigkeit	nicht ausgeführt		1,4	1,1	0,55	0,55	—	—	—	0,55	0,24
		ausgeführt	mit Kontinuität	234	176	88	35	—	—	—	93	40
			ohne Kontinuität	230	172	86	86	—	—	—	92	
	Mit doppelter Genauigkeit	nicht ausgeführt		—	—	—	—	—	—	—	—	—
		ausgeführt	mit Kontinuität	—	—	—	—	—	—	—	93	40
			ohne Kontinuität	—	—	—	—	—	—	—	92	
LDE<>	Mit einfacher Genauigkeit	mit Kontinuität		231	174	87	35	—	—	—	92	40
		ohne Kontinuität		234	176	88	88	—	—	—		
	Mit doppelter Genauigkeit	mit Kontinuität		—	—	—	—	—	—	—	92	40
		ohne Kontinuität		—	—	—	—	—	—	—		

Anweisung	Verarbeitung (Operand)			Verarbeitungszeiten (µs)								
				QnA-Serie				System Q				
				Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H
ANDE<>	Mit einfacher Genauigkeit	nicht ausgeführt		1,4	1,1	0,55	0,55	—	—	—	0,55	0,24
		ausgeführt	mit Kontinuität	230	172	86	34	—	—	—	92	40
			ohne Kontinuität	234	176	88	88	—	—	—	93	
	Mit doppelter Genauigkeit	nicht ausgeführt		—	—	—	—	—	—	—	—	—
		ausgeführt	mit Kontinuität	—	—	—	—	—	—	—	92	40
			ohne Kontinuität	—	—	—	—	—	—	—		
ORE<>	Mit einfacher Genauigkeit	nicht ausgeführt		1,4	1,1	0,55	0,55	—	—	—	0,55	0,24
		ausgeführt	mit Kontinuität	231	174	87	35	—	—	—	93	40
			ohne Kontinuität	234	176	88	88	—	—	—	92	
	Mit doppelter Genauigkeit	nicht ausgeführt		—	—	—	—	—	—	—	—	—
		ausgeführt	mit Kontinuität	—	—	—	—	—	—	—	93	40
			ohne Kontinuität	—	—	—	—	—	—	—	92	
LDE>	Mit einfacher Genauigkeit	mit Kontinuität		231	174	87	35	—	—	—	92	40
		ohne Kontinuität		234	176	88	88	—	—	—		
	Mit doppelter Genauigkeit	mit Kontinuität		—	—	—	—	—	—	—	92	40
		ohne Kontinuität		—	—		—	—	—	—		
ANDE>	Mit einfacher Genauigkeit	nicht ausgeführt		1,4	1,1	0,55	0,55	—	—	—	0,55	0,24
		ausgeführt	mit Kontinuität	230	172	86	34	—	—	—	92	40
			ohne Kontinuität	234	176	88	88	—	—	—	93	
	Mit doppelter Genauigkeit	nicht ausgeführt		—	—	—	—	—	—	—	—	—
		ausgeführt	mit Kontinuität	—	—	—	—	—	—	—	92	40
			ohne Kontinuität	—	—	—	—	—	—	—		
ORE>	Mit einfacher Genauigkeit	nicht ausgeführt		1,4	1,1	0,55	0,55	—	—	—	0,55	0,24
		ausgeführt	mit Kontinuität	231	174	87	34	—	—	—	93	40
			ohne Kontinuität	234	176	88	88	—	—	—	92	
	Mit doppelter Genauigkeit	nicht ausgeführt		—	—	—	—	—	—	—	—	—
		ausgeführt	mit Kontinuität	—	—	—	—	—	—	—	93	40
			ohne Kontinuität	—	—	—	—	—	—	—	92	
LDE<=	Mit einfacher Genauigkeit	mit Kontinuität		235	177	89	34	—	—	—	93	40
		ohne Kontinuität		231	174	87	88	—	—	—	92	
	Mit doppelter Genauigkeit	mit Kontinuität		—	—	—	—	—	—	—	93	40
		ohne Kontinuität		—	—	—	—	—	—	—	92	
ANDE<=	Mit einfacher Genauigkeit	nicht ausgeführt		1,4	1,1	0,55	0,55	—	—	—	0,55	0,24
		ausgeführt	mit Kontinuität	234	176	88	34	—	—	—	92	40
			ohne Kontinuität	230	172	86	86	—	—	—		
	Mit doppelter Genauigkeit	nicht ausgeführt		—	—	—	—	—	—	—	—	—
		ausgeführt	ohne Kontinuität	—	—	—	—	—	—	—	92	40
				—	—	—	—	—	—	—		
ORE<=	Mit einfacher Genauigkeit	nicht ausgeführt		1,4	1,1	0,55	0,55	—	—	—	0,55	0,24
		ausgeführt	mit Kontinuität	234	176	88	34	—	—	—	92	40
			ohne Kontinuität	230	172	86	86	—	—	—		
	Mit doppelter Genauigkeit	nicht ausgeführt		—	—	—	—	—	—	—	—	—
		ausgeführt	mit Kontinuität	—	—	—	—	—	—	—	92	40
			ohne Kontinuität	—	—	—	—	—	—	—		

Anweisung	Verarbeitung (Operand)			Verarbeitungszeiten (µs)								
				QnA-Serie				System Q				
				Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H
LDE<	Mit einfacher Genauigkeit	mit Kontinuität		231	174	87	35	—	—	—	92	40
		ohne Kontinuität		234	176	88	88	—	—	—		
	Mit doppelter Genauigkeit	mit Kontinuität		—	—	—	—	—	—	—	92	92
		ohne Kontinuität		—	—	—	—	—	—	—		
ANDE<	Mit einfacher Genauigkeit	nicht ausgeführt		1,4	1,1	0,55	0,55	—	—	—	0,55	0,24
		ausgeführt	mit Kontinuität	230	172	86	34	—	—	—	92	40
			ohne Kontinuität	234	176	88	88	—	—	—		
	Mit doppelter Genauigkeit	nicht ausgeführt		—	—	—	—	—	—	—	—	—
		ausgeführt	mit Kontinuität	—	—	—	—	—	—	—	92	40
			ohne Kontinuität	—	—	—	—	—	—	—		
ORE<	Mit einfacher Genauigkeit	nicht ausgeführt		1,4	1,1	0,55	0,55	—	—	—	0,55	0,24
		ausgeführt	mit Kontinuität	231	174	87	34	—	—	—	93	40
			ohne Kontinuität	234	176	88	88	—	—	—	92	
	Mit doppelter Genauigkeit	nicht ausgeführt		—	—	—	—	—	—	—	—	—
		ausgeführt	mit Kontinuität	—	—	—	—	—	—	—	93	40
			ohne Kontinuität	—	—	—	—	—	—	—	92	
LDE>=	Mit einfacher Genauigkeit	mit Kontinuität		235	177	89	35	—	—	—	93	40
		ohne Kontinuität		231	174	87	87	—	—	—	92	
	Mit doppelter Genauigkeit	mit Kontinuität		—	—	—	—	—	—	—	93	40
		ohne Kontinuität		—	—		—	—	—	—	92	
ANDE>=	Mit einfacher Genauigkeit	nicht ausgeführt		1,4	1,1	0,55	0,55	—	—	—	0,55	0,24
		ausgeführt	mit Kontinuität	234	176	88	34	—	—	—	92	40
			ohne Kontinuität	231	174	87	87	—	—	—		
	Mit doppelter Genauigkeit	nicht ausgeführt		—	—	—	—	—	—	—	—	—
		ausgeführt	mit Kontinuität	—	—	—	—	—	—	—	92	40
			ohne Kontinuität	—	—	—	—	—	—	—		
ORE>=	Mit einfacher Genauigkeit	nicht ausgeführt		1,4	1,1	0,55	0,55	—	—	—	0,55	0,24
		ausgeführt	mit Kontinuität	234	176	88	34	—	—	—	92	40
			ohne Kontinuität	230	172	86	86	—	—	—		
	Mit doppelter Genauigkeit	nicht ausgeführt		—	—	—	—	—	—	—	—	—
		ausgeführt	mit Kontinuität	—	—	—	—	—	—	—	92	40
			ohne Kontinuität	—	—	—	—	—	—	—		
LD$=	mit Kontinuität			97	73	37	37	—	—	—	38	16
	ohne Kontinuität			81	61	31	31	—	—	—	34	15
AND$=	nicht ausgeführt			1,4	1,1	0,55	0,55	—	—	—	0,56	0,23
	ausgeführt	mit Kontinuität		96	72	36	36	—	—	—	39	17
		ohne Kontinuität		81	61	31	31	—	—	—	32	14
OR$=	nicht ausgeführt			1,4	1,1	0,55	0,55	—	—	—	0,56	0,24
	ausgeführt	mit Kontinuität		97	73	37	37	—	—	—	40	17
		ohne Kontinuität		80	60	30	30	—	—	—	33	14
LD$<>	mit Kontinuität			83	62	31	31	—	—	—	32	14
	ohne Kontinuität			97	73	37	37	—	—	—	40	17
AND$<>	nicht ausgeführt			1,4	1,1	0,55	0,55	—	—	—	0,56	0,23
	ausgeführt	mit Kontinuität		80	60	30	30	—	—	—	33	14
		ohne Kontinuität		96	72	36	36	—	—	—	39	17
OR$<>	nicht ausgeführt			1,4	1,1	0,55	0,55	—	—	—	0,56	0,24
	ausgeführt	mit Kontinuität		81	61	31	31	—	—	—	32	14
		ohne Kontinuität		96	72	36	36	—	—	—	39	17

Anweisung	Verarbeitung (Operand)		Verarbeitungszeiten (µs)								
			QnA-Serie				System Q				
			Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H
LD$>	mit Kontinuität		83	62	31	31	—	—	—	32	14
	ohne Kontinuität		97	73	37	37	—	—	—	40	17
AND$>	nicht ausgeführt		1,4	1,1	0,55	0,55	—	—	—	0,56	0,23
	ausgeführt	mit Kontinuität	80	60	30	30	—	—	—	33	14
		ohne Kontinuität	96	72	36	36	—	—	—	39	17
OR$>	nicht ausgeführt		1,4	1,1	0,55	0,55	—	—	—	0,56	0,24
	ausgeführt	mit Kontinuität	81	61	31	31	—	—	—	32	14
		ohne Kontinuität	96	72	36	36	—	—	—	39	17
LD$<=	mit Kontinuität		97	73	37	37	—	—	—	40	17
	ohne Kontinuität		81	61	31	31	—	—	—	32	14
AND$<=	nicht ausgeführt		1,4	1,1	0,55	0,55	—	—	—	0,56	0,23
	ausgeführt	mit Kontinuität	96	72	36	36	—	—	—	39	17
		ohne Kontinuität	81	61	31	31	—	—	—	32	14
OR$<=	nicht ausgeführt		1,4	1,1	0,55	0,55	—	—	—	0,56	0,24
	ausgeführt	mit Kontinuität	97	73	37	37	—	—	—	40	17
		ohne Kontinuität	80	60	30	30	—	—	—	33	14
LD$<	mit Kontinuität		81	61	31	31	—	—	—	32	14
	ohne Kontinuität		97	73	37	37	—	—	—	40	17
AND$<	nicht ausgeführt		1,4	1,1	0,55	0,55	—	—	—	0,56	0,23
	ausgeführt	mit Kontinuität	80	60	30	30	—	—	—	32	14
		ohne Kontinuität	96	72	36	36	—	—	—	39	16
OR$<	nicht ausgeführt		1,4	1,1	0,55	0,55	—	—	—	0,56	0,24
	ausgeführt	mit Kontinuität	81	61	31	31	—	—	—	32	14
		ohne Kontinuität	96	72	36	36	—	—	—	39	16
LD$>=	mit Kontinuität		97	73	37	37	—	—	—	40	17
	ohne Kontinuität		81	61	31	31	—	—	—	32	14
AND$>=	nicht ausgeführt		1,4	1,1	0,55	0,55	—	—	—	0,56	0,23
	ausgeführt	mit Kontinuität	96	72	36	36	—	—	—	39	16
		ohne Kontinuität	81	61	31	31	—	—	—	32	14
OR$>=	nicht ausgeführt		1,4	1,1	0,55	0,55	—	—	—	0,56	0,24
	ausgeführt	mit Kontinuität	97	73	37	37	—	—	—	39	17
		ohne Kontinuität	80	60	30	30	—	—	—	32	14
BKCMP= BKCMP=P	n = 1		120	90	45	45	130	105	97	48	21
	n = 96		367	276	138	138	205	175	165	142	61
BKCMP<> BKCMP<>P	n = 1		123	92	46	46	130	105	98	48	21
	n = 96		346	260	130	130	210	180	165	150	65
BKCMP> BKCMP>P		n = 1	123	92	96	96	130	105	97	48	21
		n = 96	366	275	138	138	210	180	165	142	61
BKCMP>= BKCMP>=P		n = 1	121	91	46	46	130	105	98	48	21
		n = 96	386	290	145	145	205	175	165	150	65
BKCMP< BKCMP<P		n = 1	121	91	96	96	130	105	98	48	21
		n = 96	366	275	138	138	210	180	165	158	68
BKCMP<= BKCMP<=P		n = 1	121	91	46	46	130	105	97	48	21
		n = 96	348	261	131	131	205	175	165	150	65
+ (s,d) +P (s,d)			2,4	1,8	0,9	0,9	1,0	0,80	0,50	0,39	0,17
+ (s1,s2,d) +P (s1,s2,d)			2,7	2,0	1,0	1,0	1,2	0,96	0,60	0,47	0,20
- (s,d) -P (s,d)			2,4	1,8	0,9	0,9	1,0	0,80	0,50	0,39	0,17
- (s1,s2,d) -P (s1,s2,d)			2,6	2,0	1,0	1,0	1,2	0,96	0,60	0,47	0,20

Anweisung	Verarbeitung (Operand)	Verarbeitungszeiten (µs)								
		QnA-Serie				System Q				
		Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H
D+ (s,d) D+P (s,d)		2,8	2,1	1,1	1,1	1,3	1,04	0,65	0,71	,031
D+ (s1,s2,d) D+P (s1,s2,d)		3,2	2,4	1,2	1,2	1,5	1,2	0,75	0,79	0,34
D-(s,d) D-P(s,d)		2,8	2,1	1,1	1,1	1,3	1,04	0,65	0,71	0,30
D- (s1,s2,d) D-P (s1,s2,d)		3,2	2,4	1,2	1,2	1,5	1,2	0,75	0,79	0,34
x (s1,s2,d) xP (s1,s2,d)		2,8	2,1	1,1	1,1	1,1	0,88	0,55	0,47	0,20
/ (s1,s2,d) /P (s1,s2,d)		6,8	5,1	2,6	2,6	19	16	15	2,7	1,2
Dx (s1,s2,d) DxP (s1,s2,d)		20	15	7,5	7,5	41	34	31	7,9	3,4
D/ (s1,s2,d) D/P (s1,s2,d)		36	27	13,5	13,5	28	23	21	14	6,1
B+ (s,d) B+P (s,d)		5,5	4,1	2,1	2,1	34	28	26	2,2	1,0
B+ (s1,s2,d) B+P (s1,s2,d)		13	9,6	4,8	4,8	47	39	37	5,0	2,2
B- (s,d) B-P (s,d)		5,2	3,9	2,0	2,0	34	28	26	2,0	0,9
B- (s1,s2,d) B-P (s1,s2,d)		13	9,4	4,7	4,7	48	40	38	4,9	2,1
DB+ (s,d) DB+P (s,d)		29	22	11	11	58	48	44	12	5,0
DB+ (s1,s2,d) DB+P (s1,s2,d)		32	24	12	12	60	49	46	12	5,3
DB- (s,d) DB-P (s,d)		29	22	11	11	59	48	45	11	4,8
DB- (s1,s2,d) DB-P (s1,s2,d)		32	24	12	12	60	51	45	12	5,2
Bx (s1, s2, d) BxP (s1, s2, d)		9,4	7,1	3,6	3,6	42	35	33	3,7	1,6
B/ (s1, s2, d) B/P (s1, s2, d)		9,4	7,1	3,6	3,6	48	40	37	3,8	1,6

Anweisung	Verarbeitung (Operand)		Verarbeitungszeiten (µs)								
			QnA-Serie				System Q				
			Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H
DBx (s1, s2, d) DBxP (s1, s2, d)			62	46	23	23	140	120	110	24	10
DB/ (s1, s2, d) DB/P (s1,s2,d)			69	52	26	26	83	69	65	27	12
E+ (s, d) E+P (s, d)	Mit einfacher Genauigkeit	s = 0, d = 0	54	40	20	35	—	—	—	1,8	0,78
		$s = 2^{127}$, $d = 2^{127}$	524	394	197						
	Mit doppelter Genauigkeit	s = 0, d = 0	—	—	—	—	—	—	—	203	87
		$s = 2^{127}$, $d = 2^{127}$	—	—	—	—					
E+ (s1, s2, d) E+P (s1, s2, d)	Mit einfacher Genauigkeit	s1 = 0, s2 = 0	54	40	20	35	—	—	—	2,4	1,1
		$s1 = 2^{127}$, $s2 = 2^{127}$	524	394	197						
	Mit doppelter Genauigkeit	s = 0, d = 0	—	—	—	—	—	—	—	209	90
		$s = 2^{127}$, $d = 2^{127}$	—	—	—	—					
E- (s, d) E-P (s, d)	Mit einfacher Genauigkeit	s = 0, d = 0	54	40	20	35	—	—	—	1,8	0,78
		$s = 2^{127}$, $d = 2^{127}$	515	387	194						
	Mit doppelter Genauigkeit	s = 0, d = 0	—	—	—	—	—	—	—	202	87
		$s = 2^{127}$, $d = 2^{127}$	—	—	—	—					
E- (s1, s2, d) E-P (s1, s2, d)	Mit einfacher Genauigkeit	s1 = 0, s2 = 0	55	41	21	36	—	—	—	2,4	1,1
		$s1 = 2^{127}$, $s2 = 2^{127}$	520	391	146						
	Mit doppelter Genauigkeit	s = 0, d = 0	—	—	—	—	—	—	—	210	90
		$s = 2^{127}$, $d = 2^{127}$	—	—	—	—					
Ex (s1, s2, d) ExP (s1, s2, d)	Mit einfacher Genauigkeit	s1 = 0, s2= 0	55	41	21	36	—	—	—	2,4	1,1
		$s1 = 2^{127}$, $s2 = 2^{127}$	567	426	218						
	Mit doppelter Genauigkeit	s = 0, d = 0	—	—	—	—	—	—	—	222	96
		$s = 2^{127}$, $d = 2^{127}$	—	—	—	—					
E/ (s1, s2, d) E/P (s1, s2, d)	Mit einfacher Genauigkeit	s1 = 0, s2 = 1	149	112	56	37	—	—	—	12	5,2
		$s1 = 2^{127}$, $s2 = -2^{126}$	1109	834	417						
	Mit doppelter Genauigkeit	s = 0, d = 0	—	—	—	—	—	—	—	369	159
		$s1 = 2^{127}$, $s2 = -2^{126}$	—	—	—	—					
$+ (s, d) $+P (s, d)			179	134	67	67	—	—	—	68	29
$+ (s1, s2, d) $+P (s1, s2, d)			206	155	78	78	—	—	—	81	35
INC INCP			1,9	1,4	0,7	0,7	0,70	0,56	0,35	0,32	0,14
DINC DINCP			2,3	1,7	0,9	0,9	0,90	0,72	0,45	0,47	0,20
DEC DECP			1,9	1,4	0,7	0,7	0,70	0,56	0,35	0,32	0,14
DDEC DDECP			2,3	1,7	0,9	0,9	0,90	0,72	0,45	0,47	0,20
BCD BCDP			2,7	2,0	1,0	1,0	20	16	15	1,1	0,48
DBCD DBCDP			7,9	5,9	3,0	3,0	26	21	20	3,2	1,4
BIN BINP			2,7	2,0	1,0	1,0	19	16	15	1,0	0,44
DBIN DBINP			4,8	3,6	1,8	1,8	22	18	17	1,9	0,82

Anweisung	Verarbeitung (Operand)		Verarbeitungszeiten (µs)								
			QnA-Serie				System Q				
			Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H
INT INTP	Mit einfacher Genauigkeit	s = 0	20	15	7,5	7,5	—	—	—	3,2	1,4
		s = 32766,5	54	40	20	20					
	Mit doppelter Genauigkeit	s = 0	—	—	—	—	—	—	—	22	9,3
		s = 32766,5	—	—	—	—					
DINT DINTP	Mit einfacher Genauigkeit	s = 0	20	15	7,5	7,5	—	—	—	2,5	1,1
		s = 1234567890,3	59	44	22	22					
	Mit doppelter Genauigkeit	s = 0	—	—	—	—	—	—	—	24	10
		s = 1234567890,3	—	—	—	—					
FLT FLTP	Mit einfacher Genauigkeit	s = 0	27	20	10	10	—	—	—	2,1	0,92
		s = $7FFF_H$	55	41	21	21					
	Mit doppelter Genauigkeit	s = 0	—	—	—	—	—	—	—	22	9,6
		s = $7FFF_H$	—	—	—	—					
DFLT DFLTP	Mit einfacher Genauigkeit	s = 0	28	21	11	11	—	—	—	2,1	0,88
		s = $7FFFFFFF_H$	56	42	21	21					
	Mit doppelter Genauigkeit	s = 0	—	—	—	—	—	—	—	26	11
		s = $7FFFFFFF_H$	—	—	—	—					
DBL DBLP			12	8,6	4,3	4,3	19	16	15	4,5	1,9
WORD WORDP			12	9,0	4,5	4,5	23	19	17	4,7	2,0
GRY GRYP			12	9,0	4,5	4,5	19	16	15	4,7	2,0
DGRY DGRYP			14	10	5,0	5,0	23	19	17	5,3	2,3
GBIN GBINP			46	34	17	17	52	42	40	18	7,7
DGBIN DGBINP			83	62	31	31	110	88	84	32	14
NEG NEGP			9,3	7	3,5	3,5	16	13	12	3,6	1,6
DNEG DNEGP			11	8,2	4,1	4,1	19	17	15	4,3	1,8
ENEG ENEGP			9,8	7,4	3,7	3,7	—	—	—	3,9	1,7
BKBCD (s, d, n) BKBCDP (s, d, n)	n = 1		102	76	38	38	78	63	57	38	17
	n = 96		272	204	102	102	315	275	250	99	43
BKBIN (s, d, n) BKBINP (s, d, n)	n = 1		102	76	38	38	74	61	57	38	17
	n = 96		272	204	102	102	285	255	230	99	43
MOV MOVP	s = D0, d = D1		0,7	0,5	0,3	0,3	0,70	0,56	0,35	0,24	0,10
	s = D0, d = J1/W1		392[1]	305[1]	176[1]	176[1]	155	130	120	140	160
			391[2]	299[2]	165[2]	165[2]					
DMOV DMOVP	s = K4X0, d = D1		2,4	1,8	0,9	0,9	0,90	0,72	0,45	0,47	0,20
	s = K4X0, d = J1/W1		400[1]	313[1]	183[1]	183[1]	165	135	120	147	64
			395[2]	301[2]	167[2]	167[2]					
EMOV EMOVP			12	8,6	4,3	4,3	—	—	—	0,63	0,27
$MOV $MOVP			100	75	38	38	46	38	35	40	17
							98	80	73		
CML CMLP			2,0	1,5	0,8	0,8	0,70	0,56	0,35	0,40	0,17

[1] Diese Verarbeitungszeiten gelten bei Verwendung des Hauptbaugruppenträgers A38B/A1S38B und eines Erweiterungsbaugruppenträgers.
[2] Diese Verarbeitungszeiten gelten bei Einsatz des Hauptbaugruppenträgers A38HB/A1S38HB.

Anweisung	Verarbeitung (Operand)	Verarbeitungszeiten (µs)								
		QnA-Serie				System Q				
		Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H
DCML DCMLP		2,4	1,8	0,9	0,9	0,90	0,72	0,45	0,55	0,24
BMOV (s, d, n)	n = 1	43	32	16	16	27	21	20	17	7,1
BMOVP (s, d, n)	n = 96	81	61	31	31	72	62	53	32	14
FMOV (s, d, n)	n = 1	18	13	6,5	6,5	23	19	17	6,7	2,9
FMOVP (s, d, n)	n = 96	36	27	14	14	48	41	36	14	6,1
XCH XCHP		3,1	2,3	1,2	1,2	7,6	6,3	5,7	1,3	0,54
DXCH DXCHP		3,1	2,3	1,2	1,2	9,5	8,0	7,1	1,3	0,54
BXCH (d1, d2, n)	n = 1	77	58	29	29	62	51	48	31	13
BXCHP (d1, d2, n)	n = 96	213	160	80	80	165	140	125	84	36
SWAP SWAPP		9,2	6,9	3,5	3,5	17	14	13	3,7	1,6
CJ		7,8	5,8	2,9	2,9	10	8,5	8,1	3,2	1,4
SCJ		7,8	5,8	2,9	2,9	10	8,5	8,1	3,2	1,4
JMP		8,0	6,0	3,0	3,0	11	8,5	8,1	3,2	1,4
GOEND		2,0	1,5	0,75	0,75	3,3	2,9	2,8	0,39	0,34
EI		3,1	2,3	1,2	1,2	14	11	11	1,3	0,54
DI		2,3	1,7	0,9	0,9	13	12	11	0,95	0,41
IMASK		8,1	6,5	3,3	3,3	41	34	35	11	4,6
IRET		4,0	3,0	1,5	1,5	205	170	155	1,6	0,68
RFS RFSP	s= X, n = 1	31,3	23,4	11,7	11,7	55	46	43	6,7	4,7
	s = Y, n = 1					54	45	41		
	s = X, n = 96	97,6	72,8	36,4	36,4	79	64	59	19	13
	s = Y, n =96					73	61	56		
UDCNT1		42,6	31,8	15,9	15,9	—	—	—	15	6,5
UDCNT2		44,6	33,3	16,7	16	—	—	—	16	6,8
TTMR		25,9	19,3	9,7	9,7	—	—	—	10	4,4
STMR		41,7	31,1	15,6	15,6	—	—	—	20	7,1
ROTC		66,1	49,3	24,7	24,7	—	—	—	26	11
RAMP		45,4	33,9	17,0	17,0	—	—	—	18	7,7
SPD		48,9	36,5	18,3	18,3	—	—	—	19	8,3
PLSY		26,9	20,1	10,1	10,1	—	—	—	10	4,5
PWM		32,8	24,5	12,3	12,3	—	—	—	9,1	3,9
MTR		29,2	21,8	10,9	10,9	—	—	—	11	4,9
WAND (s, d) WANDP (s, d)		2,4	1,8	0,9	0,9	1,0	0,80	0,50	0,39	0,17
WAND (s1, s2, d) WANDP (s1, s2, d)		9,5	7,1	3,6	3,6	1,2	0,96	0,60	0,47	0,20
DAND (s, d) DANDP (s, d)		3,0	2,3	1,2	1,2	1,3	1,04	0,65	0,71	0,31
DAND (s1, s2, d) DANDP (s1, s2, d)		19	14	7,0	7,0	1,5	1,2	0,75	0,79	0,34

Anweisung	Verarbeitung (Operand)	Verarbeitungszeiten (µs)								
		QnA-Serie				System Q				
		Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H
BKAND (s1, s2, d, n)	n = 1	89	67	34	34	110	87	79	36	16
BKANDP (s1, s2, d, n)	n = 96	184	138	69	69	185	155	140	74	32
WOR (s, d) WORP (s, d)		2,4	1,8	0,9	0,9	1,0	0,80	0,50	0,40	0,17
WOR (s1, s2, d) WORP (s1, s2, d)		9,5	7,1	3,6	3,6	1,2	0,96	0,60	0,47	0,20
DOR (s, d) DORP (s, d)		3,0	2,3	1,2	1,2	1,3	1,04	0,65	0,71	0,31
DOR (s1, s2, d) DORP (s1, s2, d)		19	14	7,0	7,0	1,5	1,2	0,75	0,79	0,34
BKOR (s1, s2, d, n)	n = 1	89	67	34	34	110	87	81	36	16
BKORP (s1, s2, d, n)	n = 96	184	138	69	69	185	155	140	74	32
WXOR (s, d) WXORP (s, d)		2,4	1,8	0,9	0,9	1,0	0,80	0,50	0,39	0,17
WXOR (s1, s2, d) WXORP (s1, s2, d)		17,2	7,1	3,6	3,6	1,2	0,96	0,60	0,47	0,20
DXOR (s, d) DXORP (s, d)		3,0	2,3	1,2	1,2	1,3	1,04	0,65	0,71	0,31
DXOR (s1, s2, d) DXORP (s1, s2, d)		19	14	7,0	7,0	1,5	1,2	0,75	0,79	0,34
BKXOR (s1, s2, d, n)	n = 1	89	67	34	34	110	87	81	36	16
BKXORP (s1, s2, d, n)	n = 96	184	138	69	69	183	155	140	74	32
WXNR (s, d) WXNRP (s, d)		2,4	1,8	0,9	0,9	1,0	0,80	0,50	0,40	0,17
WXNR (s1, s2, d) WXNRP (s1, s2, d)		9,5	7,1	3,6	3,6	1,2	0,96	0,60	0,47	0,20
DXNR (s,d) DXNRP (s,d)		3,0	2,3	1,2	1,2	1,3	1,04	0,65	0,71	0,31
DXNR (s1,s2,d) DXNRP (s1,s2,d)		24	18	9	9	1,5	1,2	0,75	0,79	0,34
BKXNR (s1, s2, d, n)	n = 1	89	67	34	34	110	87	82	36	16
BKXNR (s1, s2, d, n)	n = 96	184	138	69	69	185	155	140	74	32

Anweisung	Verarbeitung (Operand)	Verarbeitungszeiten (µs)								
		QnA-Serie				System Q				
		Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H
ROR (d, n)	n = 1	5,0	3,8	1,9	1,9	13	11	9,7	2,0	0,85
RORP (d, n)	n = 15	5,0	3,8	1,9	1,9	13	11	9,7	2,0	0,85
RCR (d, n)	n = 1	4,0	3,0	1,5	1,5	15	12	12	1,6	0,68
RCRP (d, n)	n = 15	4,0	3,0	1,5	1,5	15	13	12	1,6	0,68
ROL (d, n)	n = 1	5,0	3,8	1,9	1,9	13	11	10	2,0	0,85
ROLP (d, n)	n = 15	5,0	3,8	1,9	1,9	13	11	10	2,0	0,85
RCL (d, n)	n = 1	4,0	3,0	1,5	1,5	15	13	12	1,6	0,68
RCLP (d, n)	n = 15	4,0	3,0	1,5	1,5	16	13	12	1,6	0,68
DROR (d, n)	n = 1	9,8	7,4	3,7	3,7	15	12	12	3,9	1,7
DRORP (d, n)	n = 31	10	7,8	3,9	3,9	15	13	12	4,0	1,7
DRCR (d, n)	n = 1	11	8,1	4,1	4,1	17	14	14	4,3	1,8
DRCRP (d, n)	n = 31	11	8,3	4,2	4,2	18	16	15	4,3	1,9
DROL (d, n)	n = 1	9,8	7,4	3,7	3,7	14	13	12	3,9	1,7
DROLP (d, n)	n = 31	10	7,8	3,9	3,9	14	13	12	4,0	1,7
DRCL (d, n)	n = 1	11	8,1	4,1	4,1	18	15	14	4,3	1,8
DRCLP (d, n)	n = 31	11	8,3	4,2	4,2	20	17	16	4,3	1,9
SFR (d, n)	n = 1	4,4	3,3	1,7	1,7	13	10	9,7	1,7	0,75
SFRP (d, n)	n = 15	5,0	3,8	1,9	1,9	13	11	9,5	2,0	0,85
SFL (d, n)	n = 1	4,4	3,3	1,7	1,7	12	10	9,5	1,7	0,75
SFLP (d, n)	n = 15	5,0	3,8	1,9	1,9	12	9,8	9,5	2,0	0,85
BSFLR (d, n)	n = 1	51	38	19	19	42	35	33	20	8,6
BSFLRP (d, n)	n = 96	60	45	23	23	69	58	54	24	10
BSFL (d, n)	n = 1	49	37	19	19	41	34	32	20	8,5
BSFLP (d, n)	n = 96	58	44	22	22	63	53	50	23	10
DSFR (d, n)	n = 1	3,6	2,6	1,3	1,3	19	16	15	1,3	0,58
DSFRP (d, n)	n = 96	63	47	24	24	71	61	53	25	11
DSFL (d, n)	n = 1	3,6	2,6	1,3	1,3	19	16	15	1,3	0,58
DSFLP (d, n)	n = 96	65	49	25	25	70	60	52	26	11
BSET (d, n)	n = 1	20	15	7,5	7,5	27	22	20	7,6	3,3
BSETP (d, n)	n = 15	20	15	7,5	7,5	27	22	20	7,6	3,3
BRST (d, n)	n = 1	20	15	7,5	7,5	27	22	21	7,6	3,3
BRSTP (d, n)	n = 15	20	15	7,5	7,5	27	22	21	7,6	3,3
TEST (s1, s2, d) TESTP (s1, s2, d)		21	16	8,0	8,0	35	30	27	8,2	3,5

Anweisung	Verarbeitung (Operand)			Verarbeitungszeiten (µs)								
				QnA-Serie				System Q				
				Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H
DTEST (s1, s2, d) DTESTP (s1, s2, d)				24	18	9,0	9,0	37	31	28	9,2	3,9
BKRST (s, n) BKRST (s, n)	n = 1			45	34	17	17	49	41	38	18	7,8
	n = 96			49	37	19	19	64	54	50	19	8,2
SER (s1, s2, d, n) SERP (s1, s2, d, n)	n = 1	Übereinstimmung		58	44	22	22	56	54	42	22	9,6
		keine Übereinstimmung		57	43	21	21	56	54	42	21	8,9
	n = 96	Übereinstimmung		293	220	110	110	280	240	220	115	49
		keine Übereinstimmung		340	256	128	128	280	240	220	133	57
DSER (s1, s2, d, n) DSERP (s1, s2, d, n)	n = 1	Übereinstimmung		61	46	23	23	71	67	53	23	9,9
		keine Übereinstimmung		58	44	22	22	71	67	54	23	9,7
	n = 96	Übereinstimmung		354	266	133	133	495	415	375	142	61
		keine Übereinstimmung		354	266	133	133	500	415	375	132	57
SUM SUMP	s = 0			9,8	7,4	3,7	3,7	32	26	25	3,9	1,7
	$s = FFFF_H$							27	22	21		
DSUM DSUMP	s = 0			12	9,0	4,5	4,5	54	44	42	4,7	2,0
	$s = FFFFFFFF_H$			31	23	12	12	54	44	42	12	5,0
DECO (s, d, n) DECOP (s, d, n)	n = 2			48	36	18	18	60	50	46	20	8,6
	n = 8			62	47	24	24	80	65	61	27	12
ENCO (s, d, n) ENCOP (s, d, n)	n = 2		M1 = EIN	52	39	20	20	66	55	51	21	9,1
			M4 = EIN	52	39	20	20	66	54	51	21	9,1
	n = 8		M1 = EIN	65	49	25	25	90	76	71	28	12
			M256 = EIN	65	49	25	25	76	74	71	26	11
SEG SEGP				3,2	2,4	1,2	1,2	8,0	6,8	6,1	1,3	0,54
DIS (s, d, n) DISP (s, d, n)	n = 1			46	34	17	17	47	39	36	18	7,7
	n = 4			51	38	19	19	53	43	40	19	8,3
UNI (s, d, n) UNIP (s, d, n)	n = 1			53	40	20	20	54	44	41	21	8,9
	n = 4			57	43	22	22	60	49	46	23	9,7
NDIS (s1, d, s2) NDISP (s1, d, s2)				104	78	39	39	92	76	38	41	18
NUNI (s1, d, s2) NUNIP (s1, d, s2)				105	79	40	40	47	39	36	42	18
WTOB (s, d, n) WTOBP (s, d, n)	n = 1			125	94	47	47	56	46	42	47	20
	n = 96			257	193	97	97	190	155	145	99	43
BTOW (s, d, n) BTOWP (s, d, n)	n = 1			121	91	46	46	56	46	42	45	19
	n = 96			233	175	88	88	190	155	145	89	38
MAX (s, d, n) MAXP (s, d, n)	n = 1			43	32	16	16	48	40	36	17	7,1
	n = 96			318	239	120	120	300	240	235	136	59
MIN (s, d, n) MINP (s, d, n)	n = 1			43	32	16	16	48	40	36	17	7,1
	n = 96			436	326	163	163	300	240	235	159	69

Anweisung	Verarbeitung (Operand)	Verarbeitungszeiten (µs)								
		QnA-Serie				System Q				
		Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H
DMAX (s, d, n)	n = 1	71	53	27	27	52	43	39	27	12
DMAXP (s, d, n)	n = 96	427	321	161	161	600	490	460	181	78
DMIN (s, d, n)	n = 1	71	53	27	27	52	43	39	27	12
DMINP (s, d, n)	n = 96	268	201	101	101	585	475	445	112	48
SORT (s1, n, s2, d1, d2)	n = 1	43	32	16	16	66	55	50	16	7,1
	n = 96	40*	30*	15*	15*	105	86	80	14	6,2
DSORT (s1, n, s2, d1, d2)	n = 1	44	33	17	17	98	57	52	17	7,1
	n = 96	43*	32*	16*	16*	115	96	88	16	6,8
*Verlängerung der Zykluszeit durch die Ausführung der Anweisung										
WSUM (s, d, n)	n = 1	41,5	31,1	15,6	15,6	52	43	40	16,4	7,1
WSUMP (s, d, n)	n = 96	173,2	129,9	65	65	175	140	135	68,4	29,5
DWSUM (s, d, n)	n = 1	47,9	35,9	18	18	61	51	46	18,9	8,2
DWSUMP (s, d, n)	n = 96	330	247,5	123,8	123,8	515	420	395	130,4	56,1
FOR	n = 0	5,2	3,9	2,0	2,0	11	8,9	8,1	2,3	1,0
NEXT		8,0	6,0	3,0	3,0	8,8	7,3	6,8	3,3	1,4
BREAK BREAKP		26	19	9,5	9,5	37	30	28	11	4,6
CALL (pn)	interner Dateipointer	5,1	3,8	1,9	1,9	17	14	13	2,1	0,88
CALLP (pn)	allgemeiner Pointer	85	64	32	32				33	14
CALL (pn s1 bis s5) CALLP (pn s1 bis s5)		348	261	131	131	245	200	190	135	58
RET	zurück zum Ursprungsprogramm	7,5	5,6	2,8	2,8	16	13	12	2,9	1,3
	zurück zum anderen Programm	51	38	19	19	—	—	—	20	8,5
FCALL (pn)	interner Dateipointer	8,8	6,6	3,3	3,3	29	24	22	3,6	1,6
FCALLP (pn)	allgemeiner Pointer	48	36	18	18				20	8,7
FCALL (pn S1 bis S5) FCALLP (pn S1 bis S5)		388	254	127	127	250	205	190	134	57
ECALL (pn) ECALLP (pn)		187	140	70	70	—	—	—	77	33
ECALL (pn S1 bis S5) ECALLP (pn S1 bis S5)		515	387	144	144	—	—	—	162	70
EFCALL (pn) EFCALLP (pn)		188	141	71	71	—	—	—	78	34
EFCALL (pn S1 bis S5) EFCALLP (pn S1 bis S5)		516	388	194	194	—	—	—	200	86
COM		137	103	52	52	110	77	72	55	16
IX		31	23	12	12	65	54	51	12	5,2
IXEND		12	8,9	4,5	4,5	30	26	25	4,7	2,0
IXDEV	Anzahl der Kontakte: 1	127	95	46	46	145	120	110	48	21
	Anzahl der Kontakte: 14	238	179	85	85	770	630	585	93	40

Anweisung	Verarbeitung (Operand)		Verarbeitungszeiten (µs)								
			QnA-Serie				System Q				
			Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H
IXSET	Anzahl der Kontakte: 1		127	95	46	46	145	120	110	48	21
	Anzahl der Kontakte: 14		238	179	85	85	770	630	585	93	40
FIFW FIFWP	Anzahl der Datensätze: 1		27	20	10	10	36	32	28	11	4,5
	Anzahl der Datensätze: 96										
FIFR FIFRP	Anzahl der Datensätze: 1		34	25	13	13	45	41	36	13	5,6
	Anzahl der Datensätze: 96		79	59	30	30	93	82	70	32	14
FPOP FPOPP	Anzahl der Datensätze: 1		46	34	17	17	40	37	32	16	7,0
	Anzahl der Datensätze: 96										
FINS FINSP	Anzahl der Datensätze: 1		48	36	18	18	53	44	38	20	8,4
	Anzahl der Datensätze: 96		96	72	36	36	100	89	76	36	15
FDEL FDELP	Anzahl der Datensätze: 1		47	35	18	18	60	50	43	19	7,5
	Anzahl der Datensätze: 96		97	73	37	37	110	95	82	39	15
PR	SM701 ON	1 Zeichen	83	62	31	31	—	—	—	33	11
		32 Zeichen	123	92	46	46				48	18
	SM701 OFF		54	40	20	20				21	7,8
PRC			400	301	151	151	—	—	—	181	16
LED	bei Anzeige		223	167	84	84	—	—	—	—	—
	Anzeige komplett		79	59	30	30					
LEDC	bei Anzeige		559	420	210	210	—	—	—	—	—
	Anzeige komplett		413	310	155	155					
LEDR	keine Anzeige → keine Anzeige		18	13	6,5	6,5	—	—	—	0,40	0,17
	LED Anweisung ausführen → keine Anzeige		205	154	77	77				103	44
CHKST			15	11	5,5	5,5	—	—	—	5,8	2,5
CHK (Fehler-kontrolle)	kein Fehler an Kontakt 1		61	46	23	23	—	—	—	24	10
	kein Fehler an Kontakt 150		4232	3182	1591	1591				1676	721
	Fehler an Kontakt 1		224	168	84	84				88	38
CHKCIR	10 Fehlerprüfnetzwerke		15	11	5,5	5,5	—	—	—	5,8	2,5
SLT	alle internen Operanden		2399	1804	902	902	—	—	—	—	—
	File-Register 8 kByte		7254	5454	2727	2727					
	Beendung der SLT Ausführung		15	11	5,5	5,5					
SLTR			1,1	0,8	0,4	0,4	—	—	—	—	—
STRA	Start		47	35	18	18	—	—	—	—	—
	Beendung der STRA Ausführung		15	11	5,5	5,5					
STRAR			1,1	0,8	0,4	0,4	—	—	—	—	—
PTRA			15	11	5,5	5,5	—	—	—	—	—
PTRAR			15	11	5,5	5,5	—	—	—	—	—
PTRAEXE PTRAEXEP	Anweisungsausführung		1,6	1,2	0,6	0,6	—	—	—	—	—
	Programmüberwachung		169	127	64	64					
BINDA BINDAP	s = 1		40	30	15	15	—	—	—	15	6,7
	s = -32768		60	45	23	23				24	10
DBINDA DBINDAP	s = 1		63	47	44	44	—	—	—	43	18
	s = -2147483648		217	163	82	82				86	37
BINHA BINHAP	s = 1		46	34	17	17	—	—	—	18	7,7
	$s = FFFF_H$		48	36	18	18				19	8,2
DBINHA DBINHAP	s = 1		59	44	22	22	—	—	—	23	10
	$s = FFFFFFFF_H$		62	46	23	23				24	10
BCDDA BCDDAP	s = 1		58	43	22	22	—	—	—	23	9,8
	s = 9999		54	40	20	20				21	8,9
DBCDDA DBCDDAP	s = 1		61	46	23	23	—	—	—	22	9,5
	s = 99999999		75	56	28	28				29	13
DABIN DABINP	s = 1		133	100	50	50	—	—	—	57	25
	s = -32768		145	109	55	55				58	28
DDABIN DDABINP	s = 1		241	181	91	91	—	—	—	92	40
	s = -2147483648		268	201	101	101				106	46

Anweisung	Verarbeitung (Operand)		Verarbeitungszeiten (µs)								
			QnA-Serie				System Q				
			Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H
HABIN HABINP	s = 1		32	24	12	12	—	—	—	13	5,8
	$s = FFFF_H$		38	28	14	14				15	6,4
DHABIN DHABINP	s = 1		54	40	20	20	—	—	—	22	9,5
	$s = FFFFFFFF_H$		63	47	24	24				25	11
DABCD DABCDP	s = 1		36	27	14	14	—	—	—	16	6,9
	s = 9999		42	31	16	16				17	7,2
DDABCD DDABCDP	s = 1		63	47	24	24	—	—	—	25	11
	s = 99999999		75	56	28	28				29	13
COMRD COMRDP			36	27	14	14	—	—	—	40	16
LEN LENP	1 Zeichen		48	36	18	18	—	—	—	18	8,0
	96 Zeichen		229	172	86	86				86	37
STR STRP			132	99	50	50	—	—	—	53	23
DSTR DSTRP			285	214	107	107	—	—	—	123	53
VAL VALP			258	194	97	97	—	—	—	95	41
DVAL DVALP			402	302	151	151	—	—	—	166	72
ESTR ESTRP			1337	1005	503	503	—	—	—	564	243
EVAL EVALP	Gleitkommaformat		242	182	91	91	—	—	—	100	43
	Exponentialformat		306	230	115	115				127	55
ASC (s, d, n) ASCP (s, d, n)	n = 1		164	123	62	62	—	—	—	64	28
	n = 96		780	586	293	293				289	125
HEX (s, d, n) HEXP (s, d, n)	n = 1		161	121	61	61	—	—	—	60	26
	n = 96		826	621	311	311				343	148
RIGHT (s, d, n) RIGHTP (s, d, n)	n = 1		131	98	49	49	—	—	—	49	21
	n = 96		354	266	133	133				131	56
LEFT (s, d, n) LEFTP (s, d, n)	n = 1		129	97	49	49	—	—	—	50	21
	n = 96		354	266	133	133	—	—	—	131	56
MIDR MIDRP			141	106	53	53	—	—	—	53	23
MIDW MIDWP			341	256	128	128	—	—	—	128	55
INSTR INSTRP	keine Übereinstimmung		156	117	59	59	—	—	—	58	25
	Übereinstimmung	erstes	141	106	53	53				55	24
		letztes	155	116	58	58				58	25
EMOD EMODP			1313	987	494	494	—	—	—	527	227
EREXP EREXPP			4423	3325	1663	1663	—	—	—	1656	713
SIN SINP	Einfache Genauigkeit		4921	3700	1850	35	—	—	—	115	50
	Doppelte Genauigkeit		—	—	—	—				1945	837
COS COSP	Einfache Genauigkeit		6462	4858	2429	35	—	—	—	122	53
	Doppelte Genauigkeit		—	—	—	—				2618	1127
TAN TANP	Einfache Genauigkeit		6515	4898	2449	38	—	—	—	123	53
	Doppelte Genauigkeit		—	—	—	—				2618	1127
ASIN ASINP	Einfache Genauigkeit		890	669	335	44	—	—	—	111	48
	Doppelte Genauigkeit		—	—	—	—				2491	1072

Anweisung	Verarbeitung (Operand)		Verarbeitungszeiten (µs)								
			QnA-Serie				System Q				
			Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H
ACOS ACOSP	Einfache Genauigkeit		801	602	301	44	—	—	—	115	49
	Doppelte Genauigkeit		—	—	—	—				2367	1019
ATAN ATANP	Einfache Genauigkeit		7818	5878	2939	39	—	—	—	157	68
	Doppelte Genauigkeit		—	—	—	—				3140	1352
RAD RADP	Einfache Genauigkeit		465	349	175	31	—	—	—	17	7,2
	Doppelte Genauigkeit		—	—	—	—				24	10
DEG DEGP	Einfache Genauigkeit		492	369	185	31	—	—	—	17	7,2
	Doppelte Genauigkeit		—	—	—	—				23	9,9
SQR SQRP	Einfache Genauigkeit		4520	3398	1699	39	—	—	—	28	12
	Doppelte Genauigkeit		—	—	—	—				1812	780
EXP EXPP	Einfache Genauigkeit	s = -10	5871	4414	2207	37	—	—	—	—	—
		s = 1	5950	4474	2237					129	56
	Doppelte Genauigkeit	s = -10	—	—	—	—				—	—
		s = 1	—	—	—	—				2386	1026
LOG LOGP	Einfache Genauigkeit	s = 1	1191	896	448	37	—	—	—	113	49
		s = 10	6839	5142	2571					—	—
	Doppelte Genauigkeit	s = 1	—	—	—	—				2146	924
		s = 10	—	—	—	—				—	—
RND RNDP			10	7,5	3,8	3,8	—	—	—	3,9	1,7
SRND SRNDP			8,8	6,6	3,3	3,3	—	—	—	3,5	1,5
BSQR BSQRP	s = 0		16	12	6,0	6,0	—	—	—	6,2	2,7
	s = 9999		97	73	37	37				38	16
BDSQR BDSQRP	s = 0		17	13	6,5	6,5	—	—	—	6,2	2,7
	s = 99999999		88	66	33	33				38	16
BSIN BSINP			30	22	11	11	—	—	—	12	5,1
BCOS BCOSP			32	24	12	12	—	—	—	12	5,2
BTAN BTANP			30	22	11	11	—	—	—	12	5,2
BASIN BASINP			52	39	20	20	—	—	—	20	8,7
BACOS BACOSP			53	40	20	20	—	—	—	21	9,0
BATAN BATANP			56	42	21	21	—	—	—	22	9,6
LIMIT LIMITP			24	18	9,0	9,0	34	28	26	10	4,3
DLIMIT DLIMITP			28	21	11	11	41	34	30	11	4,7
BAND BANDP			24	18	9,0	9,0	33	28	25	9,8	4,2
DBAND DBANDP			28	21	11	11	40	34	30	11	4,9
ZONE ZONEP			24	18	9,0	9,0	31	25	24	9,1	3,9
DZONE DZONEP			28	21	11	11	37	29	28	11	4,6
RSET RSETP			19	14	7,0	7,0	—	18	16	6,8	2,9
QDRSET QDRSETP			322	242	121	121	—	—	—	205	88
QCDSET QCDSETP			218	164	82	82	—	—	—	147	63
DATERD DATERDP			36	27	14	14	30	25	23	13	5,5

Anweisung	Verarbeitung (Operand)	Verarbeitungszeiten (µs)								
		QnA-Serie				System Q				
		Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H
DATEWR DATEWRP		42	31	16	16	69	57	54	15	6,4
DATE+ DATE+P	ohne Stellenzuwachs	60	45	23	23	47	39	36	13	5,4
	mit Stellenzuwachs	60	45	23	23	50	42	38	13	5,4
DATE- DATE-P	ohne Stellenzuwachs	59	44	22	22	47	40	36	12	5,2
	mit Stellenzuwachs	60	45	23	23	50	42	38	12	5,2
SECOND SECONDP		27	20	10	10	28	24	22	10	4,5
HOUR HOURP		31	23	12	12	38	32	29	12	5,2
MSG	1 Zeichen	7,2	5,4	2,7	2,7	—	—	—	3,0	1,3
	32 Zeichen	7,4	5,6	2,8	2,8					
PKEY	Lesezeit	51	38	19	19	—	—	—	20	8,6
	kein Empfang	48	36	18	18				19	8,2
PSTOP PSTOPP		122	92	46	46	—	—	—	79	34
POFF POFFP		120	90	45	45	—	—	—	79	34
PSCAN PSCANP		122	92	46	46	—	—	—	75	32
PLOW PLOWP		124	93	47	47	—	—	—	80	34
WDT WDTP		12	8,7	4,4	4,4	18	15	14	5,9	2,6
DUTY		1,6	1,2	0,6	0,6	41	36	32	9,3	4,0
ZRRDB ZRRDBP		19	14	6,9	6,9	—	24	22	7,9	3,4
ZRWRB ZRWRBP		21	16	7,8	7,8	—	27	24	9,4	4,0
ADRSET ADRSETP		13	9,3	4,7	4,7	23	19	18	4,9	2,1
KEY		43,4	32,4	16,2	16,2	—	—	—	17	7,3
ZPUSH ZPUSHP		27,6	20,6	10,3	10,3	38	33	30	11	4,7
ZPOP ZPOPP		12,7	9,5	4,8	4,8	37	31	29	5,1	2,2
EPROMWR EPROMWRP		62,6	46,7	23,4	23,4	—	—	—	—	—
ZCOM		4296,6	3206,4	1603,2	1603,2	105	82	80	691	289
READ		770,6	575,1	287,6	287,6	—	—	—	554	260
SREAD		858,9	641,0	320,5	320,5	—	—	—	588	278
WRITE		791,9	591,0	295,5	295,5	—	—	—	582	273
SWRITE		848,6	633,3	316,6	316,6	—	—	—	625	295
SEND		575,7	429,6	214,8	214,8	—	—	—	—	—
RECV		375,9	280,5	140,3	140,3	—	—	—	—	—
REQ		527,4	393,6	196,8	196,8	—	—	—	—	—
ZNFR		982,1	732,9	366,5	366,5	—	—	—	—	—
ZNTO		989,3	738,3	369,2	369,2	—	—	—	—	—
ZNRD	MELSECNET/10	598,6	446,7	223,4	223,4	—	—	—	—	—
	MELSECNET (II)	649,2	484,5	242,3	242,3	—	—	—	—	—
ZNWR	MELSECNET/10	614,3	458,4	229,2	229,2	—	—	—	—	—
	MELSECNET (II)	665,6	496,7	248,4	248,4	—	—	—	—	—
RFRP		590,9	441,0	220,5	220,5	—	—	—	—	—
RTOP		588,8	439,4	219,7	219,7	—	—	—	—	—
S.STMODE		—	—	—	22,6	—	—	—	—	—
S.CGMODE		—	—	—	19,4	—	—	—	—	—
S.TRUCK		—	—	—	43,7	—	—	—	—	—

Anweisung	Verarbeitung (Operand)			Verarbeitungszeiten (µs)								
				QnA-Serie				System Q				
				Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H
S.SPREF	Pufferspeicherkopfadresse = 0 (Mit A68AD)			—	—	—	407,1	—	—	—	—	—
	Pufferspeicherkopfadresse = 0 (Mit A62DA)			—	—	—	331,4	—	—	—	—	—
UNIRD	n = 1			—	—	—	—	96	80	74	79	34
	n = 16							440	370	340		
TRACE	Start			—	—	—	—	—	—	—	176	76
	Beendung der STRA Ausführung										6,3	2,7
TRACER				—	—	—	—	—	—	—	19	8,2
FWRITE				—	—	—	—	—	—	—	84	36
FREAD				—	—	—	—	—	—	—	79	34
PLOADP				—	—	—	—	—	—	—	58	25
PUNLOADP				—	—	—	—	—	—	—	272	117
PSWAPP				—	—	—	—	—	—	—	308	133
RBMOV	Übertragungszeit		1 Wort	—	—	—	—	—	—	—	69	29
			1000 Worte								580	308
COM[1]	Mit autom. Auffrischung des gemeinsamen Speicherberechs	Bereichsgröße = 2 k Worte (0,5 k Worte für jede CPU)		—	—	—	—	—	—	—	720	660
		Bereichsgröße = 4 k Worte (1 k Worte für jede CPU									860	730
	Ohne autom. Auffrischung des gemeinsamen Speicherberechs										43	20
FROM[1]	Aus dem gemeinsamen Speicher einer anderen CPU lesen	n3 = 0		—	—	—	—	—	—	—	59	29
		n3 = 1000									530	500
	Aus dem Pufferspeicher eines Sondermoduls lesen[2]	n3 = 1	Hauptbaugruppenträger								51	24
			Erweiterungsbaugruppenträger								54	27
		n3 = 1000	Hauptbaugruppenträger								540	480
			Erweiterungsbaugruppenträger								1100	1050
S.TO	s2 = 1			—	—	—	—	—	—	—	74	33
	s2 = 256										126	54

[1] Falls die Anweisung in mehreren CPU-Modulen eines Multi-CPU-Systems gleichzeitig ausgeführt wird, verlängert sich die Bearbeitungszeit. Diese Verlängerung kann mit den folgenden Formeln berechnet werden.
Bei einem System, das nur aus einem Hauptbaugruppenträger besteht:
Verlängerung der Ausführungszeit [µs] = 0,54 x (Anzahl der Adressen) x (Anzahl der CPU-Module)

Bei einem System, das aus Haupt- und Erweiteungsbaugruppenträgern besteht:
Verlängerung der Ausführungszeit [µs] = 1,30 x (Anzahl der Adressen) x (Anzahl der CPU-Module)

[2] Die Ausführungszeiten sind identisch bei Modulen, die der CPU zugeordnet sind, welche die Anweisung ausführt und Modulen, die einer anderen CPU zugeordnet sind.

Anweisung	Verarbeitung (Operand)	Verarbeitungszeiten (µs								
		QnA-Serie				System Q				
		Q2A	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H
FROM (n1, n2, d, n3) FROMP (n1, n2, d, n3)	n3 = 1	253[1]	217[1]	160[1]	160[1]	—	—	—	—	—
		252[2]	210[2]	154[2]	154[2]	—	—	—	—	—
		—	—	—	—	125[3]	105[3]	93[3]	47[4]	22[4]
	n3 = 1000	4514[1]	4286[1]	4150[1]	4150[1]	—	—	—	—	—
		2855[2]	2127[2]	2038[2]	2038[2]	—	—	—	—	—
		—	—	—	—	740[3]	695[3]	685[3]	476[4]	437[4]
DFRO (n1, n2, d, n3) DFROP (n1, n2, d, n3)	n3 = 1	260[1]	221[1]	165[1]	165[1]	—	—	—	—	—
		257[2]	214[2]	156[2]	156[2]	—	—	—	—	—
		—	—	—	—	130[3]	110[3]	100[3]	51[4]	24[4]
	n3 = 500	4543[1]	4271[1]	4082[1]	4082[1]	—	—	—	—	—
		2883[2]	2129[2]	2064[2]	2064[2]	—	—	—	—	—
		—	—	—	—	745[3]	695[3]	675[3]	478[4]	437[4]
TO (n1, n2, d, n3) TOP (n1, n2, d, n3)	n3 = 1	276[1]	217[1]	162[1]	162[1]	—	—	—	—	—
		254[2]	211[2]	154[2]	154[2]	—	—	—	—	—
		—	—	—	—	120[3]	105[3]	92[3]	48[4]	20[4]
	n3 = 1000	4500[1]	4319[1]	4188[1]	4188[1]	—	—	—	—	—
		2878[2]	2155[2]	2043[2]	2043[2]	—	—	—	—	—
		—	—	—	—	735[3]	680[3]	645[3]	479[4]	412[4]
DTO (n1, n2, d, n3) DTOP (n1, n2, d, n3)	n3 = 1	260[1]	221[1]	165[1]	165[1]	—	—	—	—	—
		257[2]	216[2]	157[2]	157[2]	—	—	—	—	—
		—	—	—	—	130[3]	110[3]	99[3]	50[4]	23[4]
	n3 = 500	4471[1]	4315[1]	4198[1]	4198[1]	—	—	—	—	—
		2819[2]	2172[2]	2062[2]	2062[2]	—	—	—	—	—
		—	—	—	—	740[3]	680[3]	640[3]	457[4]	416[4]

[1] Diese Verarbeitungszeiten gelten bei Verwendung des Hauptbaugruppenträgers A38B/A1S38B und eines Erweiterungsbaugruppenträgers.
[2] Diese Verarbeitungszeiten gelten bei Einsatz des Hauptbaugruppenträgers A38HB/A1S38HB.
[3] Die Verarbeitungszeit ist abhängig vom Typ des Erweiterungsbaugruppenträgers, von der Anzahl der Steckplätze und der Anzahl der installierten Module.
[4] Diese Verarbeitungszeiten gelten bei Einsatz des Hauptbaugruppenträgers Q312B und dem Modul QJ71C24 auf Steckplatz 0.

A.2.2 Verarbeitungszeiten der MELSEC A-Serie

Die Verarbeitungszeit einer Anweisung richtet sich nach der angewandten Verarbeitungsmethode der Ein- und Ausgangssignale:

- Direktverarbeitung = ○
- Verarbeitung nach dem Prozessabbild = ◎

Anweisung	Verarbeitung (Operand)					Verarbeitungszeiten (µs)				
						A	AnN,AnS		A2A, A2AS	A3A
						○	◎	○	◎	◎
								ohne X, Y / mit X, Y		
LD, LDI, AND	X					2,3	1,0	2,3	0,2	0,15
ANI, OR, ORI	Y, M, L, B, F, T, C					1,3	1,0	1,0	0,2	0,15
ANB, ORB, MPS, MRD, MPP						1,3	1,0	1,0	0,2	0,15
OUT	Y	gleichbleibender Zustand				2,3	1,0	2,3	0,4	0,3
		wechselnder Zustand (AUS/EIN)				2,3	1,0	2,3	0,4	0,3
	M (ohne Sondermerker)	gleichbleibender Zustand				1,3	1,0	1,0	0,4	0,3
		wechselnder Zustand (AUS/EIN)				1,3	1,0	1,0	0,4	0,3
	Sondermerker						37	37	0,8	0,6
	F	nicht ausgeführt				66	AnN: 61 AnS: 62	61	6,6	5,0
		ausgeführt	angezeigt			700	AnN: 633 AnS: 270	AnN: 633 AnS: 267	99	74
		ausgeführt	Anzeige abgeschlossen							
	T	Ausführungszeit der Anweisung				1,3	1,0	1,0	0,4	0,3
		END-Verarbeitung	nicht ausgeführt			1,3	0	0	0,23	0,18
		ausgeführt	nach Ablauf			15	11	11	4,5	3,3
			zusätzlich mit		K	30	24	24	7,7	5,7
					D	36	30	30	8,3	6,2
	C	Ausführungszeit der Anweisung				1,3	1,0	1,0	0,4	0,3
		END-Verarbeitung	nicht ausgeführt			1,3	0	0	0,27	0,2
		ausgeführt	nicht gezählt			14	0	0	0,27	0,2
			nach Ablauf			14	0	0	0,27	0,2
			zusätzlich mit		K	28	25	25	4,2	3,1
					D	33	30	30	4,8	3,6

Anweisung	Verarbeitung (Operand)			Verarbeitungszeiten (µs)				
				A	AnN,AnS		A2A, A2AS	A3A
				○	◎	○	◎	◎
						ohne X, Y / mit X, Y		
SET	Y	nicht ausgeführt		2,3	1,0	2,3	0,4	0,3
		ausgeführt	gleichbleibender Zustand	2,3	1,0	2,3	0,4	0,3
			wechselnder Zustand	2,3	1,0	2,3	0,4	0,3
	M, L, S, B	nicht ausgeführt		3,7	1,0	1,0	0,4	0,3
		ausgeführt	gleichbleibender Zustand	41	1,0	1,0	0,4	0,3
			wechselnder Zustand	41	1,0	1,0	0,4	0,3
	Sonder-merker M, B	nicht ausgeführt			2,0	3,0	0,8	0,6
		ausgeführt			32	32	0,8	0,6
	F	nicht ausgeführt		3,7	AnN: 3,0 AnS: 2,7	AnN: 3,0 AnS: 3,2	2,0	1,5
		ausgeführt	angezeigt	730	AnN: 638 AnS: 232	AnN: 638 AnS: 237	99	74
			Anzeige abgeschlossen					
RST	Y	nicht ausgeführt		2,3	1,0	2,3	0,4	0,3
		ausgeführt	gleichbleibender Zustand	2,3	1,0	2,3	0,4	0,3
			wechselnderZustand (AUS/ EIN)	2,3	1,0	2,3	0,4	0,3
	M, L, S, B	nicht ausgeführt		3,7	1,0	1,0	0,4	0,3
		ausgeführt	gleichbleibender Zustand	41	1,0	1,0	0,4	0,3
			wechselnder Zustand (AUS/ EIN)	41	1,0	1,0	0,4	0,3
	Sonder-merker M, B	nicht ausgeführt			3,0	3,0	0,8	0,6
		ausgeführt			32	32	0,8	0,6
	F	nicht ausgeführt		3,7	AnN: 3,0 AnS: 3,6	3,0	2,0	1,5
		ausgeführt	angezeigt	680	AnN: 477 AnS: 296	AnN: 477 AnS: 283	150	112
			Anzeige abgeschlossen					
	T, C	nicht ausgeführt		3,7	3,0	3,0	1,4	1,1
		ausgeführt		57	43	43	5,6	4,2
	D, W, A0, A1, V, Z	nicht ausgeführt		3,7	3,0	3,0	1,4	1,1
		ausgeführt		34	28	28	8,4	6,3
	R	nicht ausgeführt		3,7	3,0	3,0	1,4	1,1
		ausgeführt		41	35	35	4,6	3,5
PLS PLF	Y	nicht ausgeführt		65	59	61	2,2	1,7
		ausgeführt	EIN	68	62	63	2,2	1,7
			AUS	64	60	62	2,2	1,7
	M, L, B, F	nicht ausgeführt		64	59	59	2,2	1,7
		ausgeführt	EIN	67	62	62	2,2	1,7
			AUS	63	61	61	2,2	1,7
SFT SFTP	Y	nicht ausgeführt		3,7	3,0	3,0	1,4	1,1
		ausgeführt		49	38	39	4,4	3,3
	M, L, B, F	nicht ausgeführt		3,7	3,0	3,0	1,4	1,1
		ausgeführt		48	38	38	4,4	3,3
MC	Y	nicht ausgeführt		85	43	44	1,2	0,9
		ausgeführt		50	39	41	1,2	0,9
	M, L, S, B, F	nicht ausgeführt		84	43	43	1,2	0,9
		ausgeführt		49	39	39	1,2	0,9
MCR				35	26	26	0,6	0,45
FEND END	M9084 = AUS			2400	2150	2150	435	327
	M9084 = EIN			2400	2060	2060	285	214
NOP				1,3	1,0	1,0	0,2	0,15
LD=				95	70	70 / 87	3,8	2,9
AND=				96	61	62 / 81	2,6	2,0

Anweisung	Verarbeitung (Operand)	Verarbeitungszeiten (µs)					
		A	AnN,AnS			A2A, A2AS	A3A
		○	◎	○		◎	◎
				ohne X, Y	mit X, Y		
OR=		94	67	66	85	2,8	2,1
LD<>		98	69	69	86	4,1	3,1
AND<>		92	60	60	79	2,6	2,0
OR<>		96	66	66	84	2,8	2,1
LD>		96	67	67	84	4,1	3,1
AND>		92	60	60	79	2,6	2,0
OR>		98	66	65	83	2,8	2,1
LD<=		100	71	71	88	4,1	3,1
AND<=		94	61	61	80	2,6	2,0
OR<=		100	69	68	86	2,8	2,1
LD<		96	69	69	86	4,1	3,1
AND<		92	59	60	79	2,6	2,0
OR<		96	66	65	84	2,8	2,1
LD>=		100	71	71	88	4,1	3,1
AND>=		94	61	61	81	2,6	2,0
OR>=		100	69	68	86	2,8	2,1
LDD=		238	133	134	119	10	7,7
ANDD=		231	124	125	210	5,9	4,4
ORD=		236	133	133	218	6,3	4,7
LDD<>		235	131	132	217	10	7,7
ANDD<>		239	129	129	215	5,9	4,4
ORD<>		234	129	129	214	6,1	4,6
LDD>		238	133	133	219	9,7	7,3
ANDD>		240	131	131	217	5,8	4,4
ORD>		236	131	130	219	6,0	4,5
LDD<=		244	137	136	222	9,7	7,3
ANDD<=		238	127	128	213	5,8	4,4
ORD<=		246	137	136	221	6,0	4,5
LDD<		238	133	133	219	9,7	7,3
ANDD<		241	131	131	217	5,8	4,4
ORD<		236	131	130	215	6,0	4,5
LDD>=		243	137	137	222	9,7	7,3
ANDD>=		238	127	128	213	5,8	4,4
ORD>=		246	137	136	221	6,0	4,5
+ (s,d) +P (s,d)		72	44	45	59	2,8	2,1
+ (s1,s2,d) +P (s1,s2,d)		112	77	77	103	3,2	2,4
- (s,d) -P (s,d)		74	45	45	59	2,8	2,1
- (s1,s2,d) -P (s1,s2,d)		123	79	79	107	3,2	2,4
D+ (s,d) D+P (s,d)		110	69	69	90	4,0	3,0
D+ (s1,s2,d) D+P (s1,s2,d)		140	99	99	246	4,6	3,5
D-(s,d) D-P(s,d)		110	69	69	90	4,0	3,0

Anweisung	Verarbeitung (Operand)	Verarbeitungszeiten (µs)					
		A	AnN,AnS			A2A, A2AS	A3A
		○	◎	○		◎	◎
				ohne X, Y	mit X, Y		
D- (s1,s2,d) D-P (s1,s2,d)		141	99	99	130	4,6	3,5
x (s1,s2,d) xP (s1,s2,d)		135	94	95	168	3,4	2,6
/ (s1,s2,d) /P (s1,s2,d)		144	102	103	99	11	8,6
Dx (s1,s2,d) DxP (s1,s2,d)		429	341	340	370	20	15
D/ (s1,s2,d) D/P (s1,s2,d)		289	393	394	412	36	27
B+ (s,d) B+P (s,d)		210	123	123	183	6,4	4,8
B+ (s1,s2,d) B+P (s1,s2,d)		217	129	129	192	6,2	4,7
B- (s,d) B-P (s,d)		210	125	125	185	34	25
B- (s1,s2,d) B-P (s1,s2,d)		212	133	133	203	32	23
DB+ (s,d) DB+P (s,d)		320	175	176	280	14	11
DB+ (s1,s2,d) DB+P (s1,s2,d)		321	187	186	294	14	11
DB- (s,d) DB-P (s,d)		318	175	175	280	31	23
DB- (s1,s2,d) DB-P (s1,s2,d)		322	185	186	294	29	22
Bx (s1, s2, d) BxP (s1, s2, d)		410	299	300	358	14	11
B/ (s1, s2, d) B/P (s1, s2, d)		422	235	236	274	89	67
DBx (s1, s2, d) DBxP (s1, s2, d)		1158	941	939	1044	11	8,0
DB/ (s1, s2, d) DB/P (s1,s2,d)		998	896	894	954	62	47

[1] Mit einfacher Genauigkeit
[2] Mit doppelter Genauigkeit

Anweisung	Verarbeitung (Operand)	Verarbeitungszeiten (µs)					
		A	AnN,AnS			A2A, A2AS	A3A
		○	◎	○		◎	◎
				ohne X, Y	mit X, Y		
INC INCP		46	29	29	38	2,0	1,5
DINC DINCP		66	42	42	132	2,4	1,8
DEC DECP		48	31	31	39	2,0	1,5
DDEC DDECP		66	42	42	54	2,4	1,8
BCD BCDP		110	82	83	90	3,0	2,3
DBCD		329	219	220	284	13	9,5
DBCDP		329	219	220	284	13	9,5
BIN BINP		104	79	78	86	3,0	2,3
DBIN DBINP		311	215	216	280	6,0	4,5
NEG NEGP		105	50	49	86	8,6	6,5
MOV MOVP		72	47	47	57	17	13
DMOV DMOVP		104	67	67	87	20	15
CML CMLP		68	43	43	57	2,4	1,8
DCML DCMLP		130	74	75	108	3,2	2,4
BMOV (s1, s2, n) BMOVP (s1, s2, n)		7498	399	400	7144	1444	1083
FMOV (s1, s2, n) FMOVP (s1, s2, n)		1118	229	228	1029	1427	1070
XCH XCHP		102	60	61	84	2,8	2,1
DXCH DXCHP		170	107	107	141	4,2	3,2
CJ	ohne Index-Vergabe	49	39	39		6,6	5,0
	mit Index-Vergabe		48	48		6,6	5,0
SCJ	ohne Index-Vergabe	54	71	71		6,6	5,0
	mit Index-Vergabe		81	81		6,6	5,0
JMP		50	39	39		6,6	5,0
EI		195	38	38		3,0	2,3
DI		46	66	66		3,2	2,4
IRET		249	120	120		3,4	2,6
WAND (s, d) WANDP (s, d)		90	60	59	72	2,8	2,1
WAND (s1, s2, d) WANDP (s1, s2, d)		179	96	96	152	7,6	5,7

Anweisung	Verarbeitung (Operand)	Verarbeitungszeiten (µs)					
		A	AnN,AnS			A2A, A2AS	A3A
		○	◎	○		◎	◎
				ohne X, Y	mit X, Y		
DAND (s, d) DANDP (s, d)		276	140	139	240	13	9,5
DAND (s1, s2, d) DANDP (s1, s2, d)							
WOR (s, d) WORP (s, d)		90	61	60	72	2,8	2,1
WOR (s1, s2, d) WORP (s1, s2, d)		176	97	96	152	7,6	5,7
DOR (s, d) DORP (s, d)		276	140	139	240	13	9,5
DOR (s1, s2, d) DORP (s1, s2, d)							
WXOR (s, d) WXORP (s, d)		91	60	59	72	2,8	2,1
WXOR (s1, s2, d) WXORP (s1, s2, d)		178	97	96	152	7,6	5,7
DXOR (s, d) DXORP (s, d)		274	140	139	240	13	9,5
DXOR (s1, s2, d) DXORP (s1, s2, d)							
WXNR (s, d) WXNRP (s, d)		89	64	62	74	3,0	2,3
WXNR (s1, s2, d) WXNRP (s1, s2, d)		177	98	96	152	7,8	5,9
DXNR (s,d) DXNRP (s,d)		277	142	140	241	15	11
DXNR (s1,s2,d) DXNRP (s1,s2,d)							
ROR (n) RORP (n)	n = 3/5 (n = 5 bei allen AnN CPUs, n = 3 bei allen CPUs außer AnN)	66	52	51	51	5,8	4,4
RCR (n) RCRP (n)	n = 3/5 (n = 5 bei allen AnN CPUs, n = 3 bei allen CPUs außer AnN)	74	59	59	59	6,4	4,8

Anweisung	Verarbeitung (Operand)	Verarbeitungszeiten (µs)					
		A	AnN,AnS			A2A, A2AS	A3A
		○	◎	○		◎	◎
				ohne X, Y	mit X, Y		
ROL (n) ROLP (n)	n = 3/5 (n = 5 bei allen AnN CPUs, n = 3 bei allen CPUs außer AnN)	68	54	53	53	5,8	4,4
RCL (n) RCLP (n)	n = 3/5 (n = 5 bei allen AnN CPUs, n = 3 bei allen CPUs außer AnN)	74	57	57	57	6,4	4,8
DROR (n) DRORP (n)	n = 3/5 (n = 5 bei allen AnN CPUs, n = 3 bei allen CPUs außer AnN)	97	70	70	69	11	8,3
DRCR (n) DRCRP (n)	n = 3/5 (n = 5 bei allen AnN CPUs, n = 3 bei allen CPUs außer AnN)	95	72	72	72	12	9,2
DROL (n) DROLP (n)	n = 3/5 (n = 5 bei allen AnN CPUs, n = 3 bei allen CPUs außer AnN)	101	70	69	69	10	7,8
DRCL (n) DRCLP (n)	n = 3/5 (n = 5 bei allen AnN CPUs, n = 3 bei allen CPUs außer AnN)	98	68	68	68	12	8,7
SFR (s, n) SFRP (s, n)	n = 3/5 (n = 5 bei allen AnN CPUs, n = 3 bei allen CPUs außer AnN)	102	74	72	83	5,0	3,8
SFL (d, n) SFLP (d, n)	n = 5	106	74	73	84	4,8	3,6
BSFR (d, n) BSFRP (d, n)	n = 5	145	124	123	124	29	22
BSFL (d, n) BSFLP (d, n)	n = 5	158	134	133	134	28	21
DSFR (d, n) DSFRP (d, n)	n = 5	133	118	116	—	18,8	14,1
DSFL (d, n) DSFLP (d, n)	n = 5	134	118	17	—	22	17
BSET (d, n) BSETP (d, n)	n = 5	107	90	90	—	9,6	7,2
BRST (d, n) BRSTP (d, n)	n = 5	114	97	96	—	9,6	7,2
SER (s1, s2, d, n) SERP (s1, s2, d, n)	n = 5	230	200	200	—	33	25
SUM SUMP		164	115	114	131	15	11
DSUM DSUMP		267	200	199	231	34	25

Anweisung	Verarbeitung (Operand)	Verarbeitungszeiten (µs)					
		A	AnN,AnS			A2A, A2AS	A3A
		○	◎	○		◎	◎
				ohne X, Y	mit X, Y		
DECO (s, d, n) DECOP (s, d, n)	n = 2	249	164	163	216	32	24
ENCO (s, d, n) ENCOP (s, d, n)	n = 2	478	164	163	195	41	31
SEG		170	91	91	155	6,4	4,8
DIS (s, d, n)	n = 4	180	154	153	—	25	19
DISP (s, d, n)	n = 4	180	154	153	120	25	19
UNI (s, d, n)	n = 4	159	131	131	—	31	24
UNIP (s, d, n)	n = 4	159	131	131	—	31	24
FOR		64	53	53		5,8	4,4
NEXT		2532	41	41		8,0	6,0
CALL (pn)	ohne Index-Vergabe	74	74	74		10	7,8
	mit Index-Vergabe		78	78		10	7,8
CALLP (pn)	ohne Index-Vergabe	74	70	70		10	7,8
	mit Index-Vergabe		78	78		10	7,8
RET		249	50	50		7,0	5,3
CHG	M9084 = AUS	8546	2420	2420		—	—
	M9084 = EIN	—	2340	2340		—	—
SUB SUBP	ohne Index-Vergabe	90	79	79		—	—
	mit Index-Vergabe	—	85	85		—	—
FIFW FIFWP		340	101	101	123	20	15
FIFR FIFRP		202	118	118	134	69	52
PR		—	226	226	226	74	56
PRC		—	141	141	141	37	28
LED		170	203	203	203	100	75
LEDC		210	265	265	265	142	106
LEDR		520	638	638	638	106	80
LEDA		170	202	202	202	—	—
LEDB		172	211	211	211	—	—
CHK (Fehler-kontrolle)	1 Eingangskontakt	—	—	AnN: 771 AnS: 240		33	25
	50 Eingangskontakte	—	—	AnN: 3380 AnS: 3905		1257	943
	100 Eingangskontakte	—	—	AnN: 6687 AnS: 7820		2503	1877
	150 Eingangskontakte	—	—	AnA: 10137 AnS: 11470		3753	2815
CHK (Erzeugung eines Flip Flop)		—	121	121	121	—	—
SLT	Operandenspeicher	—	8448	8448	8448	2915	2186
	Operandenspeicher + R	—	24598	24598	24598	9996	7497
SLTR		—	29	29	29	6,6	5,0
STRA		—	30	30	30	5,0	3,8
STRAR		—	28	28	28	5,0	3,8
STC		—	28	28	28	2,4	1,8
CLC		—	31	31	31	2,4	1,8

Anweisung	Verarbeitung (Operand)	Verarbeitungszeiten (µs)					
		A	AnN,AnS			A2A, A2AS	A3A
		○	◎	○		◎	◎
				ohne X, Y	mit X, Y		
ASC		140	120	120	120	3,4	2,6
WDT WDTP		—	64	64	64	5,0	3,8
DUTY		—	68	68	68	14	11
LRDP (n1, s, d, n2)	n2 = 1	—	190	190	190	42	32
	n2 = 32		190	190	190	42	32
LWTP (n1, d, s, n2)	n2 = 1	—	200	200	200	49	37
	n2 = 32		446	446	446	89	66
RFRP (n1, n2, d, n3)	n3 = 1	—	172	172	172	32	24
	n3 = 32		172	172	172	63	47
RTOP (n1, n2, s, n3)	n3 = 1	—	176	176	176	68	51
	n3 = 32		176	176	176	34	26

Anweisung	Verarbeitung (Operand A-Serie)	Verarbeitungszeiten (µs)						
		A	AnN,AnS		A2A, A2AS,A2U		A3A, A3U	
		○	○ + ◎		◎		◎	
			ohne X, Y	mit X, Y	ohne X, Y	mit X, Y	ohne X, Y	mit X, Y
FROM (n1, n2, d, n3)	n3 = 1	—	439	524	237	261	178	196
FROMP (n1, n2, d, n3)	n3 = 1000/112 [1]	—	6609	2358	5749	2789	4312	2092
DFRO (n1, n2, d, n3)	n3 = 1	—	449	529	244	266	183	199
DFROP (n1, n2, d, n3)	n3 = 500/56 [2]	—	6609	2109	5669	1669	4252	1252
TO (n1, n2, d, n3)	n3 = 1	—	449	539	243	266	182	200
TOP (n1, n2, d, n3)	n3 = 1000/112 [1]	—	6609	3918	5773	2117	4330	1588
DTO (n1, n2, d, n3)	n3 = 1	—	454	544	240	266	180	199
DTOP (n1, n2, d, n3)	n3 = 500/56 [2]	—	6609	1609	5747	1501	4310	1126

[1] CPUs ohne X und Y: n3 = 1000; übrige CPUs mit X und Y: n3 = 112

[2] CPUs ohne X und Y: n3 = 500; übrige CPUs mit X und Y: n3 = 56.

A.3 Vergleich der CPUs

In diesem Abschnitt werden die Leistungsdaten, d.h. verwendbare Operanden, Verarbeitungsmodi, Sondermerker usw. der einzelnen CPUs (Q-, Q4AR-, QnA-, AnU-, AnA-, AnN-, AnS-) tabellarisch aufgeführt.

A.3.1 Verwendbare Operanden

Operanden		System Q		Q-Serie		A-Serie			
		Q00J, Q00, Q01	Qn, QnH, QnPH	Q4AR	QnA	AnU-CPU	AnA-CPU	AnN-CPU	AnS-CPU
Anzahl der Ein-/Ausgänge		Q00J: 256 Q00: 1024 Q01: 1024	4096 für Q02(H), Q06H, Q12H, Q12PH, Q25H und Q25PH	4096	Q2A: 512 Q2A-S1: 1024 Q3A: 2048 Q4A: 4096	A2U : 512 A2U-S1: 1024 A3U: 2048 A4U : 4096	A2A : 512 A2A-S1:1024 A3A: 2048	A1N: 256 A2N: 512 A2N-S: 1024 A3N: 2048	A1S: 256 A1S-S1: 512 A2S: 512 A2S-S1: 1024
Merker		8192 [1]		8192	8192 [1]	Gesamt 8192	Gesamt 8192	Gesamt 2048	256
Latch-Merker		2048 [1]	8192 [1]	8192	8192 [1]				1048 [1]
Schrittmerker	Ablaufprogramm	—		8192	—				0 [1]
	AS	2048	8192		8192	—	—	—	
Fehlermerker		1024 [1]	2048 [1]	2048	2048 [1]	2048	2048	256	256
Flankengesteuerte Merker		1024 [1]	2048 [1]	2048	2048 [1]	—	—	—	—
Link-Merker		2048 [1]	8192 [1]	8192	8192 [1]	8192	4096	1024	1024
Sonder-Link-Merker		1024	2048 [1]	2048	2048 [1]	56	56	56	—
Timer		512 [1]	2048 [1]	2048	2048 [1]	Gesamt 2048	Gesamt 2048	Gesamt 256	256 [1]
Remanente Timer		0 [1]	0 [1]	0 [1]	0 [1]				0 [1]
Counter		512 [1]	1024 [1]	1024	1024 [1]	1024	1024	256	256 [1]
Datenregister		11136 [1]	12288 [1]	12288	12288 [1]	8192	6144	1024	1024
Link-Register		2048 [1]	8192 [1]	8192	8192 [1]	8192	4096	1024	1024
Sonder-Link-Register		1024 [1]	2048 [1]	2048	2048 [1]	56	56	56	—
Funktionseingang		16 (FX0 bis FXF)	16 (FX0 bis FXF)	16 (FX0 bis FXF)	16 (FX0 bis FXF)	—	—	—	—
Funktionsausgang		16 (FX0 bis FXF)	16 (FY0 bis FYF)	16 (FY0 bis FYF)	16 (FY0 bis FYF)	—	—	—	—
Diagnosemerker		1024	2048	2048	2048	256	256	256	256
Funktionsregister		5 (FD0 bis FD4)	16 (FD0 bis FD15)	16 (FD0 bis FD15)	16 (FD0 bis FD15)	—	—	—	—
Diagnoseregister		1024	2048	2048	2048	256	256	256	256
Direkt adressierbare Link-Operanden		Bezeichnet durch J□\□		Bezeichnet durch J□\□		—	—	—	—

Operanden		System Q		Q-Serie		A-Serie			
		Q00J, Q00, Q01	Qn, QnH, QnPH	Q4AR	QnA	AnU-CPU	AnA-CPU	AnN-CPU	AnS-CPU
Direkt adressierbare Operanden		Bezeichnet durch U□\G□		Bezeichnet durch U□□\G□□		—	—	—	—
Index-Register	Z	10 (Z0 bis Z15)	16 (Z0 bis Z15)	16 (Z0 bis Z15)	16 (Z0 bis Z15)	7 (Z, Z1 bis Z6)	7 (Z, Z1 bis Z6)	1 (Z)	1 (Z)
	V[2]	—		—		7 (V, V1 bis V6)	7 (V, V1 bis V6)	1 (V)	1 (V)
File-Register		Q00JCPU: 0 Q00- und Q01CPU: 32767	32767 pro Block (R0 bis R32767) 1042432 (ZR0 bis ZR1042432)	32767 pro Block (R0 bis R32767) 1042432 (ZR0 bis ZR1042432)	32767 pro Block (R0 bis R32767) 1042432 (ZR0 bis ZR1042432)	8192 pro Block (R0 bis R8191)	8192 pro Block (R0 bis R8191)	8192 pro Block (R0 bis R8191)	0 [1]
Akkumulatoren[3]		—		—		2	2	2	2
Nesting		15	15	15	15	8	8	8	8
Pointer		300	4096	4096	4096	256	256	256	256
Interrupt-Pointer		128	256	48	48	32	32	32	32
AS-Blöcke		—	320	320	320	—	—	—	—
AS-Transitions-Operanden		—	512	512	512	—	—	—	—
Dezimalkonstanten		K-2147483648 bis K2147483647		K-2147483648 bis K2147483647		K-2147483648 bis K2147483647			
Hexadezimal-konstanten		H0 bis HFFFFFFFF		H0 bis HFFFFFFFF		H0 bis HFFFFFFFF			
Gleitkommazahl-konstanten		—	E±1,17549-38 bis E±3,40282+38	E±1,17549-38 bis E±3,40282+38		—	—	—	—
Zeichenfolgen		„QnA CPU“, „ABCD“ [4]	„QnA CPU“, „ABCD“	„QnA CPU“, „ABCD“		—	—	—	—

[1] Die Anzahl der Operandenadressen kann in den Parametern verändert werden.

[2] Die Q-/QnA-CPU benutzt V als Flankenmerker

[3] Anweisungen, die Akkumulatoren mit AnN-CPUs, AnA-CPUs und AnU-CPUs benutzen, haben andere Formate als bei Q-/QnA-CPUs.

[4] Kann bei einer Q00JCPU, Q00CPU und Q01CPU nur in Verbindung mit der $MOV-Anweisung verwendet werden.

A.3.2 E/A-Verarbeitungsmodi

E/A-Verarbeitungsmodus			Typ der CPU				
			Q	QnA/ QnAR	AnU	AnA	AnN
Verarbeitung	nach dem Prozessabbild		●		●	●	●[2]
	Direktverarbeitung der Ein-/ Ausgänge	Teilaktualisierungs-Anweisung	●		●	●	●
		Erweiterte-Anweisungen[1]	—		●	●	—
		Direktadressierbare Eingänge	●		—	—	—
		Direktadressierbare Ausgänge	●		—	—	—
Direktverarbeitung			—		—	—	●[2]

[1] Die DOUT-, DSET- und SRST-Anweisungen sind erweiterte Anweisungen für direkt adressierbare Ausgänge. Es sind keine erweiterten Anweisungen für direkt adressierbare Eingänge.

[2] Das Umschalten zwischen der Verarbeitung nach dem Prozessabbild und der Direktverarbeitung erfolgt bei einer AnN-CPU mittels DIP-Schalter.

A.3.3 Datentypen

Gesetzte Daten		Q-CPU	QnA/QnAR-CPU	AnU-CPU	AnA-CPU	AnN-CPU
Bit-Daten	Bit-Operand	●		●	●	●
	Wort-Operand	● (Bit-Adressierung erforderlich)		—	—	—
16-Bit-Datenwort	Bit-Operand	● (Anzahl der Stellen festlegen)		● (Anzahl der Stellen festlegen)	● (Anzahl der Stellen festlegen)	● (Anzahl der Stellen festlegen)
	Wort-Operand	●		●	●	●
32-Bit-Datenwort	Bit-Operand	● (Anzahl der Stellen festlegen)		● (Anzahl der Stellen festlegen)	● (Anzahl der Stellen festlegen)	● (Anzahl der Stellen festlegen)
	Wort-Operand	●		●	●	●
Gleitkommazahlen		●[1]		●	●	—
Zeichenfolgen		●[2]		—	—	—

[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

[2] Zeichenfolgen können bei einer Q00JCPU, Q00CPU oder Q01CPU nur in Verbindung mit der $MOV-Anweisung verwendet werden.

HINWEIS *Weiterführende Informationen zu diesem Thema finden Sie im Abs. 3.5 „Datentypen“ dieser Programmieranleitung.*

A.3.4 Timer-Vergleich

Timer-Funktionen

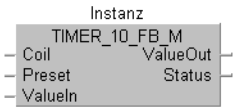
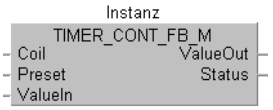
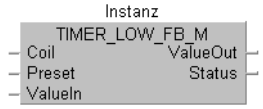
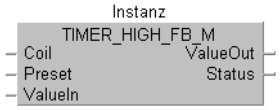
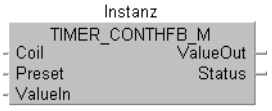
Name	Funktion			Q-CPU QnA/QnAR-CPU	AnU-CPU*	AnA-CPU*	AnN/AnS-CPU*
Low-Speed-Timer	Messbereich			100 ms (Voreinstellung) Dieser Wert kann in den Parametern innerhalb des Bereiches von 10 bis 100 ms eingestellt werden. Dieser Wert dient als Faktor, mit dem der Sollwert (TV) multipliziert wird.	● (Fest eingestellt auf 100 ms)		
	Programmierung (GX IEC Developer)	TIMER_M (normale/erweiterte Timer)	Übergabe des Sollwertes und Aufruf (Start) des Timers	●	●	●	●
		TIMER_VALUE_M (nur erweiterte Timer)	Übergabe des Sollwertes	●	●	●	●
		TIMER_START_M (nur erweiterte Timer)	Aufruf (Start) des Timers	●	●	●	●
	Programmierung (GX Developer)	Übergabe des Sollwertes und Aufruf (Start) des Timers		OUT Tn Sollwert	OUT Tn Sollwert		
High-Speed-Timer	Messbereich			10 ms (Voreinstellung) Dieser Wert kann in den Parametern innerhalb des Bereiches von 10 bis 100 ms eingestellt werden. Dieser Wert dient als Faktor, mit dem der Sollwert (TV) multipliziert wird..	● Fest eingestellt auf 10 ms		
	Programmierung (GX IEC Developer)	TIMER_M (normale/erweiterte Timer)	Übergabe des Sollwertes und Aufruf (Start) des Timers	●	●	●	●
		TIMER_VALUE_M (nur erweiterte Timer)	Übergabe des Sollwertes	●	●	●	●
		TIMER_START_M (nur erweiterte Timer)	Aufruf (Start) des Timers	●	●	●	●
	Programmierung (GX Developer)	Übergabe des Sollwertes und Aufruf (Start) des Timers		OUTH Tn Sollwert	OUT Tn Sollwert		

Timer-Funktionen

<table>
<tr><th>Name</th><th colspan="3">Funktion</th><th>Q-CPU
QnA/QnAR-CPU</th><th>AnU-CPU*</th><th>AnA-CPU*</th><th>AnN/AnS-CPU*</th></tr>
<tr><td rowspan="5">remanenter Low-Speed-Timer</td><td colspan="3">Messbereich</td><td>100 ms (Voreinstellung)
Dieser Wert kann in den Parametern innerhalb des Bereiches von 10 bis 100 ms eingestellt werden.
Dieser Wert dient als-Faktor, mit dem der Sollwert (TV) multipliziert wird.</td><td colspan="3">●
(Fest eingestellt auf 100 ms)</td></tr>
<tr><td rowspan="3">Programmierung (GX IEC Developer)</td><td>TIMER_H_M (normale/erweiterte Timer)</td><td>Übergabe des Sollwertes und Aufruf (Start) des Timers</td><td>●</td><td>●</td><td>●</td><td>●</td></tr>
<tr><td>TIMER_VALUE_M (nur erweiterte Timer)</td><td>Übergabe des Sollwertes</td><td>●</td><td>●</td><td>●</td><td>●</td></tr>
<tr><td>TIMER_START_M (nur erweiterte Timer)</td><td>Aufruf (Start) des Timers</td><td>●</td><td>●</td><td>●</td><td>●</td></tr>
<tr><td>Programmierung (GX Developer)</td><td colspan="2">Übergabe des Sollwertes und Aufruf (Start) des Timers</td><td>OUT STn Sollwert</td><td colspan="3">OUT Tn Sollwert</td></tr>
<tr><td rowspan="5">remanenter High-Speed-Timer</td><td colspan="3">Messbereich</td><td>10 ms (Voreinstellung)
Dieser Wert kann in den Parametern innerhalb des Bereiches von 1 bis 100 ms eingestellt werden.
Dieser Wert dient als-Faktor, mit dem der Sollwert (TV) multipliziert wird.</td><td colspan="3">—</td></tr>
<tr><td rowspan="3">Programmierung (GX IEC Developer)</td><td>TIMER_H_M (normale/erweiterte Timer)</td><td>Übergabe des Sollwertes und Aufruf (Start) des Timers</td><td>●</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>TIMER_VALUE_M (nur erweiterte Timer)</td><td>Übergabe des Sollwertes</td><td>●</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>TIMER_START_M (nur erweiterte Timer)</td><td>Aufruf (Start) des Timers</td><td>●</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>Programmierung (GX Developer)</td><td colspan="2">Übergabe des Sollwertes und Aufruf (Start) des Timers</td><td>OUTH STn Sollwert</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td colspan="2">Einstellbereich desSollwertes</td><td colspan="2"></td><td>1 bis 32767</td><td colspan="3">1 bis 32767</td></tr>
<tr><td colspan="2">Verarbeitung des Sollwertes 0</td><td colspan="2"></td><td>Sofort EIN</td><td colspan="3">Kein Maximum (gibt keine Zeit aus)</td></tr>
<tr><td rowspan="4">indizierte Adressierung</td><td>Kontakt</td><td colspan="2"></td><td rowspan="2">Möglich (nur Z0 und Z1 sind nutzbar)</td><td>Möglich</td><td colspan="2">Nicht möglich</td></tr>
<tr><td>Spule</td><td colspan="2"></td><td>Nicht möglich</td><td colspan="2">Nicht möglich</td></tr>
<tr><td>Sollwert</td><td colspan="2"></td><td>Nicht möglich</td><td>Nicht möglich</td><td colspan="2">Nicht möglich</td></tr>
<tr><td>Istwert</td><td colspan="2"></td><td>Möglich (Z0 bis Z15 sind nutzbar; Z0 bis Z9 bei Q00J-, Q00- und Q01CPU)</td><td>Möglich</td><td colspan="2">Möglich</td></tr>
<tr><td colspan="2">Aktualisierung des Ist-Wertes</td><td colspan="2" rowspan="2"></td><td rowspan="2">Bei Ausführung der OUT Tn-Anweisung</td><td colspan="3" rowspan="2">Nach der END-Verarbeitung</td></tr>
<tr><td colspan="2">Kontakt EIN/AUS Verarbeitung</td></tr>
</table>

* Die Startadresse der unterschiedlichen Timer muss im GX IEC Developer im Dialogfenster „Timer/Counter Bereich“ festgelegt werden.

Timer-Funktionsbausteine im GX IEC Developer

Name	Funktionsbaustein	Typ der CPU					
		Q	QnA/ QnAR	AnU	AnA	AnN	AnS
10-ms-Timer		—	●	●	●	●	●
100-ms-Timer		—	●	●	●	●	●
Remanenter Timer		—	●	●	●	●	●
Low-Speed-Timer		—	●	—	—	—	—
High-Speed-Timer		—	●	—	—	—	—
Remanenter High-Speed-Timer		—	●	—	—	—	—

Timer-Funktionsbausteine (Erläuterung)

Begriff im Funktionsbau-stein	Bedeutung		Bezeichnung bei normalen Timern	Bezeichnung bei remanenten Timern
Coil	Spule	Einschaltbedingung für den Timer	TC	STC
Preset	Sollwert		TValue	TValue
ValueIn	Anfangswert	Normalfall: 0	—	—
ValueOut	Istwert		TN	STN
Status	Kontakt	Ausgangskontakt wird geschaltet	TS	STS

Versehen Sie den Funktionsbaustein mit dem im Header festgelegten Instanzennamen, und ordnen Sie die Eingangs- und Ausgangsvariablen zu.

Hinweise zur Verwendung von Timern

Bei Ausführung der OUT(H) T-Anweisung wird der Einstellwert des Timers aktualisiert und die Timerspule gesetzt oder rückgesetzt. Wenn der Istwert des Timers größer oder gleich dem Sollwert ist, wird die Timerspule gesetzt.
In einem Programm, in dem ein Timer von einen anderen Timer gestartet wird, ist es erforderlich, dass die Anweisungen für den Timer, der später gestartet wird, zuerst bearbeitet werden. Wenn z.B. die Spule von T1 den Timer T2 startet, werden die Anweisungen für T2 im Programm vor denen von T1 gelegt.

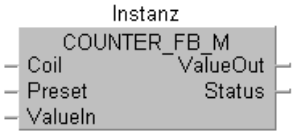
So wird verhindert, dass alle Timerspulen im selben Zyklus gesetzt werden. Dies ist der Fall, wenn bei schnellen Timern der Sollwert kleiner als die Zykluszeit ist oder wenn bei langsamen Timern der Sollwert auf „1“ eingestellt ist und die Anweisungen des zweiten Timers nach dem startenden Timer bearbeitet werden.

A.3.5 Counter-Vergleich

Counter-Funktionen

Funktion			Typ der CPU					
			Q	QnA/ QnAR	AnU	AnA	AnN	AnS
Programmierung (GX IEC Developer)	Counter_M	Übergabe des Sollwertes und Aufruf (Start) des Counters	●		●	●	●	●
	Counter_Start_M	Übergabe des Sollwertes	●		●	●	●	●
	Counter_Value_M	Aufruf (Start) des Counters	●		●	●	●	●
Programmierung (GX Developer)	OUT Cn Sollwert	Übergabe des Sollwertes und Aufruf (Start) des Counters	●		●	●	●	●
indizierte Adressierung	Kontakt		Möglich (nur Z0 und Z1 sind nutzbar)		Möglich		Nicht möglich	
	Spule		Möglich (nur Z0 und Z1 sind nutzbar)		Möglich		Nicht möglich	
	Sollwert		Nicht möglich		Nicht möglich		Nicht möglich	
	Istwert		Möglich (Z0 bis Z15 sind nutzbar; Z0 bis Z9 bei Q00J-, Q00- und Q01CPU)		Möglich		Möglich	
Aktualisierung des Ist-Wertes			Bei Ausführung der OUT Cn-Anweisung		Nach der END-Verarbeitung			
Kontakt EIN/AUS Verarbeitung								

Counter-Funktionsbausteine

Name	Funktionsbaustein	Typ der CPU					
		Q	QnA/ QnAR	AnU	AnA	AnN	AnS
Counter		—	●	●	●	●	●

Counter-Funktionsbausteine (Erläuterung)

Begriff im Funktionsbaustein	Bedeutung		Counter-Bezeichnung
Coil	Spule	Einschaltbedingung für den Counterprozess	CC
Preset	Sollwert		CValue
ValueIn	Anfangswert	Normalfall: 0	—
ValueOut	Istwert		CN
Status	Kontakt	Ausgangskontakt wird geschaltet, wenn der Funktionsbaustein abgearbeitet ist	CS

A.3.6 Vergleich der Display-Anweisungen

Anweisung	Q-CPU	QnA/QnAR-CPU	AnU-CPU	AnA-CPU	AnN-CPU	AnS-CPU
PR[1]	SM701 ist nicht gesetzt: Ausgabe bis zum Erreichen von 00_H SM701 ist gesetzt: Ausgabe von 16 Zeichen		M9049 nicht gesetzt: Ausgabe bis zum Erreichen von 00_H M9049 gesetzt: Ausgabe von 16 Zeichen			
PRC[1]	SM701 ist nicht gesetzt: Ausgabe von 32 Zeichen Kommentar SM701 ist gesetzt: Ausgabe von den 16 höherwertigen Zeichen		Ausgabe von 16 Zeichen Kommentar			

[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

A.3.7 Zur MELSEC A-Serie äquivalente Q-Serie- und System Q-Befehle

Da die Q/QnA-CPU keine Akkumulatoren (A0, A1) verwendet, hat sich das Format der AnU, AnN und AnN CPU-Anweisungen geändert, die Akkumulatoren verwenden.

Funktion	Q-CPU / QnA-CPU		AnU-CPU / AnA-CPU / AnN-CPU	
	Anweisungsformat	Bemerkung	Anweisungsformat	Bemerkung
16-Bit-Rotation nach rechts	ROR (d, n)	d: Rotationsdaten	ROR (n)	Rotationsdaten sind in A0 gesetzt
	RCR (d, n)	d: Rotationsdaten SM700 wird als Carry-Flag genutzt	RCR (n)	Rotationsdaten sind in A0 gesetzt M9012 wird als Carry-Flag genutzt
16-Bit-Rotation nach links	ROL (d, n)	d: Rotationsdaten	ROL (n)	Rotationsdaten sind in A0 gesetzt
	RCL (d, n)	d: Rotationsdaten SM700 wird als Carry-Flag genutzt	RCL (n)	Rotationsdaten sind in A0 gesetzt M9012 wird als Carry-Flag genutzt
32-Bit-Rotation nach rechts	DROR (d, n)	d: Rotationsdaten	DROR (n)	Rotationsdaten sind in A0 und A1 gesetzt
	DRCR (d, n)	d: Rotationsdaten SM700 wird als Carry-Flag genutzt	DRCR (n)	Rotationsdaten sind in A0 und A1 gesetzt M9012 wird als Carry-Flag genutzt
32-Bit-Rotation nach links	DROL (d, n)	d: Rotationsdaten	DROL (n)	Rotationsdaten sind in A0 und A1 gesetzt
	DRCL (d, n)	d: Rotationsdaten SM700 wird als Carry-Flag genutzt	DRCL (n)	Rotationsdaten sind in A0 und A1 gesetzt M9012 wird als Carry-Flag genutzt
16-Bit-Datensuche	SER (s1, s2, d, n)	Das Suchergebnis wird unter den in D bis D+1 angegebenen Operanden gespeichert.	SER (s1, s2, n)	Das Suchergebnis wird in A0 und A1 gespeichert
32-Bit-Datensuche	DSER (s1, s2, d, n)	Das Suchergebnis wird unter den in D bis D+1 angegebenen Operanden gespeichert.	DSER (s1, s2, n)	Das Suchergebnis wird in A0 und A1 gespeichert M9012 wird als Carry-Flag genutzt
16-Bit-Datenbit-Kontrolle	SUM (s, d)	Das Kontrollergebnis wird unter den in D angegebenen Operanden gespeichert.	SUM (s)	Das Kontrollergebnis wird in A0 gespeichert
32-Bit-Datenbit-Kontrolle	DSUM (s, d)	Das Kontrollergebnis wird unter den in D angegebenen Operanden gespeichert.	DSUM (s)	Das Kontrollergebnis wird in A0 gespeichert
Teilaktualisierung	RFS (s, n)	Erweitere Anweisung hinzugefügt	SEG (d, n)	Nur wenn M9052 gesetzt ist
8 Zeichen ASCII-Umwandlung	$MOV (s, d)		ASC (d)	
Setzen des Carry-Flags	SET (SM700)	Keine erweiterte Anweisung	STC	
Rücksetzen des Carry-Flags	RST (SM700)	Keine erweiterte Anweisung	CLC	
Sprung zum Programmende	GOEND	Erweitere Anweisung hinzugefügt	CJ (P255)	P255: Kennzeichnung der End-Anweisung
CHK-Anweisung[1]	CHKST CHK	CHKST-Anweisung hinzugefügt	CJ (Pn) CHK (P255)	

[1] Nicht für Q00JCPU, Q00CPU und Q01CPU

A.3.8 Vergleich zwischen QnA-/Q2AS-CPUs und CPUs des MELSEC System Q

Die folgenden Anweisungen für ein Q-CPU des MELSEC System Q sind neu im Vergleich mit einer MELSEC QnA-/Q2AS-CPU.

Funktion	Anweisung
Modul-Informationen lesen	UNIRD
Trace setzen	TRACE
Trace rücksetzen	TRACER
Daten in eine Datei schreiben	S.FWRITE
Daten aus einer Datei lesen	S.FREAD
Programm von einer Speicherkarte laden	PLOAD
Programm löschen	PUNLOAD
Programm löschen und Programm von einer Speicherkarte laden	PSWAP
Übertragung von Binärdatenblöcken	PBMOV
In den geneimsamen Speicherbereich schreiben (Nur Q02-, Q02H-, Q06H-, Q12H- und Q025HCPU ab Version B)	S.TO

Die Anweisungen, die in der folgenden Tabelle aufgeführt sind, gibt es bei einer CPU des System Q nicht mehr.

Funktion	Anweisung
Schreiben von Daten in ein EEPROM-Register	EROMWR
Abtastüberwachung (Sampling Trace) setzen (kann durch TRACE ersetzt werden)	STRA
Abtastüberwachung (Sampling Trace) rücksetzen (kann durch TRACER ersetzt werden)	STRAR
Status Latch setzen	SLT
Status Latch rücksetzen	SLTR
Programmüberwachung (Program Trace) setzen	PTRA
Programmüberwachung (Program Trace) rücksetzen	PTRAR
Programmüberwachung (Program Trace) ausführen	PTRAEXE PTRAEXEP
ASCII-Zeichen auf LED-Display ausgeben	LED
Kommentare auf LED-Display ausgeben	LEDC

Bei den folgenden Anweisungen sind bei der Verwendung von Programmen einer QnA-/Q2AS-CPU für eine CPU des MELSEC System Q Unterschiede zu beachten.

Funktion	Anweisung
Setzen von Ausgängen, Setzen/Rücksetzen eines Operanden (nur intern)	OUT, SET, RST
Auslesen von Kommentaren	COMRD
ASCII-Ausgabe	PRC
Anzeige löschen	LEDR
BIN-Konvertierung	BIN
BIN-Konvertierung	DBIN
Lesen von Uhr-Daten	DATERD
Schreiben von Uhr-Daten	DATEWR
Bit-Schemen der Ausführungsbedingungen der Interrupt-Programme	IMASK
Aktualisierungsanweisung für Netzwerk- und Schnittstellendaten	COM
Netzwerk Datenaktualisierung	ZCOM
Lesen von Routing-Informationen	RTREAD
Schreiben von Routing-Informationen	RTWRITE
PID-Regelung einstellen	PIDINT
PID-Regelung	PIDCONT
Einphasiger Auf-/Abwärtszähler	UDCNT1
Zweiphasiger Auf-/Abwärtszähler	UDCNT2
Impulszähler	SPD
Impulsausgang mit einstellbarer Anzahl von Impulsen	PLSY
Pulsweiten-Modulation	PWM

HINWEISE *Wenn ein Programm einer QnA-CPU, das auf Sondermodule zugreift, in ein Programm für eine CPU des System Q konvertiert wird, sind die folgenden Punkte zu beachten:*

- *Die CPU-Module des System Q sind im Q-Modus nicht kompatibel zu Sondermodulen und Netzwerkmodulen der A/AnS-Serie. Ändern Sie die Programme ab, indem Sie die FROM- und TO-Anweisungen benutzen, wenn diese Module weiter verwendet werden.*
- *Wenn Sondermodule der QnA-, Q2AS-, A- oder AnS-Serie gegen Sondermodule des System Q ausgetauscht werden, können bestimmte Anweisungen weiter verwendet werden. Nähere Informationen hierzu können Sie den Handbüchern der jeweiligen Sondermodule entnehmen.*

A.4 Übersicht der Sondermerker

A.4.1 Liste der Diagnosemerker (MELSEC Q-Serie und MELSEC System Q)

Diagnosemerker (SM) sind interne Merker deren Anwendung in der SPS festgelegt ist. Aus diesem Grund können sie nicht wie interne Merker in Ablaufprogrammen verwendet werden. Sie können jedoch zur Steuerung der CPU ein- oder ausgeschaltet werden.

HINWEISE *Die Diagnosemerker SM1200 bis SM1255 werden bei einer QnA-CPU verwendet. Bei einer Q-CPU sind diese Merker nicht belegt.*

Die Diagnosemerker ab SM 1500 sind für die Q4AR-CPU reserviert

In dieser Tabelle werden die Einträge für die Tabellen auf den folgenden Seiten erläutert.

Tabellenüberschrift	Bedeutung
Adresse	● Zeigt die Adresse des Diagnosemerkers.
Name	● Zeigt den Namen des Diagnosemerkers.
Bedeutung	● Kurzerläuterung der Bedeutung des Diagnosemerkers.
Beschreibung	● Beinhaltet detaillierte Informationen zur Bedeutung des Diagnosemerkers.
Gesetzt vom (wenn gesetzt)	Gibt Aufschluss darüber, ob der Diagnosemerker vom System oder vom Benutzer gesetzt wurde. <Gesetzt vom> S : Durch das System gesetzt B : Durch den Benutzer gesetzt (im Ablaufprogramm oder Prüfmodus eines Peripheriegerätes) S/B : Durch das System und den Benutzer gesetzt Wird nur gezeigt wenn die Einstellung durch das System vorgenommen wurde. <Wenn gesetzt> END-Verarbeitung : Wird während jeder END-Verarbeitung gesetzt Initialisierung : Wird nur während der Initialisierung gesetzt (beim Einschalten des Netzteils oder beim Umschalten der CPU vom STOP-Modus in den RUN-Modus) Zustandsänderung : Wird nur nach Auftreten einer Zustandsänderung gesetzt Fehler : Wird nur nach Auftreten eines Fehlers gesetzt Anweisungsausführung : Wird gesetzt, wenn die Anweisung ausgeführt wird Anforderung : Wird nur gesetzt, wenn eine Benutzeranforderung ansteht (durch SM, etc.)
A-CPU M9[ ] [ ] [ ]	● Zeigt Diagnosemerker M9 [ ] [ ] [ ] korresspondierend zur A-CPU. (Änderung und Schreibweise, wenn es inhaltliche Änderungen gab.) ● Wird mit „Neu“ gekennzeichnet, wenn er der Q-CPU neu hinzugefügt wurde.
Gültig für:	● Gibt an, für welche CPU dieser Sondermerker zur Verfügung steht. ●: Gilt für alle CPU-Typen Q-CPU: Gilt nur für alle CPU-Module des System Q QnA: Gilt für die CPUs der QnA-Reihe und die Q2AS-CPU CPU-Typ: Gilt nur für diese CPU (z. B. Q4AR-CPU) Rem: Gültig für dezentrale MELSECNET/H E/A-Module

(1) Informationen zur Fehlerdiagnose

Adresse	Name	Bedeutung	Beschreibung	Gesetzt vom (wenn gesetzt)	A-CPU M9[ ] [ ] [ ]	Gültig für:
SM0	Diagnosefehler	AUS: Kein Fehler EIN: Fehler	Wird auf EIN gesetzt, wenn das Diagnoseergebnis einen Fehler aufweist (inklusive externer Diagnosen). Nach Beseitigung des Fehlers bleibt der Merker gesetzt.	S (Fehler)	Neu	
SM1	Selbstdiagnose-fehler	AUS: Kein Fehler mit der Selbstdiagnose erkannt EIN: Fehler	Wird auf EIN gesetzt, wenn das Selbstdiagnoseergebnis einen Fehler aufweist. Nach Beseitigung des Fehlers bleibt der Merker gesetzt.	S (Fehler)	M9008	
SM5	Allgemeine Fehlerinforma-tionen	AUS: Keine allgemeinen Fehlerinformationen EIN: Allgemeine Fehlerinformationen	Wird bei gesetztem SM0 und vorhandener allgemeiner Fehlerinformation auf EIN gesetzt	S (Fehler)	Neu	● Rem
SM16	Spezielle Fehlerinformation	AUS: Keine speziellen Fehlerinformationen EIN: Spezielle Fehlerinformationen	Wird bei gesetztem SM0 und vorhandener spezieller Fehlerinformation auf EIN gesetzt	S (Fehler)	Neu	
SM50	Fehler zurücksetzten	AUS → EIN: Fehler löschen	Ein Fehler wird zurückgesetzt. Weitere Informationen sind im Absatz 5.3.6 zu finden.	B	Neu	
SM51	Niedrige Batterie-spannung (Latch-Merker)	AUS: Normal EIN: Spannung abgefallen	Die Spannung der Pufferbatterie von CPU oder Speicherkarte ist unter ihren Minimalwert gesunken. Der Merker bleibt nach dem Austausch der Batterie gesetzt. Der Merkerzustand ist identisch mit BAT. ALARM LED.	S (Fehler)	M9007	●
SM52	Niedrige Batterie-spannung	AUS: Normal EIN: Spannung abgefallen	Die Spannung der Pufferbatterie ist unter ihren Minimalwert gesunken. Der Merker wird nach dem Austausch der Batterie zurückgesetzt.	S (Fehler)	M9006	
SM53	Spannungsabfall bei der Versorgungs-spannung	AUS: Normal EIN: Spannung abgefallen	Die Eingangsspannung des Wechselspannungsnetzteils ist für weniger als 20 ms abgefallen. Das Rücksetzen erfolgt mit dem Aus- und Wiedereinschalten der Versorgungsspannung.	S (Fehler)	M9005	●
			Die Eingangsspannung des Netzteils mit Gleichspannungsein-gang ist für weniger als 10 ms abgefallen. Das Rücksetzen erfolgt mit dem Aus- und Wiedereinschalten der Versorgungsspannung.			Q-CPU
			Die Eingangsspannung des Netzteils mit Gleichspannungsein-gang ist für weniger als 1 ms abgefallen. Das Rücksetzen erfolgt mit dem Aus- und Wiedereinschalten der Versorgungsspannung.			QnA-CPU
SM54	Fehler im MELSECNET/ MINI	AUS: Normal EIN: Fehler	Der Merker wird gesetzt, wenn ein Link-Fehler in einem installierten AJ71PT32 (S3)-Modul aufgetreten ist. Der Merker bleibt auch nach Wegfall der Störung gesetzt.	S (Fehler)	M9004	QnA-CPU
SM56	Verarbeitungs-fehler	AUS: Normal EIN: Verarbeitungsfehler	Der Merker wird gesetzt, wenn ein Verarbeitungsfehler aufgetreten ist. Der Merker bleibt auch nach Wegfall der Störung gesetzt.	S (Fehler)	M9011	●

(1) Informationen zur Fehlerdiagnose

Adresse	Name	Bedeutung	Beschreibung		Gesetzt vom (wenn gesetzt)	A-CPU M9[ ] [ ] [ ]	Gültig für:
SM60	Sicherung defekt	AUS: Normal EIN: Modul mit defekter Sicherung	Der Merker wird gesetzt, sobald bei einem der Ausgangsmodule die Sicherung als defekt erkannt wurde. Der Merker bleibt auch nach Rückkehr in den Normalzustand gesetzt.		S (Fehler)	M9000	● Rem
SM61	Vergleichsfehler E/A-Module	AUS: Normal EIN: Vergleichsfehler	Der aktuelle Status der E/A-Module unterscheidet sich von der registrierten Information nach Einschalten der Versorgungsspannung. Der Vergleich der E/A-Module wird auch mit einer Remote-Station durchgeführt.		S (Fehler)	M9002	
SM62	Fehlermerker-kennung	AUS: Keine Kennung EIN: Kennung	Wird gesetzt, wenn nur ein Fehlermerker F gesetzt wurde.		S (Anweisungs-ausführung)	M9009	●
SM80	Mittels CHK-Anweisung erkannter Fehler	AUS: Normal EIN: Fehler	Wird gesetzt, wenn ein Fehler mittels CHK-Anweisung erkannt wurde. Der Merker bleibt auch nach Wegfall des Fehlers gesetzt.		S (Anweisungs-ausführung)	Neu	QnA-CPU, Q-CPU außer Q00J-, Q00- und Q01CPU
SM90	Start des WDT (Watchdog-Timer) zur Überwachung der Transitionen (Nur aktiv wenn ein AS-Programm existiert)	AUS: Kein Start (WDT ist zurückgesetzt) EIN: Start (WDT wird gestartet)	Korrespondierend zu SD90	Der Merker wird gesetzt, wenn die Messung mittels WDT begonnen hat. Mit Rücksetzen des Merkers wird der WDT zurückgesetzt.	B	M9108	
SM91			Korrespondierend zu SD91			M9109	
SM92			Korrespondierend zu SD92			M9110	
SM93			Korrespondierend zu SD93			M9111	
SM94			Korrespondierend zu SD94			M9112	
SM95			Korrespondierend zu SD95			M9113	
SM96			Korrespondierend zu SD96			M9114	
SM97			Korrespondierend zu SD97			Neu	
SM98			Korrespondierend zu SD98			Neu	
SM99			Korrespondierend zu SD99			Neu	
SM100	Verwendung der seriellen Kommunikation.	AUS: Serielle Kommunikation wird nicht benutzt EIN: Serielle Kommunikation wird verwendet	Der Merker zeigt an, ob die serielle Kommunikation in den Systemeinstellungen angewählt wurde.		S (Initialisierung)	Neu	Q00J-, Q00- und Q01CPU
SM101	Verwendetes Protokoll bei der seriellen Kommunikation.	AUS: Protokoll für Programmiergeräte EIN: MC-Protokoll	Dieser Merker gibt an, mit welchem Protokoll die serielle Kommunikation abgewickelt wird.		S (Bei der seriellen Kommunikation)	Neu	
SM110	Fehlerhaftes Protokoll	AUS: Kein Fehler EIN: Fehler	Wird gesetzt, wenn versucht wurde, mit einem falschen Protokoll Daten über die serielle Schnittstelle auszutauschen. Bleibt gesetzt, auch wenn danach mit dem korrekten Protokoll kommuniziert wird.		S (Fehler)	Neu	
SM111	Status der Kommunikation	AUS: Kein Fehler EIN: Fehler	Wird gesetzt, wenn die tatsächlich verwendete Art der seriellen Kommunikation von den Einstellungen abweicht. Bleibt gesetzt, auch wenn nach dieser Fehlermeldung mit der korrekten Betriebsart kommuniziert wird.		S (Fehler)	Neu	
SM112	Fehlerinformationen löschen	EIN: Diagnosemerker und -register löschen	Beim Übergang von AUS nach EIN werden SM110, SM111, SD110 und SD111 gelöscht.		B	Neu	

(1) Informationen zur Fehlerdiagnose

Adresse	Name	Bedeutung	Beschreibung	Gesetzt vom (wenn gesetzt)	A-CPU M9[ ] [ ] [ ]	Gültig für:
SM113	Daten-Überlauf	AUS: Kein Fehler EIN: Fehler	Dieser Merker wird gesetzt, wenn bei der seriellen Kommunikation ein Datenüberlauf aufgetreten ist.	S (Fehler)	Neu	Q00J-, Q00- und Q01CPU
SM114	Paritäts-Fehler	AUS: Kein Fehler EIN: Fehler	Der gesetzte Merker zeigt einen Paritätsfehler bei der seriellen Kommunikation an.	S (Fehler)	Neu	
SM115	Rahmen-Fehler	AUS: Kein Fehler EIN: Fehler	Dieser Merker wird gesetzt, wenn bei der seriellen Kommunikation der Datenrahmen fehlerhaft ist.	S (Fehler)	Neu	

(2) Systeminformationen

Adresse	Name	Bedeutung	Beschreibung	Gesetzt vom (wenn gesetzt)	A-CPU M9[ ] [ ] [ ]	Gültig für:
SM202	LED-AUS-Befehl	AUS → EIN : LED AUS	Die mit den Bits von SD202 korrespondierenden LEDs erlöschen beim Wechsel von AUS nach EIN.	B	Neu	● außer Q00J-, Q00- und Q01CPU
SM203	Kennung des STOP-Status	STOP-Status	Wird gesetzt, wenn die CPU stoppt.	S (Zustandsänderung)	M9042	●
SM204	Kennung des Pause-Modus	Pause-Status	Wird gesetzt, wenn die CPU im Pause-Modus ist.	S (Zustandsänderung)	M9041	
SM205	Kennung des STEP-RUN-Modus	STEP-RUN-Modus	Wird gesetzt, wenn die CPU im STEP-RUN-Modus ist.	S (Zustandsänderung)	M9054	● außer Q00J-, Q00- und Q01CPU
SM206	Ausführungsbedingung Pause-Status	AUS: Nicht möglich EIN: Möglich	Die CPU geht in den PAUSE-Modus über, wenn der PAUSE-Fernkontakt und der Merker gesetzt sind.	B	M9040	●
	Status des Operandentest	AUS: Operandentest wurde noch nicht ausgeführt EIN: Operandentest wurde ausgeführt	Dieser Merker gibt den Status des Operandentests an, der mit Hilfe der Programmier-Software ausgeführt werden kann.	S (Anforderung)	Neu	Q00J-, Q00- und Q01CPU Rem
SM210	Setzanforderung für Uhrdaten	AUS: Keine Verarbeitung EIN: Anforderung	Bei gesetztem Merker werden die Uhrdaten nach Ausführung der END-Anweisung in den Registern SD210 bis SD213 gespeichert und an die Uhr übertragen.	B	M9025	●
SM211	Uhrdaten-Fehler	AUS: Normal EIN: Fehler	Der Merker ist gesetzt, wenn ein Fehler in den Werten der Uhrdaten, die in SD210 bis SD213 gespeichert sind, vorhanden ist. Der Merker ist nicht gesetzt, wenn kein Fehler vorhanden ist.	S (Anforderung)	M9026	
SM212	Uhrdatenanzeige	AUS: Keine Verarbeitung EIN: Anzeige	Die Uhrdaten aus den Registern SD210 bis SD213 werden gelesen und mit Monat, Tag, Stunde, Minute und Sekunde über die LED-Anzeige an der CPU ausgegeben. (Nur bei Q3A-CPU und Q4A-CPU möglich)	B	M9027	Q3A-, Q4A-, Q4AR-CPU
SM213	Leseanforderung für Uhrdaten	AUS: Keine Verarbeitung EIN: Anforderung	Bei gesetztem Merker werden die Uhrdaten in die Register SD210 bis SD213 als BCD-Werte eingelesen.	B	M9028	● Rem

(2) Systeminformationen

Adresse	Name	Bedeutung	Beschreibung	Gesetzt vom (wenn gesetzt)	A-CPU M9[ ] [ ] [ ]	Gültig für:
SM240	Reset-Merker CPU 1	AUS: Kein Reset EIN: Reset an CPU 1 ausgeführt	Beim Zurücksetzen der CPU 1 oder beim Entfernen der CPU vom Baugruppenträger wird dieser Merker gesetzt. Die anderen CPUs des Multi-CPU-Systems werden ebenfalls zurückgesetzt.	S (Zustandsänderung)	Neu	Q02-, Q02H-, Q06H-, Q12H-, Q25H-CPU ab Ver. B
SM241	Reset-Merker CPU 2	AUS: Kein Reset EIN: Reset an CPU 2 ausgeführt	Beim Zurücksetzen der CPU 2 oder beim Entfernen der CPU vom Baugruppenträger wird dieser Merker gesetzt. Bei den anderen CPUs des Multi-CPU-Systems wird die Fehlermeldung MULTI CPU DOWN (Fehlercode 7000) gemeldet.	S (Zustandsänderung)	Neu	
SM242	Reset-Merker CPU 3	AUS: Kein Reset EIN: Reset an CPU 3 ausgeführt	Beim Zurücksetzen der CPU 3 oder beim Entfernen der CPU vom Baugruppenträger wird dieser Merker gesetzt. Bei den anderen CPUs des Multi-CPU-Systems wird die Fehlermeldung MULTI CPU DOWN (Fehlercode 7000) gemeldet	S (Zustandsänderung)	Neu	
SM243	Reset-Merker CPU 4	AUS: Kein Reset EIN: Reset an CPU 4 ausgeführt	Beim Zurücksetzen der CPU 4 oder beim Entfernen der CPU vom Baugruppenträger wird dieser Merker gesetzt. Bei den anderen CPUs des Multi-CPU-Systems wird die Fehlermeldung MULTI CPU DOWN (Fehlercode 7000) gemeldet	S (Zustandsänderung)	Neu	
SM244	Fehler-Merker CPU 1	AUS: Kein Fehler EIN: Fehler bei CPU 1, der die CPU stoppt	Der gesetzte Merker zeigt an, dass ein Fehler aufgetreten ist, der die CPU gestoppt hat. Im fehlerfreiem Zustand oder bei einem Fehler, der keinen STOP verursacht, wird der Merker zurückgesetzt.	S (Zustandsänderung)	Neu	
SM245	Fehler-Merker CPU 2	AUS: Kein Fehler EIN: Fehler bei CPU 2, der die CPU stoppt		S (Zustandsänderung)	Neu	
SM246	Fehler-Merker CPU 3	AUS: Kein Fehler EIN: Fehler bei CPU 3, der die CPU stoppt		S (Zustandsänderung)	Neu	
SM247	Fehler-Merker CPU 4	AUS: Kein Fehler EIN: Fehler bei CPU 4, der die CPU stoppt		S (Zustandsänderung)	Neu	
SM250	Max. geladene E/A-s lesen	AUS: Keine Verarbeitung EIN: Lesen	Dieser Merker wird gesetzt, wenn die max. geladene Anzahl von E/As in das Register SD250 ausgelesen wird.	B	Neu	● außer Q00J-, Q00- und Q01CPU Rem
SM251	Änderungskennung der E/A-Module	AUS: Keine Änderung EIN: Änderung	Nach der Festlegung einer Kopfadresse für ein bestimmtes E/A-Modul in Register SD251 kann dieses Modul durch Setzen von SD251 Online geschaltet werden (nur ein Modul pro Ausführung). Eine Änderung des Betriebszustands eines E/A-Moduls ist mit Hilfe von SD251 auch während des Testbetriebs über ein Programmiergerät oder im STOP-Zustand der CPU möglich. Der Betriebszustand der CPU darf bis zur Vollendung der Online-Schaltung des E/A-Moduls nicht verändert werden.	B (END-Verarbeitung)	M9054	Q2A(S1), Q3A-, Q4A-, Q4AR-CPU
SM252	E/A-Austausch erlaubt	AUS: Austausch gesperrt EIN: Austausch möglich	Wird gesetzt, wenn der E/A-Austausch erlaubt ist.	S (END-Verarbeitung)	Neu	
SM254	Anweisung zum Refresh für alle Stationen	AUS: Refresh nur für die 1. Station EIN: Refresh für alle Stationen	Gültig für den Batch-Refresh und für den langsamen Zyklus. Dieser Merker wird gesetzt, wenn ein Refresh für alle Stationen erfolgen soll.	S (END-Verarbeitung)	Neu	Q-CPU außer Q00J-, Q00- und Q01CPU

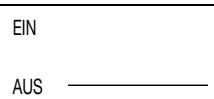
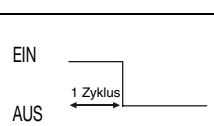
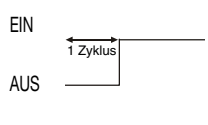
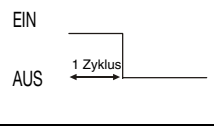
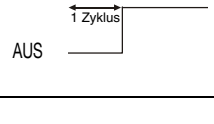
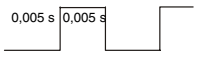
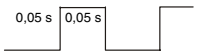
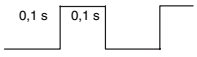
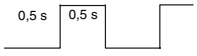
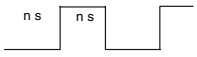
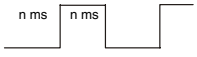
(2) Systeminformationen

Adresse	Name	Bedeutung	Beschreibung	Gesetzt vom (wenn gesetzt)	A-CPU M9[ ] [ ] [ ]	Gültig für:
SM255	MELSECNET/10 Informationen über Modul 1	AUS: Betriebsbereites Netzwerk EIN: Standby-Netzwerk	Wird bei einem Standby-Netzwerk gesetzt. (Ist keine Festlegung zwischen „standby“ und „aktiv“ getroffen, wird „aktiv“ vorausgesetzt.)	S (Initialisierung)	Neu	● außer Q00J-, Q00- und Q01CPU
SM256		AUS: Liest EIN: Liest nicht	Für die Aktualisierung zwischen Link-Modul und CPU (B, W etc.) wird angegeben, ob vom Link-Modul gelesen wird.	B	Neu	
SM257		AUS: Schreibt EIN: Schreibt nicht	Für die Aktualisierung zwischen CPU und Link-Modul (B, W etc.) wird bestimmt, ob ins Link-Modul geschrieben wird.	B	Neu	
SM260	MELSECNET/10 Informationen über Modul 2	AUS: Betriebsbereites Netzwerk EIN: Standby-Netzwerk	Wird bei einem Standby-Netzwerk gesetzt. (Ist keine Festlegung zwischen standby und aktiv getroffen, wird aktiv vorausgesetzt.)	S (Initialisierung)	Neu	
SM261		AUS: Liest EIN: Liest nicht	Für die Aktualisierung zwischen Link-Modul und CPU (B, W etc.) wird angegeben, ob vom Link-Modul gelesen wird.	B	Neu	
SM262		AUS: Schreibt EIN: schreibt nicht	Für die Aktualisierung zwischen CPU und Link-Modul (B, W etc.) wird bestimmt, ob ins Link-Modul geschrieben wird.	B	Neu	
SM265	MELSECNET/10 Informationen über Modul 3	AUS: Betriebsbereites Netzwerk EIN: Standby-Netzwerk	Wird bei einem Standby-Netzwerk gesetzt. (Ist keine Festlegung zwischen „standby“ und „aktiv“ getroffen, wird „aktiv“ vorausgesetzt.)	S (Initialisierung)	Neu	
SM266		AUS: Liest EIN: Liest nicht	Für die Aktualisierung zwischen Link-Modul und CPU (B, W etc.) wird angegeben, ob vom Link-Modul gelesen wird.	B	Neu	
SM267		AUS: Schreibt EIN: Schreibt nicht	Für die Aktualisierung zwischen CPU und Link-Modul (B, W etc.) wird bestimmt, ob ins Link-Modul geschrieben wird.	B	Neu	
SM270	MELSECNET/10 Informationen über Modul 4	AUS: Betriebsbereites Netzwerk EIN: Standby-Netzwerk	Wird bei einem Standby-Netzwerk gesetzt. (Ist keine Festlegung zwischen standby und aktiv getroffen, wird aktiv vorausgesetzt.)	S (Initialisierung)	Neu	
SM271		AUS: Liest EIN: Liest nicht	Für die Aktualisierung zwischen Link-Modul und CPU (B, W etc.) wird angegeben, ob vom Link-Modul gelesen wird.	B	Neu	
SM272		AUS: Schreibt EIN: Schreibt nicht	Für die Aktualisierung zwischen CPU und Link-Modul (B, W etc.) wird bestimmt, ob ins Link-Modul geschrieben wird.	B	Neu	
SM280	CC-Link Fehler	AUS: Normal EIN: Fehler	Wird gesetzt, wenn bei einem der installierten QJ61QBT11 ein CC-Link Fehler erkannt wurde. Wird rückgesetzt, wenn der normale Betriebszustand wieder hergestellt wird.	S (Zustandsänderung)	Neu	Q-CPU außer Q00J-, Q00- und Q01CPU
			Wird gesetzt, wenn bei einem der installierten A1(1S)J61QBT11 ein CC-Link Fehler erkannt wurde. Bleibt gesetzt, wenn der normale Betriebszustand wieder hergestellt wird	S (Fehler)	Neu	QnA-CPU
SM315	Für Kommunikation reservierte Zeit freigeben	AUS: Ohne Verzögerung EIN: Mit Verzögerung	Bei gesetztem Merker wird die Ausführung der END-Anweisung um die in SD315 eingetragene Zeit verzögert, wenn keine Kommunikation abgewickelt wird. Die Zykluszeit verlängert sich entsprechend. Bei zurückgesetztem Merker wird die END-Anweisung unverzögert ausgeführt, wenn keine Kommunikation abgewickelt wird.	B	Neu	Q00J-, Q00- und Q01CPU

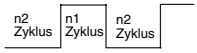
(2) Systeminformationen

Adresse	Name	Bedeutung	Beschreibung	Gesetzt vom (wenn gesetzt)	A-CPU M9[ ] [ ] [ ]	Gültig für:
SM320	Programm der Ablaufsprache vorhanden	AUS: Programm der Ablaufsprache nicht vorhanden EIN: Programm der Ablaufsprache vorhanden	Ist gesetzt, wenn ein Programm der Ablaufsprache korrekt registriert ist, und nicht gesetzt, wenn es nicht registriert ist. Wird nicht gesetzt, wenn die erweiterte Anweisung der Ablaufsprache nicht richtig ist.	S (Initialisierung)	M9100	
SM321	Start/Stopp Programm der Ablaufsprache	AUS: Programm der Ablaufsprache stoppt EIN: Programm der Ablaufsprache startet	Der Initialisierungswert wird auf den gleichen Wert gesetzt wie SM900. (Wird automatisch gesetzt, wenn ein Programm der Ablaufsprache vorhanden ist.) Das Programm der Ablaufsprache wird nicht ausgführt, wenn vor der Programmverarbeitung der Merker auf AUS steht. Wenn der Merker nachträglich von AUS auf EIN wechselt, startet das Programm der Ablaufsprache. Wenn der Merker nachträglich von EIN auf AUS wechselt, stoppt das Programm der Ablaufsprache.	S/B (Initialisierung)	M9101 Geändertes Format	
SM322	Startzustand des Programms der Ablaufsprache	AUS: Initialstart EIN: Neustart	Der Initialwert ist in Anhängigkeit von den Parametern entweder auf EIN oder auf AUS gesetzt. Steht der Merker auf AUS,werden alle Ausführungs-zustände von dem Zeitpunkt an , an dem das Programm der Ablaufsprache gestoppt wurde, zurückgesetzt und starten von dem Initialisierungsschritt aus, von dem die Startanforderung gemacht wurde. Steht der Merker auf EIN, startet das Programm wieder von dem Punkt aus, an dem es beim Stopp aktiv war. (EIN ist nur möglich, wenn in den Parametern die Wiederaufnahme des Programms eingestellt wurde.) SM902 ist nicht automatisch als Latch vorgesehen.	S/B Initialisierung)	M9102 Geändertes Format	
SM323	vorhanden sein/ nicht vorhanden sein von fortlaufenden Transitionen im gesamten Block	AUS: Transition unwirksam EIN: Transition wirksam	Wenn nicht gesetzt, wird in jedem Zyklus eine vorhandene Transition abgearbeitet. Ist der Merker gesetzt, werden in jedem Zyklus alle fortlaufenden Transitionen abgearbeitet. Um die einzelnen Blöcke zu unterscheiden, erhält das Bit einer fortlaufenden Transition in einem Block eine Priorität. (Die Unterscheidung wird vor dem Start eines Blocks überprüft.)	B	M9103	● außer Q00J-, Q00- und Q01CPU
SM324	Verhinderungs-kennung einer fortlaufenden Transition	AUS: Bei ausgeführter Transition EIN: Keine Transition wird ausgeführt	Bei Ausführung einer wirksamen fortlaufenden Transition wird der Merker gesetzt. Zurückgesetzt wird er, wenn die Ausführung beendet ist. Im Normalzustand wird der Merker gesetzt, wenn die Transition nicht wirksam ist.	S (Anweisungs-ausführung)	M9104	
SM325	Ausgabemodus bei einem Blockstopp	AUS: AUS EIN: Wird beibehalten	Bei einem Blockstopp wird der aktive Verarbeitungsschritt ausgegeben. Bei AUS werden alle Ausgänge zurückgesetzt. Bei EIN werden alle Ausgänge beibehalten.	S (Zustandsänderung)	M9196	
SM326	Ausgabemodus für AS-Programm	AUS: Ausgänge löschen EIN: Zustände beibehalten	Mit diesem Merker wird das Verhalten der Ausgänge angewählt, nachdem ein bestehendes AS-Programm geändert wurde und die CPU gestartet wird.	B	Neu	
SM327	Ausgabemodus bei Ausführung der End-Anweisung	AUS: AUS EIN: Wird beibehalten	Mit diesem Merker wird ausgewählt, ob der Zustand der Ausgänge beibehalten wird, wenn der letzte Programmschritt ausgeführt wird. Ist der Merker nicht gesetzt, werden alle Ausgänge rückgesetzt.Bei gesetztem Merker bleibt der Zustand der Ausgänge erhalten.	S (Initialisierung) B	Neu	
SM330	Betriebsart für Programme mit langsamer Ausführung	AUS: Asynchron EIN: Synchron	Asynchron: Die Anweisungen für ein Programm mit langsamer Ausführung werden während der noch im Zyklus zur Verfügung stehenden Zeit fortgesetzt. Synchron: Die Anweisungen für ein Programm mit langsamer Ausführung werden, auch wenn noch genug Zeit vorhanden ist, im nächsten Zyklus begonnen.	B (END-Verarbeitung)	Neu	

(3) Systemtakte und Counter

Adresse	Name	Bedeutung	Beschreibung	Gesetzt vom (wenn gesetzt)	A-CPU M9[ ] [ ] [ ]	Gültig für:
SM400	Immer EIN		Dieser Sondermerker ist immer gesetzt (EIN).	S (END-Verarbeitung)	M9036	
SM401	Immer AUS		Dieser Sondermerker ist immer zurückgesetzt (AUS).	S (END-Verarbeitung)	M9037	
SM402	EIN nur für einen Programmzyklus nach RUN		Nach dem Setzen von RUN wird das Programm für einen Programmzyklus auf EIN gesetzt. Dieses Verfahren kann nur von Programmen genutzt werden, die einmal pro Programmzyklus ausgeführt werden.	S (END-Verarbeitung)	M9038	●
SM403	AUS nur für einen Programmzyklus nach RUN		Nach dem Setzen von RUN wird das Programm für einen Programmzyklus auf AUS gesetzt. Dieses Verfahren kann nur von Programmen genutzt werden, die einmal pro Programmzyklus ausgeführt werden.	S (END-Verarbeitung)	M9039	
SM404	EIN nur für einen Programmzyklus nach RUN		Nach dem Setzen von RUN wird das Programm für einen Programmzyklus auf EIN gesetzt. Dieser Kontakt kann nur von Programmen genutzt werden, die eine langsame Programmausführung beherrschen.	S (END-Verarbeitung)	Neu	● außer Q00J-, Q00- und Q01CPU
SM405	AUS nur für einen Programmzyklus nach RUN		Nach dem Setzen von RUN wird das Programm für einen Programmzyklus auf AUS gesetzt. Dieser Kontakt kann nur von Programmen genutzt werden, die eine langsame Programmausführung beherrschen.	S (END-Verarbeitung)	Neu	
SM409	0,01 s Takt		Wiederholte Änderung zwischen EIN und AUS während eines 10ms-Intervalls Nach Abschaltem des Netzteils oder Zurücksetzen der CPU wird der Merker automatisch von AUS auf EIN gesetzt.	S (Zustandsänderung)	Neu	Q-CPU, außer Q00J-, Q00- und Q01CPU
SM410	0,1 s Takt		Wiederholte Änderung zwischen EIN und AUS während eines vorbestimmten Intervalls. Wird auch während STOP weiter ausgeführt. Nach Abschalten des Netzteils oder Zurücksetzen der CPU wird der Merker automatisch von AUS auf EIN gesetzt.	S (Zustandsänderung)	M9030	●
SM411	0,2 s Takt				M9031	
SM412	1 s Takt				M9032	
SM413	2 s Takt				M9033	
SM414	2 x n s Takt		Wechselt zwischen EIN und AUS entsprechend der in SD414 eingestellten Anzahl von Sekunden.	S (Zustandsänderung)	M9034 Geändertes Format	
SM415	2 x n ms Takt		Wechselt zwischen EIN und AUS entsprechend der in SD415 eingestellten Anzahl von Millisekunden.	S (Zustandsänderung)	Neu	Q-CPU, außer Q00J-, Q00- und Q01CPU

(3) Systemtakte und Counter

Adresse	Name	Bedeutung	Beschreibung	Gesetzt vom (wenn gesetzt)	A-CPU M9[ ] [ ] [ ]	Gültig für:
SM420	Zeittakt Nr. 0		Der Merker wiederholt Wechsel zwischen EIN/AUS innerhalb eines festen Abtastintervalls. Nach Abschalten des Netzteils oder Zurücksetzen der CPU wird der Merker automatisch von AUS auf EIN gesetzt. Die EIN/AUS-Intervalle werden mit einer DUTY-Anweisung eingestellt.	S (END-Verarbeitung)	M9020	●
SM421	Zeittakt Nr.1				M9021	
SM422	Zeittakt Nr. 2				M9022	
SM423	Zeittakt Nr. 3				M9023	
SM424	Zeittakt Nr. 4				M9024	
SM430	Zeittakt Nr. 5		Die Merker SM420 bis SM424 sind für die Nutzung von Programmen mit langsamer Ausführung.	S (END-Verarbeitung)	Neu	● außer Q00J-, Q00- und Q01CPU
SM431	Zeittakt Nr. 6					
SM432	Zeittakt Nr. 7					
SM433	Zeittakt Nr. 8					
SM434	Zeittakt Nr. 9					

(4) Zyklusinformation

Adresse	Name	Bedeutung	Beschreibung	Gesetzt vom (wenn geetzt)	A-CPU M9[ ] [ ] [ ]	Gültig für:
SM510	Ausführungskennung für Programme mit geringer Geschwindigkeit	AUS: Fertig oder nicht ausgeführt EIN: Programm wird ausgeführt	Wird gesetzt, wenn Programme mit geringer Geschwindigkeit ausgeführt werden.	S (END-Verarbeitung)	Neu	● außer Q00J-, Q00- und Q01CPU
SM551	Liest während des Serviceintervalls Module aus	AUS: Nicht ausgeführt EIN: Liest	Wenn dieser Merker von AUS auf EIN wechselt, wird das Serviceintervall ausgeführt, das unter SD550 festgelegt ist. Es werden SD550 bis SD551 ausgelesen.	B	Neu	

(5) Speicherkarten

Adresse	Name	Bedeutung	Beschreibung	Gesetzt vom (wenn gesetzt)	A-CPU M9[ ] [ ] [ ]	Gültig für:
SM600	Speicherkarte A Verwendbarkeits-kennung	AUS: nicht verwendbar EIN: verwendbar	Ist gesetzt, wenn die Speicherkarte A fertig zur Verwendung durch den Benutzer ist.	S (Initialisierung)	Neu	● außer Q00J-, Q00- und Q01CPU
SM601	Speicherkarte A Schutzkennung	AUS: ungeschüzt EIN: geschüzt	Wird gesetzt wenn der Schutzschalter der Speicherkarte A gesetzt ist.	S (Initialisierung)	Neu	
SM602	Kennung Laufwerk 1	AUS: Laufwerk 1 nicht vorhanden EIN: Laufwerk 1 vorhanden	Wird gesetzt wenn das Laufwerk 1 (Karte 1 RAM-Bereich) vorhanden ist.	S (Initialisierung)	Neu	
SM603	Kennung Laufwerk 2	AUS: Laufwerk 2 nicht vorhanden EIN: Laufwerk 2 vorhanden	Wird gesetzt, wenn das Laufwerk 2 (Karte 1 ROM-Bereich) vorhanden ist.	S (Initialisierung)	Neu	
SM604	Benutzungskennung Speicherkarte A	AUS: Nicht in Benutzung EIN: In Benutzung	Wird gesetzt, wenn die Speicherkarte A in Benutzung ist.	S (Initialisierung)	Neu	
SM605	Verriegelungskennung Speicherkarte A	AUS: Entfernen/Einsetzen möglich EIN: Entfernen /Einsetzen verboten	Wird gesetzt, wenn die Speicherkarte A nicht entfernt oder eingesetzt werden darf.	B	Neu	
SM620	Verwendbarkeits-kennung Speicherkarte B	AUS: Nicht verwendbar EIN: Verwendbar	Immer EIN.	S (Initialisierung)	Neu	Q-CPU
			Ist gesetzt, wenn die Speicherkarte B einsatzbereit ist.	S (Initialisierung)	Neu	Q2A(S1), Q3A, Q4A, Q4AR
SM621	Schutzkennung Speicherkarte B	AUS: Ungeschüzt EIN: Geschüzt	Immer EIN.	S (Initialisierung)	Neu	Q-CPU
			Wird gesetzt, wenn der Schalter zum Schutz der Speicherkarte B eingeschaltet ist.	S (Initialisierung)	Neu	Q2A(S1), Q3A, Q4A, Q4AR
SM622	Kennung Laufwerk 3	AUS: Laufwerk 3 nicht vorhanden EIN: Laufwerk 3 vorhanden	Immer EIN.	S (Initialisierung)	Neu	Q-CPU
			Wird gesetzt, wenn das Laufwerk 3 (Karte 2 RAM-Bereich) vorhanden ist.	S (Initialisierung)	Neu	Q2A(S1), Q3A, Q4A, Q4AR
SM623	Kennung Laufwerk 4	AUS: Laufwerk 4 nicht vorhanden EIN: Laufwerk 4 vorhanden	Immer EIN.	S (Initialisierung)	Neu	Q-CPU
			Wird gesetzt, wenn das Laufwerk 4 (Karte 2 ROM-Bereich) vorhanden ist.	S (Initialisierung)	Neu	Q2A(S1), Q3A, Q4A, Q4AR
SM624	Benutzungskennung Speicherkarte B	AUS: Nicht in Benutzung EIN: In Benutzung	Wird gesetzt, wenn die Speicherkarte B in Benutzung ist.	S (Initialisierung)	Neu	
SM625	Verriegelungskennung Speicherkarte B	AUS: Entfernen/Einsetzen möglich EIN: Entfernen /Einsetzen verboten	Wird gesetzt, wenn die Speicherkarte B nicht entfernt oder eingesetzt werden darf.	B	Neu	

(5) Speicherkarten

Adresse	Name	Bedeutung	Beschreibung	Gesetzt vom (wenn gesetzt)	A-CPU M9[ ] [ ] [ ]	Gültig für:
SM640	File-Register-Gebrauch	AUS: File-Register nicht in Benutzung EIN: File-Register in Benutzung	Wird gesetzt, wenn das File-Register benutzt wird.	S (Zustandsänderung)	Neu	●
SM650	Kommentargebrauch	AUS: Kommentar nicht in Benutzung EIN: Kommentar in Benutzung	Wird gesetzt, wenn die Kommentardatei benutzt wird.	S (Zustandsänderung)	Neu	● außer Q00J-, Q00- und Q01CPU
SM660	Boot-Vorgang	AUS: Interne Speicherausführung EIN: Boot-Vorgang im Vollzug	Wird gesetzt, solange der Boot-Vorgang läuft. Wird zurückgesetzt, wenn der Boot-Wahlschalter nicht gesetzt ist.	S (Zustandsänderung)	Neu	●
SM672	File-Register-Zugriffsbereichs-kennung Speicherkarte A	AUS: Innerhalb des Zugriffsbereichs EIN: Außerhalb des Zugriffsbereichs	Wird gesetzt, wenn ein Zugriff außerhalb des Bereichs vom File-Register R der Speicherkarte A versucht wurde. (Gesetzt innerhalb der END-Verarbeitung) Wird vom Anwenderprogramm zurückgesetzt.	S/B	Neu	● außer Q00J-, Q00- und Q01CPU
SM673	File-Register-Zugriffsbereichs-kennung Speicherkarte B	AUS: Innerhalb des Zugriffsbereichs EIN: Außerhalb des Zugriffsbereichs	Wird gesetzt, wenn ein Zugriff außerhalb des Bereichs vom File-Register R der Speicherkarte B versucht wurde. (Gesetzt innerhalb der END-Verarbeitung) Wird vom Anwenderprogramm zurückgesetzt.	S/B	Neu	Q2A(S1), Q3A, Q4A, Q4AR

(6) Diagnosemerker, die sich auf Anweisungen beziehen

Adresse	Name	Bedeutung	Beschreibung	Gesetzt vom (wenn gesetzt)	A-CPU M9[ ] [ ] [ ]	Gültig für:
SM700	Übertragskennung	AUS: Übertrag AUS EIN: Übertrag EIN	Die Übertragskennung wird in anwendungs-spezifischen Anweisungen benutzt.	S (Anwendungs-ausführung)	M9012	●
SM701	Auswahl von auszugebenden Zeichen	AUS: Ausgabe von 16 Zeichen EIN: Ausgabe bis zum NUL-Zeichen	Ist der Merker gesetzt, werden 16 Zeichen im ASCII-Code ausgegeben. Ist der Merker nicht gesetzt, wird die Ausgabe bis zum NUL-Zeichen (00_H) durchgeführt.	B	M9049	● außer Q00J-, Q00- und Q01CPU
SM702	Suchmethode	AUS: Nächste Suche EIN: 2-teilige Suche	Vorbestimmte Methode, die von Suchanweisungen benutzt wird. Die Daten müssen für die zweiteilige Suche geordnet sein.	B	Neu	●
SM703	Sortierbefehl	AUS: Aufsteigend EIN: Absteigend	Die Sortierbefehle benötigen eine Festlegung, ob aufsteigend oder absteigend sortiert werden soll.	B	Neu	
SM704	Blockweiser Vergleich	AUS: Nichts Passendes gefunden EIN: Alles stimmt überein	Wird gesetzt, wenn alle Bedingungen der Daten mit der BKCMP-Anweisung übereinstimmen.	S (Anwendungs-ausführung)	Neu	

(6) Diagnosemerker, die sich auf Anweisungen beziehen

Adresse	Name	Bedeutung	Beschreibung	Gesetzt vom (wenn gesetzt)	A-CPU M9[ ] [ ] [ ]	Gültig für:
SM707	Ausführungsmodus für Anweisungen mit Gleitkommazahlen.	AUS: Geschwindigkeits orientiert EIN: Genauigkeits orientiert	Wenn SM707 nicht gesetzt ist, werden Anweisungen mit Gleitkommazahlen mit hoher Geschwindigkeit ausgeführt. Bei gesetztem SM707 werden Befehle, die sich auf Gleitkommazahlen beziehen, mit hoher Genauigkeit ausgeführt.	B	Neu	Q4AR
SM710	Prüfpriorität in der CHK-Anweisung	AUS: Kontaktzustands-priorität EIN: Fehlerprüfnetz-werkpriorität	Ist der Merker nicht gesetzt, verbleibt die Einstellung. Ist der Merker gesetzt, wird die CHK-Priorität aktualisiert.	S (Anwendungs-ausführung)	Neu	● außer Q00J-, Q00- und Q01CPU
SM711	Anzeige der geteilten Übertragung	AUS: Während anderer Verarbeitung EIN: Während der geteilten Verarbeitung	Steht die CPU mit einem AD57(S1)-Modul in Verbindung, wird der Merker bei einer Übertragung mit geteiltem Bildschirm auf EIN gesetzt. Ist die Übertragung abgeschlossen, wird der Merker zurückgesetzt.	S (Anwendungs-ausführung)	M9065	QnA
SM712	Auswahl der Übertragungs-verarbeitung	AUS: Batch-Übertragung EIN: geteilte Übertra-gung	In Verbindung mit dem AD57(S1)-Modul, wird dieser Merker gesetzt, wenn sich die Bildschirmmaske für die Übertragung geteilt hat.	S (Anwendungs-ausführung)	M9066	
SM714	Betriebssignal (BUSY-Signal) des Registrationsbereichs der Kommunikations-anforderung	AUS: Kommunikations-anforderung an ein Remote-Terminalmodul ist möglich EIN: Kommunikations-anforderung an ein Remote-Terminal-Modul ist nicht möglich	Wird zur Festlegung benötigt, ob Kommunikations-anfragen zu einem Remote-Terminal- Modul, das mit einem AJ71PT32-S3-Modul oder einer A2C-CPU verbunden ist, ausgeführt werden oder nicht.	S (Anwendungs-ausführung)	M9081	
SM715	EI-Kennung	AUS: Während DI EIN: Während EI	Wird gesetzt, wenn die EI-Anweisung ausgeführt wird.	S (Anwendungs-ausführung)	Neu	●
SM720	Kommentar wurde gelesen	AUS: Lesen der Kommentare nicht abgeschlossen EIN: Lesen der Kommentare abgeschlossen	Dieser Merker wird nach Ausführung der COMRD- oder PRC-Anweisung nur für einen Zyklus gesetzt.	S (Zustandsänderung)	Neu	Q-CPU, außer Q00J-, Q00- und Q01CPU
SM721	Auf ein File wird zugegriffen	AUS: Auf File wird nicht zugegriffen EIN: Auf File wird zugegriffen	Während auf ein File mit den Anweisungen S.FWRITE, S.FREAD, COMRD, PRC oder LEDC zugegriffen wird, wird dieser Merker gesetzt.	S (Zustandsänderung)	Neu	Q-CPU
SM722	Fehlermeldung für die Anweisungen BIN und DBIN sperren	AUS: Fehlermeldung ist freigegeben EIN: Fehlermeldung ist gesperrt	Um die Meldung eines Ausführungsfehlers bei den Befehlen BIN oder DBIN zu sperren, muss dieser Merker gesetzt werden.	B	Neu	
SM730	BUSY-Signal für CC-Link Kommunikations-anforderungsbereich	AUS: Anforderung zum Datenaustausch mit Sondermodul freigegeben EIN: Anforderung zum Datenaustausch gesperrt	Dieser Merker wird benutzt, um das Signal zur Anforderung der Kommunikation mit dem Sonder-moduls, das an dem A(1S)J61QBT11 angeschlossen ist, freizugeben oder zu sperren.	S (Anwendungs-ausführung)	Neu	QnA

(6) Diagnosemerker, die sich auf Anweisungen beziehen

Adresse	Name	Bedeutung	Beschreibung	Gesetzt vom (wenn gesetzt)	A-CPU M9[ ] [ ] [ ]	Gültig für:
SM736	Kennung der PKEY-Anweisungs-ausführung	AUS: Anweisung wird nicht ausgeführt EIN: Anweisung wird ausgeführt	Wird gesetzt, wenn die PKEY-Anweisung ausgeführt wird. Wird zurückgesetzt, wenn die Zeichenfolge CR empfangen wird oder die Zeichenfolge 32 Zeichen erreicht hat.	S (Anwendungs-ausführung)	Neu	● außer Q00J-, Q00- und Q01CPU
SM737	Empfangskennung einer Tastatureingabe der PKEY-Anweisung	AUS: Tastatureingabe-Empfang möglich EIN: Tastatureingabe-Empfang nicht möglich	Wird gesetzt, wenn eine Tastatureingabe gemacht wurde. Wird zurückgesetzt wenn, die Tastatureingabe von der CPU gespeichert wurde.	S (Anwendungs-ausführung)	Neu	
SM738	Empfangskennung der MSG-Anweisung	AUS: Anweisung wird nicht ausgeführt EIN: Anweisung wird ausgeführt	Wird gesetzt, wenn die MSG-Anweisung ausgeführt wird.	S (Anwendungs-ausführung)	Neu	
SM774	PID-Regelung mit stoßfreier Regel-charakteristik	AUS: Erzwingt Übereinstimmung EIN: Übereinstimmung wird nicht erzwungen	Im manuellen Betrieb wird festgelegt, ob der Istwert zur Übereinstimmung mit dem Sollwert gezwungen wird.	B	Neu	
SM775	Auswahl der Link-Aktualisierungs-Verarbeitung während der Ausführung der COM-Anweisung	AUS: Link-Aktualisierung wird ausgeführt EIN: Link-Aktualisierung wird nicht ausgeführt	Ist der Merker nicht gesetzt, wird eine Aktualisierung der Netzwerk- und Schnittstellen-Daten (Link Refresh) und eine Gesamtdatenverarbeitung (END-Verarbeitung) ausgeführt. Ist der Merker gesetzt, wird nur eine Gesamtdaten-verarbeitung (END-Verarbeitung) ausgeführt.	B	Neu	●
SM776	Lokale Geräte freigeben oder sperren bei der Anweisung CALL	AUS: Lokale Geräte gesperrt EIN: Lokale Geräte freigegeben	Mit diesem Merker wird festgelegt, ob lokale Geräte in dem Programm, das mit der Anweisung CALL aufgerufen wird, freigegeben oder gesperrt werden.	B (Zustandsänderung)	Neu	● außer Q00J-, Q00- und Q01CPU
SM777	Lokale Geräte während eines Interrupt-programmes freigeben oder sperren	AUS: Lokale Geräte gesperrt EIN: Lokale Geräte freigegeben	Mit diesem Merker wird festgelegt, ob lokale Geräte während eines Interruptprogrammes freigegeben oder gesperrt werden.	B (Zustandsänderung)	Neu	
SM780	CC-Link bezogene Anweisungen sind ausführbar	AUS: CC-Link bezogene Anweisungen sind ausführbar EIN: CC-Link bezogene Anweisungen sind nicht ausführbar	Wenn die Anzahl der CC-Link bezogenen Anweisungen, die gleichzeitig ausgeführt werden können, 32 erreicht, wird dieser Merker gesetzt. Der Merker wird rückgesetzt, wenn die Zahl der Aweisungen 32 wieder unterschreitet.	S (Zustandsänderung)	Neu	QnA

(7) Fehlerbeseitigung

Adresse	Name	Bedeutung	Beschreibung	Gesetzt vom (wenn gesetzt)	A-CPU M9[ ] [ ] [ ]	Gültig für:
SM800	Vorbereitung eines Trace	AUS: nicht vorbereitet EIN: Vorbereitung beendet	Wird gesetzt, wenn die Vorbereitung eines Trace abgeschlossen ist.	S (Zustandsänderung)	Neu	Q-CPU, außer Q00J-, Q00- und Q01CPU
	Vorbereitung eines Sampling Trace			S (Zustandsänderung)		QnA-CPU
SM801	Trace startet	AUS: Trace/Sampling Trace stoppen EIN: Trace/Sampling Trace starten	Der Merker wird zum Starten des Trace/Sampling Trace gesetzt und zum Stoppen zurückgesetzt. (In Verbindung mit allen zurückgesetzten Diagnosemerkern)	B	M9047	Q-CPU, außer Q00J-, Q00- und Q01CPU
	Sampling Trace startet			B		QnA-CPU
SM802	Trace während der Ausführung	AUS: Trace/Sampling Trace stoppen EIN: Trace/Sampling Trace starten	Wird während der Ausführung des Trace/Sampling Trace gesetzt.	S (Zustandsänderung)	M9046	Q-CPU, außer Q00J-, Q00- und Q01CPU
	Sampling Trace während der Ausführung			S (Zustandsänderung)		QnA-CPU
SM803	Trace Trigger	AUS → EIN: Start	Der Merker entspricht in seiner Funktion der TRACE-Anweisung. Der Sampling Trace-Trigger wird gesetzt, wenn der Merker von AUS auf EIN wechselt	B	M9044	Q-CPU, außer Q00J-, Q00- und Q01CPU
	Sampling Trace-Trigger		Der Merker entspricht in seiner Funktion der STRA-Anweisung. Der Sampling Trace-Trigger wird gesetzt, wenn der Merker von AUS auf EIN wechselt.	B		QnA-CPU
SM804	Nach dem Trace-Trigger	AUS: Nicht nach dem Trigger EIN: Nach dem Trigger	Der Merker wird nach dem Trace-Trigger gesetzt.	S (Zustandsänderung)	Neu	Q-CPU, außer Q00J-, Q00- und Q01CPU
	Nach dem Sampling Trace-Trigger		Der Merker wird nach dem Sampling Trace-Trigger gesetzt.	S (Zustandsänderung)		QnA-CPU
SM805	Trace beendet		Der Merker wird nach Beendung des Trace gesetzt.	S (Zustandsänderung)	M9043	Q-CPU, außer Q00J-, Q00- und Q01CPU
	Sampling Trace beendet	AUS: Während eines Trace/Sampling Trace EIN: Trace/Sampling Trace beendet	Der Merker wird nach Beendung des Sampling Trace gesetzt.	S (Zustandsänderung)		QnA-CPU

(7) Fehlerbeseitigung

Adresse	Name	Bedeutung	Beschreibung	Gesetzt vom (wenn gesetzt)	A-CPU M9[ ] [ ] [ ]	Gültig für:
SM806	Status Latch Vorbereitung	AUS: Nicht vorbereitet EIN: Fertig	Wird gesetzt, wenn die Vorbereitung abgeschlossen ist.	S (Zustandsänderung)	Neu	QnA-CPU
SM807	Status-Latch-Befehl	AUS → EIN: Latch	Ausführung des Status-Latch-Befehl	B	Neu	
SM808	Status Latch vollständig	AUS: Staus Latch nicht beendet EIN: Staus Latch beendet	Wird gesetzt, sobald ein Status Latch beendet ist.	S (Zustandsänderung)	M9055	
SM809	Status Latch frei	AUS → EIN: frei	Nächstes Status Latch ist möglich.	B	Neu	
SM810	Programm-überwachung wird vorbereitet	AUS: Nicht vorbereitet EIN: Vorbereitung beendet	Wird gesetzt, wenn die Vorbereitung zur Programmüberwachung abgeschlossen ist.	S (Zustandsänderung)	Neu	
SM811	Start der Programm-überwachung	AUS: Programm-überwachung stoppen EIN: Programm-überwachung starten	Der Merker wird zum Starten der Programm-überwachung gesetzt und zum Stoppen zurück-gesetzt. (In Verbindung mit allen zurückgesetzten Diagnosemerkern.)	B	Neu	
SM812	Programm-überwachung wird ausgeführt	AUS: Programm-überwachung stoppen EIN: Programm-überwachung starten	Wird bei Ausführung der Programmüberwachung gesetzt.	S (Zustandsänderung)	Neu	QnA-CPU
SM813	Programm-überwachungs-Trigger	AUS → EIN: Start	Der Merker entspricht in seiner Funktion der PTRA-Anweisung . Der Programmüberwachungs-Trigger wird gesetzt, wenn der Merker von AUS auf EIN wechselt.	B	Neu	
SM814	Nach dem Programm-überwachungs-Trigger	AUS: Nicht nach dem Trigger EIN: Nach dem Trigger	Der Merker wird nach dem Programm-überwachungs-Trigger gesetzt.	S (Zustandsänderung)	Neu	
SM815	Programm-überwachung beendet	AUS: Während einer Programm-überwachung EIN: Programm-überwachung beendet	Der Merker wird nach Beendung der Programm-überwachung gesetzt.	S (Zustandsänderung)	Neu	
SM820	Vorbereitung der Schrittüberwachung	AUS: Nicht vorbereitet EIN: Vorbereitung beendet	Wird gesetzt, wenn die Vorbereitung zur Schrittüberwachung abgeschlossen ist.	S (Zustandsänderung)	Neu	

(7) Fehlerbeseitigung

Adresse	Name	Bedeutung	Beschreibung	Gesetzt vom (wenn gesetzt)	A-CPU M9[ ] [ ] [ ]	Gültig für:
SM821	Schrittüberwachung startet	AUS: Schrittüber-wachung stoppen EIN: Schrittüber-wachung starten	Der Merker wird zum Starten des Schrittüberwachung gesetzt und zum Stoppen zurückgesetzt. (In Verbindung mit allen zurückgesetzten Diagnosemerkern.)	B	M9182 Geändertes Format	● außer Q00J-, Q00- und Q01CPU
SM822	Schrittüberwachung wird ausgeführt	AUS: Schrittüber-wachung stoppen EIN: Schrittüber-wachung starten	Wird bei Ausführung der Schrittüberwachung gesetzt. Wird bei abgeschlossener oder ausgesetzter Schrittüberwachung zurückgesetzt.	S (Zustandsänderung)	M9181	
SM823	Schrittüberwachung hat einen Block getriggert	AUS: Nicht nach dem Trigger EIN: Nach dem ersten Trigger	Nachdem der erste Block, in dem die Schrittüberwachung ausgeführt wurde, getriggert wurde, wird der Merker gesetzt. Er wird zurückgesetzt, wenn die Schrittüber-wachung beginnt.	S (Zustandsänderung)	Neu	
SM824	Schrittüberwachung hat alle Blöcke getriggert	AUS: Nicht nach dem Trigger EIN: nach allen Triggern	Nachdem alle Blöcke, in denen die Schrittüberwachung ausgeführt wurde, getriggert wurden, wird der Merker gesetzt. Er wird zurückgesetzt, wenn die Schrittüber-wachung beginnt.	S (Zustandsänderung)	Neu	
SM825	Schrittüberwachung ist abgeschlossen	AUS: Während der Schrittüberwachung EIN: Schrittüber-wachung beendet	Der Merker wird nach Beendung der Schritt-überwachung gesetzt. Der Merker wird bei Beginn der Schrittüberwachung zurückgesetzt.	S (Zustandsänderung)	M9180	
SM826	Fehler beim Trace	AUS: Normal EIN: Fehler	Der Merker wird gesetzt, wenn während des Trace ein Fehler auftritt.	S (Zustandsänderung)	Neu	Q-CPU, außer Q00J-, Q00- und Q01CPU
	Fehler im Sampling Trace		Tritt während des Sampling Trace ein Fehler auf, wird der Merker gesetzt.	S (Zustandsänderung)		QnA-CPU
SM827	Fehler im Status Latch	AUS: Normal EIN: Fehler	Tritt während des Status Latch ein Fehler auf, wird der Merker gesetzt.	S (Zustandsänderung)	Neu	
SM828	Fehler in der Programmüberwachung	AUS: Normal EIN: Fehler	Tritt während des Programmüberwachung ein Fehler auf, wird der Merker gesetzt.	S (Zustandsänderung)	Neu	

(8) Latch-Bereich

Adresse	Name	Bedeutung	Beschreibung	Gesetzt vom (Wenn gesetzt)	A-CPU M9[ ] [ ] [ ]	Gültig für:
SM900	Bearbeitete Datei während eines Spannungsabfalls	AUS: Kein Datei während Spannungsabfall EIN: Datei während Spannungsabfall bearbeitet	Wenn während der Bearbeitung einer Datei die Spannung abgefallen ist, wird dieser Merker gesetzt.	S/B (Zustandsänderung)	Neu	QnA-CPU
SM910	PKEY Registrations-Flag	AUS: Tastatureingabe nicht registriert EIN: Tastatureingabe registriert	Bei Registrieung einer Tastatureingabe wird dieser Merker gesetzt. Zurückgesetzt wird er, wenn die Tastatureingabe nicht registriert wurde.	S (Anweisungsaus-führung)	Neu	● außer Q00J-, Q00- und Q01CPU

(9) Übereinstimmungen zwischen Sondermerkern (A-Serie) und Diagnosemerkern (QnA-Serie/System Q)

Bei der Umstellung von der MELSEC A-Serie zur MELSEC Q-Serie oder dem System Q entsprechen die Sondermerker M9000 bis M9255 (MELSEC A-Serie) den Diagnosemerkern SM1000 bis SM1255 (MELSEC Q-Serie).
Diese Diagnosemerker werden alle durch das System gesetzt und können nicht durch ein Anwenderprogramm verändert werden. Benutzer, die diese Merker setzen oder rücksetzen wollen, sollten ihre Programme so ändern, dass nur reine QnA-Diagnosemerker verwendet werden. Eine Ausnahme bilden die Sondermerker M9084 und M9200 bis M9255. Wenn diese Merker vor der Umstellung zur MELSEC Q-Serie/System Q gesetzt und rückgesetzt werden konnten, so ist das nach der Umstellung auch mit den entsprechenden Diagnosemerkern SM1084 und SM1200 bis SM1255 möglich.
Detaillierte Informationen zu den Sondermerkern der A-Serie können den Handbüchern zu den CPUs und den Netzwerken „MELSECNET“ und „MELSECNET/B“ entnommen werden.

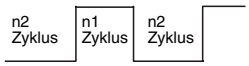
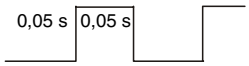
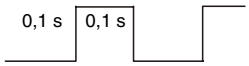
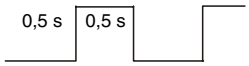
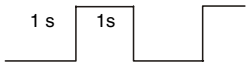
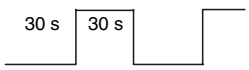
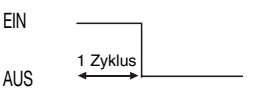
HINWEIS Die Verarbeitungszeit kann sich bei der Q-CPU verlängern, wenn umgewandelte Sondermerker verwendet werden. Wählen Sie in der Programmier-Software bei den SPS-Parametern auf der Karteikarte „SPS-System“ die Option „*A-SPS: Verw. Sondermerker/Sonderregister von SM/SD 1000*“ ab, wenn keine umgewandelten Sondermerker benutzt werden.

Wenn ein aquivalenter Diagnosemerker für eine System Q- oder QnA-CPU angegeben ist, sollte das Programm geändert und dieser Merker verwendet werden. Wenn kein äquivalenter System Q-/QnA-Diagnosemerker angegeben ist, kann der Merker verwendet werden, der nach der Umstellung angegeben wird.

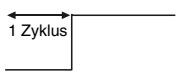
Liste der Sondermerker und Diagnosemerker

A-CPU Sondermerker	Diagnosemerker nach der Umstellung	Äquivalente System Q-/QnA-Diagnosemerker	Name	Bedeutung	Gülig für
M9000	SM1000	—	Sicherung defekt	AUS: Normal EIN: Defekt	Q-/QnA-CPU
M9002	SM1002	—	Vergleichsfehler E/A-Modul	AUS: Normal EIN: Fehler	
M9004	SM1004	—	Fehler im Master-Modul des MELSECNET MINI	AUS: Normal EIN: Fehler	QnA-CPU
M9005	SM1005	—	Spannungsabfall in der Netzspannung	AUS: Normal EIN: Spannung abgefallen	Q-/QnA-CPU
M9006	SM1006	—	Niedrige Batteriespannung	AUS: Normal EIN: Spannung abgefallen	
M9007	SM1007	—	Niedrige Batteriespannung (Latch-Merker)	AUS: Normal EIN: Spannung abgefallen	
M9008	SM1008	SM1	Fehlererkennung nach Selbstdiagnose	AUS: Normal EIN: Fehler	
M9009	SM1009	SM62	Fehlermelderkennung	AUS: Keine Kennung EIN: Kennung	
M9011	SM1011	SM56	Fehlererkennung im Programmablauf	AUS: Normal EIN: Fehler	
M9012	SM1012	SM700	Carry Flag (Übertragmerker)	AUS: Carry aus EIN: Carry ein	
M9016	SM1016	Keine Funktion bei Q-/QnA-CPU	Löschkennung gespeicherter Operandendaten	AUS: Keine Ausführung EIN: Löschvorgang	

Liste der Sondermerker und Diagnosemerker

A-CPU Sondermerker	Diagnosemerker nach der Umstellung	Äquivalente System Q-/QnA-Diagnosemerker	Name	Bedeutung	Gülig für
M9017	SM1017	Keine Funktion bei einer System Q- oder QnA-CPU	Löschkennung gespeicherter Operandendaten	AUS: Keine Ausführung EIN: Löschvorgang	Q-/QnA-CPU
M9020	SM1020		Zeittakt Nr. 0		
M9021	SM1021		Zeittakt Nr. 1		
M9022	SM1022		Zeittakt Nr. 2		
M9023	SM1023		Zeittakt Nr. 3		
M9024	SM1024		Zeittakt Nr. 4		
M9025	SM1025		Setzanforderung für Uhrdaten	AUS: Keine Verarbeitung EIN: Anforderung	
M9026	SM1026		Uhrdatenfehler	AUS: Normal EIN: Fehler	
M9027	SM1027		Uhrdatenanzeige	AUS: Keine Verarbeitung EIN: Anforderung	
M9028	SM1028		Leseanforderung für Uhrdaten	AUS: Normal EIN: Fehler	
M9029	SM1029	Keine Funktion bei einer System Q- oder QnA-CPU	Stapelverarbeitung von Daten einer Kommunikations-anforderung	AUS: Stapelverarbeitung wird nicht ausgeführt EIN: Stapelverarbeitung wird ausgeführt	
M9030	SM1030		Taktgeber 0,1 Sekunden		
M9031	SM1031		Taktgeber 0,2 Sekunden		
M9032	SM1032		Taktgeber 1 Sekunden		
M9033	SM1033		Taktgeber 2 Sekunden		
M9034	SM1034		Taktgeber 1 Minute		
M9036	SM1036		Ständig EIN		
M9037	SM1037		Ständig AUS		
M9038	SM1038		EIN für 1 Zyklus nur nach RUN		

Liste der Sondermerker und Diagnosemerker

A-CPU Sondermerker	Diagnosemerker nach der Umstellung	Äquivalente System Q-/QnA-Diagnosemerker	Name	Bedeutung	Gülig für
M9039	SM1039	—	AUS nur für 1 Zyklus nach RUN	EIN  AUS	Q-/QnA-CPU
M9040	SM1040	SM206	Pausenbedingung	AUS: PAUSE nicht möglich EIN : PAUSE möglich	
M9041	SM1041	SM204	Kennung des PAUSE-Status	AUS: Keine PAUSE EIN: Während einer PAUSE	
M9042	SM1042	SM203	Kennnung des STOP-Status	AUS: Kein STOP EIN: Bei STOP	
M9043	SM1043	SM805	Sampling Trace beendet	AUS: Während eines Sampling Trace EIN: Nach Ende des Sampling Trace	
M9044	SM1044	SM803	Sampling Trace	0 → 1 Gleicht der Ausführung der STRA-Anweisung 1 → 0 Gleicht der Ausführung der STRAR-Anweisung	
M9045	SM1045	Keine Funktion bei einer System Q- oder QnA-CPU	Watchdog-Timer zurücksetzen	AUS: Kein Zurücksetzen EIN: Watchdog-Timer wird zurückgesetzt	
M9046	SM1046	SM802	Sampling Trace	AUS: Überwachung ist nicht aktiv EIN: Überwachung ist aktiv	
M9047	SM1047	SM801	Vorbereitung des Sampling Trace	AUS: Sampling Trace stoppen EIN: Sampling Trace starten	
M9049	SM1049	SM701	Anzahl der ausgegebenen Zeichen	AUS: Ausgabe bis zum NUL-Code EIN: Ausgabe von 16 Zeichen	
M9051	SM1051	Keine Funktion bei einer System Q- oder QnA-CPU	Unterdrückung der CHG-Anweisung	AUS: Ausführung möglich EIN: Ausführung nicht möglich	
M9052	SM1052	Keine Funktion bei einer System Q- oder QnA-CPU	Umschaltung der SEG-Anweisung	AUS: 7-Segmentanzeige EIN: E/A-Teilaktualisierung	
M9054	SM1054	SM205	Kennung des STEP RUN	AUS: Andere Betriebsart EIN: STEP RUN	
M9055	SM1055	SM808	Kennung des Status Latch	AUS: Nicht beendet EIN: Beendet	QnA-CPU
M9055	SM1055	SM808	Kennung des Status Latch	AUS: Nicht beendet EIN: Beendet	
M9056	SM1056	Keine Funktion bei einer System Q- oder QnA-CPU	Anforderung von P, I für das Hauptprogramm	AUS: Keine Anforderung EIN: Anforderung von P, I	Q-/QnA-CPU
M9057	SM1057		Anforderung von P, I für das Unterprogramm	AUS: Keine Anforderung EIN: Anforderung von P, I	
M9058	SM1058		Hauptprogramm P, I abgeschlossen	Kurzzeitig EIN wenn P, I abgeschlossen ist	
M9059	SM1059		Unterprogramm P, I abgeschlossen	Kurzzeitig EIN wenn P, I abgeschlossen ist	
M9060	SM1060		Anforderung von P, I für das Unterprogramm 2	AUS: Keine Anforderung EIN: Anforderung von P, I	
M9061	SM1061		Anforderung von P, I für das Unterprogramm 3	AUS: Keine Anforderung EIN: Anforderung von P, I	

Liste der Sondermerker und Diagnosemerker

A-CPU Sondermerker	Diagnosemerker nach der Umstellung	Äquivalente System Q-/QnA-Diagnosemerker	Name	Bedeutung	Gülig für
M9065	SM1065	SM711	Kennung der schrittweisen Übertragung	AUS: Andere Verarbeitung EIN: schrittweise Übertragung	QnA-CPU
M9066	SM1066	SM712	Umschaltung der Transferverarbeitung	AUS: Stapel-Transfer EIN: Schrittweise Übertragung	
M9070	SM1070	Keine Funktion bei einer System Q- oder QnA-CPU	A8UPU/A8PUJ benötigte Suchzeit	AUS: Lesezeit nicht verkürzt EIN: Lesezeit verkürzt	Q-/QnA-CPU
M9081	SM1081	SM714	Kommunikationsanforderung an ein Remote-Sondermodul	AUS: Anforderung ist möglich EIN: Anforderung ist nicht möglich	QnA-CPU
M9084	SM1084	Keine Funktion bei einer System Q- oder QnA-CPU	Fehlerkontrolle	AUS: Fehlerkontrolle gegeben EIN: Keine Fehlerkontrolle	Q-/QnA-CPU
M9091	SM1091		Anweisungsfehlerkennung	AUS: Normal EIN: Fehler	
M9094	SM1094	SM251	Änderungskennung der E/A-Module	AUS: Änderung EIN: Keine Änderung	QnA-CPU
M9100	SM1100	SM320	Vorhandensein/Fehlen eines Programms der Ablaufsprache	AUS: Programme der Ablaufsprache nicht verwendet EIN: Programme der Ablaufsprache werden verwendet	Q-/QnA-CPU
M9101	SM1101	SM321	Start/Stopp eines Programms der Ablaufsprache	AUS: Programme der Ablaufsprache stoppen EIN: Programme der Ablaufsprache starten	
M9102	SM1102	SM322	Startzustand eines Programms der Ablaufsprache	AUS: Initialstart: EIN: Fortsetzen	
M9103	SM1103	SM323	Vorhandensein/Fehlen fortlaufender Transitionen	AUS: Transition ohne Wirkung EIN: Transition wirksam	
M9104	SM1104	SM324	Anzeige-Flag der fortlaufenden Transition	AUS: Bei abgearbeiteter Transition EIN: Keine Transition	
M9108	SM1108	SM90	Schritt Transition Watchdog-Timer startet (äquivalent zu D9108)	AUS: Watchdog-Timer zurückgesetzt EIN: Zurückgesetzter Watchdog-Timer startet	
M9109	SM1109	SM91	Schritt Transition Watchdog-Timer startet (äquivalent zu D9109)		
M9110	SM1110	SM92	Schritt Transition Watchdog-Timer startet (äquivalent zu D9110)		
M9111	SM1111	SM93	Schritt Transition Watchdog-Timer startet (äquivalent zu D9111)		
M9112	SM1112	SM94	Schritt Transition Watchdog-Timer startet (äquivalent zu D9112)		
M9113	SM1113	SM95	Schritt Transition Watchdog-Timer startet (äquivalent zu D9113)		
M9114	SM1114	SM96	Schritt Transition Watchdog-Timer startet (äquivalent zu D9114)		

Liste der Sondermerker und Diagnosemerker

A-CPU Sondermerker	Diagnosemerker nach der Umstellung	Äquivalente System Q-/QnA-Diagnosemerker	Name	Bedeutung	Gülig für
M9180	SM1180	SM825	Vollendungs-Flag der Abtastüberwachung des aktiven Schritts	AUS: Abtastüberwachung startet EIN: Abtastüberwachung vollendet	Q-/QnA-CPU
M9181	SM1181	SM822	Ausführungs-Flag der Abtastüberwachung des aktiven Schritts	AUS: Abtastüberwachung wird nicht ausgeführt EIN: Abtastüberwachung wird im Moment ausgeführt	
M9182	SM1182	SM821	Erlaubnis der Abtastüberwachung des aktiven Schritts	AUS: Abtastüberwachung nicht möglich/ausgesetzt EIN: Abtastüberwachung möglich	
M9196	SM1196	SM325	Ausgabe des Arbeitsschritts nach einem Blockstopp	AUS: Auagänge AUS EIN: Ausgänge EIN	
M9197 . M9198	SM1197 . SM1198	Keine Funktion bei einer System Q- oder QnA-CPU	Umschaltung zwischen defekter Sicherung und Vergleichsfehler E/A-Fehler	Die Anzeige ändert sich in Abhängigkeit von der Kombination der Zustände der Merker M9197 und M9198	
M9199	SM1199		Online Aufzeichnung der Sampling Trace Status Latch-Daten	AUS : Führt keine Datenaufzeichnung durch EIN: Führt die Datenaufzeichnung durch	
M9200	SM1200	—	Empfang der LRDP-Anweisung	AUS: Nicht empfangen EIN: Empfangen	QnA-CPU
M9201	SM1201	—	Verarbeitung der LRDP-Anweisung	AUS: Unvollständig EIN: Vollständig	
M9202	SM1202	—	Empfang der LWTP-Anweisung	AUS: Nicht empfangen EIN: Empfangen	
M9203	SM1203	—	Verarbeitung der LWTP-Anweisung	AUS: Unvollständig EIN: Vollständig	
M9204	SM1204	—	Verarbeitung der LRDP-Anweisung	AUS: Unvollständig EIN: Fertig	
M9205	SM1205	—	Verarbeitung der LWTP-Anweisung	AUS: Unvollständig EIN: Fertig	
M9206	SM1206	—	Fehler in den Link-Parametern der Host-Station	AUS: Normal EIN: Fehler	
M9207	SM1207	—	Übereinstimmung der Link-Parameter zwischen mehreren Master-Stationen	AUS: Normal EIN: Keine Übereinstimmung	
M9208	SM1208	—	Übertragungsbereich von B und W für die Master-Station in der unteren Ebene	AUS: Zur 2. und 3. Ebene EIN: Nur zur 2. Ebene	
M9208	SM1208	—	Übertragungsbereich von B und W für die Master-Station in der unteren Ebene	AUS: Zur 2. und3. Ebene EIN: Nur zur 2. Ebene	
M9209	SM1209	—	Überprüfung der Link-Parameter (nur für Master-Stationen in der unteren Ebene)	AUS: Überprüfung EIN: Keine Überprüfung	
M9210	SM1210	—	Fehler in der Link-Karte in der Lokalen-Station	AUS: Normal EIN: Fehler	
M9211	SM1211	—	Fehler in der Link-Karte der Master-Station	AUS: Normal EIN: Fehler	

Liste der Sondermerker und Diagnosemerker

A-CPU Sondermerker	Diagnosemerker nach der Umstellung	Äquivalente System Q-/QnA-Diagnosemerker	Name	Bedeutung	Gülig für
M9224	SM1224	—	Link-Status	AUS: Online EIN: Offline	QnA-CPU
M9225	SM1225	—	Fehler in der Vorwärtsschleife	AUS: Normal EIN: Fehler	
M9226	SM1226	—	Fehler in der Rückwärtsschleife	AUS: Normal EIN: Fehler	
M9227	SM1227	—	Teststatus der Schleife	AUS: Kein Test EIN: Test der Vor- oder Rückwärts-schleife	
M9232	SM1232	—	Betriebszustand einer lokalen Station	AUS: RUN oder STEP RUN EIN: STOP oder PAUSE	
M9233	SM1233	—	Fehlererkennung für eine lokale Station	AUS: Normal EIN: Fehler	
M9235	SM1235	—	Parameter-Fehler in einer lokalen oder Remote-E/A-Station	AUS: Normal EIN: Fehler	
M9236	SM1236	—	Initialisierungszustand einer lokalen oder Remote-E/A-Station	AUS: Keine Übertragung EIN: Datenübertragung	
M9237	SM1237	—	Fehler in einer lokalen oder Remote-E/A-Station	AUS: Normal EIN: Fehler	
M9238	SM1238	—	Fehler in der Schleife einer lokalen oder Remote-E/A-Station	AUS: Normal EIN: Fehler	
M9240	SM1240	—	Link-Status	AUS: Online EIN: Offline	
M9241	SM1241	—	Fehler in der Vorwärtsschleife	AUS: Normal EIN: Fehler	
M9242	SM1242	—	Fehler in der Rückwärtsschleife	AUS: Normal EIN: Fehler	
M9243	SM1243	—	Übertragung über Rückleitung	AUS: Keine Ausführung EIN: Ausführung	
M9246	SM1246	—	Empfangsstatus von Daten	AUS: Daten wurden empfangen EIN: Daten wurden nicht empfangen	
M9247	SM1247	—	Empfangsstatus von Daten	AUS: Daten wurden empfangen EIN: Daten wurden nicht empfangen	
M9250	SM1250	—	Empfangsstatus von Parametern	AUS: Parameter wurden empfangen EIN: Parameter nicht empfangen	
M9251	SM1251	—	Unterbrechung in der Übertragung	AUS: Normal EIN: Unterbrechung	
M9252	SM1252	—	Teststatus der Schleife	AUS: Kein Test EIN: Test der Vorwärts- oder Rückwärtsschleife	
M9253	SM1253	—	Betriebszustand der Master-Station	AUS: RUN oder STEP RUN EIN: STOP oder PAUSE	
M9254	SM1254	—	Betriebszustand einer anderen lokalen Station	AUS: RUN oder STEP RUN EIN: STOP oder PAUSE	
M9255	SM1255	—	Fehlererkennung für andere lokale Stationen	AUS: Normal EIN: Fehler	

A.4.2 Liste der Sondermerker (A-Serie)

Sondermerker sind interne Merker, die für eine Vielzahl von Anwendungsmöglichkeiten, wie Fehleranzeigen, spezielle Funktionen usw., bestimmt sind. Die nachfolgende Tabelle enthält eine Übersicht sämtlicher Sondermerker der MELSEC A-Serie mit der Beschreibung ihrer Verwendungszwecke.

Generell lassen sich zwei Arten von Sondermerkern unterscheiden:

- Sondermerker, die automatisch durch die CPU gesetzt werden und vom Anwender nur ausgeschaltet (zurückgesetzt) werden können.
- Sondermerker, die funktionsbedingt unter bestimmten Vorraussetzungen gesetzt oder rückgesetzt werden können.

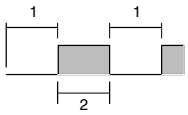
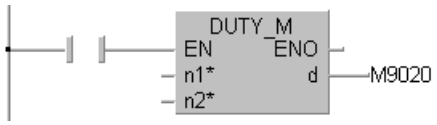
HINWEIS *Der Einsatzbereich der Sondermerker in einem Ablaufprogramm ist entsprechend zu prüfen.*

Merker, die in der Spalte „Adressen“ mit einer ❶, ❷ oder ❸ markiert sind, können nicht beliebig gesetzt bzw. zurückgesetzt werden. Die entsprechenden Hinweise hierzu befinden sich im Anschluss an diese Tabelle auf Seite 40.

Inwieweit ein Sondermerker in Verbindung mit einer bestimmten CPU einsetzbar ist, ist der folgenden Tabelle der Spalte CPU zu entnehmen.

CPU		Bedeutung
		Uneingeschränkt für alle CPU-Typen
○	(AnA- CPU)	Nicht für die angegebene(n)CPU(s)
●	A2C- CPU	Ausschließlich für die angegebene(n) CPU(s)

Liste der Sondermerker

Adresse	Bedeutung	Zustand	Beschreibung	CPU	
❶ M9000	Sicherung defekt	AUS: Normal EIN: defekt	Der Merker wird gesetzt, sobald bei einem oder mehreren Modulen die Sicherung als defekt erkannt wird. Der Merker bleibt auch nach Rückkehr in den Normalzustand gesetzt.	○	(Nicht für A2C-CPU)
❶ M9002	Vergleichsfehler E/A-Module	AUS: Normal EIN: Fehler	Der aktuelle Status des A2C-Netzwerkes ist von dem Status nach Einschalten der Versorgungsspannung verschieden. Der Merker bleibt auch nach Wegfall der Störung gesetzt. Ein Rücksetzen des Merkers ist nur nach Rücksetzen der Register D9116 bis D9123 möglich.	○	(Nicht für A2C-CPU)
M9004	Fehler im Master-Modul des MELSECNET/MINI	AUS: Normal EIN: Fehler	Der Merker wird bei einer Störung an einem der Master-Module vom Typ AJ71PT32(S3) im MELSECNET/MINI gesetzt. Der Merker bleibt auch nach Wegfall der Störung gesetzt.	●	Nur für AnA-, AnAS-, AnU-CPUs
❶ M9005	Spannungsabfall in der Netzspannung	AUS: Normal EIN: Spannungs-abfall	Der Merker wird bei einem Spannungsabfall von max. 20 ms gesetzt. Ein Rücksetzen ist nach dem Aus- und Wiedereinschalten der Spannungsversorgung möglich.		
M9006	Niedrige Batteriespannung	AUS: Normal EIN: Spannung abgefallen	Die Batteriespannung der Pufferbatterie ist unter ihren Minimalwert gesunken. Der Merker wird nach dem Austausch der Batterie zurückgesetzt.		
❶ M9007	Niedrige Batteriespannung (Latch-Merker)	AUS: Normal EIN: Spannung abgefallen	Die Batteriespannung der Pufferbatterie ist unter ihren Minimalwert gesunken. Der Merker bleibt nach dem Austausch der Batterie gesetzt.		
❶ M9008	Fehlererkennung nach Selbstdiagnose	AUS: Normal EIN: Fehler	Die CPU hat über ihre Selbstdiagnosefunktion einen Fehler erkannt.		
M9009	Fehlermelder-kennung	AUS: keine Kennung EIN: Kennung	Der Merker wird nach Ausführung einer OUT F- oder SET F-Anweisung gesetzt. Das Rücksetzen des Merkers erfolgt, sobald der Inhalt von D9124 auf „0“ gesetzt ist.		
M9010	Fehlererkennung im Programm-ablauf	AUS: Normal EIN: Fehler	Im Verlauf der Programmverarbeitung ist ein Fehler bei der Ausführung einer Applikationsanweisung aufgetreten. Nach Beseitigung des Fehlers wird der Merker zurückgesetzt.	○	(Nicht für A3H-, A3M-, AnA-, AnAS,- AnU-CPUs)
❶ M9011	Fehlererkennung im Programmablauf	AUS: Normal EIN: Fehler	Im Verlauf der Programmverarbeitung ist ein Fehler bei der Ausführung einer Applikationsanweisung aufgetreten. Nach Beseitigung des Fehlers bleibt der Merker gesetzt.		
M9012	Carry Flag (Übertrag)	AUS: Carry aus EIN: Carry ein	Merker zur Kennung des Carry Flag, das in den Applikationsanweisungen eingesetzt wird.		
M9016	Löschkennung gespeicherter Operandendaten	AUS: keine Ausführung EIN: Löschvorgang	Alle Operandendaten (mit Ausnahme von Sondermerkern und Sonderregistern) werden im Remote-RUN-Betrieb der SPS gelöscht, wenn M9016 = „1“ ist.		
M9017	Löschkennung gespeicherter Operandendaten	AUS: keine Ausführung EIN: Löschvorgang	Alle nicht zwischengespeicherten Operandendaten (mit Ausnahme von Sondermerkern und Sonderregistern) werden im Remote-RUN-Betrieb der SPS gelöscht, wenn M9017 = „1“ ist.		
M9020	Zeittakt Nr. 0	 [1] Zyklus n2 [2] Zyklus n1	Das Ein- und Ausschalten der Merker erfolgt zyklusabhängig durch eine DUTY-Anweisung (z.B. 2 Zyklen EIN, 3 Zyklen AUS, 2 EIN, 3 AUS usw.) Nach dem Einschalten oder einem RESET beginnt der Merker im Zustand „0“. 		
M9021	Zeittakt Nr. 1				
M9022	Zeittakt Nr. 2				
M9023	Zeittakt Nr. 3				
M9024	Zeittakt Nr. 4				
❷ M9025	Setzanforderung für Uhrdaten	AUS: keine Verarbeitung EIN: Anforderung	Bei gesetztem Merker werden nach Ausführung der END-Anweisung die Uhrdaten aus den Datenregistern D9025 bis D9028 gelesen und an das Uhr-Element übertragen.	●	Nur für AnN-, AnS-CPUs

Liste der Sondermerker

Adresse	Bedeutung	Zustand	Beschreibung	CPU	
M9026	Uhrdatenfehler	AUS: Normal EIN: Fehler	Der Merker wird bei fehlerhaften Uhrdaten (Register D9025 bis D9028) gesetzt.	●	Nur für AnN-, AnS-CPUs
M9027	Uhrdatenanzeige	AUS: keine Verarbeitung EIN: Anzeige	Aus den Datenregistern D9025 bis D9028 werden die Uhrdaten gelesen und mit Monat, Tag, Stunde, Minute und Sekunde über die LED-Anzeige an der CPU ausgegeben.	●	Nur für A3N-CPU, A3A-CPU
❷ M9028	Leseanforderung für Uhrdaten	AUS: keine Verarbeitung EIN: Anforderung	In die Datenregister D9025 bis D9028 werden die Uhrdaten in BCD-Form eingelesen.	●	Nur für AnN-, AnS-CPUs
M9030	Taktgeber 0,1 Sekunden	AUS: 0,05 s EIN: 0,05 s	Taktgeber für Zeittakte von 0,1 bis 60 Sekunden. Die Ein- und Ausschaltvorgänge erfolgen nicht pro Programmdurchlauf, sondern zeitabhängig während eines Zyklus entsprechend des gewählten Zeittaktes. Die Taktgeber werden unmittelbar nach dem Einschalten der Spannungsversorgung oder einem Reset gestartet.		
M9031	Taktgeber 0,2 Sekunden	AUS: 0,1 s EIN: 0,1 s			
M9032	Taktgeber 1 Sekunden	AUS: 0,5 s EIN: 0,5 s			
M9033	Taktgeber 2 Sekunden	AUS: 1 s EIN: 1 s			
M9034	Taktgeber 1 Minute	AUS: 30 s EIN: 30 s			
M9036	Scheinkontakt (Schließer)	Ständig EIN	Die Merker M9036 bis M9039 sind sogenannte Scheinkontakte (Dummys) für den Initialisierungsteil und für Applikationsanweisungen im Ablaufprogramm. Während das Setzen von M9036 und M9037 unabhänggig von der Stellung des Schlüsselschalters an der CPU erfolgt, werden M9038 und M9039 in Abhängigkeit des RUN/STOP-Status der CPU gesetzt bzw. zurückgesetzt. M9038 wird für einen Programmzyklus nach dem Umschalten in den RUN-Zustand nach dem Umschalten aus dem STOP-Zustand eingeschaltet. Mit M9039 verhält es sich umgekehrt.		
M9037	Scheinkontakt (Öffner)	Ständig AUS			
M9038	RUN-Kennung (Schließer)	EIN: für einen Zyklus nach RUN AUS: nach 1 Zyklus			
M9039	RUN-Kennung (Öffner)	EIN: nach 1 Zyklus AUS: für einen Zyklus RUN			
M9040	Pausenbedingung	AUS: PAUSE nicht möglich EIN: PAUSE möglich	Ein Umschalten in den PAUSE-Zustand der CPU (durch Schalten des Schlüsselschalters auf PAUSE oder nach Anliegen eines externen PAUSE-Signals) ist nur bei gesetztem Merker M9040 möglich. M9041 wird gesetzt, sobald sich die CPU im PAUSE-Zustand befindet.		
M9041	Kennung des PAUSE-Status	AUS: keine PAUSE EIN: während einer PAUSE			
M9042	Kennnung des STOP-Status	AUS: kein STOP EIN: bei STOP	Der Merker wird gesetzt, sobald der Schlüsselschalter an der CPU auf STOP geschaltet wird.		
M9043	Sampling Trace beendet	AUS: während eines Sampling Trace EIN: nach Ende des Sampling Trace	Das Setzen des Merkers erfolgt mit Ausführung einer STRA-Anweisung nach Beendigung des Sampling Trace. Mit der Ausführung einer STRAR-Anweisung wird der Merker zurückgesetzt.	○	(Nicht für A1-CPU, A1N- CPU)
M9044	Sampling Trace	0→1: entspricht der STRA-Anw. 1→0: entspricht der STRAR-Anw.	Der Merker entspricht in seiner Funktion der STRA- und STRAR-Anweisung (M9044 kann über ein Programmiergerät gesetzt werden). Das Setzen des Merkers entspricht der Ausführung der STRA-Anweisung und das Rücksetzen der Ausführung der STRAR-Anweisung. Der Zustand der Verarbeitung des Sampling Trace ist hier von dem Wert in D9044 abhängig („0“ für eine schrittweise Abtastung mit einem Inkrement von 10 ms).	●	Nur für A2C-CPU
M9046	Sampling Trace	AUS: Normal EIN: während des Sampling Trace	Kennung des Sampling Trace; wird bei Ausführung gesetzt.	○	(Nicht für A1-CPU, A1N-CPU)
M9047	Vorbereitung eines Sampling Trace	AUS: Sampling Trace stoppen EIN: Sampling Trace starten	Der Merker wird zum Starten des Sampling Trace gesetzt und zum Stoppen zurückgesetzt.	○	(Nicht für A1-CPU, A1N-CPU)

Liste der Sondermerker

Adresse	Bedeutung	Zustand	Beschreibung	CPU	
M9049	Anzahl der ausgegebenen Zeichen	AUS: Ausgabe bis zum Null-Code EIN: Ausgabe von 16 Zeichen	Ist M9049 nicht gesetzt, werden alle Zeichen bis zum Null-Code (00_H) ausgegeben. Bei gesetztem Merker wird ein ASCII-Code mit 16 Zeichen ausgegeben.	○	(Nicht für A-CPUs, A2C-CPU)
❷ M9050	Ausführungsbedingung der CHG-Anweisung	AUS: keine Änderung EIN: Änderung	Der Merker wird als Ausführungsbedingung der CHG-Anweisung zur Änderung des Verarbeitungsergebnisses gesetzt (näheres siehe Abs. 7.6.8)	●	Nur für A3-CPU
M9051	Unterdrückung der CHG-Anweisung	AUS: Ausführung möglich EIN: Ausführung nicht möglich	Eine CHG-Anweisung kann nur bei gesetztem Merker ausgeführt werden. Der Merker ist vor einer Programmumschaltung zu setzen. Nach Beendigung des Programms wird der Merker selbsttätig zurückgesetzt.	●	Nur für A-CPUs, A3N-CPU
❷ M9052	Umschaltung der SEG-Anweisung	AUS: 7-Segementanzeige EIN: E/A-Teilaktualisierung	Bei gesetztem Merker hat die SEG-Anweisung die Funktion zur 7-Segmentanzeige gegebener Daten. Ist der Merker nicht gesetzt, erfolgt bei Ausführung der SEG-Anweisung eine Teilaktualisierung der in der Anweisung definierten E/As (näheres siehe Abs. 6.7.2 und 7.5.5).	○	(Nicht für A-CPUs, A2C-CPU)
❷ M9053	Umschaltung der EI-/DI-Anweisung	AUS: Bedingung für Link Refresh EIN: Bedingung für Interrupt-Progr.	Der Merker ändert die Funktion der EI-/DI-Anweisung. Bei gesetztem Merker dient die Anweisung als Ausführungsbedingung für ein Interrupt-Programm. Ist der Merker nicht gesetzt, dient die EI-/DI-Anweisung als Ausführungsbedingung für eine Netzwerkaktualisierung (näheres siehe Abs. 6.6.1 und 7.5.5).	●	Nur für AnS-CPUs, A2C-CPU, AnN-CPUs
M9054	Kennung für STEP RUN	AUS: andere Betriebsart EIN: STEP RUN	Der Merker wird nach dem Umschalten des Schlüsselschalters auf STEP RUN gesetzt.	○	(Nicht für A2C-CPU)
M9055	Kennung des Status Latch	AUS: nicht beendet EIN: beendet	Der Merker wird gesetzt, sobald ein Status Latch beendet ist. Das Rücksetzten erfolgt durch eine Rücksetzanweisung.	○	(Nicht für A1-CPU, A1N-CPU)
M9056	Anforderung von P, I für das Hauptprogramm	AUS: keine Anforderung EIN: Anforderung von P, I	Anforderung für die Pointer P, I nach einer Programmumschaltung (z.B. für ein Unterprogramm während der Hauptprogrammverarbeitung). Die Merker werden selbsttätig zurückgesetzt, sobald die Anforderung der Pointer abgeschlossen ist.	●	Nur für A3-, A3N-, A3H-, A3M-, A3A-CPUs
M9057	Anforderung von P, I für die Unterprogramme	AUS: keine Anforderung EIN: Anforderung von P, I			
M9060	Fehler in einem Remote-Sondermodul	AUS: Normal EIN: Fehler	Der Merker wird gesetzt, sobald ein fehlerhaftes Remote-Sondermodul erkannt wird (ein Kommunikationsfehler liegt vor, wenn eine Kommunikation auch nach einer Anzahl von Wiederholungsversuchen, die mit D9174 vorgegeben wird, nicht möglich ist). Nach Beseitigung der Fehlerursache und automatischer Wiederangliederung der fehlerhaften Station ins Netzwerk wird der Merker zurückgesetzt. Ohne automatische Wiederangliederung bleibt der Merker gesetzt. Eine Unterbrechung der Kommunikation bei Erkennung eines Fehlers führt nicht zum Setzen/Rücksetzen des Merkers.	●	Nur für A2C-CPU
M9061	Kommunikationsfehler	AUS: Normal EIN: Fehler	Der Merker wird gesetzt, wenn ein Fehler in der Kommunikation mit einem Remote-Modul auftritt. Die Ursache für einen Kommunikationsfehler kann ein Fehler in den Initialisierungsdaten, in der Verkabelung oder ein ausgeschaltetes Remote-Modul sein. Nach Beseitigung der Fehlerursache und automatischer Wiederangliederung der fehlerhaften Station ins Netzwerk wird der Merker zurückgesetzt. Ohne automatische Wiederangliederung bleibt der Merker gesetzt	●	Nur für A2C-CPU
M9065	Kennung der schrittweisen Übertragung	AUS: Andere Verarb. EIN: Schrittweise Übertragung	Der Merker ist während der schrittweisen Übertragung der Bildschirmmasken zum AD57(S1)/AD58 gesetzt. Nach Übertragungsende wird der Merker zurückgesetzt.	●	Nur für AnA-, AnAS-, AnU-CPUs
❷ M9066	Umschaltung der Transferverarbeitung	AUS: Stapel-Transfer EIN: Schrittweise Übertragung	Ist M9066 gesetzt erfolgt die Übertragung der Bildschirmmasken zum AD57(S1)/AD58 schrittweise. Ist der Merker nicht gesetzt, erfolgt die Übertragung stapelweise.	●	Nur für AnA-, AnAS-, AnU-CPUs

Liste der Sondermerker

Adresse	Bedeutung	Zustand	Beschreibung	CPU	
M9067	Fehler in einem E/A-Modul	AUS: Normal EIN: Fehler	Der Merker wird gesetzt, sobald ein fehlerhaftes E/A-Modul erkannt wird (ein Kommunikationsfehler liegt vor, wenn eine Kommunikation auch nach einer Anzahl von Wiederholungsversuchen, die mit D9174 vorgegeben wird, nicht möglich ist). Nach Beseitigung der Fehlerursache und automatischer Wiederangliederung der fehlerhaften Station ins Netzwerk wird der Merker zurückgesetzt. Ohne automatische Wiederangliederung bleibt der Merker gesetzt. Eine Unterbrechung der Kommunikation bei Erkennung eines Fehlers führt nicht zum Setzen/Rücksetzen des Merkers.	●	Nur für A2C-CPU
M9068	Testbetrieb	AUS: Normalbetrieb EIN: Prüfung der Verbindungen	Der Merker ist gesetzt, wenn im Netzwerk ein Prüfung der Verbindungen zu den einzelnen E/A-Modulen und Remote-Sondermodulen durchgeführt wird. Bei normaler Netzwerkskommunikation ist der Merker nicht gesetzt.	●	Nur für A2C-CPU
M9069	Verarbeitung nach einem Kommunikationsfehler	AUS: alle Ausgänge werden abgeschaltet EIN: normale Ausgabe	Mit diesem Merker kann die Verarbeitung der Ausgangssignale während eines Kommunikationsfehlers bestimmt werden. Ist der Merker nicht gesetzt, werden sämtliche Ausgänge bei Erkennung eines Kommunikationsfehlers ausgeschaltet. Ist der Merker dagegen gesetzt, wird der Istzustand der Ausgänge zum Zeitpunkt des Fehlers beibehalten.	●	Nur für A2C-CPU
M9081	Kommunikations-anforderung an ein Remote-Sondermodul	AUS: Anforderung ist möglich EIN: Anforderung ist nicht möglich	Der Merker zeigt an, inwieweit eine Kommunikationsanforderung an ein Remote-Sondermodul (verbunden mit einem Computer-Link-Modul oder einer A2C-CPU) möglich ist.	●	Nur für A2C-CPU, AnA-CPUs, AnU-CPUs
M9082	Überreinstimmung der gesetzten Stationsnummern mit den tatsächlichen Gegebenheiten	AUS: Übereinstimmung ist gegeben EIN: keine Übereinstimmung	Der Merker wird gesetzt, wenn die letzte Stationsnummer eines im Netzwerk befindlichen Remote-Moduls nicht mit der in der Initialisierung vorgegebenen Anzahl von Stationen übereinstimmt. Ein Rücksetzten des Merkers erfolgt nach dem Umschalten von STOP auf RUN, wenn die in der Initialisierung vorgegebene Anzahl von Stationen mit der tatsächlich im Netzwerk befindlichen Anzahl von Stationen übereinstimmt.	●	Nur für A2C-CPU
M9084	Fehlerkontrolle	AUS: Fehlerkontrolle gegeben EIN: Keine Fehlerkontrolle	Nach Verarbeitung der END-Anweisung prüft die CPU alle Sicherungen auf Defekte, E/A-Module auf Vergleichsfehler sowie die Pufferbatterie auf korrekte Spannung. Im Fehler- oder Störungsfall wird eine Fehlermeldung ausgegeben. Mit M9084 kann die Fehlerkontrolle zur Verkürzung der END-Verarbeitungszeit ausgeschaltet werden.	○	(Nicht für A2C-CPU, AnA-, AnAS-, AnU-CPUs)
M9086	RUN-Kennung für BASIC-Progr.	AUS: BASIC-Progr. wird nicht verarbeitet EIN: BASIC-Progr. wird verarbeitet	Der Merker ist während der Programmverarbeitung des A3M-BASIC-Programms gesetzt und bei Verarbeitungsstopp zurückgesetzt.	●	Nur für A3M-CPU
M9087	PAUSE-Kennung für BASIC-Progr.	AUS: Programmverarbeitung möglich EIN: Programmverarbeitung nicht möglich	Der Merker legt fest, ob eine Weiterverarbeitung des A3M-BASIC-Programms möglich ist, wenn sich die CPU im PAUSE-Zustand befindet. Ist der Merker nicht gesetzt wird das BASIC-Programm im PAUSE-Zustand weiterverarbeitet. Ist der Merker gesetzt, wird die Programmverarbeitung des BASIC-Programms ebenfalls unterbrochen.	●	Nur für A3M-CPU
M9089	Ausgabe an ERR-Klemme	AUS: kein Signal an ERR-Ausgang EIN: Signal an ERR-Ausgang	Der Merker wird gesetzt, wenn über das Ablaufprogramm eine Ausgabe über die ERR-Klemmen erfolgt ist. Ein Rücksetzten des Merkers ist nur möglich, wenn M9089 und M9090 gleichzeitig rückgesetzt werden.	●	Nur für A2C-CPU
M9090	Ausgabe an ERR-Klemme	AUS: kein Signal an ERR-Ausgang EIN: Signal an ERR-Ausgang	Der Merker wird gesetzt, wenn ein Fehler im MELSECNET/MINI oder im Ablaufprogramm auftritt (bei einem Stopp der Verarbeitung). Das Rücksetzten erfolgt, sobald der Fehler im Netzwerk behoben bzw. das Ablaufprogramm wiederhergestellt ist.	●	Nur für A2C-CPU
❶ M9091	Fehlerkennung für detaillierten Fehler im Prozessablauf	AUS: kein Fehler EIN: Fehler	Nach dem Auftreten eines Fehlers im Prozessablauf wird in einigen Fällen ein detaillierter Fehlercode in das Datenregister D9091 geschrieben und M9091 gesetzt. Der Merker bleibt auch nach Behebung der Fehlerursache gesetzt.	●	Nur für AnA-CPUs, AnU-CPUs
	Fehlerkennung für Unterprogrammaufruf aus dem Mikrocomputerbereich	AUS: kein Fehler EIN: Fehler	M9091 wird gesetzt, wenn während der Verarbeitung eines Mikrocomputerprogramms ein Fehler auftritt. Der Merker bleibt auch nach Behebung der Fehlerursache gesetzt.	○	(Nicht für AnS-CPUs, AnA-CPUs, AnU-CPUs)

Liste der Sondermerker

Adresse	Bedeutung	Zustand	Beschreibung	CPU	
❷❸ M9094	Änderungskennung der E/A-Module	AUS: Änderung EIN: keine Änderung	Nach der Festlegung einer Kopfadresse für ein bestimmtes E/A-Modul in Register D9094 kann dieses Modul durch Setzen von M9094 Online geschaltet werden (nur ein Modul pro Ausführung). Eine Änderung des Betriebszustands eines E/A-Moduls ist mit Hilfe von M9094 auch während des Testbetriebs über ein Programmiergerät oder im STOP-Zustand der CPU möglich. Der Betriebszustand der CPU darf bis zur Vollendung der Online-Schaltung des E/A-Moduls nicht verändert werden.	○	(Nicht für A-CPUs, AnS-CPUs, A3H-CPU)

HINWEIS *Nach dem Ausschalten der Versorgungsspannung, einem Löschen des Zwischenspeichers oder einem RESET werden sämtliche Sondermerker zurückgesetzt.*

Ein Umschalten des Schlüsselschalters auf die STOP-Position bewirkt kein Rücksetzen der Merker. Die Ist-Zustände werden beibehalten.

Die mit ❶ gekennzeichneten Sondermerker bleiben auch dann gesetzt, wenn der Normalzustand nach Beseitigung der Fehlerursache wiederhergestellt ist. Ein Rücksetzen dieser Sondermerker ist wie folgt möglich:

- Einfügen einer Programmzeile in das Ablaufprogramm, die den Sondermerker mittels RST-Anweisung bei einer bestimmten Eingangsbedingung zurücksetzt.
- Rücksetzen über ein Programmiergerät.
- Rücksetzen der CPU durch Umschalten des Schlüsselschalters an der CPU auf RESET.

Die mit ❷ gekennzeichneten Sondermerker werden ausschließlich über das Ablaufprogramm gesetzt und rückgesetzt.

Die mit ❸ gekennzeichneten Sondermerker werden im Testbetrieb eines Programmiergerätes gesetzt und rückgesetzt.

A.4.3 Übersicht der Sondermerker im Link-Betrieb (Nur A-Serie)

Diese Sondermerker (im Link-Betrieb) werden durch unterschiedliche Faktoren während der Datenkommunikation in einem Netzwerk gesetzt bzw. rückgesetzt. Ihr Zustand ändert sich nach dem Auftreten eines Fehlers in der Programmverarbeitung.

Die Verarbeitung der Sondermerker im Link-Betrieb ist davon abhängig, ob sich die CPU in einer Master- oder einer lokalen Station befindet.

Sondermerker im Link-Betrieb in der Master-Station

Adresse	Name	Bedeutung	Beschreibung
M9200	Empfang der LRDP-Anweisung	AUS: Nicht empfangen EIN: Empfangen	Der Merker wird in Abhängigkeit des Empfangs der LRDP-Anweisung (Wortoperanden lesen) gesetzt und kann im Programmverlauf als Verriegelung dieser Anweisung programmiert werden. Das Rücksetzen des Merkers muss mit einer RST-Anweisung erfolgen.
M9201	Verarbeitung der LRDP-Anweisung	AUS: Unvollständig EIN: Vollständig	Der Merker wird in Abhängigkeit des Verarbeitungszustands der LRDP-Anweisung (Wortoperanden lesen) gesetzt und kann im Programmverlauf als Eingangsbedingung zum Rücksetzen der Sondermerker M9200 und M9201 programmiert werden. Das Rücksetzen des Merkers muss mit einer RST-Anweisung erfolgen.
M9202	Empfang der LWTP-Anweisung	AUS: Nicht empfangen EIN: Empfangen	Der Merker wird in Abhängigkeit des Empfangs der LWTP-Anweisung (Wortoperanden schreiben) gesetzt und kann im Programmverlauf als Verriegelung dieser Anweisung programmiert werden. Das Rücksetzen des Merkers muss mit einer RST-Anweisung erfolgen.
M9203	Verarbeitung der LWTP-Anweisung	AUS: Unvollständig EIN: Vollständig	Der Merker wird in Abhängigkeit des Verarbeitungszustands der LWTP-Anweisung (Wortoperanden schreiben) gesetzt und kann im Programmverlauf als Eingangsbedingung zum Rücksetzen der Sondermerker M9202 und M9203 programmiert werden. Das Rücksetzen des Merkers muss mit einer RST-Anweisung erfolgen.
M9206	Fehler in den Link-Parametern der Host-Station	AUS: Normal EIN: Fehler	Abhängig von der Einstellung der Link-Parameter in der Host-Station.
M9207	Übereinstimmung der Link-Parameter zwischen mehreren Master-Stationen	AUS: Normal EIN: Keine Übereinstimmung	Abhängig von der Übereinstimmung der Link-Parameter (Operanden B und W) aus der Master-Station von Ebene 2 mit der Master-Station von Ebene 3 (nur bei einem Netzwerk mit 3 Ebenen).
M9208	Übertragungsbereich von B und W für die Master-Station in der unteren Ebene	AUS: Zur 2. und 3.Ebene EIN: Nur zur 2. Ebene	Der Merker legt fest, ob die Link-Daten der Operanden B und W von der Master-Station in der 1. Ebene in die Stationen der unteren Ebenen (Unterstationen) übertragen werden sollen. Eine Übertragung erfolgt nur dann zu den Unterstationen, wenn M9208 nicht gesetzt ist.
M9209	Überprüfung der Link-Parameter (nur für Master-Station in der unteren Ebene)	AUS: Überprüfung EIN: Keine Überprüfung	Der Sondermerker wird gesetzt, wenn die Link-Operanden (B und W) aus der oberen Ebene nicht mit den Link-Operanden (B und W) aus der unteren Ebene auf Übereinstimmung verglichen werden sollen. Ist M9209 nicht gesetzt, werden die Link-Parameter der oberen und unteren Ebene ständig überprüft.
M9210	Fehler in der Link-Karte der Master-Station	AUS: Normal EIN: Fehler	Abhängig von einem Hardware-Fehler oder dem Fehlen der MELSECNET(/B)-Link-Karte. Die Verarbeitung erfolgt durch die CPU.
M9224	Link-Status	AUS: Online EIN: Offline	Der Merker ist im Offline-Betrieb und bei Durchführung eines Station-zu-Station-Tests oder im Prüfschleifentest gesetzt.
M9225*	Fehler in der Vorwärtsschleife	AUS: Normal EIN: Fehler	Abhängig von einem Fehlerzustand in der Vorwärtsschleife.
M9226*	Fehler in der Rückwärtsschleife	AUS: Normal EIN: Fehler	Abhängig von einem Fehlerzustand in der Rückwärtsschleife.
M9227*	Teststatus der Schleife	AUS: Test der Vor- oder Rückwärtsschleife EIN: Kein Test	Abhängig von der Durchführung eines Schleifentests der Vor- oder Rückwärtsschleife.
M9232	Betriebszustand einer lokalen Station	AUS: RUN oder STEP-RUN EIN: STOP oder PAUSE	Abhängig von dem Betriebszustand (RUN, STOP oder PAUSE) einer lokalen Station.

Sondermerker im Link-Betrieb in der Master-Station

Adresse	Name	Bedeutung	Beschreibung
M9233	Fehlererkennung für eine lokale Station	AUS: Normal EIN: Fehler	Abhängig von der Fehlererkennung einer lokalen Station in einer anderen Station.
M9235	Parameter-Fehler in einer lokalen oder Remote-EA-Station	AUS: Normal EIN: Fehler	Abhängig von der Erkennung eines Fehlers in den Link-Parametern einer lokalen oder Remote-EA-Station.
M9236	Initialisierungs-zustand einer lokalen oder Remote-EA-Station	AUS: Keine Übertragung EIN: Datenübertragung	Abhängig von der Übertragung der Initialisierungsdaten (wie z.B. Parameter) zwischen Master-Station und einer lokalen oder Remote-EA-Station.
M9237	Fehler in einer lokalen oder Remote-EA-Station	AUS: Normal EIN: Fehler	Abhängig von dem Fehlerzustand einer lokalen oder Remote-EA-Station.
M9238*	Fehler in der Schleife einer lokalen oder Remote-EA-Station	AUS: Normal EIN: Fehler	Abhängig von dem Fehlerzustand in einer Vorwärts- oder Rückwärtsschleife einer lokalen oder Remote-EA-Station.

* Die gekennzeichneten Sondermerker sind nicht im MELSECNET/B einsetzbar.

Sondermerker im Link-Betrieb in einer lokalen Station

Adresse	Bedeutung	Bedeutung	Beschreibung
M9204	Verarbeitung der LRDP-Anweisung	AUS: Unvollständig EIN: Vollständig	Der Merker wird in Abhängigkeit des Verarbeitungszustands der LRDP-Anweisung in der lokalen Station gesetzt.
M9205	Verarbeitung der LWTP-Anweisung	AUS: Unvollständig EIN: Vollständig	Der Merker wird in Abhängigkeit des Verarbeitungszustands der LWTP-Anweisung in der lokalen Station gesetzt.
M9211	Fehler in der Link-Karte der lokalen Station	AUS: Normal EIN: Fehler	Abhängig von einem Hardware-Fehler oder dem Fehlen der MELSECNET(/B) Link-Karte. Die Verarbeitung erfolgt durch die CPU.
M9240*	Link-Status	AUS: Online EIN: Offline	Der Merker ist im Offline-Betrieb und bei Durchführung eines Station-zu-Station-Tests oder im Prüfschleifentest gesetzt.
M9241*	Fehler in der Vorwärts-schleife	AUS: Normal EIN: Fehler	Abhängig von einem Fehlerzustand in der Vorwärtsschleife.
M9242*	Fehler in der Rückwärts-schleife	AUS: Normal EIN: Fehler	Abhängig von einem Fehlerzustand in der Rückwärtsschleife.
M9243	Übertragung über Rück-leitung	AUS: Keine Ausführung EIN: Ausführung	Der Sondermerker ist gesetzt, wenn die Station selbst eine Datenübertragung über die Rückleitung ausführt.
M9246	Empfangsstatus von Daten	AUS: Daten wurden empfangen EIN: Daten nicht empfangen	Abhängig von der Datenübertragung zwischen der Master-Station und der lokalen Station.
M9247	Empfangsstatus von Daten	AUS: Daten wurden empfangen EIN: Daten nicht empfangen	Abhängig von der Datenübertragung zwischen der Master-Station einer bestimmten Ebene und der lokalen Station.
M9250	Empfangsstatus von Parametern	AUS: Parameter wurden empfangen EIN: Parameter nicht empfangen	Abhängig von dem Übertragungszustand der Parameterdaten aus der Master-Station.
M9251	Unterbrechung in der Übertragung	AUS: Normal EIN: Unterbrechung	Abhängig von dem Übertragungszustand an einer lokalen Station.
M9252	Teststatus der Schleife	AUS: Kein Test EIN: Test der Vor- oder Rückwärtsschleife	Abhängig von der Durchführung eines Schleifentests der Vor- oder Rückwärtsschleife.
M9253	Betriebszustand der Master-Station	AUS: RUN oder STEP RUN EIN: STOP oder Pause	Abhängig von dem Betriebszustand (RUN, STOP oder Pause) der Master-Station.
M9254	Betriebszustand einer anderen lokalen Station	AUS: RUN oder STEP RUN EIN: STOP oder Pause	Abhängig von dem Betriebszustand (RUN, STOP oder Pause) einer lokalen Station.
M9255	Fehlererkennung für andere lokale Stationen	AUS: Normal EIN: Fehler	Abhängig von der Fehlererkennung einer lokalen Station in einer anderen Station.

* Die gekennzeichneten Sondermerker sind nicht im MELSEC NET/B einsetzbar.

A.5 Übersicht der Sonderregister

A.5.1 Übersicht der Diagnoseregister (MELSEC Q-Serie und System Q)

Die Diagnoseregister SD sind interne Register mit einer festgelegten Aufgabe innerhalb der SPS.

Aus diesem Grund ist es nicht möglich, diese Register in der gleichen Weise von Ablaufprogrammen nutzen zu lassen wie normale Register.
Zur Steuerung der CPU ist es jedoch möglich, Daten in diese Register zu schreiben.
Die in den Diagnoseregistern gespeicherten Daten werden im binären Format abgespeichert, es sei denn, es wird ein anderes Format gefordert.

In dieser Tabelle werden die Einträge für die Tabellen auf den folgenden Seiten erläutert.

Tabellenüberschrift	Bedeutung
Adresse	● Zeigt die Adresse des Diagnoseregisters.
Name	● Zeigt den Namen des Diagnoseregisters.
Bedeutung	● Kurzerläuterung der Bedeutung des Diagnoseregisters.
Beschreibung	● Beinhaltet detaillierte Informationen zur Bedeutung des Diagnoseregisters.
Gesetzt vom (wenn gesetzt)	Gibt Aufschluss darüber, ob das Diagnosemerker vom System oder vom Benutzer gesetzt wurde. <Gesetzt vom> S : Durch das System gesetzt B : Durch den Benutzer gesetzt (im Ablaufprogramm oder Prüfmodus eines Peripheriegerätes) S/B : Durch das System und den Benutzer gesetzt Wird nur gezeigt, wenn die Einstellung durch das System vorgenommen wurde. <Wenn gesetzt> END-Verarbeitung : Wird während jeder END-Verarbeitung gesetzt Initialisierung : Wird nur während der Initialisierung gesetzt (bei Einschalten des Netzteils oder beim Umschalten der CPU vom STOP-Modus in den RUN-Modus) Zustandsänderung : Wird nur nach Auftreten einer Zustandsänderung gesetzt Fehler : Wird nur nach Auftreten eines Fehlers gesetzt Anweisungsausführung : Wird gesetzt, wenn die Anweisung ausgeführt wird Anforderung : Wird nur gesetzt, wenn eine Benutzeranforderung ansteht (durch SM, etc.)
Korrespondierende A-CPU Register D9 [ ] [ ] [ ]	● Zeigt Sonderregister D9 [ ] [ ] [ ] korresspondierend zur A-CPU. (Änderung und Schreibweise, wenn es inhaltliche Änderungen gab.) ● Wird mit „Neu“ gekennzeichnet, wenn er der Q-/QnA-CPU neu hinzugefügt wurde.
Gültig für:	● Gibt an, für welche CPU dieser Sondermerker zur Verfügung steht. ●: Gilt für alle CPU-Typen Q-CPU: Gilt nur für die CPUs des MELSEC System Q QnA: Gilt für die CPUs der MELSEC QnA-Serie und die Q2AS-CPU CPU-Typ: Gilt nur für diese CPU (z. B. Q4AR-CPU) Rem: Gültig für dezentrale MELSECNET/H E/A-Module

HINWEISE Detaillierte Informationen zu dem folgenden Thema finden Sie den Handbüchern:

- Networks → Melsecnet/10/10H/25H Network System Reference Manual for QnA
- SFC → Q-/QnA-CPU Programming Manual (SFC)

Allgemeine Fehlerinformationen

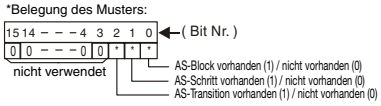
Die Sonderregister SD1200 bis SD1255 werden bei einer QnA-CPU verwendet. Bei einer CPU des MELSEC System Q sind diese Register nicht belegt.
Die Sonderregister ab SD 1500 sind für die Q4AR-CPU reserviert.

Liste der Diagnoseregister

(1) Informationen zur Diagnose

Adresse	Name	Bedeutung	Beschreibung	Gesetzt von (wenn gesetzt)	A-CPU-Register D9[][][]	Gültig für:
SD0	Diagnosefehler	Diagnose-Fehlercode	● Der Fehlercode der von der Diagnosefunktion entdeckten Fehler wird im binären Format gespeichert. ● Die Inhalte sind identisch mit den letzten Fehlerereignis-informationen.	S (Fehler)	D9008 Geändertes Format	● Rem
SD1	Uhrzeit des Auftretens eines Diagnosefehlers	Uhrzeit des Auftretens eines Diagnosefehlers	● Jahr (die letzten zwei Stellen) und Monat, in dem die Daten von SD0 aktualisiert wurden. Die Daten werden im zweistelligen BCD-Code abgespeichert. Beispiel: **Oktober 1995 = 9510** b15 b8 b7 b0 Jahr (0 bis 99) | Monat (1 bis 12)	S (Fehler)	Neu	
SD2			● Tag und Stunde, an dem die Daten von SD0 aktualisiert wurden. Die Daten werden im zweistelligen BCD-Code abgespeichert. Beispiel: **25. 22 Uhr = 2522** b15 b8 b7 b0 Tag (1 bis 31) | Stunde (0 bis 23)			
SD3			● Minute und Sekunde, in der die Daten von SD0 aktualisiert wurden. Die Daten werden im zweistelligen BCD-Code abgespeichert. Beispiel: **35 min 48s = 3548** b15 b8 b7 b0 Minute (0 bis 59) | Sekunde (0 bis 59)			
SD4	Kategorien der Fehler-informationen	Kategorie-Codes der Fehlerinformationen	Mit Hilfe der Kategorie-Codes ist die Auswertung möglich, welche Art der Information in dem Bereich der allgemeinen Fehlerinformation (SD5 - SD15) und dem Bereich der spezifischen Fehlerinformation (SD16 - SD26) gespeichert sind b15 b8 b7 b0 Spezifische Fehlerinformationen | Allgemeine Fehlerinformationen ● Die Kategorie-Codes der allgemeinen Fehlerinformation werden wie folgt gespeichert: 0: Kein Fehler 1: Stations-/Modul-/CPU-/Baugruppenträger-Nummer 2: File-Name/ Laufwerksname 3: Zeit (eingestellter Wert) 4: Lokalisierung des Programmfehlers 5: Grund der Umschaltung (nur bei einer Q4ARCPU) ● Die Kategorie-Codes der spezifischen Fehlerinformation werden wie folgt gespeichert: 0: Kein Fehler 1: (Offen) 2: File-Name/ Laufwerksname 3: Zeit (tatsächlich gemessener Wert) 4: Lokalisierung des Programmfehlers 5: Nummer des Parameters 6: Nummer des Fehlermerkers 7: Nummer der Funktionsstörung der Prüfanweisung	S (Fehler)	Neu	

Liste der Diagnoseregister (Fortsetzung)

<table>
<tr><th>Adresse</th><th>Name</th><th>Bedeutung</th><th>Beschreibung</th><th>Gesetzt von (wenn gesetzt)</th><th>A-CPU-Register D9 [ ] [ ] [ ]</th><th>Gültig für:</th></tr>
<tr><td>SD5
SD6
SD7
SD8
SD9
SD10
SD11
SD12
SD13
SD14
SD15</td><td colspan="2">Allgemeine Fehlerinformation</td><td>
● Die zu den Fehlercodes (SD0) korrespondierenden allgemeinen Informationen werden hier gespeichert.
● Die folgenden 5 Arten von Informationen werden gespeichert:
(1) Stations-/Modulnummer
<table>
<tr><th>Nummer</th><th>Bedeutung</th></tr>
<tr><td>SD5</td><td>Stations-/Modulnummer</td></tr>
<tr><td>SD6</td><td>E/A-Nummer</td></tr>
<tr><td>SD7
SD8
SD9
SD10
SD11
SD12
SD13
SD14
SD15</td><td>Frei</td></tr>
</table>
(2) File-Name/Laufwerksname
Beispiel: File-Name = ABCDEFGH.IJK
<table>
<tr><th>Nummer</th><th>Bedeutung</th></tr>
<tr><td>SD5</td><td>Laufwerk</td></tr>
<tr><td>SD6
SD7
SD8
SD9</td><td>File-Name
(ASCII-Code: 8 Zeichen)</td></tr>
<tr><td>SD10</td><td>Erweiterung 2E_H(.)</td></tr>
<tr><td>SD11</td><td>(ASCII-Code: 3 Zeichen)</td></tr>
<tr><td>SD12
SD13
SD14
SD15</td><td>Frei</td></tr>
</table>
b15 b0
<table>
<tr><td>B</td><td>A</td></tr>
<tr><td>D</td><td>C</td></tr>
<tr><td>F</td><td>E</td></tr>
<tr><td>H</td><td>G</td></tr>
<tr><td>I</td><td>.</td></tr>
<tr><td>K</td><td>J</td></tr>
</table>
(3) Zeit (eingestellter Wert)
<table>
<tr><th>Nummer</th><th>Bedeutung</th></tr>
<tr><td>SD5</td><td>Zeit: 1 µs-Schritte (0 – 999 µs)</td></tr>
<tr><td>SD6</td><td>Zeit: 1 ms-Schritte (0 – 65535 ms)</td></tr>
<tr><td>SD7
SD8
SD9
SD10
SD11
SD12
SD13
SD14
SD15</td><td>Frei</td></tr>
</table>
(4) Lokalisierung des Programmfehlers
<table>
<tr><th>Nummer</th><th>Bedeutung</th></tr>
<tr><td>SD5
SD6
SD7
SD8</td><td>File-Name
(ASCII-Code: 8 Zeichen)</td></tr>
<tr><td>SD9</td><td>Erweiterung 2E_H (.)</td></tr>
<tr><td>SD10</td><td>(ASCII-Code: 3 Zeichen)</td></tr>
<tr><td>SD11</td><td>Muster*</td></tr>
<tr><td>SD12</td><td>Block-Nr.</td></tr>
<tr><td>SD13</td><td>Schritt-/Transitions-Nr.</td></tr>
<tr><td>SD14</td><td>Ablaufschritt-Nr. (L)</td></tr>
<tr><td>SD15</td><td>Ablaufschritt-Nr. (H)</td></tr>
</table>
*Belegung des Musters:


</td><td>S
(Fehler)</td><td>Neu</td><td>●</td></tr>
</table>

Liste der Diagnoseregister (Fortsetzung)

Adresse	Name	Bedeutung	Beschreibung	Gesetzt von (wenn gesetzt)	A-CPU-Register D9 [ ] [ ] [ ]	Gültig für:

Bedeutung der Erweiterungen der File-Namen:

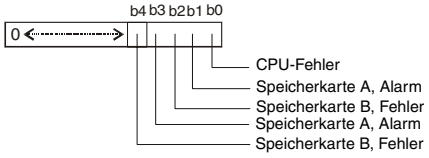
SD10 (SD9)	SD11 (SD10)		Erweiterung	Datei-Typ
Höherwertiges Byte	Niederwertiges Byte	Höherwertiges Byte		
51H	50H	41H	QPA	Parameter
51H	50H	47H	QPG	Programme
51H	43H	44H	QCD	Operandenkommentare
51H	44H	49H	QDI	Anfangswerte von Operanden
51H	44H	52H	QDR	File-Register
51H	44H	53H	QDS	Simulationsdaten
51H	44H	4CH	QDL	Lokale Operanden
51H	54H	53H	QTS	Daten von Sampling Trace (Nur QnA-CPU)
51H	54H	4CH	QTL	Status-Latch-Daten (Nur QnA-CPU)
51H	54H	50H	QTP	Daten von Programm Trace (QnA-CPU)
51H	54H	52H	QTR	Trace-File für AS-Programme
51H	46H	44H	QFD	Fehlerdaten

Liste der Diagnoseregister (Fortsetzung)

<table>
<tr><th>Adresse</th><th>Name</th><th>Bedeutung</th><th>Beschreibung</th><th>Gesetzt von (wenn gesetzt)</th><th>A-CPU-Register D9 [ ] [ ] [ ]</th><th>Gültig für:</th></tr>
<tr><td>SD16
SD17
SD18
SD19
SD20
SD21
SD22
SD23
SD24
SD25
SD26</td><td colspan="2">Allgemeine Fehlerinformation</td><td>
● Die zu den Fehlercodes (SD0) korrespondierenden allgemeinen Informationen werden hier gespeichert.
● Die folgenden 6 Arten von Informationen werden hier gespeichert:

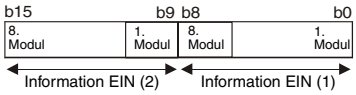
(1) File-Name/Laufwerksname
Beispiel:)
File-Name =
ABCDEFGH.IJK
<table>
<tr><th>Nummer</th><th>Bedeutung</th></tr>
<tr><td>SD16</td><td>Laufwerk</td></tr>
<tr><td>SD17
SD18
SD19
SD20</td><td>File-Name
(ASCII-Code: 8 Zeichen)</td></tr>
<tr><td>SD21</td><td>Erweiterung 2E_H(.)</td></tr>
<tr><td>SD22</td><td>(ASCII-Code: 3 Zeichen)</td></tr>
<tr><td>SD23
SD24
SD25
SD26</td><td>Frei</td></tr>
</table>
b15 b0
<table>
<tr><td>B</td><td>A</td></tr>
<tr><td>D</td><td>C</td></tr>
<tr><td>F</td><td>E</td></tr>
<tr><td>H</td><td>G</td></tr>
<tr><td>I</td><td>.</td></tr>
<tr><td>K</td><td>J</td></tr>
</table>
(2) Zeit (eingestellter Wert)
<table>
<tr><th>Nummer</th><th>Bedeutung</th></tr>
<tr><td>SD16</td><td>Zeit: 1 µs-Schritte (0 – 999 µs)</td></tr>
<tr><td>SD17</td><td>Zeit: 1 ms-Schritte (0 – 65535 ms)</td></tr>
<tr><td>SD18
SD19
SD20
SD21
SD22
SD23
SD24
SD25
SD26</td><td>Frei</td></tr>
</table>
(3) Lokalisierung des Programmfehlers
<table>
<tr><th>Nummer</th><th>Bedeutung</th></tr>
<tr><td>SD16
SD17
SD18
SD19</td><td>File-Name
(ASCII-Code: 8 Zeichen)</td></tr>
<tr><td>SD20</td><td>Erweiterung 2E_H (.)</td></tr>
<tr><td>SD21</td><td>(ASCII-Code: 3 Zeichen)</td></tr>
<tr><td>SD22</td><td>Muster*</td></tr>
<tr><td>SD23</td><td>Block-Nr.</td></tr>
<tr><td>SD24</td><td>Schritt-/Transitions-Nr.</td></tr>
<tr><td>SD25</td><td>Ablaufschritt-Nr. (L)</td></tr>
<tr><td>SD26</td><td>Ablaufschritt-Nr. (H)</td></tr>
</table>
*Belegung des Musters:
15 14 – – – 4 3 2 1 0 ← (Bit Nr.)
0 0 – – – 0 0 * * *
nicht verwendet
AS-Block vorhanden (1) / nicht vorhanden (0)
AS-Schritt vorhanden (1) / nicht vorhanden (0)
AS-Transition vorhanden (1) / nicht vorhanden (0)
</td><td>S
(Fehler)</td><td>Neu</td><td>●</td></tr>
</table>

Liste der Diagnoseregister (Fortsetzung)

<table>
<tr><th>Adresse</th><th>Name</th><th>Bedeutung</th><th>Beschreibung</th><th>Gesetzt von (wenn gesetzt)</th><th>A-CPU-Register D9 [ ] [ ] [ ]</th><th>Gültig für:</th></tr>
<tr><td>SD16
SD17
SD18
SD19
SD20
SD21
SD22
SD23
SD24
SD25
SD26</td><td colspan="2">Spezifische Fehlerinformationen</td><td>(4) Parameter-Nr (5). Fehlermerker- Nr./ CHK-Anweisungs-Fehlfunktions-Nr.
<table><tr><th>Nummer</th><th>Bedeutung</th></tr><tr><td>SD16</td><td>Parameter-Nr.</td></tr><tr><td>SD17</td><td rowspan="10">Nicht belegt</td></tr><tr><td>SD18</td></tr><tr><td>SD19</td></tr><tr><td>SD20</td></tr><tr><td>SD21</td></tr><tr><td>SD22</td></tr><tr><td>SD23</td></tr><tr><td>SD24</td></tr><tr><td>SD25</td></tr><tr><td>SD26</td></tr></table><table><tr><th>Nummer</th><th>Bedeutung</th></tr><tr><td>SD16</td><td>Parameter-Nr.</td></tr><tr><td>SD17</td><td rowspan="10">Nicht belegt</td></tr><tr><td>SD18</td></tr><tr><td>SD19</td></tr><tr><td>SD20</td></tr><tr><td>SD21</td></tr><tr><td>SD22</td></tr><tr><td>SD23</td></tr><tr><td>SD24</td></tr><tr><td>SD25</td></tr><tr><td>SD26</td></tr></table>(6) Parametrierfehler bei Sondermodulen (Nur bei CPUs des MELSEC System Q)
<table><tr><th>Nummer</th><th>Bedeutung</th></tr><tr><td>SD16</td><td>Parameter-Nr.</td></tr><tr><td>SD17</td><td>Fehlercode für Sondermodul</td></tr><tr><td>SD18</td><td rowspan="10">Nicht belegt</td></tr><tr><td>SD19</td></tr><tr><td>SD20</td></tr><tr><td>SD21</td></tr><tr><td>SD22</td></tr><tr><td>SD23</td></tr><tr><td>SD24</td></tr><tr><td>SD25</td></tr><tr><td>SD26</td></tr></table></td><td>S (Fehler)</td><td>Neu</td><td>●</td></tr>
<tr><td>SD50</td><td>Rücksetzen eines Fehlers</td><td>Fehlernummer des zurückgesetzten Fehlers</td><td>Speichert die Fehlernummer des zurückgesetzten Fehlers</td><td>B</td><td>Neu</td><td></td></tr>
<tr><td>SD51</td><td>Batteriespannung zu niedrig (Latch-Merker)</td><td>Bit-Muster, das anzeigt, wo die Batteriespannung abgefallen ist</td><td>● Die entsprechenden Bits werden gesetzt, wenn die Batteriespannung sinkt.
● Dieses Bit bleibt gesetzt, auch wenn die Batteriespannung wieder ihren Normalwert erreicht hat.
 

Bei einer Q00J-, Q00- u. Q01CPU wird nur Bit 0 gesetzt.</td><td>S (Fehler)</td><td>Neu</td><td></td></tr>
<tr><td>SD52</td><td>Batteriespannung niedrig</td><td>Bit-Muster, das anzeigt, wo die Batteriespannung abgefallen ist</td><td>● Funktionweise wie in SD51beschrieben (siehe oben)
● Dieses Bit wird rückgesetzt, nachdem die Batterie ihren Normalwert erreicht hat.</td><td>S (Fehler)</td><td>Neu</td><td></td></tr>
<tr><td>SD53</td><td>Abfall der Versorgungsspannung</td><td>Häufigkeit der Spannungsabfälle</td><td>● In dieses Register wird bei jedem Spannungsabfall, bei dem die Nennspannung während des Betriebs um mehr als 20% sinkt, eine „1“ addiert. Der Wert wird in binärer Form abgelegt.</td><td>S (Fehler)</td><td>D9005</td><td>● Rem</td></tr>
</table>

Liste der Diagnoseregister (Fortsetzung)

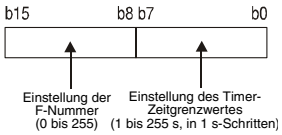
Adresse	Name	Bedeutung	Beschreibung	Gesetzt von (wenn gesetzt)	A-CPU-Register D9 [ ] [ ] [ ]	Gültig für:
SD54	MINI-Link-Fehler	Status der Fehlerkennung	(1) Das relevante Stations-Bit wird gesetzt, wenn eine der Kopfadressen eines installierten MINI (-S3)-Modules X(n+0) /X(n+20), X(n+6)/()n+26), X(n+7)/(n+27) oder X(n+8)/Xn+28) gesetzt wird. (2) Das relevante Bit wird gesetzt, wenn die Kommunikation zwischen den installierten MINI (-S3)-Modulen und der CPU nicht möglich ist. 	S (Fehler)	D9004 Geändertes Format	QnA-CPU
SD60	Nummer der defekten Sicherung	Nummer des Moduls, dessen Sicherung defekt ist	Der hier gespeicherte Wert ist die unterste Stationsadresse des Moduls, dessen Sicherung defekt ist, geteilt durch 16.	S (Fehler)	D9000	● Rem
SD61	Vergleichsfehler mit E/A-Modul	Nummer des Moduls, bei dem der Vergleichsfehler vorliegt	Die untere Moduladresse, bei dem der Vergleichsfehler zuerst erkannt wurde.	S (Fehler)	D9002	
SD62	Fehlermerker-Nr.		Hier wird die zuerst erkannte Fehlernummer gespeichert.	S (Anweisungs-ausführung)	D9009	●
SD63	Anzahl der Fehlermerker		Speichert die Anzahl der Fehlermerker.	S (Anweisungs-ausführung)	D9124	

Liste der Diagnoseregister (Fortsetzung)

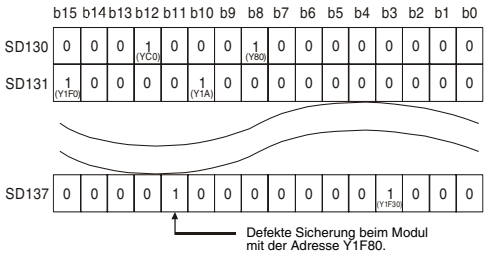
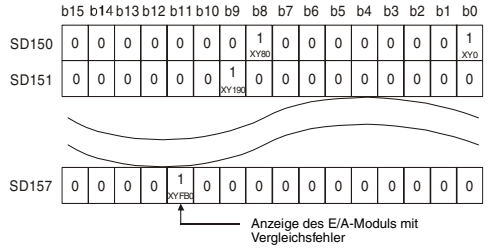
<table>
<tr><th>Adresse</th><th>Name</th><th>Bedeutung</th><th>Beschreibung</th><th>Gesetzt von (wenn gesetzt)</th><th>A-CPU-Register D9 [ ] [ ] [ ]</th><th>Gültig für:</th></tr>
<tr><td>SD64</td><td rowspan="16">Tabelle der erkannten Fehlermerker-nummer</td><td rowspan="16">Nummer der erkannten Fehlermerker</td><td rowspan="16">Wenn einer der Fehlermerker über eine OUT F oder SET F-Anweisung gesetzt wird, wird die Fehlermerkeradresse des gesetzten Fehlermerkers in binärer Form in die Register SD64 bis SD79 geschrieben.
Eine Fehlermerkeradresse, die über eine RST F-Anweisung rückgesetzt wird, wird aus dem Registerbereich gelöscht. Der Inhalt der nachfolgenden Datenregister wird anschließend um ein Register hochgeschoben.
Bei Ausführung einer LEDR-Anweisung wird der Inhalt von SD64 bis SD79 um ein Bit nach oben verschoben.
(Dieser Vorgang wird auch ausgeführt, wenn der Schlüsselschalter an der CPU (Q3A/Q4A) auf RESET geschaltet wird.)
Liegen mehr als 16 Fehlermeldungen vor, wird der 17. Fehlermerker nicht in den Registern SD64 bis SD79 gespeichert.

SET SET SET SET SET SET SET SET SET SET SET SET SET LEDR
F50 F25 F19 F25 F15 F70 F65 F38 F110F151F210
SD62 | 0 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 99 | — Erkannte Adressen
SD63 | 0 | 1 | 2 | 3 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 8 | — Anzahl der erkannten Fehlermerker
SD64 | 0 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 99
SD65 | 0 | 0 | 25 | 25 | 99 | 99 | 99 | 99 | 99 | 99 | 99 | 99 | 15
SD66 | 0 | 0 | 0 | 99 | 0 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15
SD67 | 0 | 0 | 0 | 0 | 0 | 0 | 70 | 70 | 70 | 70 | 70 | 70 | 65
SD68 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 65 | 65 | 65 | 65 | 65 | 38
SD69 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 38 | 38 | 38 | 38 | 110
SD70 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 110 | 110 | 110 | 151
SD71 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 151 | 151 | 210
SD72 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 210 | 0
SD73 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0
SD74 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0
SD75 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0
SD76 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0
SD77 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0
SD78 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0
SD79 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0
— Erkannte Adressen</td><td rowspan="16">S (Anweisungs-ausführung)</td><td>D9125</td><td rowspan="16">●</td></tr>
<tr><td>SD65</td><td>D9126</td></tr>
<tr><td>SD66</td><td>D9127</td></tr>
<tr><td>SD67</td><td>D9128</td></tr>
<tr><td>SD68</td><td>D9129</td></tr>
<tr><td>SD69</td><td>D9130</td></tr>
<tr><td>SD70</td><td>D9131</td></tr>
<tr><td>SD71</td><td>D9132</td></tr>
<tr><td>SD72</td><td>Neu</td></tr>
<tr><td>SD74</td><td>Neu</td></tr>
<tr><td>SD75</td><td>Neu</td></tr>
<tr><td>SD76</td><td>Neu</td></tr>
<tr><td>SD77</td><td>Neu</td></tr>
<tr><td>SD78</td><td>Neu</td></tr>
<tr><td>SD79</td><td>Neu</td></tr>
<tr><td>SD80</td><td colspan="2">Fehlercode der CHK-Anweisung</td><td>Der von der CHK-Anweisung erkannte Fehlercode wird im BCD-Format gespeichert.</td><td>S (Anweisungs-ausführung)</td><td>Neu</td><td>● außer Q00J-, Q00- und Q01CPU</td></tr>
</table>

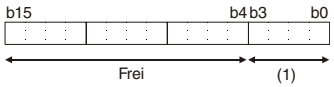
Liste der Diagnoseregister (Fortsetzung)

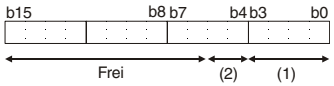
Adresse	Name	Bedeutung	Beschreibung		Gesetzt von (wenn gesetzt)	ACPU-Register D9 [ ] [ ] [ ]	Gültig für:
SD90	WDT-Einstellwert zur Überwachung von Schritten und Transitionen (Nur möglich wenn ein AS-Programm vorhanden ist.)	Fehlermerkernummer für den Timer-Einstellwert und Zeitüberschreitungsfehler	Korrespondiert mit SM90	● Fehlermerkernummer, die gesetzt wird, wenn eine falsche WDT-Schritt-Überwachungszeit eingegeben oder ein WDT-Zeitüberschreitungsfehler ansteht.  ● Der Timer startet, wenn die Diagnosemerker SM90 bis SM99 gesetzt sind und ein Schritt aktiv ist. Wird die Weiterschaltbedingung des relevanten Schrittes nicht während des eingestellten Zeitwertes erfüllt, wird der Fehlermerker (F) gesetzt.	B	D9108	● außer Q00J-Q00- und Q01CPU
SD91			Korrespondiert mit SM91			D9109	
SD92			Korrespondiert mit SM92			D9110	
SD93			Korrespondiert mit SM93			D9111	
SD94			Korrespondiert mit SM94			D9112	
SD95			Korrespondiert mit SM95			D9113	
SD96			Korrespondiert mit SM96			D9114	
SD97			Korrespondiert mit SM97			Neu	
SD98			Korrespondiert mit SM98			Neu	
SD99			Korrespondiert mit SM99			Neu	
SD100	Übertragungsgeschwindigkeit	Speicher für die eingestellte Übertragungsgeschwindigkeit der seriellen Schnittstelle	K96: 9600 Bit/s, K192: 19,2 kBit/s, K384: 38,4 kBit/s, K576: 57,6 kBit/s, K1152: 115,2 kBit/s		S (Beim Einschalten der Versorgungspannung oder nach einem Reset)	Neu	Q00J-Q00- und Q01CPU
SD101	Kommunikationseinstellungen	Speicher für die Einstellungen zur seriellen Kommunikation	Bit 4 = AUS: Ohne Prüfsumme Bit 4 = EIN: Mit Prüfsumme Bit 5 = AUS: Online-Programmänderungen nicht zugelassen Bit 5 = EIN: Online-Programmänderungen erlaubt Die anderen Bits haben keine Bedeutung.			Neu	
SD102	Wartezeit	Speicher für die Wartezeit bei der seriellen Kommunikation	0: Keine Wartezeit 1 bis F_H: Wartezeit in Einheiten von 10 ms. Voreinstellung: 0			Neu	
SD105	Übertragungsgeschwindigkeit für CH1 (RS 232)	Speicher für die eingestellte Übertragungsgeschwindigkeit.	K3: 300 Bit/s, K6: 600 Bit/s, K24: 2400 Bit/s, K48: 4800 Bit/s, K96: 9600 Bit/s, K192: 19,2 kBit/s, K384: 38,4 kBit/s, K576: 57,6 kBit/s, K1152: 115,2 kBit/s		S	Neu	Q-CPU außer Q00J-Q00- und Q01CPU
SD110	Sendeergebnis	Fehlercode beim Senden von Daten	Falls beim Senden von Daten mittels serieller Kommunikation ein Fehler aufgetreten ist, wird hier der Fehlercode gespeichert.		S (Fehler)	Neu	Q00J-Q00- und Q01CPU
SD111	Empfangsergebnis	Fehlercode beim Empfang von Daten	Falls beim Empfang von Daten mittels serieller Kommunikation ein Fehler aufgetreten ist, wird hier der Fehlercode gespeichert.		S (Fehler)	Neu	
SD120	Fehlernummer bei Ausfall der externen Versorgungsspannung	Nummer des Moduls, dessen externe Spannungsversorgung ausgefallen ist	Die niedrigste Adresse des Moduls des System Q, dessen Versorgungsspannung ausgefallen ist, wird gespeichert. (In Vorbereitung)		S (Fehler)	Neu	Q-CPU außer Q00J-Q00- und Q01CPU

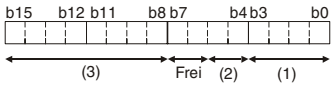
Liste der Diagnoseregister (Fortsetzung)

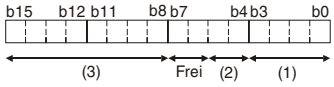
Adresse	Name	Bedeutung	Beschreibung	Gesetzt von (wenn gesetzt)	ACPU-Register D9 [ ] [ ] [ ]	Gültig für:
SD130 SD131 SD132 SD133 SD134 SD135 SD136 SD137	Module mit defekter Sicherung	Das Bit-Muster (16 Bit) zeigt die Module mit defekter Sicherung an 0 : Keine defekte Sicherung 1 : Defekte Sicherung vorhanden	● Die Anzahl der Ausgangsmodule mit defekter Sicherung wird als Bit-Muster von 16 Bit gespeichert. (Wenn die Modulnummer in den Parametern gesetzt ist, wird diese Nummer gespeichert.) ● Es werden auch defekte Sicherungen in Ausgangsmodulen von Remote-Stationen erfasst. ● Nach Austausch der defekten Sicherung wird das entsprechende Bit nicht automatisch rückgesetzt. Das Bit muss durch Rücksetzen der Fehlermeldung gelöscht werden. 	S (Fehler)	Neu	Q00J-, Q00- und Q01CPU
SD150 SD151 SD152 SD153 SD154 SD155 SD156 SD157	E/A-Module mit Vergleichsfehler	Das Bit-Muster (16 Bit), zeigt die Module mit Vergleichsfehler an 0 : Keine E/A-Vergleichsfehler 1 : E/A-Vergleichsfehler vorhanden	● Ist der aktuelle Status eines E/A-Moduls von dem vorgegebenen Status nach Einschalten der Versorgungsspannung verschieden, werden die E/A-Modulinformationen in dem Register gespeichert. (Wenn die Modulnummer in den Parametern gesetzt ist, wird diese Nummer gespeichert.) ● Es werden auch E/A-Modulinformationen erkannt . 	S (Fehler)	Neu	

(2) Systeminformationen

<table>
<tr><th>Adresse</th><th>Name</th><th>Bedeutung</th><th>Beschreibung</th><th>Gesetzt von (wenn gesetzt)</th><th>A-CPU-Register D9 [ ] [ ] [ ]</th><th>Gültig für:</th></tr>
<tr><td rowspan="4">SD200</td><td rowspan="4">Schaltzustand</td><td rowspan="4">Zustand des CPU-Betriebsarten-schalters</td><td>● Der Zustand des Betriebsartenschalters wird im folgenden Format gespeichert:
 

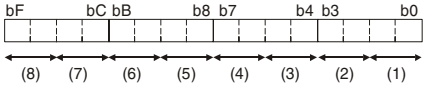
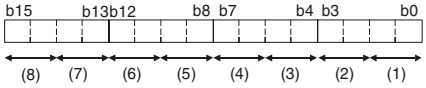
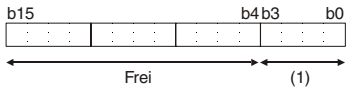
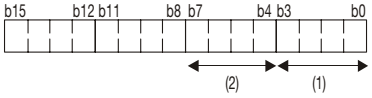
(1) Betriebsart — Immer 1: STOP</td><td>S (Kontinuierlich)</td><td>Neu</td><td>Remote</td></tr>
<tr><td>● Der Zustand des Betriebsartenschalters wird im folgenden Format gespeichert:
 

(1) Betriebsart — (0): RUN; (1): STOP
(2) Speicherkarten-schalter — Immer AUS</td><td rowspan="3">S (END-Verarbeitung)</td><td>Neu</td><td>Q00J-, Q00- und Q01CPU</td></tr>
<tr><td>● Der Zustand des Betriebsartenschalters wird im folgenden Format gespeichert:
 

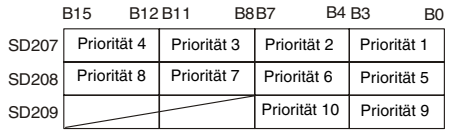
(1) Betriebsart — (0): RUN; (1): STOP; (2): L.CLR
(2) Speicherkarten-schalter — b4: Speicherkarte A, b5: Speicherkarte B, 0: AUS, 1: EIN
(3) DIP-Schalter — Die Bits b8 bis b11 entsprechen den Schaltern SW1 bis SW5. 0: AUS, 1: EIN</td><td>Neu</td><td>Q-CPU außer Q00J- Q00- und Q01CPU</td></tr>
<tr><td>● Der Zustand des CPU-Schlüsselschalters wird im folgenden Format gespeichert
 

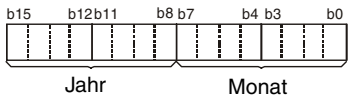
(1) Betriebsart — (0): RUN; (1): STOP; (2): L.CLR
(2) Speicherkarten-schalter — b4: Speicherkarte A, b5: Speicherkarte B, 0: AUS, 1: EIN
(3) DIP-Schalter — Die Bits b8 bis b11 entsprechen den Schaltern SW1 bis SW5. 0: AUS, 1: EIN</td><td>S (END-Verarbeitung)</td><td>Neu</td><td>QnA-CPU</td></tr>
</table>

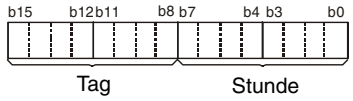
(2) Systeminformationen

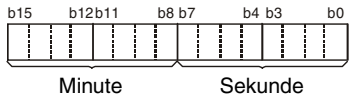
Adresse	Name	Bedeutung	Beschreibung	Gesetzt von (wenn gesetzt)	A-CPU-Register D9 [ ] [ ] [ ]	Gültig für:
SD201	LED-Zustand	Zustand der CPU-LED-Anzeige	● Die unten aufgeführten Informationen beziehen sich auf die LED-Anzeigen der CPU.  (1) : RUN (5) : BOOT (2) : ERROR (6) : Frei (3) : USER (7) : Frei (4) : BAT.ALARM (8) : Betriebsart Die Betriebsart wird in folgenden Bit-Muster gespeichert: 0: AUS, 1: GRÜN, 2: ORANGE Bei einer Q00J-, Q00- oder Q01CPU stehen nur die Bereiche 1 und 2 zur Verfügung.	S (Zustandsänderung)	Neu	Q-CPU
			● Die unten aufgeführten Informationen beziehen sich auf die LED-Anzeigen der CPU und werden im folgenden Bit-Muster gespeichert: ● AUS bei 0; EIN bei 1; Blinken bei 2  (1) : RUN (5) : BOOT (2) : ERROR (6) : Card A (Speicherkarte A) (3) : USER (7) : Card B (Speicherkarte B) (4) : BAT.ALARM (8) : Frei	S (Zustandsänderung)	Neu	QnA-CPU
SD202	LED AUS	Bit-Muster der ausgeschalteten LEDs	● Speichert Bit-Muster ausgeschalteter LEDs (Nur mit USER und BOOT LED möglich) ● AUS bei 0, EIN bei 1	B	Neu	QnA-CPU
SD203	Verarbeitungs-zustand der CPU		● Der Verarbeitungszustand wird wie unten angegeben gespeichert  (1) Verarbeitungszustand von dezentralen E/A-Modulen: Immer 2: STOP	S (Kontinurierlich)	Neu	Remote
			● Der CPU-Verarbeitungsstatus wird wie unten angegeben gespeichert.  (1) : Verarbeitungs-zustand der CPU: 0 : RUN; 1 : STEP-RUN (nicht bei Q00J-, Q00- und Q01CPU); 2 : STOP; 3 : PAUSE (2) : STOP/PAUSE verursacht durch: 0 : Betriebsartenschalter; 1 : Remote-Kontakt; 2 : Peripheriegerät, Computerverbindung oder andere Remote-Quellen; 3 : interne Programmanweisungen; 4 : Fehler Hinweis: Es wird nur der zuerst aufgetretene Fehler angezeigt.	S (END-Verarbeitung)	D9015 (Geändertes Format)	●

(2) Systeminformationen

<table>
<tr><th>Adresse</th><th>Name</th><th>Bedeutung</th><th>Beschreibung</th><th>Gesetzt von (wenn gesetzt)</th><th>A-CPU-Register D9 [ ] [ ] [ ]</th><th>Gültig für:</th></tr>
<tr><td>SD206</td><td>Ausführungsart des Operanden-test</td><td>Anzeige des ausge-führten Operanden-test</td><td>● Beim Operandentest mittels eines Programmiergerätes wird hier eingetragen, welche Operanden getestet werden.
0: Kein Operandentest aktiviert
1: Während der Prüfung der Eingänge (X)
2: Während der Prüfung der Ausgänge (Y)
3: Während der Prüfung der Ein-/Ausgänge (X/Y)</td><td>S
(Anforderung)</td><td>Neu</td><td>Remote</td></tr>
<tr><td>SD207</td><td rowspan="3">Anzeigepriorität der ERR-LED</td><td>Priorität 1 bis 4</td><td rowspan="3">● Tritt ein Fehler auf, wird er auf dem LED-Display (blinkt) entsprechend der Fehlernummer in den vorliegenden Registern angezeigt.
● Die Einstellbereiche der Anzeigeprioritäten sehen wie folgt aus.
 

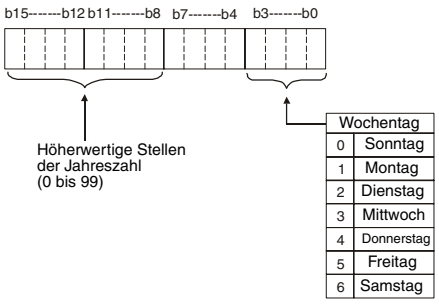
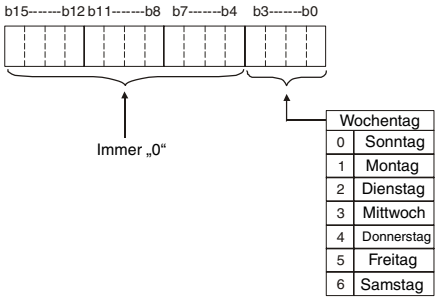
Werkseinstellung: (4321H)
(8765 H)
(00A9 H)
● Ist die „0“ eingestellt, erfolgt keine Anzeige. Aber auch wenn eine „0“ eingestellt ist, werden die Informationen über den Fehler, der die CPU gestoppt hat (einschließlich der Parametereinstellungen) durch die LED angezeigt.</td><td rowspan="3">B</td><td>D9038</td><td rowspan="3">● außer Q00J-, Q00-und Q01CPU</td></tr>
<tr><td>SD208</td><td>Priorität 5 bis 8</td><td>D9039 (Geändertes Format)</td></tr>
<tr><td>SD209</td><td>Priorität 9 bis 10</td><td>Neu</td></tr>
<tr><td>SD210</td><td rowspan="3">Uhr-Daten</td><td>Uhr-Daten
(Jahr, Monat)</td><td>● Das Jahr (letzten 2 Stellen) und der Monat werden im BCD-Code in Register SD210 gespeichert:
 

Beispiel: Juli 1993 = 9307</td><td rowspan="3">S/B
(Anforderung)</td><td>D9025</td><td rowspan="3">● Rem</td></tr>
<tr><td>SD211</td><td>Uhr-Daten
(Tag, Stunde)</td><td>● Der Tag und die Stunden werden im BCD-Code in Register SD211 gespeichert:
 

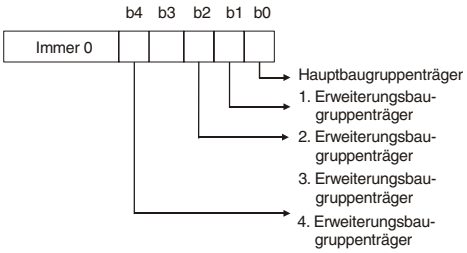
Beispiel: 31., 10 Uhr = 3110</td><td>D9026</td></tr>
<tr><td>SD212</td><td>Uhr-Daten
(Minute, Sekunde)</td><td>● Die Minuten und die Sekunden werden in Register SD212 BCD-codiert gespeichert.
 

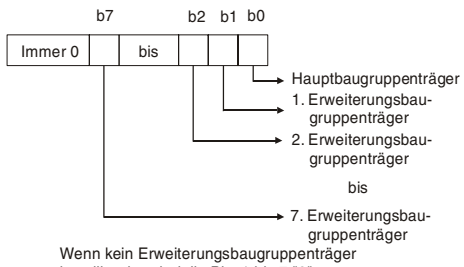
Beispiel: 35 min, 48s = 3548</td><td>D9027</td></tr>
</table>

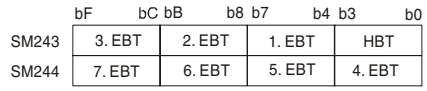
(2) Systeminformationen

<table>
<tr><th>Adresse</th><th>Name</th><th>Bedeutung</th><th>Beschreibung</th><th>Gesetzt von (wenn gesetzt)</th><th>A-CPU-Register D9 [ ] [ ] [ ]</th><th>Gültig für:</th></tr>
<tr><td rowspan="2">SD213</td><td rowspan="2">Uhr-Daten</td><td rowspan="2">Uhr-Daten (Wochentag)</td><td>● Der Wochentag wird im BCD-Code in Register SD213 gespeichert.
 
</td><td>S/B (Anforderung</td><td rowspan="2">D9028</td><td>Q-CPU Rem</td></tr>
<tr><td>● Der Wochentag wird im BCD-Code in Register SD213 gespeichert.
 
</td><td>S/B (Anforderung)</td><td>QnA-CPU</td></tr>
<tr><td>SD220
SD221
SD222
SD223
SD224
SD226
SD227</td><td>Daten des LED-Displays</td><td>Anzeigedaten des Displays</td><td>● Die ASCII-Daten (16 Zeichen) des LED-Displays werden in den unten aufgeführten Registern gespeichert.
b15 bis b8 b7 bis b0
SD220: 15. Zeichen von rechts | 16. Zeichen von rechts
SD221: 13. Zeichen von rechts | 14. Zeichen von rechts
SD222: 11. Zeichen von rechts | 12. Zeichen von rechts
SD223: 9. Zeichen von rechts | 10. Zeichen von rechts
SD224: 7. Zeichen von rechts | 8. Zeichen von rechts
SD225: 5. Zeichen von rechts | 6. Zeichen von rechts
SD226: 3. Zeichen von rechts | 4. Zeichen von rechts
SD227: 1. Zeichen von rechts | 2. Zeichen von rechts</td><td>S (Zustandsänderung)</td><td>Neu</td><td>●</td></tr>
<tr><td>SD240</td><td>Betriebsart des Baugruppenträgers</td><td>0: Automatischer Betrieb
1: Detaillierter Betrieb</td><td>Dieses Register dient zur Speicherung der Betriebsart des Baugruppenträgers.</td><td>S (Initialisierung)</td><td>Neu</td><td rowspan="2">Q-CPU Rem</td></tr>
<tr><td>SD241</td><td>Anzahl der Erweiterungsbaugruppenträger</td><td>0: Nur Hauptbaugruppenträger
1 bis 7: Anzahl der Erweiterungsbaugruppenträger</td><td>In diesem Register wird die Anzahl der installierten Erweiterungsbaugruppenträger gespeichert.</td><td>S (Initialisierung)</td><td>Neu</td></tr>
</table>

(2) Systeminformationen

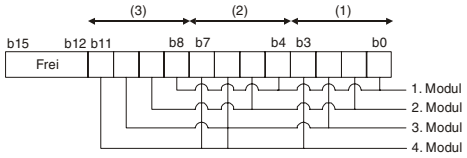
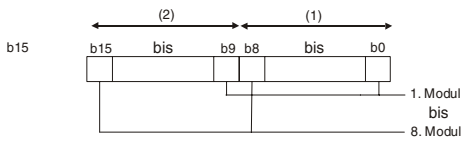
<table>
<tr><th>Adresse</th><th>Name</th><th>Bedeutung</th><th>Beschreibung</th><th>Gesetzt von (Wenn gesetzt)</th><th>A-CPU-Register D9 [ ] [ ] [ ]</th><th>Gültig für:</th></tr>
<tr><td rowspan="2">SD242</td><td rowspan="2">Unterscheidung zwischen A- und Q-Baugruppen-träger</td><td rowspan="2">0: Baugruppen-träger vom Typ QA[ ] [ ]B ist installiert (Betriebsart A)
1: Baugruppen-träger vom Typ Q[ ] [ ]B ist installiert (Betriebsart Q)</td><td> 

Wenn kein Erweiterungsbaugruppenträger installiert ist, sind die Bits 1 bis 4 "0".</td><td rowspan="2">S (Initialisierung)</td><td rowspan="2">Neu</td><td>Q00J-, Q00- und Q01CPU</td></tr>
<tr><td> 

Wenn kein Erweiterungsbaugruppenträger installiert ist, sind die Bits 1 bis 7 "0"</td><td>Q02- Q06H- Q12H- Q25H- CPU; Rem</td></tr>
<tr><td>SD243</td><td rowspan="2">Anzahl der Steckplätze auf den Baugruppenträgern</td><td rowspan="2">Anzahl der Steckplätze auf den Baugruppenträgern
Bei einer Q00J-, Q00- oder Q01CPU sind die Positionen für den 5. bis 7. EBT mit Null belegt.</td><td rowspan="2"> 

Die Anzahl der Steckplätze wird für den Hauptbaugruppen (HBT) und die Erweiterungsbaugruppenträger (EBT) in den entsprechenden Bereichen abgelegt.</td><td rowspan="2">S (Initialisierung)</td><td rowspan="2">Neu</td><td rowspan="2">Q-CPU</td></tr>
<tr><td>SD244</td></tr>
<tr><td>SD250</td><td>Maximum an E/As geladen</td><td>Maximale Anzahl an E/As geladen</td><td>Wird SM250 gesetzt, werden zu den oberen beiden Stellen der letzten geladenen E/A-Modul-Adresse der Wert 1 addiert und als Binärwert gespeichert.</td><td>S (END-Verarbeitung)</td><td>Neu</td><td>●</td></tr>
<tr><td>SD251</td><td>Adresse eines auszuwechseln-den E/A-Moduls</td><td>Startadresse des E/A-Moduls</td><td>Das Register D9094 speichert die oberen beiden Stellen der Startadresse eines E/A-Moduls, das während des Online-Betriebs aus dem Baugruppenträger entfernt bzw. eingesetzt wird, und speichert sie als Binärwert ab.</td><td>B</td><td>D9094</td><td>Q2A(S1)-, Q3A- Q4A- Q4AR- CPU</td></tr>
<tr><td>SD253</td><td>Übertragungs-gechwindigkeit für RS422</td><td>0: 9600 Bit/s
1: 19,2 kBit/s
2: 38,4 kBit/s</td><td>Das Register speichert den Wert für die Übertragungs-geschwindigkeit der RS422-Schnittstelle.</td><td>S (Bei Änderung)</td><td>Neu</td><td>QnA-CPU</td></tr>
</table>

(2) Systeminformationen

Adresse	Name	Bedeutung		Beschreibung	Gesetzt von (wenn gesetzt)	A-CPU-Register D9 [ ] [ ] [ ]	Gültig für:
SD254	MELSECNET/10 Information	Anzahl der installierten Module		Zeigt die Anzahl der installierten Module im MELSECNET/10 an.	S (Initialisierung)	Neu	●
SD255		Informationen des ersten Moduls	E/A-Adresse	E/A-Adresse des ersten installierten Moduls im MELSECNET/10.			
SD256			Netzwerk-nummer	Netzwerkadresse des ersten installierten Moduls im MELSECNET/10.			
SD257			Gruppen-nummer	Gruppennummer des ersten installierten Moduls im MELSECNET/10.			
SD258			Stations-nummer	Stationsnummer des ersten installierten Moduls im MELSECNET/10.			
SD259			Standby-Informa-tion	Sind Standby-Stationen vorhanden, wird die Modulnummer der Standby-Station gespeichert (1 bis 4).			● außer Q00J-, Q00- und Q01CPU
SD260 – SD264		Informationen des zweiten Moduls		Die Konfiguration ist identisch mit der des ersten Moduls.			
SD265 – SD269		Informationen des dritten Moduls					
SD270 – SD274		Informationen des vierten Moduls					
SD280	CC-Link Fehler	Zustand bei erkannten Fehler		 (1) Wenn Xn0 des installierten CC-Link eingeschaltet wird, wird das der Station zugeordnete Bit gesetzt. (2) Wenn entweder Xn1 oder XnF des installierten CC-Link ausgeschaltet wird, wird das der Station zugeordnete Bit gesetzt. (3) Die Bits in diesem Bereich werden gesetzt, wenn die CPU nicht mit dem installiertem CC-Link kommunizieren kann.	S (Bei Fehler)	Neu	Q-CPU Rem
				 (1) Wenn Xn0 des installierten CC-Link eingeschaltet wird, wird das der Station zugeordnete Bit gesetzt. (2) Wenn entweder Xn1 oder XnF des installierten CC-Link ausgeschaltet wird, wird das der Station zugeordnete Bit gesetzt (3) Die Bits in diesem Bereich werden gesetzt, wenn die CPU nicht mit dem installiertem CC-Link kommunizieren kann.	S (Bei Fehler)	Neu	QnA-CPU

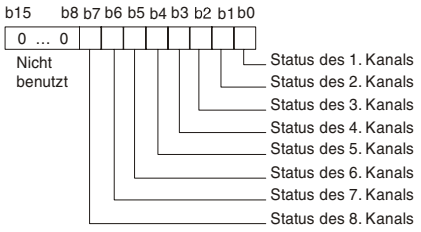
(2) Systeminformationen

Adresse	Name	Bedeutung	Beschreibung	Gesetzt von (wenn gesetzt)	A-CPU-Register D9 [ ] [ ] [ ]	Gültig für:
SD290	Operanden-zuweisung (identisch mit den Parameter-inhalten)	Anzahl der Adressen des Operanden X	Aktuell eingestellte Anzahl der Adressen der Operanden X	S (Initialisierung)	Neu	● Rem
SD291		Anzahl der Adressen des Operanden Y	Aktuell eingestellte Anzahl der Adressen der Operanden Y			
SD292		Anzahl der Adressen des Operanden M	Aktuell eingestellte Anzahl der Adressen der Operanden M			
SD293		Anzahl der Adressen des Operanden L	Aktuell eingestellte Anzahl der Adressen der Operanden L		Neu	●
SD294		Anzahl der Adressen des Operanden B	Aktuell eingestellte Anzahl der Adressen der Operanden B		Neu	● Rem
SD295		Anzahl der Adressen des Operanden F	Aktuell eingestellte Anzahl der Adressen der Operanden F		Neu	●
SD296		Anzahl der Adressen des Operanden SB	Aktuell eingestellte Anzahl der Adressen der Operanden SB		Neu	● Rem
SD297		Anzahl der Adressen des Operanden V	Aktuell eingestellte Anzahl der Adressen der Operanden V		Neu	●
SD298		Anzahl der Adressen des Operanden S	Aktuell eingestellte Anzahl der Adressen der Operanden S			
SD299		Anzahl der Adressen des Operanden T	Aktuell eingestellte Anzahl der Adressen der Operanden T			
SD300		Anzahl der Adressen des Operanden ST	Aktuell eingestellte Anzahl der Adressen der Operanden ST			
SD301		Anzahl der Adressen des Operanden C	Aktuell eingestellte Anzahl der Adressen der Operanden C			
SD302		Anzahl der Adressen des Operanden D	Aktuell eingestellte Anzahl der Adressen der Operanden D	(Initialisierung)	Neu	● Rem
SD303		Anzahl der Adressen der Operanden W	Aktuell eingestellte Anzahl der Adressen der Operanden W			
SD304		Anzahl der Adressen der Operanden SW	Aktuell eingestellte Anzahl der Adressen der Operanden SW			
SD315	Für Kommunikation reservierte Zeit	Zeit, die für Kommunikation reserviert ist.	Die hier eingetragene Zeit (Bereich 1 ms bis 100 ms) steht für die Kommunikation mit einem Programmiergerät zur Verfügung. Je größer der hier eingetragene Wert ist, um so kürzer wird die zur Kommunikaton mit anderen Geräten (z.B. serielle Kopplung) zur Verfügung stehende Reaktionszeit. Wenn der Wert ausserhalb des erlaubten Bereiches liegt, wird er so behandelt, als ob kein Wert eingetragen ist. Die Zykluszeit verlängert sich um die eingestellte Zeit.	END-Verarbeitung	Neu	Q-CPU

(2) Systeminformationen

Adresse	Name	Bedeutung		Beschreibung	Gesetzt von (wenn gesetzt)	A-CPU-Register D9 [ ] [ ] [ ]	Gültig für:
SD340	Ethernet-Information	Anzahl der installierten Module		Zeigt die Anzahl der installierten Module im Ethernet an.	S (Initialisierung)	Neu	Q-CPU Rem
SD341		Informationen des ersten Moduls	E/A-Adresse	E/A-Adresse des ersten installierten Moduls im Ethernet.			
SD342			Netzwerk-nummer	Netzwerkadresse des ersten installierten Moduls im Ethernet.			
SD343			Gruppen-nummer	Gruppennummer des ersten installierten Moduls im Ethernet.			
SD344			Stations-nummer	Stationsnummer des ersten installierten Moduls im Ethernet.			
SD345 — SD346			Frei	Nicht belegt. Bei der Q-CPU wird die Ethernet IP-Adresse des ersten Moduls im Pufferspeicher abgelegt.		Neu	Q-CPU (außer Q00J-, Q00- und Q01CPU) Rem
SD347			Frei	Nicht belegt. Bei der Q-CPU wird der Ethernet-Fehlercode des ersten Moduls mit der ERRORRD-Anweisung gelesen..			
SD348 — SD354		Informationen des zweiten Moduls		Die Konfiguration ist identisch mit der des ersten Moduls.			
SD355 — SD361		Informationen des dritten Moduls		Die Konfiguration ist identisch mit der des ersten Moduls.			
SD362 — SD368		Informationen des vierten Moduls		Die Konfiguration ist identisch mit der des ersten Moduls.			
SD340	Ethernet-Information	Anzahl der installierten Module		Zeigt die Anzahl der installierten Module im Ethernet an.	S (Initialisierung)	Neu	QnA-CPU
SD341		Informationen des ersten Moduls	E/A-Adresse	E/A-Adresse des ersten installierten Moduls im Ethernet.			
SD342			Netzwerk-nummer	Netzwerkadresse des ersten installierten Moduls im Ethernet.			
SD343			Gruppen-nummer	Gruppennummer des ersten installierten Moduls im Ethernet.			
SD344			Stations-nummer	Stationsnummer des ersten installierten Moduls im Ethernet.			
SD345 — SD346			IP-Adresse	Ethernet IP-Adresse des ersten installierten Moduls.			
SD347			Fehlercode	Ethernet-Fehlercode des ersten installierten Moduls.			
SD348 — SD354		Informationen des zweiten Moduls		Die Konfiguration ist identisch mit der des ersten Moduls.			
SD355 — SD361		Informationen des dritten Moduls					
SD362 — SD368		Informationen des vierten Moduls					

(2) Systeminformationen

Adresse	Name	Bedeutung	Beschreibung	Gesetzt von (wenn gesetzt)	A-CPU-Register D9 [ ] [ ] [ ]	Gültig für:
SD380	Empfangszustand der Ethernet-Anweisungen	Informationen des ersten Moduls	 Bit gesetzt: Empfangen (Kanal wird benutzt) Bit nicht gesetzt: Nicht empfangen (Kanal wird nicht benutzt)	S (Initialisierung)	Neu	Q-CPU
SD381		Informationen des zweiten Moduls	Die Konfiguration ist identisch mit der des ersten Moduls.			
SD382		Informationen des dritten Moduls				
SD383		Informationen des vierten Moduls				
SD392	Software-Version	Version der System-Software	Die Version der internen Systemsoftware wird im High-Byte im ASCII-Code gespeichert. Die im Low-Byte enthaltenen Daten sind nicht gültig. Software Version 'A ' wird beispielsweise im High-Byte als 41_H abgelegt. Die hier gespeicherte Versionsnummer der Systemsoftware kann von der auf dem Gehäuse des Moduls aufgedruckten Versionsnummer abweichen.	S (Initialisierung)	D9060	
SD395	CPU-Nummer	1: CPU 1 2: CPU 2 3: CPU 3 4: CPU 4	Dieses Register enthält die Nummer der CPU, wenn sie in einem Multi-CPU-System betrieben wird.	S (Initialisierung)	Neu	CPU des System Q, die für den Multi-CPU-Betrieb geeignet ist.

(3) System-Uhr/Counter

Adresse	Name	Bedeutung	Beschreibung	Gesetzt von (wenn gesetzt)	A-CPU-Register D9 [ ] [ ] [ ]	Gültig für:
SD412	1-Sekunden-Counter	Zählt in Sekundenschritten	● Mit Beginn des RUN-Betriebes der CPU beginnt der Zähler im Sekundentakt zu zählen. ● Der Zähler zählt von 0 bis 32767 aufwärts, anschließend bis -32767 abwärts und wieder zurück auf 0.	S (Zustandsänderung)	D9022	●
SD414	2n-Sekundentakt	Einheiten von 2xn Sekunden	● Speichert die Einstellwerte von n des 2xn Sekundentaktes (Voreinstellung = 30). ● Es können Werte zwischen 1 und 32767 gesetzt werden.	B	Neu	
SD415	2n-Millisekundentakt	Einheiten von 2xn Millisekunden	● Speichert die Einstellwerte von n des 2 x n Millisekundentaktes (Voreinstellung = 30). ● Es können Werte zwischen 1 und 32767 eingetragen werden.	B	Neu	Q02-Q02H-Q06H-Q12PH-Q12PH-Q25H-Q25PH-CPU
SD420	Counter der Programmzyklen	Zählt die Anzahl der Programmzyklen	● Mit Beginn des RUN-Betriebes der CPU wird der Zähler bei jedem Programmzyklus um 1 erhöht. ● Der Zähler zählt von 0 bis 32767 aufwärts, anschließend bis -32767 abwärts und wieder zurück auf 0.	S (END-Verarbeitung)	Neu	●
SD430	Counter der Programm-zyklen niedriger Verarbeitungs-geschwindigkeit	Zählt die Anzahl der Programmzyklen niedriger Verarbeitungs-geschwindigkeit	● Nach dem Einschalten der CPU in den RUN-Betrieb beginnt der Zähler bei jedem Programmzyklus um 1 zu erhöhen. ● Der Zähler zählt von 0 bis 32767 aufwärts, anschließend bis -32767 abwärts und wieder zurück auf 0. ● Kann nur bei Programmen des Ausführungstyps „Low Speed Execution“ verwendet werden.	S (END-Verarbeitung)	Neu	● außer Q00J-, Q00- und Q01CPU

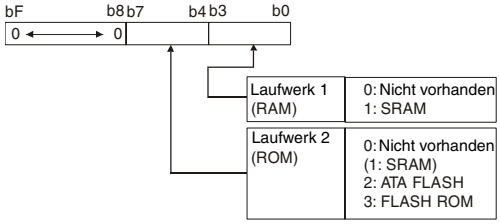
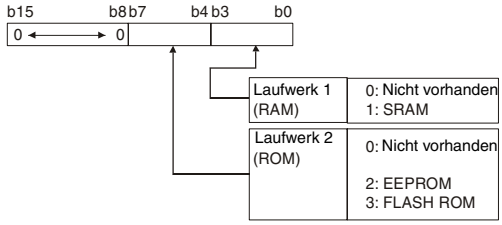
(4) Programmzyklusinformationen

Adresse	Name	Bedeutung	Beschreibung	Gesetzt von (wenn gesetzt)	A-CPU-Register D9 [ ] [ ] [ ]	Gültig für:
SD500	Programm-nummer des ausgeführten Programms	Ausführungstyp des Programms, das ausgeführt wird	● Die Programmnummer des aktuell ausgeführten Programms ist als Binärwert gespeichert.	S (Zustandsänderung)	Neu	● außer Q00J-, Q00- und Q01CPU
SD510	Programm-nummer des „Low Speed Execution“ Programms	File-Name des Programms	● Die Programmnummer des aktuell ausgeführten Programms des Typs „Low Speed Execution“ ist als Binärwert gespeichert. ● Nur bei gesetztem SM510 möglich.	S (END-Verarbeitung)	Neu	
SD520	Aktuelle Zykluszeit	Zykluszeit (Einheit 1 ms)	● Speichert die aktuelle Zeit der Programmzykluszeit (in 1-ms-Schritten) im Bereich von 0 bis 65535. ● Speichert die aktuelle Zeit der Programmzykluszeit (in 1-µs-Schritten) im Bereich von 00000 bis 900. Beispiel: Eine aktuelle Programmzykluszeit von 23,6 ms wird so gespeichert: D520 = 23 D521 = 600	S (END-Verarbeitung)	D9017 (Geändertes Format)	●
SD521		Zykluszeit (Einheit 1 µs)			Neu	
SD522	Zeit des Initialisierungs-zyklus	Zeit des Initialisierungszyklus (Einheit 1 ms)	● Speichert die Zeit des ersten Programmzyklus (in 1-ms-Schritten). ● Bereich von 0 bis 65535	S (END-Verarbeitung)	Neu	● außer Q00J-, Q00- und Q01CPU
SD523		Zeit des Initialisierungszyklus (Einheit 100 µs)	● Speichert die Zeit des ersten Programmzyklus (in 1-µs-Schritten). ● Bereich von 000 bis 900			
SD524	Minimale Zykluszeit	Minimale Zykluszeit (Einheit 1 ms)	● Speichert die minimale Programmzykluszeit (in 1-ms-Schritten). ● Bereich von 0 bis 65535	S (END-Verarbeitung)	D9018 (Geändertes Format)	●
SD525		Minimale Zykluszeit (Einheit 100µs)	● Speichert die minimale Programmzykluszeit (in 100-µs-Schritten). ● Bereich von 000 bis 900		Neu	
SD526	Maximale Zykluszeit	Maximale Zykluszeit (Einheit 1 ms)	● Speichert die maximale Programmzykluszeit (in 1-ms-Schritten) mit Ausnahme des ersten Zyklus. ● Bereich von 0 bis 65535	S (END-Verarbeitung)	D9019 (Geändertes Format)	
SD527		Maximale Zykluszeit (Einheit 100 µs)	● Speichert die maximale Programmzykluszeit (in 100-µs-Schritten) mit Ausnahme des ersten Zyklus. ● Bereich von 000 bis 900		Neu	
SD528	Zykluszeit für Programme des Ausführungs-modus „Low Speed Execution“	aktuelle Zykluszeit (Einheit 1 ms)	● Speichert die aktuelle Zykluszeit des Programms vom Typ „Low Speed Execution“ (in 1-ms-Schritten). ● Bereich von 0 bis 65535	S (END-Verarbeitung)	Neu	● außer Q00J-, Q00- und Q01CPU
SD529		aktuelle Zykluszeit (Einheit 100 µs)	● Speichert die aktuelle Zykluszeit des Programms vom Typ „Low Speed Execution“ (in 100-µs-Schritten). ● Bereich von 000 bis 900			
SD532	Minimale Zykluszeit für Programme des Ausführungs-modus „Low Speed Execution“	Minimale Zykluszeit (Einheit 1 ms)	● Speichert die minimale Programmzykluszeit des Programms vom Typ „Low Speed Execution“ (in 1-ms-Schritten). ● Bereich von 0 bis 65535	S (END-Verarbeitung)	Neu	
SD533		Minimale Zykluszeit (Einheit 100 µs)	● Speichert die minimale Programmzykluszeit des Programms vom Typ „Low Speed Execution“ (in 100-µs-Schritten). ● Bereich von 000 bis 900			
SD534	Maximale Zykluszeit für Programme des Ausführungs-modus „Low Speed Execution“	Maximale Zykluszeit (Einheit 1 ms)	● Speichert die maximale Programmzykluszeit des Programms vom Typ „Low Speed Execution“ (in 1-ms-Schritten, mit Ausnahme des 1. Zyklus. ● Bereich von 0 bis 65535	S (END-Verarbeitung)	Neu	
SD535		Maximale Zykluszeit (Einheit 100 µs)	● Speichert die maximale Programmzykluszeit des Programms vom Typ „Low Speed Execution“ (in 100-µs-Schritten) mit Ausnahme des 1. Zyklus. ● Bereich von 000 bis 900			

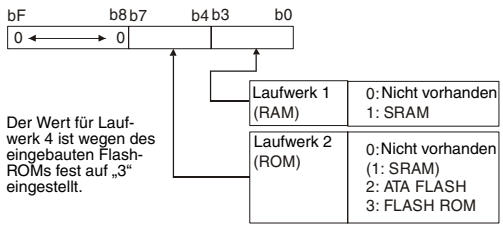
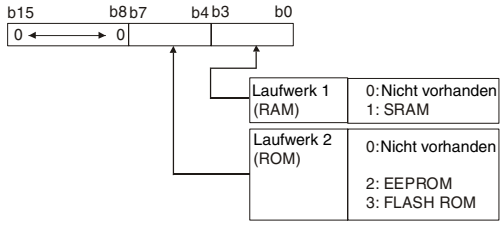
(4) Programmzyklusinformationen

Adresse	Name	Bedeutung	Beschreibung	Gesetzt von (wenn gesetzt)	A-CPU-Register D9 [ ] [ ] [ ]	Gültig für:
SD540	Zeit der END-Verarbeitung	Zeit der END-Verarbeitung (Einheit 1 ms)	● Speichert die Zeit vom Ende des letzten Programmzyklus bis zum Anfang des nächsten Zyklus (in 1-ms-Schritten). ● Bereich von 0 bis 65535	S (END-Verarbeitung)	Neu	●
SD541		Zeit der END-Verarbeitung (Einheit 100 µs)	● Speichert die Zeit vom Ende des letzten Programmzyklus bis zum Anfang des nächsten Zyklus (in 100-µs-Schritten). ● Bereich von 000 bis 900			
SD542	Wartezeit bei konstanter Zykluszeit	Wartezeit bei konstanter Zykluszeit (Einheit 1 ms)	● Speichert die Wartezeit bei gesetzter konstanter Zykluszeit (in 1-ms-Schritten). ● Bereich von 0 bis 65535	S (Erstes END)	Neu	
SD543		Wartezeit bei konstanter Zykluszeit (Einheit 100 µs)	● Speichert die Wartezeit bei gesetzter konstanter Zykluszeit (in 100-µs-Schritten). ● Bereich von 000 bis 900			
SD544	Kumulierte Ausführungszeit für Programme des Ausführungsmodus „Low Speed Execution“	Kumulierte Ausführungszeit für Programme des Ausführungstyp langsame Ausführung (Einheit 1 ms)	● Speichert die kumulierte Ausführungszeit des Programms vom Typ „Low Speed Execution“ (in 1-ms-Schritten). ● Bereich von 0 bis 65535	S (END-Verarbeitung)	Neu	● außer Q00J-, Q00- und Q01CPU
SD545		Kumulierte Ausführungszeit für Programme des Ausführungstyp „Low Speed Execution“ (Einheit 100 µs)	● Speichert die kumulierte Ausführungszeit des Programms vom Typ „Low Speed Execution“ (in 100-µs-Schritten). ● Bereich von 000 bis 900			
SD546	Ausführungszeit für Programme des Ausführungstyp „Low Speed Execution“	Ausführungszeit für Programme des Ausführungstyp „Low Speed Execution“ (Einheit 1 ms)	● Speichert die Ausführungszeit des Programms vom Typ „Low Speed Execution“ (in 1-ms-Schritten) während eines Zyklus. ● Bereich von 0 bis 65535 ● Speichert jeden Zyklus	S (END-Verarbeitung)	Neu	
SD547		Ausführungszeit für Programme des Ausführungstyp „Low Speed Execution“ (Einheit 100 µs)	● Speichert die Ausführungszeit des Programms vom Typ „Low Speed Execution“(in 100-µs-Schritten) während eines Zyklus. ● Bereich von 000 bis 900 ● Speichert jeden Zyklus			
SD548	Ausführungszeit für Programme des Ausführungstyp „Scan Execution“	Ausführungszeit für Programme des Ausführungsmodus „Scan Execution“ (Einheit 1 ms)	● Speichert die Ausführungszeit des Programms vom Typ „Scan Execution“ (in 1 ms Schritten) während eines Zyklus. ● Bereich von 0 bis 65535 ● Speichert jeden Zyklus	S (END-Verarbeitung)	Neu	●
SD549		Ausführungszeit für Programme des Ausführungstyp „Scan Execution“ (Einheit 100 µs)	● Speichert die Ausführungszeit des Programms vom Typ „Scan Execution“ (in 100 µs Schritten) während eines Zyklus. ● Bereich von 000 bis 900 ● Speichert jeden Zyklus			
SD550	Messung des Serviceintervalls für Module	Stations-/Modul-Nr.	● Setzt die E/A-Adresse des Moduls, dessen Serviceintervall gemessen wird.	B	Neu	
SD551	Serviceintervall	Serviceintervall des Moduls (Einheit 1 ms)	● Ist SM551 gesetzt, wird das Intervall gespeichert, nach dem das in SD550 benannte Modul gewartet wird (in 1-ms-Schritten). ● Bereich von 0 bis 65535	S (Anforderung)	Neu	● außer Q00J-, Q00- und Q01CPU
SD552		Serviceintervall des Moduls (Einheit 100 µs)	● Ist SM551 gesetzt, wird das Intervall gespeichert, nach dem das in SD550 benannte Modul gewartet wird (in 1-µs-Schritten). ● Bereich von 000 bis 900			

(5) Speicherkarten

Adresse	Name	Bedeutung	Beschreibung	Gesetzt von (wenn gesetzt)	A-CPU-Register D9 [ ] [ ] [ ]	Gültig für:
SD600	Typ der Speicherkarte A		Zeigt den Typ der installierten Speicherkarte A an. 	S (Bei Initialisierung und Entfernen der Speicherkarte)	Neu	Q-CPU außer Q00J-, Q00- und Q01CPU
			Zeigt den Typ der installierten Speicherkarte A an. 	S (Bei Initialisierung und Entfernen der Speicherkarte)	Neu	QnA-CPU
SD602	Kapazität von Laufwerk 1 (RAM)		Die Kapazität von Laufwerk 1 wird in 1-kB-Schritten gespeichert.	S (Bei Initialisierung und Entfernen der Speicherkarte)	Neu	● außer Q00J-, Q00- und Q01CPU
SD603	Kapazität von Laufwerk 2 (ROM)		Die Kapazität von Laufwerk 2 wird in 1-kB-Schritten gespeichert.	S (Bei Initialisierung und Entfernen der Speicherkarte)	Neu	
SD604	Nutzungsbedingungen der Speicherkarte A		● Die Nutzungsbedingungen der Speicherkarte A werden als Bit-Muster gespeichert (EIN wenn in Benutzung). ● Bedeutung dieses Bit-Musters: b0 : BOOT-Vorgang (QBT) b1 : Parameter (QPA) b2 : Operandenkommentare (QCD) b3 : Startwert von Operanden (QDI) b4 : File-R (QDR) b5 : Trace (QTS) b6 : b7 : b8 : b9 : CPU- Fehlerprotokoll (QFD) bA : Trace der Ablaufsprache (QTS) bB : Lokale Variable (QDL) bC : bD : bE : bF :	S (Zustandsänderung)	Neu	Q-CPU außer Q00J-, Q00- und Q01CPU
			● Die Nutzungsbedingungen der Speicherkarte A werden als Bit-Muster gespeichert (EIN wenn in Benutzung). ● Bedeutung dieses Bit-Musters: b0 : BOOT-Vorgang (QBT) b1 : Parameter (QPA) b2 : Operandenkommentare (QCD) b3 : Startwert von Operanden (QDI) b4 : File-R (QDR) b5 : Trace (QTS) b6 : Status-Latch (QTL) b7 : Programm-Trace (QTP) b8 : Simulationsdaten b9 : CPU- Fehlerprotokoll (QFD) b10 : Trace der Ablaufsprache (QTS) b11 : Lokale Variable (QDL) b12 : b13 : b14 : b15 :	S (Zustandsänderung)	Neu	QnA-CPU

(5) Speicherkarten

Adresse	Name	Bedeutung	Beschreibung	Gesetzt von (wenn gesetzt)	A-CPU-Register D9 [ ] [ ] [ ]	Gültig für:
SD620	Typ der Speicherkarte B		● Zeigt den Speichertyp der installierten Speicherkarte B an. 	S (Initialisierung)	Neu	Q-CPU
			● Zeigt den Speichertyp der installierten Speicherkarte B an. 	S (Initialisierung)	Neu	Q2A(S1)-, Q3A-, Q4A-, Q4AR-CPU
SD622	Kapazität von Laufwerk 3 (RAM)		● Die Kapazität des Laufwerks 3 wird in 1 kB Schritten gespeichert Bei der Q-CPU ist dieser Wert wegen des 61 kB RAM fest auf ‘61‘ eingestellt.	S (Initialisierung)	Neu	Q-CPU
			● Die Kapazität des Laufwerks 3 wird in 1 kB Schritten gespeichert.	S (Initialisierung)	Neu	Q2A(S1)-, Q3A-, Q4A-, Q4AR-CPU
SD623	Kapazität von Laufwerk 4 (ROM)		● Die Kapazität des Laufwerks 4 wird in 1 kB Schritten gespeichert.	S (Initialisierung)	Neu	Q-CPU, Q2A(S1)-, Q3A-, Q4A-, Q4AR-CPU
SD624	Nutzungsbedingungen des Laufwerks 3		● Die Nutzungsbedingungen des Laufwerks 3 wird durch Bit 4 angezeigt: Bit 4 = AUS: Nicht verwendet Bit 4 = EIN: Wird zur Speicherung von File-Registern verwendet	S (Zustandsänderung)	Neu	Q00J-, Q00- und Q01CPU
	Nutzungsbedingungen der Laufwerke 3 und 4		● Die Nutzungsbedingungen der Laufwerke 3 und 4 werden als Bit-Muster gespeichert (EIN wenn in Benutzung). ● Die Bedeutung dieses Bit-Musters wird unten beschrieben. b0 : BOOT-Vorgang (QBT) b1 : Parameter (QPA) b2 : Operandenkommentare (QCD) b3 : Startwert von Operanden (QDI) b4 : File-R (QDR) b5 : Trace (QTS) b6 : b7 : b8 : b9 : CPU- Fehlerprotokoll (QFD) bA : Trace der Ablaufsprache (QTS) bB : Lokale Variable (QDL) bC : bD : bE : bF :	S (Zustandsänderung)	Neu	Q-CPU außer Q00J-, Q00- und Q01CPU
	Nutzungsbedingungen der Speicherkarte B		● Die Nutzungsbedingungen der Speicherkarte B werden als Bit-Muster gespeichert (EIN wenn in Benutzung). ● Die Bedeutung dieses Bit-Musters wird unten beschrieben. b0 : BOOT-Vorgang (QBT) b1 : Parameter (QPA) b2 : Operandenkommentare (QCD) b3 : Startwert von Operanden (QDI) b4 : File-R (QDR) b5 : Trace (QTS) b6 : Status-Latch (QTL) b7 : Programm-Trace (QTP) b8 : Simulationsdaten b9 : CPU- Fehlerprotokoll (QFD) b10 : Trace der Ablaufsprache (QTS) b11 : Lokale Variable (QDL) b12 : b13 : b14 : b15 :	S (Zustandsänderung)	Neu	Q2A(S1)-, Q3A-, Q4A-, Q4AR-CPU

(5) Speicherkarten

<table>
<tr><th>Adresse</th><th>Name</th><th>Bedeutung</th><th>Beschreibung</th><th>Gesetzt von (wenn gesetzt)</th><th>A-CPU-Register D9 [ ] [ ] [ ]</th><th>Gültig für:</th></tr>
<tr><td>SD640</td><td>Laufwerk des File-Registers</td><td>Laufwerksnummer</td><td>Speichert die Nummer des Laufwerks, das vom File-Register benutzt wird.</td><td>S (Zustandsänderung)</td><td>Neu</td><td rowspan="6">●</td></tr>
<tr><td>SD641
SD642
SD643
SD644
SD645
SD646</td><td colspan="2">File-Register File-Name</td><td>Speichert die durch Parameter oder QCDSET-Anweisung angegebenen File-Register und File-Name (mit Erweiterung) als ASCII-Code.
b15 b8 b7 b0
SD641: 2. Zeichen | 1. Zeichen
SD642: 4. Zeichen | 3. Zeichen
SD643: 6. Zeichen | 5. Zeichen
SD644: 8. Zeichen | 7. Zeichen
SD645: 1. Zeichen der Erweiterung | 2EH (.)
SD646: 3. Zeichen der Erweiterung | 2. Zeichen der Erweiterung</td><td>S (Zustandsänderung)</td><td>Neu</td></tr>
<tr><td>SD647</td><td colspan="2">Kapazität der File-Register</td><td>Datenkapazität des aktuell ausgewählten File-Registers in 1-k-Wort-Einheiten</td><td>S (Zustandsänderung)</td><td>Neu</td></tr>
<tr><td>SD648</td><td colspan="2">Blocknummer des File-Registers</td><td>Speichert die aktuell ausgewählte File-Register-Blocknummer.</td><td>S (Zustandsänderung)</td><td>D9035</td></tr>
<tr><td>SD650</td><td colspan="2">Kommentarlaufwerk</td><td>Speichert die durch Parameter oder QCDSET-Anweisung angegebene Laufwerksnummer des Kommentarlaufwerks.</td><td>S (Zustandsänderung)</td><td>Neu</td><td rowspan="3">● außer Q00J-, Q00- und Q01CPU</td></tr>
<tr><td>SD651
SD652
SD653
SD654
SD655
SD656</td><td colspan="2">Name des Kommentar-Files</td><td>Speichert den durch Parameter oder die QCDSET-Anweisung angegebenen File-Namen (mit Erweiterung) im ASCII-Code.
b15 b8 b7 b0
SD651: 2. Zeichen | 1. Zeichen
SD652: 4. Zeichen | 3. Zeichen
SD653: 6. Zeichen | 5. Zeichen
SD654: 8. Zeichen | 7. Zeichen
SD655: 1. Zeichen der Erweiterung | 2EH (.)
SD656: 3. Zeichen der Erweiterung | 2. Zeichen der Erweiterung</td><td>S (Zustandsänderung)</td><td>Neu</td></tr>
<tr><td>SD660</td><td rowspan="7">Für den Boot-Vorgang benanntes File</td><td>Laufwerksnummer, auf dem sich das für den Boot-Vorgang benannte File befindet</td><td>Speichert die Nummer des Laufwerks, auf dem sich das File befindet, das für den (*.QBT) Bootvorgang benannt wurde.</td><td>S (Initialisierung)</td><td>Neu</td></tr>
<tr><td>SD661
SD662
SD663
SD664
SD665
SD666</td><td>Name des Files, das für den Boot-Vorgang benannt ist</td><td>Speichert den File-Namen des Files, das für den Boot-Vorgang benannt ist (*.QBT).
b15 b8 b7 b0
SD661: 2. Zeichen | 1. Zeichen
SD662: 4. Zeichen | 3. Zeichen
SD663: 6. Zeichen | 5. Zeichen
SD664: 8. Zeichen | 7. Zeichen
SD665: 1. Zeichen der Erweiterung | 2EH (.)
SD666: 3. Zeichen der Erweiterung | 2. Zeichen der Erweiterung</td><td>S (Initialisierung)</td><td>Neu</td></tr>
</table>

(6) Anweisungsbezogene Register

Adresse	Name	Bedeutung	Beschreibung	Gesetzt von (wenn gesetzt)	A-CPU-Register D9 [ ] [ ] [ ]	Gültig für:
SD705 SD706	Bit-Schema		Während der Blockverabeitung wird SM705 gesetzt. Dies ermöglicht die Nutzung des unter SD705 gespeicherten Bit-Schemas, (bei Verwendung von Doppelworten ist es unter SD705 und SD706 gespeichert), um es auf alle zu verarbeitenden Daten des Blocks anzuwenden.	B	Neu	● außer Q00J-, Q00- und Q01CPU
SD714	Anzahl der freien Kommunikationsanforderungen im Registrationsbereich	0 bis 32	Anzahl der freien Blöcke im Kommunikationsanforderungsbereich für Remote-Sondermodule, die mit einem AJ71PT32-S verbunden sind.	S (während der Ausführung)	M9081	QnA-CPU
SD715 SD716 SD717	Bit-Schema der IMASK-Anweisung	Bit-Schema	Bei Verwendung der IMASK-Anweisung wird das Bit-Schema genutzt. b15 ..... b0 SD715 I15 I14 I13 I12 I11 I10 I9 I8 I7 I6 I5 I4 I3 I2 I1 I0 SD716 I31 I30 I29 I28 I27 I26 I25 I24 I23 I22 I21 I20 I19 I18 I17 I16 SD717 I47 I46 I45 I44 I43 I42 I41 I40 I39 I38 I37 I36 I35 I34 I33 I32	S (während der Ausführung)	Neu	●
SD718 SD719	Akkumulator		Diese Register emulieren die Akkumulatoren der MELSEC A-Serie.	S/B	Neu	
SD720	Programmnummerzuweisung für PLOAD-Anweisung		Dieses Register speichert die Programmnummer, die dem mit einer PLOAD-Anweisung geladenen Programm zugeteilt werden soll. Programmnummern von 1 bis 124 können vergeben werden.	B	Neu	Q-CPU
SD730	Anzahl der freien CC-Link Kommunikationsanforderungen im Registrationsbereich	0 bis 32	Speichert die Anzahl der freien Blöcke im CC-Link Kommunikationsanforderungsbereich für Remote-Sondermodule, die mit einem A(1S)J61QBT61 verbunden sind	S (während der Ausführung)	Neu	QnA-CPU
SD736	PKEY-Eingabe		Dieses Diagnoseregister speichert temporär Tastatureingabedaten in der Weise der PKEY-Anweisung	S (während der Ausführung)	Neu	● außer Q00J-, Q00- und Q01CPU

(6) Anweisungsbezogene Register

<table>
<tr><th>Adresse</th><th>Name</th><th>Bedeutung</th><th>Beschreibung</th><th>Gesetzt von (wenn gesetzt)</th><th>A-CPU-Register D9 [ ] [ ] [ ]</th><th>Gültig für:</th></tr>
<tr><td>SD738
SD739
SD740
SD741
SD742
SD743
SD744
SD745
SD746
SD747
SD748
SD749
SD750
SD751
SD752
SD753
SD754
SD755
SD756
SD757
SD758
SD759
SD760
SD761
SD762
SD763
SD764
SD765
SD766
SD767
SD768
SD769</td><td colspan="2">Mitteilungsspeicher</td><td>Speichert Mitteilungen, die durch die MSG-Anweisung bezeichnet sind.
b15 ← → b8 b7 ← → b0
SD738: 2. Zeichen | 1. Zeichen
SD739: 4. Zeichen | 3. Zeichen
SD740: 6. Zeichen | 5. Zeichen
SD741: 8. Zeichen | 7. Zeichen
SD742: 10. Zeichen | 9. Zeichen
SD743: 12. Zeichen | 11. Zeichen
SD744: 14. Zeichen | 13. Zeichen
SD745: 16. Zeichen | 15. Zeichen
SD746: 18. Zeichen | 17. Zeichen
SD747: 20. Zeichen | 19. Zeichen
SD748: 22. Zeichen | 21. Zeichen
SD749: 24. Zeichen | 23. Zeichen
SD750: 26. Zeichen | 25. Zeichen
SD751: 28. Zeichen | 27. Zeichen
SD752: 30. Zeichen | 29. Zeichen
SD753: 32. Zeichen | 31. Zeichen
SD754: 34. Zeichen | 33. Zeichen
SD755: 36. Zeichen | 35. Zeichen
SD756: 38. Zeichen | 37. Zeichen
SD757: 40. Zeichen | 39. Zeichen
SD758: 42. Zeichen | 41. Zeichen
SD759: 44. Zeichen | 43. Zeichen
SD760: 46. Zeichen | 45. Zeichen
SD761: 48. Zeichen | 47. Zeichen
SD762: 50. Zeichen | 49. Zeichen
SD763: 52. Zeichen | 51. Zeichen
SD764: 54. Zeichen | 53. Zeichen
SD765: 56. Zeichen | 55. Zeichen
SD766: 58. Zeichen | 57. Zeichen
SD767: 60. Zeichen | 59. Zeichen
SD768: 62. Zeichen | 61. Zeichen
SD769: 64. Zeichen | 63. Zeichen</td><td>S (während der Ausführung)</td><td>Neu</td><td>● außer Q00J-, Q00- und Q01CPU</td></tr>
<tr><td>SD774 – SD775</td><td>Grenzwerte für PID-Regelungen</td><td>0: Grenzwert für PID-Regelungen eingestellt
1: Grenzwerte nicht eingestellt</td><td>Die Zuordnung der Grenzwerte zu den einzelnen PID-Regelkreisen ist wie folgt:
b15 … b1 b0
SD774: Regelkreis 16 | bis | Regelkreis 2 | Regelkreis 1
SD775: Regelkreis 32 | bis | Regelkreis 18 | Regelkreis 17</td><td>B</td><td>Neu</td><td>Q-CPU außer Q00J-, Q00- und Q01CPU</td></tr>
<tr><td>SD780</td><td>Verbleibende Anzahl von CC-Link bezogene Anweisungen, die gleichzeitig ausgeführt werden können</td><td>0 bis 32</td><td>In diesem Register wird die verbleibende Anzahl von CC-Link bezogenen Anweisungen abgelegt, die gleichzeitig ausgeführt werden können.</td><td>B</td><td>Neu</td><td>QnA-CPU</td></tr>
<tr><td>SD781 – SD793</td><td>Bit-Schema der IMASK-Anweisung</td><td>Bit-Schema</td><td>Bei Verwendung der IMASK-Anweisung wird das Bit-Schema genutzt.
b15 … b11 b0
SD781: I63 | bis | I59 | I48
SD782: I79 | bis | I65 | I64
bis
SD793: I255 | bis | I241 | I240
Bei einer Q00J-, Q00- oder Q01CPU kann nur der Bereich SD781 (I48) bis SD785 (I127) genutzt werden.</td><td>S (während der Ausführung)</td><td>Neu</td><td>Q-CPU</td></tr>
</table>

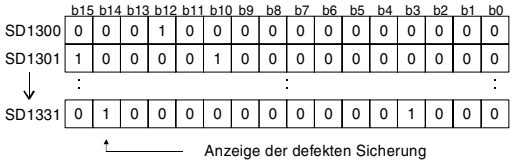
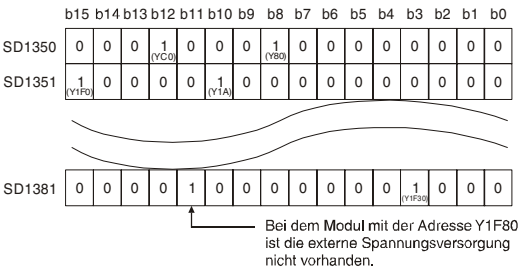
(7) Fehlerbeseitigung

<table>
<tr><th>Adresse</th><th colspan="2">Name</th><th>Bedeutung</th><th>Beschreibung</th><th>Gesetzt von (wenn gesetzt)</th><th>A-CPU-Register D9 [ ] [ ] [ ]</th><th>Gültig für:</th></tr>
<tr><td>SD806
SD807
SD808
SD809
SD810
SD811</td><td colspan="2">Status-Latch
File-Name</td><td></td><td>Der File-Name (mit Erweiterung) wird an dem Zeitpunkt, an dem der Status-Latch ausgeführt wurde, im ASCII-Code gespeichert.
b15 b8 b7 b0
SD806: 2. Zeichen | 1. Zeichen
SD807: 4. Zeichen | 3. Zeichen
SD808: 6. Zeichen | 5. Zeichen
SD809: 8. Zeichen | 7. Zeichen
SD810: 1. Zeichen der Erweiterung | 2EH (.)
SD811: 3. Zeichen der Erweiterung | 2. Zeichen der Erweiterung</td><td>S
(während der Ausführung)</td><td>Neu</td><td rowspan="2">QnA-CPU</td></tr>
<tr><td>SD812
SD813
SD814
SD815
SD816</td><td colspan="2">Status-Latch
Schritt</td><td></td><td>Die Schrittnummer wird an dem Zeitpunkt gespeichert, an dem der Status Latch ausgeführt wurde.
SD812: Muster*
SD813: Block-Nr.
SD814: Schritt-/Transitions-Nr.
SD815: Ablaufschritt-Nr. (L)
SD816: Ablaufschritt-Nr. (H)
*Belegung des Musters:
15 14 … 4 3 2 1 0 ← Bit-Nummer
0 0 … 0 0 * * *
Blockbeschreibung der Ablaufsprache vorhanden (1) / nicht vorhanden (0)
Schrittbeschreibung der Ablaufsprache vorhanden (1) / nicht vorhanden (0)
Transitionsbeschreibung der Ablaufsprache vorhanden (1) / nicht vorhanden (0)</td><td>S
(während der Ausführung)</td><td>D9055
(geändertes Format)</td></tr>
</table>

(8) Latch-Bereich

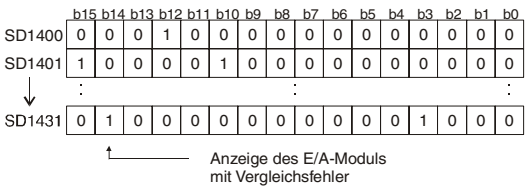
<table>
<tr><th>Adresse</th><th>Name</th><th>Bedeutung</th><th>Beschreibung</th><th>Gesetzt von (wenn gesetzt)</th><th>A-CPU-Register D9 [ ] [ ] [ ]</th><th>Gültig für:</th></tr>
<tr><td>SD900</td><td>Laufwerksnummer während eines Spannungsausfalls</td><td>Laufwerksnummer, auf dem sich das File befindet, auf das während des Spannungsausfall zugegriffen wurde.</td><td>Speichert die Nummer des Laufwerks, auf dem sich das File befindet, auf das während des Spannungsausfalls zugegriffen wurde.</td><td>S (Zustandsänderung)</td><td>Neu</td><td rowspan="26">QnA-CPU</td></tr>
<tr><td>SD901
SD902
SD903
SD904
SD905
SD906</td><td>Name des aktiven Files während eines Spannungsausfalls</td><td>File-Name des Files, auf das während eines Spannungs-ausfalls zugegriffen wurde</td><td>Speichert den File-Namen (mit Erweiterung) des Files, auf das während des Spannungsausfalls zugegriffen wurde, im ASCII-Code.
b15 b8 b7 b0
SD901: 2. Zeichen | 1. Zeichen
SD902: 4. Zeichen | 3. Zeichen
SD903: 6. Zeichen | 5. Zeichen
SD904: 8. Zeichen | 7. Zeichen
SD905: 1. Zeichen der Erweiterung | 2EH (.)
SD906: 3. Zeichen der Erweiterung | 2. Zeichen der Erweiterung</td><td>S (Zustandsänderung)</td><td>Neu</td></tr>
<tr><td>SD910
SD911
SD912
SD913
SD914
SD915
SD916
SD917
SD918
SD919
SD920
SD921
SD922
SD923
SD924
SD925</td><td colspan="2">PKEY-Eingabe</td><td>Speichert nacheinander den eingegebenen PU-Code.
b15 ⟵ b8 b7 ⟵ b0
SD910: 2. Zeichen | 1. Zeichen
SD911: 4. Zeichen | 3. Zeichen
SD912: 6. Zeichen | 5. Zeichen
SD913: 8. Zeichen | 7. Zeichen
SD914: 10. Zeichen | 9. Zeichen
SD915: 12. Zeichen | 11. Zeichen
SD916: 14. Zeichen | 13. Zeichen
SD917: 16. Zeichen | 15. Zeichen
SD918: 18. Zeichen | 17. Zeichen
SD919: 20. Zeichen | 19. Zeichen
SD920: 22. Zeichen | 21. Zeichen
SD921: 24. Zeichen | 23. Zeichen
SD922: 26. Zeichen | 25. Zeichen
SD923: 28. Zeichen | 27. Zeichen
SD924: 30. Zeichen | 29. Zeichen
SD925: 32. Zeichen | 31. Zeichen</td><td>S (Während der Ausführung)</td><td>Neu</td></tr>
</table>

(9) Module mit defekter Sicherung oder fehlender externer Spannungsversorgung

Adresse	Name	Bedeutung	Beschreibung	Gesetzt von (wenn gesetzt)	A-CPU-Register D9 [ ] [ ] [ ]	Gültig für:
SD1300 SD1301 SD1302 SD1303 SD1304 SD1305 SD1306 SD1307 SD1308 SD1309 – SD1330 SD1331	Module mit defekter Sicherung	Das Bit-Muster (16 Bit) zeigt die Module mit defekter Sicherung an 0 : Keine defekte Sicherung 1 : Defekte Sicherung vorhanden	● Die Anzahl der Ausgangsmodule mit defekter Sicherung wird als Bit-Muster von 16 Bit gespeichert. (Wenn die Modulnummer in den Parametern gesetzt ist, wird diese Nummer gespeichert.) ● Es werden auch defekte Sicherungen in Ausgangsmodulen von Remote-Stationen erfasst. ● Nach Austausch der defekten Sicherung wird das entsprechende Bit nicht automatisch rückgesetzt. Das Bit muss durch Rücksetzen der Fehlermeldung gelöscht werden. 	S (Fehler)	D9100 D9101 D9102 D9103 D9104 D9105 D9106 D9107 Neu Neu Neu	● außer Q00J-, Q00- und Q01CPU
SD1350 – SD1381	Externe Spannungsversorgung fehlt	Das Bit-Muster (16 Bit) zeigt die Module, denen die externe Spannungsversorgung fehlt 0 : Externe Spannungsversorgung fehlt 1 : Externe Spannungsversorgung ist vorhanden	Die Adresse der Ausgangsmodule, bei dem die externe Spannungsversorgung fehlt, wird als Bit-Muster (Mit der Einheit 16 Bit) gespeichert. (Wenn die Modulnummer in den Parametern eingestellt ist, wird diese Nummer gespeichert.) 	S (Fehler)	Neu	Q-CPU außer Q00J-, Q00- und Q01CPU

(10) E/A-Modul Vergleich

Adresse	Name	Bedeutung	Beschreibung	Gesetzt von (wenn gesetzt)	A-CPU-Register D9 [ ] [ ] [ ]	Gültig für:
SD1400	E/A-Module mit Vergleichsfehler	Das Bit-Muster (16 Bit), zeigt die Module mit Vergleichsfehler an 0 : Keine E/A-Vergleichsfehler 1 : E/A-Vergleichsfehler vorhanden	● Ist der aktuelle Status eines E/A-Moduls von dem vorgegebenen Status nach Einschalten der Versorgungsspannung verschieden, werden die E/A-Modulinformationen in dem Register gespeichert. (Wenn die Modulnummer in den Parametern gesetzt ist, wird diese Nummer gespeichert.) ● Es werden auch E/A-Modulinformationen erkannt .	S (Fehler)	D9116	● außer Q00J-, Q00- und Q01CPU
SD1401					D9117	
SD1402					D9118	
SD1403					D9119	
SD1404					D9120	
SD1405					D9121	
SD1406					D9122	
SD1407					D9123	
SD1408					Neu	
SD1409 bis SD1430					Neu	
SD1431					Neu	



(11) Zusammenhang zwischen Sonder- (A-Serie) und Diagnoseregistern (Q-Serie/System Q)

Bei der Umstellung von der MELSEC A-Serie zur MELSEC Q-Serie entsprechen die Sonderregister D9000 bis D9255 (A-Serie) den Diagnoseregistern SD1000 bis SD1255 (Q-Serie und System Q).
Diese Diagnoseregister werden alle durch das System gesetzt und können nicht durch ein Benutzerprogramm verändert werden. Benutzer, die diese Register setzen oder rücksetzen wollen, sollten ihre Programme so abändern, dass nur reine Q-/QnA-Diagnoseregister verwendet werden.

Eine Ausnahme bilden die Sonderregister D9200 bis D9255. Da diese Register vom Benutzer beschrieben werden konnten, ist dies auch nach der Umstellung mit den entsprechenden Diagnoseregistern SD1200 bis SD1255 möglich.
Detaillierte Informationen zu den Sonderregistern der A-Serie können den Handbüchern zu den CPUs und dem Netzwerk „MELSECNET“ entnommen werden.

HINWEIS Wenn ein äquivalentes Diagnoseregister für die Q-/QnA-CPU angegeben ist, sollte das Programm geändert und dieses Register verwendet werden. Wenn kein äquivalentes Q-/QnA-Diagnoseregister angegeben ist, kann das Register verwendet werden, das nach der Umstellung angegeben wird.

Liste der Sonderregister und Diagnoseregister

A-CPU Sonderregister	Diagnoseregister nach der Umstellung	Äquivalente System Q-/QnA-Diagnoseregister	Name	Bedeutung	Gültig für
D9000	SD1000	—	Sicherung defekt	Adresse des Moduls mit defekter Sicherung	Q-/QnA-CPU
D9001	SD1001	—	Sicherung defekt	Adresse des Moduls mit defekter Sicherung	
D9002	SD1002	—	Vergleichsfehler mit E/A-Modul	Adresse des Moduls, bei dem der Vergleichsfehler vorliegt	
D9004	SD1004	—	Fehler im Master-Modul eines MELSECNET/MINI S3	Speichert den gesetzten Status in den Parametern (Modul 1 bis 8)	QnA-CPU
D9005	SD1005	—	Häufigkeit der Netzspannungsausfälle	Zähler der Spannungsausfälle	Q-/QnA-CPU
D9008	SD1008	SD0	Fehlererkennung durch Selbstdiagnose	Fehlernummer der Selbstdiagnose	
D9009	SD1009	SD62	Erkennung eines Fehlermerkers	Fehlermerker F zur Kennzeichnung des externen Fehlers	
D9010	SD1010	Keine Funktion bei einer System Q- oder QnA-CPU	Fehlerhafte Programmschrittnummer	Schrittnummer, an der der Fehler aufgetreten ist	
D9011	SD1011				
D9014	SD1014		Kennzeichnung der Ein-/Ausgangs-verarbeitung	Nummer des E/A-Verarbeitungsmodus	
D9015	SD1015	SD203	Betriebszustand der CPU	Betriebszustand der CPU	
D9016	SD1016	Keine Funktion bei einer System Q- oder QnA-CPU	Programmkennung	Kennung des gerade abgearbeiteten Ablaufprogramms	
D9017	SD1017	SD520	Zykluszeit	Minimale Zykluszeit (10-ms-Einheiten)	

Liste der Sonderregister und Diagnoseregister

A-CPU Sonderregister	Diagnoseregister nach der Umstellung	Äquivalente System Q-/QnA-Diagnoseregister	Name	Bedeutung	Gültig für
D9018	SD1018	SD524	Zykluszeit	Zykluszeit (10-ms-Einheiten)	Q-/QnA-CPU
D9019	SD1019	SD526	Zykluszeit	Maximale Zykluszeit (10-ms-Einheiten)	
D9020	SD1020	Keine Funktion bei einer System Q- oder QnA-CPU	Konstante Zykluszeit	Sollwert der konstanten Zykluszeit (Benutzereinstellung in 10-ms-Schritten)	
D9021	SD1021	—	Zykluszeit	Zykluszeit (1-ms-Einheiten)	
D9022	SD1022	SD412	1-Sekunden-Zähler	1-Sekunden-Zähler	
D9025	SD1025	SD210	Uhrdaten	Uhrdaten (Jahr, Monat)	
D9026	SD1026	SD211		Uhrdaten (Tag, Stunde)	
D9027	SD1027	SD212		Uhrdaten (Minute, Sekunde)	
D9028	SD1028	SD213		Uhrdaten (Wochentag)	
D9035	SD1035	SD648	Erweitertes File-Register	Nummer des benutzten Blocks	
D9036	SD1036	Keine Funktion bei einer System Q- oder QnA-CPU	Operandenadresse der erweiterten File-Register	Operandenadressen der erweiteten File-Register zum direkten Lesen und Schreiben	
D9037	SD1037				
D9038	SD1038	SD207	Anzeigepriorität des LED-Displays	Priorität 1 bis 4	
D9039	SD1039	SD208		Priorität 5 bis 7	
D9044	SD1044	Keine Funktion bei einer System Q- oder QnA-CPU	Für das Sampling Trace	Schritt oder Zeit während des Sampling-Trace	
D9049	SD1049		Arbeitsbereich für Programme der Ablaufsprache	Blockadresse der erweiterten File-Register	
D9050	SD1050		Fehlernummer des Programms der Ablaufsprache	Der Fehlercode, der durch das Programm der Ablaufsprache erzeugt wurde	
D9051	SD1051		Block-Fehler	Adresse des Blocks, in dem der Fehler aufgetreten ist	
D9052	SD1052		Schritt-Fehler	Schrittnummer, in dem der Fehler aufgetreten ist	
D9053	SD1053		Transition Fehler	Transitionsbedingungen, in denen der Fehler aufgtreten ist	
D9054	SD1054		Ablaufschritt-Fehler	Ablaufschrittnummer, in dem der Fehler aufgetreten ist	
D9055	SD1055	SD812	Status Latch	Schrittnummer des Status-Latch	
D9060	SD1060	SD392	Software Version	Version der System-Software	QnA-CPU
D9072	SD1072	Keine Funktion bei einer System Q- oder QnA-CPU	PC-Kommunikationskontrolle	Datenkontrolle durch das Computer-Link-Modul	Q-/QnA-CPU
D9081	SD1081	SD714	Anzahl der freien Kommunikationsanforderungen im Registrationsbereich	Anzahl der freien Kommunikationsanforderungen im Registrationsbereich	QnA-CPU

Liste der Sonderregister und Diagnoseregister

A-CPU Sonderregister	Diagnoseregister nach der Umstellung	Äquivalente System Q-/QnA-Diagnoseregister	Name	Bedeutung	Gültig für
D9085	SD1085	Keine Funktion bei einer System Q- oder QnA-CPU	Register für die Zeiteinstellung des Überprüfungsintervalles	Voreingestellter Wert 10s	Q-/QnA-CPU
D9090	SD1090	Keine Funktion bei einer System Q- oder QnA-CPU	AnN: Adressbereich für eine Unterroutine des Mikrocomputer-Programms AnA: Ohne Bedeutung QnA: Gesamtzahl der Sondermodule	AnN: Adressbereich für eine Unterroutine des Mikrocomputer-Programms AnA: Ohne Bedeutung QnA: Gesamtzahl der Sondermodule	Q-/QnA-CPU
D9091	SD1091	Keine Funktion bei einer System Q- oder QnA-CPU	Detaillierter Fehlercode	Detaillierter Fehlercode der Selbstdiagnosefunktion	
D9094	SD9094	SD251	Kopfadresse eines auszuwechselnden E/A-Moduls	Kopfadresse eines auszuwechselnden E/A-Moduls	
D9100	SD1100	—	Sicherungsdefekt eines Moduls	Bit-Muster des Moduls, dessen Sicherung defekt ist.	
D9101	SD1101				
D9102	SD1102				
D9103	SD1103				
D9104	SD1104				
D9105	SD1105				
D9106	SD1106				
D9107	SD1107				
D9108	SD1108	—	Einstellung der Überwachungszeit für Schritttransfer	Einstellung der Überwachungszeit (1 bis 255 s) jeweils im niederwertigen Byte und Angabe des Fehlermerkers im höherwertigen Byte.	
D9109	SD1109				
D9110	SD1110				
D9111	SD1111				
D9112	SD1112				
D9113	SD1113				
D9114	SD1114				
D9115	SD1115				
D9116	SD1116	—	E/A-Modul mit Vergleichsfehler	Bit-Muster des Moduls mit dem E/A-Vergleichsfehler	
D9117	SD1117				
D9118	SD1118				
D9119	SD1119				
D9120	SD1120				
D9121	SD1121				
D9122	SD1122				
D9123	SD1123				
D9124	SD9124	SD63	Summe der Fehlermerker	Summe der Fehlermerker	

Liste der Sonderregister und Diagnoseregister

A-CPU Sonderregister	Diagnoseregister nach der Umstellung	Äquivalente System Q-/QnA-Diagnoseregister	Name	Bedeutung	Gültig für
D9125	SD9125	SD64	Nummer der Fehlermerker	Nummer der Fehlermerker	Q-/QnA-CPU
D9126	SD9126	SD65			
D9127	SD9127	SD66			
D9128	SD9128	SD67			
D9129	SD9129	SD68			
D9130	SD9130	SD69			
D9131	SD9131	SD70			
D9132	SD9132	SD71			
D9200	SD1200	—	Resultate der LRDP-Verarbeitung	0 : Normales Ende 2 : Einstellungsfehler der LRDP-Anweisung 3 : Fehler in angesprochener Station 4 : LRDP-Ausführung in angesprochener Station nicht möglich	QnA-CPU
D9201	SD1201	—	Resultate der LWTP-Verarbeitung	0 : Normales Ende 2 : Einstellungsfehler der LWTP-Anweisung 3 : Fehler in angesprochener Station 4 : LWTP-Ausführung in angesprochener Station nicht möglich	
D9202	SD1202	—	Verbindungstyp der lokalen Station	Speichert Bedingungen für 16 Adressen (1 – 16)	
D9203	SD1203	—		Speichert Stationen (17 – 32)	
D9241	SD1241	—		Speichert Stationen (33 – 48)	
D9242	SD1242	—		Speichert Stationen (49 – 64)	
D9204	SD1204	—	Verbindungsstatus	0: Datenübertragung über Vorwärtsschleife 1: Datenübertragung über Rückwärtsschleife 2: Datenübertragung sowohl vorwärts als auch rückwärts 3: Schleife ausschließlich über Vorwärtsschleife 4: Schleife ausschließlich über Rückwärtsschleife 5: Datenübertragung nicht möglich	
D9205	SD1205	—	Ausführende Station der Rückleitung	Station, über die die Vorwärts-Rückwärtsleitung erfolgt	
D9206	SD1206	—	Ausführende Station der Rückleitung	Station, über die die Vorwärts-Rückwärtsleitung erfolgt	
D9207	SD1207	—	Zykluszeit im Netzwerk	Maximalwert	
D9208	SD1208	—		Minimalwert	
D9209	SD1209	—		Ist-Wert	
D9210	SD1210	—	Anzahl der Wiederholungen	Wird als kumulierter Wert gespeichert	
D9211	SD1211	—	Zähler der Leitungswechsel	Wird als kumulierter Wert gespeichert	

Liste der Sonderregister und Diagnoseregister

A-CPU Sonderregister	Diagnoseregister nach der Umstellung	Äquivalente System Q-/QnA-Diagnoseregister	Name	Bedeutung	Gültig für
D9212	SD1212	—	Betriebszustand der lokalen Station	Zustand der Stationen 1 – 16	QnA-CPU
D9213	SD1213	—		Zustand der Stationen 17 – 32	
D9214	SD1214	—		Zustand der Stationen 33 – 48	
D9215	SD1215	—		Zustand der Stationen 49 – 64	
D9216	SD1216	—	Fehlererkennung der lokalen Station	Zustand der Stationen 1 – 16	
D9217	SD1217	—		Zustand der Stationen 17 – 32	
D9218	SD1218	—	Fehlererkennung der lokalen Station	Zustand der Stationen 33 – 48	
D9219	SD1219	—		Zustand der Stationen 49 – 64	
D9220	SD1220	—	Parameter einer lokalen Station, E/A-Zuordnung einer Remote-E/A-Station ist fehlerhaft	Zustand der Stationen 1 – 16	
D9221	SD1221	—		Zustand der Stationen 17 – 32	
D9222	SD1222	—		Zustand der Stationen 33 – 48	
D9223	SD1223	—		Zustand der Stationen 49 – 64	
D9224	SD1224	—	Initialisierungsübertragung zu lokalen oder Remote-E/A-Stationen	Zustand der Stationen 1 – 16	
D9225	SD1225	—		Zustand der Stationen 17 – 32	
D9226	SD1226	—		Zustand der Stationen 33 – 48	
D9227	SD1227	—		Zustand der Stationen 49 – 64	
D9228	SD1228	—	Fehler in einer Remote-E/A- oder lokalen Station	Zustand der Stationen 1 – 16	
D9229	SD1229	—		Zustand der Stationen 17 – 32	
D9230	SD1230	—		Zustand der Stationen 33 – 48	
D9231	SD1231	—		Zustand der Stationen 49 – 64	
D9232	SD1232	—	Schleifenfehler einer lokalen oder Remote-E/A-Station	Zustand der Stationen 1 – 8	
D9233	SD1233	—		Zustand der Stationen 9 – 16	
D9234	SD1234	—		Zustand der Stationen 17 – 24	
D9235	SD1235	—		Zustand der Stationen 25 – 32	
D9236	SD1236	—		Zustand der Stationen 33 – 40	
D9237	SD1237	—		Zustand der Stationen 41 – 48	
D9238	SD1238	—		Zustand der Stationen 49 – 56	
D9239	SD1239	—		Zustand der Stationen 57 – 64	
D9240	SD1240	—	Zähler der Übertragungsfehler	Zustand der Stationen 1 – 16	
D9243	SD1243	—	Stationsnummer der lokalen oder Remote-E/A-Station	Zustand der Stationen 17 – 32	
D9244	SD1244	—	Anzahl der angeschlossenen Slave-Stationen im Netzwerk	Speichert die Summe der Slave-Stationen	
D9245	SD1245	—	Zähler der Übertragungsfehler	Speichert die Summe der Übertragungsfehler	

Liste der Sonderregister und Diagnoseregister

A-CPU Sonderregister	Diagnoseregister nach der Umstellung	Äquivalente System Q-/QnA-Diagnoseregister	Name	Bedeutung	Gültig für
D9248	SD1248	—	Betriebszustand der lokalen Stationen	Zustand der Stationen 1 — 16	QnA-CPU
D9249	SD1249	—		Zustand der Stationen 17 — 32	
D9250	SD1250	—		Zustand der Stationen 33 — 48	
D9251	SD1251	—		Zustand der Stationen 49 — 64	
D9252	SD1252	—	Fehler in einer lokalen Station	Zustand der Stationen 1 — 16	
D9253	SD1253	—		Zustand der Stationen 17 — 32	
D9254	SD1254	—		Zustand der Stationen 33 — 48	
D9255	SD1255			Zustand der Stationen 49 — 64	

A.5.2 Sonderregister (Nur MELSEC A-Serie)

Sonderregister sind Datenregister für spezifische Anwendungsfälle in der CPU. Mit Ausnahme der mit ❷ gekennzeichneten Register können Sonderregister nicht beliebig mit Daten beschrieben werden. Es lassen sich folgende Unterscheidungen treffen:

- Sonderregister, die automatisch durch die CPU beschrieben werden und vom Anwender nur gelesen (und zurückgesetzt) werden können.
- Sonderregister, in die unter bestimmten Voraussetzungen Daten geschrieben werden können.

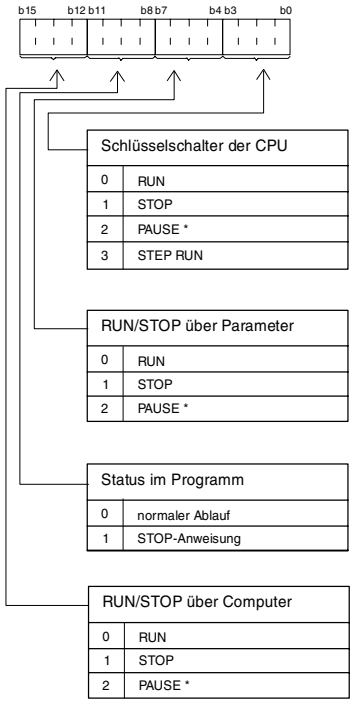
Der Einsatzbereich der Sonderregister in einem Ablaufprogramm ist entsprechend zu prüfen.

Die folgende Tabelle zeigt eine Übersicht der Sonderregister und ihre Verwendungszwecke.

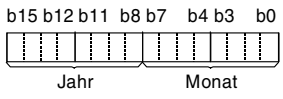
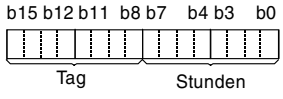
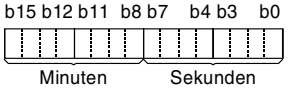
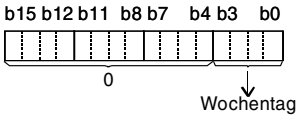
Liste der Sonderregister (MELSEC A-Serie)

Adresse	Name	Bedeutung	Beschreibung	CPU
D9000	Sicherung defekt	Adresse des Moduls, dessen Sicherung defekt ist	Wird bei einem oder mehreren Modul(en) im Netzwerk ein Defekt der Sicherung erkannt, wird die Adresse des ersten erkannten Moduls in dieses Register geschrieben. Bei mehreren Modulen wird die niedrigste Adresse in das Register geschrieben (sind beispielsweise die Sicherungen der Ausgangsmodule mit dem Adressenbereich Y50 bis 6F defekt, wird in D9000 eine „50“ geschrieben). Die Prüfung erfolgt auch für Ausgangsmodule von Remote-E/A-Stationen. Das Speichern der Moduladresse und somit auch die Anzeige über ein Programmiergerät erfolgen hexadezimal. Der Inhalt des Registers wird gelöscht, sobald die Register D9100 bis D9107 auf „0“ gesetzt sind.	
D9002	Vergleichsfehler mit EA-Modul	Adresse des Moduls, bei dem der Vergleichsfehler vorliegt	Unterscheidet sich nach dem Einschalten der Versorgungsspannung der aktuelle Status eines E/A-Moduls von dem vorgegebenen Status, wird in das Register D9002 die Adresse des Moduls geschrieben, bei dem der Vergleichsfehler zuerst erkannt wurde. Die Prüfung erfolgt auch für Ausgangsmodule von Remote-E/A-Stationen. Die Moduladresse wird hexadezimal abgelegt (die Form der Adressenspeicherung entspricht der von D9000). Der Inhalt des Registers wird gelöscht, sobald die Register D9116 bis D9123 auf „0“ gesetzt sind.	
❶ D9004	Fehler im Master-Modul eines MELSECNET/MINI-S3	Fehlerstatus	Das Register speichert nach dem Auftreten eines Fehlers im MELSECNET/MINI-S3 den Fehlerstatus des Master-Moduls im MELSECNET/MINI-S3 (AJ71PT32(S3)). Hinweise zum Bit-Muster und Fehlerstatus sind der Bedienungsanleitung zum MELSECNET/MINI-S3 zu entnehmen.	Nur für AnA-, AnAS-, AnU-CPUs
❶ D9005	Häufigkeit der Netzspannungs-abfälle	Zähler der Abfallzeiten	In dieses Register wird bei jedem Spannungsabfall, bei dem die Nennspannung während des Betriebs um mehr als 20 % abfällt, eine „1“ addiert. Der Wert wird in binärer Form abgelegt.	
❶ D9008	Fehlererkennung durch Selbstdiagnose	Fehlernummer	Hat die CPU mit Hilfe der Selbstdiagnosefunktion einen Fehler erkannt, wird in dieses Register der entsprechende Fehlercode (in binärer Form) geschrieben.	Nur für AnS-CPUs, A2C-CPU

Liste der Sonderregister (MELSEC A-Serie)

Adresse	Name	Bedeutung	Beschreibung	CPU
D9009	Erkennung eines Fehlermerkers	Fehlermerker F zur Kennzeichnung des externen Fehlers	Sobald ein Fehlermerker (F0 bis F255) durch eine OUT F- oder SET F-Anweisung gesetzt ist, wird die zuerst erkannte Fehlernummer als Binärwert in D9009 geschrieben. Der Registerinhalt kann mit Hilfe einer RST F- oder LEDR-Anweisung gelöscht werden. Sollte eine weitere F-Nummer zur Fehlererkennung vorliegen, wird diese mit der Löschanweisung von D9009 in das Register geschrieben.	(Nicht für A3N-CPU, A3M-CPU, A3A-CPU, A3H-CPU)
			Sobald ein Fehlermerker (F0 bis F255) durch eine OUT F- oder SET F-Anweisung gesetzt ist, wird die zuerst erkannte Fehlernummer als Binärwert in D9009 geschrieben. Der Registerinhalt kann mit Hilfe einer RST F- oder LEDR-Anweisung oder durch Einschalten des Schalters „INDICATOR RESET“ gelöscht werden. Sollte eine weitere Fehlermerkernummer zur Fehlererkennung vorliegen, wird diese mit der Löschanweisung von D9009 in das Register geschrieben.	Nur für A3N-CPU, A3M-CPU, A3A-CPU, A3H-CPU
D9010	Nummer des fehlerhaften Programmschrittes	Programmschrittnummer, an der der Fehler aufgetreten ist	Das Register speichert nach dem Auftreten eines Fehlers, der während der Ausführung einer Applikationsanweisung aufgetreten ist, die Programmschrittnummer der Fehlerstelle als Binärwert. Nach jedem erneuten Auftreten eines Fehlers wird der Inhalt von D9010 aktualisiert.	(Nicht für A3H-CPU, A3M-CPU)
❶ D9011	Nummer des fehlerhaften Programmschrittes	Programmschrittnummer, an der der Fehler aufgetreten ist	Das Register speichert nach dem Auftreten eines Fehlers, der während der Ausführung einer Applikationsanweisung aufgetreten ist, die Programmschrittnummer der Fehlerstelle als Binärwert. Da die Speicherung in D9011 zum gleichen Zeitpunkt erfolgt, in dem M9011 gesetzt wird, ist ein Rücksetzen des Registerinhalts auch erst nach dem Rücksetzen von M9011 möglich.	
D9014	Kennzeichnung der Ein-/Ausgangsverarbeitung	Verarbeitungsart	Die E/A-Verarbeitung wird entsprechend dem gespeicherten Registerwert durchgeführt. 0: Direktverarbeitung der E/A-Signale 1: Eingänge nach dem Prozessabbild; Ausgänge in der Direktverarbeitung 3: Ein- und Ausgänge nach dem Prozessabbild	(Nicht für A3H-CPU, A3M-CPU, An-CPUs)
D9015	Betriebszustand der CPU	Betriebsart	Der Betriebsart, in der sich die CPU befindet, wird wie folgt in D9015 gespeichert:  * Die Betriebsart PAUSE muss über M9040 freigegeben sein. Ist M9040 nicht gesetzt, kann die CPU nicht von RUN auf PAUSE umgeschaltet werden.	

Liste der Sonderregister (MELSEC A-Serie)

Adresse	Name	Bedeutung	Beschreibung	CPU
D9016	Speichereinstellung	0: ROM 1: RAM 2: EEPROM	In D9016 werden die Informationen über den ausgewählten Speicherbaustein gespeichert. Der Wert 0, 1 oder 2 wird als Binärwert abgelegt.	Nur für A1-CPU, A1N-CPU
	Programmkennung	0: Hauptprogramm (ROM) 1: Hauptprogramm (RAM) 2: Unterprogramm (RAM)	D9016 dient als Kennung des Speichers, von dem das Ablaufprogramm gerade abgearbeitet wird. Der Wert 0, 1 oder 2 wird als Binärwert abgelegt. Der Wert 2 ist nur bei einer A3N-, A3M-, A3A- oder A3H-CPU möglich.	(Nicht für A1-CPU, A1N-CPU)
D9017	Zykluszeit	Minimale Zykluszeit (Einheit: 10 ms)	In D9017 wird die minimale Programmzykluszeit als Binärwert gespeichert. Ist die tatsächliche Zykluszeit geringer als der in D9017 gespeicherte Wert, wird der alte Wert nach Ausführung der END-Verarbeitung durch den neuen Wert überschrieben.	
D9018	Zykluszeit	Zykluszeit (Einheit: 10 ms)	In D9018 wird die Programmzykluszeit als Binärwert gespeichert. Der alte Wert wird dabei nach jeder END-Verarbeitung durch den neuen Wert ersetzt.	
D9019	Zykluszeit	Maximale Zykluszeit (Einheit: 10 ms)	In D9019 wird die maximale Programmzykluszeit als Binärwert gespeichert. Ist die tatsächliche Zykluszeit größer als der in D9017 gespeicherte Wert, wird der alte Wert nach Ausführung der END-Verarbeitung durch den neuen Wert überschrieben.	
❷ D9020	Konstante Programmzykluszeit	Sollwert der konstanten Zykluszeit	In D9020 wird der Sollwert der konstanten Zykluszeit des Ablaufprogramms geschrieben. Der Wert gibt die Zykluszeit mit einem Inkrement von 10 ms vor. 0: keine Einstellung 1 bis 200: Das Programm wird in konstanten Intervallen von (Registerwert) x 10 ms ausgeführt.	(Nicht für A-CPUs)
D9021	Zykluszeit	Zykluszeit (Einheit: 1 ms)	Die Programmzykluszeit wird als Binärwert gespeichert. Der alte Wert wird dabei nach jeder END-Verarbeitung durch den neuen Wert ersetzt.	Nur für AnA-, AnAS-, AnU-CPUs
D9022	1-Sekunden-Zähler	Auf-/Abwärtszählwert	Mit Beginn des RUN-Betriebs der CPU beginnt der Zähler im Sekundentakt zu zählen. Der Zähler zählt von 0 bis 32767 aufwärts und anschließend bis -32768 abwärts.	Nur für AnA-, AnAS-, AnU-CPUs
❷ D9025	Uhrdaten	Jahr und Monat	Das Register speichert Jahr und Monat als BCD-Wert wie folgt : Beispiel: 1992, Juli = **9207** 	Nur für AnA-CPUs, AnU-CPUs, AnN-CPUs, A1S-CPU
❷ D9026		Tag und Stunde	Das Register speichert Tag und Stunde als BCD-Wert wie folgt: Beispiel 31ster, 10 Uhr = **3110** 	Nur für AnA- AnAS-, AnU-CPUs, AnN-CPUs, AnS-CPUs
❷ D9027		Minute und Sekunde	Das Register speichert Minute und Sekunde als BCD-Wert wie folgt: Beispiel 35 min., 48 s = **3548** 	Nur für AnA-, AnAS-, AnU-CPUs, AnN-CPUs, AnS-CPUs
❷ D9028		Wochentag	Das Register speichert den Wochentag als BCD-Wert wie folgt (0=Sonnntag, 1=Montag, 2=Dienstag usw.): 	Nur für AnA-, AnAS-, AnU-CPUs, AnN-CPUs, AnS-CPUs

Liste der Sonderregister (MELSEC A-Serie)

<table>
<tr><th>Adresse</th><th>Name</th><th>Bedeutung</th><th>Beschreibung</th><th>CPU</th></tr>
<tr><td>D9021
D9022
D9023
D9024
D9025
D9026
D9027
D9028
D9029
D9030
D9031
D9032
D9033
D9034</td><td>Stationsnummer der Remote-Sondermodule</td><td>1 bis 61</td><td>In diese Register werden die Stationsnummern der mit der A2C-CPU verbundenen Remote-Sondermodule geschrieben. Die Reihenfolge der Eintragungen muss nicht mit der tatsächlichen Reihenfolge im Netzwerk übereinstimmen.

Datenstruktur:

D9021 | Bereich für Remote-Sondermodul Nr.1
D9022 | Bereich für Remote-Sondermodul Nr.2
:
:
:
:
D9032 | Bereich für Remote-Sondermodul Nr.12
D9033 | Bereich für Remote-Sondermodul Nr.13
D9034 | Bereich für Remote-Sondermodul Nr.14</td><td>Nur für A2C-CPU</td></tr>
<tr><td>D9035</td><td>Attribute der Remote-Sondermodule</td><td>0: Standardprotokoll
1: ohne Protokoll</td><td>Das Register enthält die Attribute eines jeden im A2C-Netzwerk befindlichen Sondermoduls, gekennzeichnet durch „0“ oder „1“ (nähere Hinweise enthält die A2C-Hardware-Beschreibung).

Datenstruktur:

b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
Anzeige des Moduls mit Sicherungsdefekt

Bit b0 gibt das Attribut für Sondermodul 1, Bit b1 für Modul 2, Bit b3 für Modul 3 usw. an.</td><td>Nur für A2C-CPU</td></tr>
<tr><td>D9036</td><td>Summe der Stationen</td><td>1 bis 64</td><td>In das Register wird die Summe der im A2C-Netzwerk befindliche Stationen (1 bis 64) der Remote-E/A- und Sondermodule geschrieben.</td><td>Nur für A2C-CPU</td></tr>
<tr><td>D9036
D9037</td><td>Operanden-adressen der erweiterten File-Register</td><td>Operandenadresse</td><td>Operandenadressen der erweiterten File-Register zum direkten Lesen und Schreiben als Doppelwort in D9036 und D9037 als Binärwert. Die Adressen müssen, beginnend mit R0 von Block Nr.0, fortlaufend festgelegt werden. Nähere Hinweise enthält die entsprechende Hardware-Beschreibung der MELSEC A-Serie.</td><td>Nur für AnA-, AnAS-, AnU-CPU</td></tr>
<tr><td>D9038
D9039</td><td>Anzeigenpriorität der Fehler des LED-Display</td><td>Prioritätsstufe 1 bis 4
Prioritätsstufe 5 bis 7</td><td>In die vorliegenden Register wird die Anzeigenpriorität der LED-Display entsprechend der Fehlernummer geschrieben. Die Prioritätsstufe der Fehleranzeige kann durch Ändern der Bitfolge im Register vorgegeben werden.

Datenstruktur:

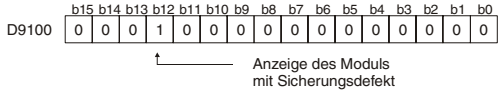
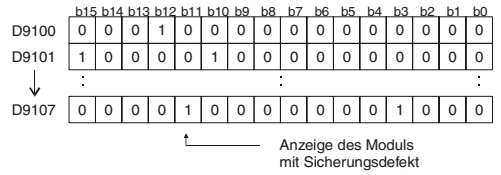
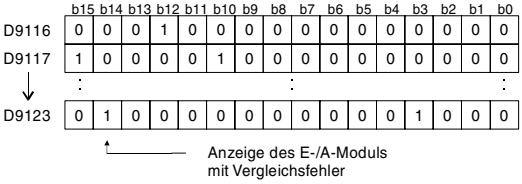
Priorität 4 | Priorität 3 | Priorität 2 | Priorität 1
Priorität 0

Weitere Details enthält die Hardware-Beschreibung der A2C-, AnA- oder AnU-Serie.</td><td>Nur für AnA-, AnAS-, AnU-CPUs, A2C-CPU, AnS-CPUs</td></tr>
<tr><td>D9055</td><td>Schrittnummer des Status Latch</td><td>Schrittnummer</td><td>Speicherung der Schrittnummer nach Ausführung eines Status Latchs.</td><td>Nur für AnA-, AnAS-, AnU-CPUs</td></tr>
</table>

Liste der Sonderregister (MELSEC A-Serie)

Adresse	Name	Bedeutung	Beschreibung	CPU
D9056 bis D9059	Kennung einer fehlerhaften Station	0: Normal 1: Fehler	Tritt ein Fehler in der Kommunikation mit einem Remote-Modul auf, wird das zugehörige Bit dieser fehlerhaften Station im Register auf „1“ gesetzt. Das entsprechende Bit wird erst dann auf „1“ gesetzt, wenn eine Kommunikation auch nach der in D9174 vorgegebenen Anzahl von Wiederholungsversuchen erfolglos war. Das Bit bleibt auch dann gesetzt, wenn der Fehler behoben und die Station wieder angegliedert wurde. Datenstruktur: b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 D9056: 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 D9057: 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 D9058: 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 D9059: 64 63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 Stationsnummern Das entsprechende Bit wird auch dann auf „1“ gesetzt, wenn die Sicherung eines Remote-E/A-Moduls defekt ist.	Nur für A2C-CPU
D9061	Kommunikations-fehler	0: Normal 1: Fehler in der Initialisierung 2: Leitungsfehler	Nach dem Setzen von M9061 (fehlerhafte Kommunikation mit einem Remote-Modul) wird in D9061 die Nummer des Fehlercodes geschrieben. Eine „1“ bedeutet, dass in den Initialisierungsdaten ein Fehler vorliegt (falsche Summe der Stationen oder falsche Anzahl von Wiederholungsversuchen). Bei einer „2“ liegt ein Leitungsdefekt vor, oder eines der Remote-Module ist nicht eingeschaltet.	Nur für A2C-CPU
D9072	PC-Kommunika-tionskontrolle	Datenkontrolle durch das Computer-Link-Modul	Selbsttest-Kontrollregister des Computer-Link-Moduls zum Schreiben/Lesen für die Datenübertragung.	Nur für AnA-, AnAS-, AnU-CPUs
D9073	Uhrdaten	Jahr und Monat	Die Funktion entspricht der von Sonderregister D9035.	Nur für A2C-CPU C24(-PRF)
D9074	Uhrdaten	Tag und Stunde	Die Funktion entspricht der von Sonderregister D9036.	
D9075	Uhrdaten	Minute und Sekunde	Die Funktion entspricht der von Sonderregister D9037.	
D9076	Uhrdaten	Wochentag	Die Funktion entspricht der von Sonderregister D9038.	
D9081	Anzahl der Kommunikations-anforderungen mit einem Remote-Sondermodul	0 bis 32	In D9081 wird die Anzahl der Kommunikationsanforderungen, die über FROM-/TO-Anweisungen mit einem Remote-Sondermodul erfolgen sollen, gespeichert. Die Anzahl der Anforderungen wird jedesmal, wenn die Kommunikation mit einem Sondermodul beendet ist, um 1 verringert.	Nur für AnA-, AnAS-, AnU-CPUs, A2C-CPU
D9082	Stationsnummer der letzten Station im Netzwerk	Nummer der letzten Station	In das Register D9082 wird die Stationsnummer des letzen im A2C-Netzwerk befindlichen Remote-Moduls geschrieben.	Nur für A2C-CPU
D9090	Adressenbereich für eine Unterroutine des Mikrocomputer-Programms	Abhängig von dem verwendetem Mikrocomputer-Programm	Nähere Hinweise sind dem entsprechenden Mikrocomputer-Programmpaket zu entnehmen.	(Nicht für AnA-, AnAS-, AnU-CPUs
D9091	Anweisungsfehler	Fehlercode des Anweisungsfehlers	Im Register wird ein Fehlercode gespeichert, der die Ursache eines Anweisungs-fehlers spezifiziert.	Nur für AnA- AnAS, AnU-CPUs
D9091	Fehlercode für Mikrocomputer-Programm	Abhängig von dem verwendeten Mikrocomputer-Programm	Nähere Hinweise sind dem entsprechenden Mikrocomputer-Programmpaket zu entnehmen.	(Nicht für AnA- AnAS, AnU-CPUs)
❷❸ D9094	Adresse eines auszuwechselnden E/A-Moduls	Startadresse des E/A-Moduls	Das Register D9094 speichert die ersten beiden Stellen der Startadresse eines E/A-Moduls, das während des Online-Betriebs aus dem Baugruppenträger entfernt bzw. eingesetzt wird (z.B. Eingangsmodul X2F0 = H2F). Die Speicherung erfolgt als Binärwert.	Nur für AnN-CPUs, AnA- AnAS-, AnU-CPUs

Liste der Sonderregister (MELSEC A-Serie)

Adresse	Name	Bedeutung	Beschreibung	CPU
❶ D9100	Sicherungsdefekt eines Moduls	Bit-Muster der Moduladressen	Wird bei einem Modul eine defekte Sicherung erkannt, wird in dem Bit-Muster von D9100 das Bit auf „1“ gesetzt, das dem entsprechenden E/A-Modul zugeordnet ist (Zuordnung über Parameter).  Nach Austausch der Sicherung und Wiederinbetriebnahme des Moduls muss das Rücksetzen des Bits über das Ablaufprogramm erfolgen.	Nur für A1S-CPU
❶ D9100 ❶ D9101 ❶ D9102 ❶ D9103 ❶ D9104 ❶ D9105 ❶ D9106 ❶ D9107	Sicherungsdefekt eines Moduls	Bit-Muster der Moduladressen	Wird bei einem Modul eine defekte Sicherung erkannt, wird die Adresse dieses Moduls in das Bit-Muster von D9100 bis D9107 geschrieben (in Einheiten zu 16). Das Bit-Muster entspricht den über Parameter voreingestellten E/A-Adressen.  Nach Wiederherstellung des Normalzustands werden die entsprechenden Bits nicht automatisch auf „0“ gesetzt. Das Rücksetzen muss daher über das Ablaufprogramm erfolgen.	(Nicht für A1S-CPU, bei AnS-CPUs nur D9100 bis D9103)
❶ D9116	E/A-Module mit Vergleichsfehler	Bit-Muster der Moduladressen	Ist der aktuelle Status eines E/A-Moduls von dem vorgegebenen Status nach Einschalten der Versorgungsspannung verschieden, wird in dem Bit-Muster von D9116 das Bit auf 1 gesetzt, das dem entsprechenden E/A-Modul zugeordnet ist (Zuordnung über Parameter). Nach Wiederherstellung des Normalzustandes muss das Rücksetzen des Bits auf 0 über das Abalufprogramm erfolgen.	Nur für A1S-CPU
D9116 D9117 D9118 D9119 D9120 D9121 D9122 D9123	E/A-Module mit Vergleichsfehler	Bit-Muster der Moduladressen	Ist der aktuelle Status eines E/A-Moduls von dem vorgegebenen Status nach Einschalten der Versorgungsspannung verschieden, wird in dem Bit-Muster von D9116 bis D123 das Bit auf „1“ gesetzt, das dem entsprechenden E/A-Modul zugeordnet ist. Das in Einheiten zu 16 Bits zusammengesetzte Bit-Muster entspricht den über Parameter voreingestellten E/A-Adressen.  Nach Wiederherstellung des Normalzustands werden die entsprechenden Bits nicht automatisch auf „0“ gesetzt. Das Rücksetzen muss daher über das Ablaufprogramm erfolgen.	(Nicht für A1S-CPU, bei AnS-CPUs nur D9116 bis D9119)
D9124	Summe der Fehlermerker	Summe der Fehlermerker	Wird ein Fehlermerker (F0 bis F255) über eine OUT F- oder SET F-Anweisung gesetzt, erhöht sich der Wert in D9124 um „1“. Bei Ausführung einer RST F- oder SET F-Anweisung verringert sich der Registerwert wieder um „1“. Die Summe der Fehlermerker wird in D9124 in binärer Form abgelegt. Der Maximalwert von D9124 beträgt deshalb 8.	

Liste der Sonderregister (MELSEC A-Serie)

<table>
<tr><th>Adresse</th><th>Name</th><th>Bedeutung</th><th>Beschreibung</th><th>CPU</th></tr>
<tr><td>D9125
–
D9132</td><td>Nummer der Fehlermerker</td><td></td><td>Wenn einer der Fehlermerker F0 bis F255 über eine OUT F- oder SET F-Anweisung gesetzt wird, wird die Fehlermerkeradresse des gesetzten Fehlermerkers in binärer Form in die Register D9125 bis D9132 geschrieben.

Eine Fehlermerkeradresse, die über eine RST F-Anweisung rückgesetzt wird, wird aus dem Registerbereich gelöscht. Der Inhalt der nachfolgenden Datenregister wird anschließend in dieses Register übertragen.

Bei Ausführung einer LEDR-Anweisung wird der Inhalt von D9125 bis D9132 um ein Bit nach oben verschoben.

Liegen mehr als 8 Fehlermeldungen vor, wird der 9. Fehlermerker nicht in den Registern D9125 bis D9132 gespeichert.

SET F50 | SET F25 | SET F19 | SET F25 | SET F15 | SET F70 | SET F65 | SET F38 | SET F110 | SET F151 | SET F210 | LEDR
D9009: 0 50 50 50 50 50 50 50 50 50 50 50 99
D9124: 0 1 2 3 2 3 4 5 6 7 8 8 8
D9125: 0 50 50 50 50 50 50 50 50 50 50 50 99
D9126: 0 0 25 25 99 99 99 99 99 99 99 99 15
D9127: 0 0 0 99 0 15 15 15 15 15 15 15 70
D9128: 0 0 0 0 0 0 70 70 70 70 70 70 65
D9129: 0 0 0 0 0 0 0 65 65 65 65 65 38
D9130: 0 0 0 0 0 0 0 0 38 38 38 38 110
D9131: 0 0 0 0 0 0 0 0 0 110 110 110 151
D9132: 0 0 0 0 0 0 0 0 0 0 151 151 210</td><td></td></tr>
<tr><td>D9133
–
D9140</td><td>Informationen über die im A2C-System eingesetzten Remote-Module</td><td>00: Kein E/A- oder Sondermodul an dieser Stationsnummer oder keine Initialisierung
01: A2C-Eingangs- oder Sondermodul
10: A2C-Ausgangsmodul
11: Remote-Sondermodul</td><td>Die Register D9133 bis D9140 enthalten Informationen über die im A2C-System befindlichen Remote-Module. Die entsprechenden Daten sind den Stationsnummern der Module zugeordnet und als 2-Bit-Information abgelegt.

Datenstruktur:

b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0
D9133: 8 7 6 5 4 3 2 1
D9134: 16 15 14 13 12 11 10 9
D9135: 16 23 22 21 20 19 18 17
↓ : : :
D9139: 56 55 54 53 52 51 50 49
D9140: 64 63 62 61 60 59 58 57
— Stationsnummern —</td><td>Nur für A2C-CPU</td></tr>
</table>

Liste der Sonderregister (MELSEC A-Serie)

Adresse	Name	Bedeutung	Beschreibung	CPU
D9141 – D9172	Wiederholungszähler	Anzahl der Wiederholungen	In die Register D9141 bis D9172 wird - bezogen auf jede Station - die Anzahl der Wiederholungszugriffe geschrieben, die bei einem Kommunikationsfehler mit einer Remote-Station von der CPU aus unternommen wurden (die Anzahl der Wiederholungsversuche wird in D9174 festgelegt). Bei fehlerfreier Wiederaufnahme der Kommunikation wird das entsprechede Bit auf „0“ gesetzt. Die Datenstruktur der Stationsnummern ergibt sich wie folgt: b15 - b8 b7 - b0 D9141: Station 2 | Station 1 D9142: Station 4 | Station 3 D9143: Station 6 | Station 5 ↓ : : : D9171: Station 62 | Station 61 D9172: Station 64 | Station 63 Der Wiederholungszähler einer Station besteht aus 8 Bit. Während im ersten Bit der Status (0 = Normal, 1 = Fehler) gespeichert wird, enthalten die restlichen 7 Bit die Anzahl der Zugriffsversuche. b(n+7) b(n+6) b(n+5) b(n+4) b(n+3) b(n+2) b(n+1) b(n+0) 0/1 | | | | | | | | Anzahl der Wiederholungszugriffe Der Wert n wird durch die Stationsnummer des Moduls festgelegt. 1, 3, 5, ... 61, 63 ungerade Station: b0 bis b7 (n = 0) 2, 4, 6, ... 62, 64 gerade Station: b8 bis b15 (n = 8)	Nur für A2C-CPU
D9173	Betriebsart	Nummerncode der Betriebsart: 0: Automatische Wiederaufnahme 1: Keine automatische Wiederaufnahme 2: Kommunikation bei Online-Fehler stoppen 3: Hardware-Test	Eingestellte Betriebsart Betriebart 0 (Automatische Wiederaufnahme): Fehlerhafte Stationen werden aus dem Netz abgekoppelt. Die Kommunikation mit den übrigen Stationen wird fortgeführt. Ist der Fehler behoben, wird der Betrieb mit der entsprechenden Station automatisch wieder aufgenommen. Betriebsart 1 (Keine automatische Wiederaufnahme): Fehlerhafte Stationen werden aus dem Netz abgekoppelt. Die Kommunikationmit den übrigen Stationen wird fortgeführt. Ist der Fehler behoben, kann der Betrieb mit der entsprechenden Station erst nach erneutem Hochfahren aufgenommen werden. Betriebsart 2 (Kommunikation bei Online-Fehler stoppen): Tritt ein Fehler in einem Modul auf, wird die Kommunikation mit sämtlichen Stationen abgebrochen. Ist der Fehler behoben, kann der Betrieb mit der entsprechenden Station erst nach erneutem Hochfahren aufgenommen werden. Betriebsart 3 (Hardware-Test): Prüfung von Hardware und Verbindungskabeln	
D9174	Anzahl der Wiederholungszugriffe	Anzahl der Wiederholungen (0 bis 32)	In Register D9174 wird die Anzahl der Wiederholungszugriffe geschrieben, die bei einer fehlerhaften Kommunikation mit einem Remote-Modul erfolgen sollen (der Standardwert lautet 5). Ist eine Kommunikation mit einem Remote-Modul auch nach der hier vorgegebenen Anzahl von Wiederholzugriffen nicht möglich, erfolgt die Ausgabe einer Fehlermeldung.	
D9175	Wiederholungszähler bei einem Fehler im Netzwerk	Anzahl der Wiederholungen	Tritt ein Fehler in der Kommunikation mit einem Remote-Modul auf (Zeitüberschreitung), wird die Anzahl der Wiederholungsversuche in Register D9175 gespeichert. Der Registerwert wird auf „0“ gesetzt, sobald die Kommunikation mit den Remote-Modulen wieder fehlerfrei hergestellt ist.	

Liste der Sonderregister (MELSEC A-Serie)

<table>
<tr><th>Adresse</th><th>Name</th><th>Bedeutung</th><th>Beschreibung</th><th>CPU</th></tr>
<tr><td>D9180
–
D9193</td><td>Fehlercode der Remote-Sonder-Module</td><td>Fehlercode
(0: Normal)</td><td>Nach dem Setzen von M9060 (Fehlererkennung) wird der Fehlercode des fehlerhaften Remote-Sondermoduls in die Register D9180 bis D9193 geschrieben.

Die Stationsnummern der Remote-Sondermodule werden in D9021 bis D9034 festgelegt. Die Speicherbereiche für die Fehlercodes sind wie folgt gegliedert.

D9180 Remote-Sondermodul Nr.1
D9181 Remote-Sondermodul Nr.2
D9182 Remote-Sondermodul Nr.3
↓ : : :
D9192 Remote-Sondermodul Nr.13
D9193 Remote-Sondermodul Nr.14

Das Löschen des Fehlercodes erfolgt, sobald der Schlüsselschalter von STOP auf RUN umgeschaltet wird (alle Register werden gelöscht) oder wenn das Signal Yn4 der Remote-Sondermodule eingeschaltet wird.</td><td rowspan="2">Nur für A2C-CPU</td></tr>
<tr><td>D9196
–
D9199</td><td>Kennung einer fehlerhaften Station</td><td>0: Normal
1: Fehler</td><td>Ist der Zugriff auf eine Station auch nach der in D9174 vorgegebenen Anzahl von Wiederholungsversuchen nicht möglich, wird das dieser Station zugeordnete Bit in D9196 bis D9199 auf „1“ gesetzt. Das Bit wird wieder auf „0“ gesetzt, wenn der Fehler behoben und die automatische Wiederaufnahme (D9173) gewählt wurde.

Datenstruktur:

<table>
<tr><th></th><th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>D9196</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td>D9197</td><td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td></tr>
<tr><td>D9198</td><td>48</td><td>47</td><td>46</td><td>45</td><td>44</td><td>43</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td></tr>
<tr><td>D9199</td><td>64</td><td>63</td><td>62</td><td>61</td><td>60</td><td>59</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td></tr>
</table>

Das entsprechende Bit wird auch dann auf „1“ gesetzt, wenn die Sicherung eines Remote-E/A-Moduls defekt ist.</td></tr>
</table>

HINWEISE *Das Rücksetzen der Sonderregister erfolgt nach dem Abschalten der SPS-Spannungsversorgung oder dem Schalten des Schlüsselschalters auf LATCH CLEAR oder RESET.*

Ein Umsetzen des Schlüsselschalters auf die STOP-Position bewirkt kein Rücksetzen der Register. Die Ist-Zustände werden beibehalten.

Der Inhalt der mit ❶ gekennzeichneten Sonderregister bleibt auch dann erhalten, wenn der Normalzustand wiederhergestellt ist. Ein Rücksetzen dieser Sonderregister ist wie folgt möglich:

- Einfügen einer Programmzeile in das Ablaufprogramm, die das Sonderregister mittels RST-Anweisung bei einer bestimmten Eingangsbedingung zurücksetzt.
- Rücksetzen über ein Programmiergerät.
- Rücksetzen der CPU durch Umschalten des Schlüsselschalters an der CPU auf RESET.

Die mit ❷ gekennzeichneten Sonderregister werden ausschließlich über das Ablaufprogramm gesetzt und und rückgesetzt.

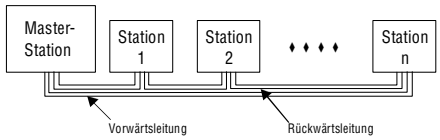
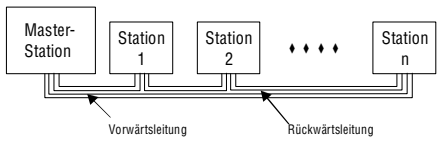
Die mit ❸ gekennzeichneten Sonderregister werden im Testbetrieb eines Programmiergerätes gesetzt und rückgesetzt.

A.5.3 Übersicht der Sonderregister im Link-Betrieb (Nur MELSEC A-Serie)

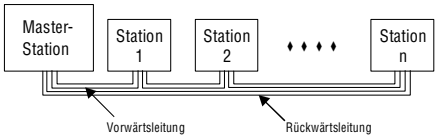
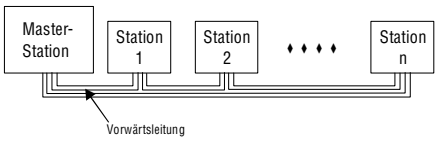
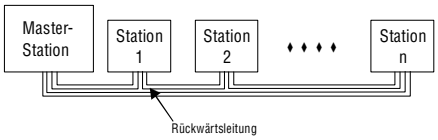
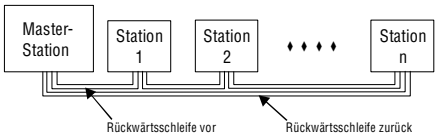
Diese Sonderregister werden durch unterschiedliche Faktoren während der Datenkommunikation in einem Netzwerk gesetzt bzw. rückgesetzt. Die Register speichern bestimmte Kommunikations- oder Fehlerzustände im Netzwerk. Störungen im Netzwerk oder in fehlerhaften Stationen können durch Auslesen der Link-Register angezeigt werden.

Die Verarbeitung der Sonderregister im Link-Betrieb ist davon abhängig, ob sich die CPU in einer Master- oder lokalen Station befindet.

Sonderregister im Link-Betrieb in der Master-Station

Adresse	Name	Bedeutung	Beschreibung
D9200	Verarbeitungsergebnis der LRDP-Anweisung	0: OK (normal) 2: Fehler in der Anweisung 3: Fehler in der angesprochenen Station 4: Ausführung nicht möglich	Das Register speichert das Ausführungsergebnis einer LRDP-Anweisung (Wortoperanden lesen). 0: Anweisung wurde fehlerfrei ausgeführt. 2: Eine Konstante oder die Quell-/Zieldaten sind falsch definiert. 3: Bei den angesprochenen Stationen handelt es sich um eine Remote-Station.
D9201	Verarbeitungsergebnis der LWTP-Anweisung	0: OK (normal) 2: Fehler in der Anweisung 3: Fehler in der angesprochenen Station 4: Ausführung nicht möglich	Das Register speichert das Ausführungsergebnis einer LWTP-Anweisung (Wortoperanden schreiben). 0: Anweisung wurde fehlerfrei ausgeführt. 2: Eine Konstante oder die Quell-/Zieldaten sind falsch definiert. 3: Bei den angesprochenen Stationen handelt es sich um eine Remote-Station.
D9202	Link-Typ einer lokalen Station (siehe auch D9241, D9242)	Zustand der Stationen 1 bis 16	Die Datenregister speichern die Kompatibilität der Slave-Stationen zum MELSECNET bzw. MELSECNET/II. Ist eine Station kompatibel zum MELSECNET/II, wird in das der Station zugeordnete Bit des Sonderregisters eine „1“ geschrieben. Ist sie kompatibel zum MELSECNET, wird eine „0“ gesetzt.
D9203		Zustand der Stationen 17 bis 32	
D9204 (siehe auch nächste Seite)	Link-Status	0: Datenübertragung über Vorwärtsschleife 1: Datenübertragung über Rückwärtsschleife 2: Datenübertragung sowohl vorwärts als auch rückwärts 3: Schleife ausschließlich über Vorwärtsschleife 4: Schleife ausschließlich über Rückwärtsschleife 5: Datenübertragung nicht möglich Anmerkung: Für MELSECNET/B sind nur die Variante 0 und 5 möglich	Das Register speichert die aktuellen Zustandsdaten der Netzwerkkommunikation. -Datenübertragung der Vorwärtsschleife  -Datenübertragung über Rückwärtsschleife 

Sonderregister im Link-Betrieb in der Master-Station

Adresse	Name	Bedeutung	Beschreibung
D9204	Link-Status		-Datenübertragung vorwärts und rückwärts  -Datenübertragung ausschließlich über Vorwärtsleitung  - Datenübertragung ausschließlich über Rückwärtsleitung 
D9205*	Ausführende Station der Rückleitung	Station, über die die Vorwärts-Rückwärtsleitung erfolgt	Die Datenregister speichern die Adresse der lokalen Station oder Remote-E/A-Station, über die eine Rückwärtsleitung erfolgt  Im Beispiel oben wird in D9205 eine „1“ und in D9206 eine „3“ gespeichert. Die Werte bleiben auch dann erhalten, wenn der normale Kommunikationszustand wieder aufgenommen wurde. Das Rücksetzen der Registerwerte muss über das Ablaufprogramm oder einen CPU-RESET erfolgen.
D9206*	Ausführende Station der Rückleitung	Station, über die die Rückwärtsleitung erfolgt	
D9207*	Zykluszeit im Netzwerk	Maximale Zykluszeit	Die Datenregister speichern die Verarbeitungszeiten der Kommunikation zwischen Master-Station und den lokalen Stationen und Remote-E/A-Stationen. In einem Kommunikationszyklus werden alle über Parameter festgelegten Link-Daten, wie Eingänge (X), Ausgänge (Y), Link-Merker (B) oder Link-Register (W) mit den entsprechenden Stationen im Netzwerk ausgetauscht. Ein Kommunikationszyklus beschreibt den Zeitraum, in dem unabhängig von der Programmzykluszeit eine vollständige Datenkommunikation mit allen angeschlossenen Slave-Stationen erfolgt.
D9208*		Minimale Zykluszeit	
D9209*		Istwert der Zykluszeit	
D9210*	Wiederhohlungszähler	Summe der Wiederholungen	Nach einem Übertragungsfehler wird die Anzahl der Wiederhohlungszugriffe in D9210 gespeichert. Der Zähler stoppt am Maximalwert $FFFF_H$. Das Rücksetzen des Registerinhalts erfolgt mittels CPU-RESET.
D9211*	Zähler für Leitungswechsel	Summe der Wechsel	Nach jedem Wechsel zwischen Vorwärts- und Rückwärtsleitung wird der Wert in D9211 um „1“ erhöht.

Sonderregister im Link-Betrieb in der Master-Station

<table>
<tr><th>Adresse</th><th>Name</th><th>Bedeutung</th><th>Beschreibung</th></tr>
<tr><td>D9212</td><td rowspan="4">Betriebszustand der lokalen Station</td><td>Zustand der Stationen 1 bis 16</td><td rowspan="4">Die Datenregister speichern die Adressen der lokalen Stationen, die sich im PAUSE- oder STOP-Zustand befinden.

<table>
<tr><th></th><th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>D9212</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td>D9213</td><td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td></tr>
<tr><td>D9214</td><td>48</td><td>47</td><td>46</td><td>45</td><td>44</td><td>43</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td></tr>
<tr><td>D9215</td><td>64</td><td>63</td><td>62</td><td>61</td><td>60</td><td>59</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td></tr>
</table>
Stationsnummern

Befindet sich eine Station im PAUSE- oder STOP-Zustand, wird das zugehörige Bit auf „1“ gesetzt. Schaltet beispielsweise Station Nr. 7 in den STOP-Zustand, wird Bit 6 in D9212 auf „1“ gesetzt. Der Wert bei Auslesen von D9212 lautet dann: 64 (40H)</td></tr>
<tr><td>D9213</td><td>Zustand der Stationen 17 bis 32</td></tr>
<tr><td>D9214*</td><td>Zustand der Stationen 33 bis 48</td></tr>
<tr><td>D9215*</td><td>Zustand der Stationen 49 bis 64</td></tr>
<tr><td>D9216</td><td rowspan="4">Fehlererkennung der lokalen Station</td><td>Zustand der Stationen 1 bis 16</td><td rowspan="4">Die Datenregister speichern die Adressen fehlerhafter lokaler Stationen.

<table>
<tr><th></th><th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>D9216</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td>D9217</td><td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td></tr>
<tr><td>D9218</td><td>48</td><td>47</td><td>46</td><td>45</td><td>44</td><td>43</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td></tr>
<tr><td>D9219</td><td>64</td><td>63</td><td>62</td><td>61</td><td>60</td><td>59</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td></tr>
</table>
Stationsnummern

Sobald in einer lokalen Station ein Fehler erkannt wird, wird das zugehörige Bit auf „1“ gesetzt. Liegt beispielsweise in den Stationen 6 und 12 ein Fehler vor, werden die Bits 5 und 11 in D9216 auf „1“ gesetzt. Der Wert bei Auslesen von D9218 lautet dann: 2082 (820H).</td></tr>
<tr><td>D9217</td><td>Zustand der Stationen 17 bis 32</td></tr>
<tr><td>D9218*</td><td>Zustand der Stationen 33 bis 48</td></tr>
<tr><td>D9219*</td><td>Zustand der Stationen 49 bis 64</td></tr>
<tr><td>D9220</td><td rowspan="4">Parameterfehler einer lokalen Station oder E/A-Zuordnung einer Remote-E/A-Station ist fehlerhaft</td><td>Zustand der Stationen 1 bis 16</td><td rowspan="4">Die Datenregister speichern die Adressen lokaler Stationen, deren Daten von den Parameterdaten abweichen. Remote-E/A-Stationen, deren E/A-Zuordnung fehlerhaft ist, werden ebenfalls gespeichert.

<table>
<tr><th></th><th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>D9220</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td>D9221</td><td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td></tr>
<tr><td>D9222</td><td>48</td><td>47</td><td>46</td><td>45</td><td>44</td><td>43</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td></tr>
<tr><td>D9223</td><td>64</td><td>63</td><td>62</td><td>61</td><td>60</td><td>59</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td></tr>
</table>
Stationsnummern

Sobald ein Fehler erkannt wird, wird das zugehörige Bit auf „1“ gesetzt. Liegt beispielsweise in der lokalen Stationen 5 und in der Remote-E/A-Station Nr. 14 ein Fehler vor, werden die Bits 4 und 13 in D9220 auf „1“ gesetzt. Der Wert bei Auslesen von D9220 lautet dann: 8208 (2010H).</td></tr>
<tr><td>D9221</td><td>Zustand der Stationen 17 bis 32</td></tr>
<tr><td>D9222*</td><td>Zustand der Stationen 33 bis 48</td></tr>
<tr><td>D9223*</td><td>Zustand der Stationen 49 bis 64</td></tr>
<tr><td>D9224</td><td rowspan="4">Initialisierungsübertragung zu lokalen Stationen oder Remote-E/A-Stationen</td><td>Zustand der Stationen 1 bis 16</td><td rowspan="4">Die Datenregister speichern die Adressen lokaler Stationen und Remote-E/A-Stationen, bei denen eine Initialisierung von Seiten der zugehörigen Master-Stationen stattfinden.

<table>
<tr><th></th><th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>D9224</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td>D9225</td><td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td></tr>
<tr><td>D9226</td><td>48</td><td>47</td><td>46</td><td>45</td><td>44</td><td>43</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td></tr>
<tr><td>D9227</td><td>64</td><td>63</td><td>62</td><td>61</td><td>60</td><td>59</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td></tr>
</table>
Stationsnummern

Das zugehörige Bit der Station, die die Initialisierungsdaten eerhält, wird auf „1“ gesetzt. Befindet sich beispielsweise Station Nr. 45 im Initialisierungsprozess, wird Bit 23 von D9226 auf „1“ gesetzt. Der Wert bei Auslesen von D9226 lautet dann: 4096 (1000H).</td></tr>
<tr><td>D9225</td><td>Zustand der Stationen 17 bis 32</td></tr>
<tr><td>D9226*</td><td>Zustand der Stationen 33 bis 48</td></tr>
<tr><td>D9227*</td><td>Zustand der Stationen 49 bis 64</td></tr>
</table>

Sonderregister im Link-Betrieb in der Master-Station

<table>
<tr><th>Adresse</th><th>Name</th><th>Bedeutung</th><th>Beschreibung</th></tr>
<tr><td>D9228</td><td rowspan="4">Fehler in einer Remote E/A-Station oder lokalen Station</td><td>Zustand der Stationen
1 bis 16</td><td rowspan="4">In den Datenregisten werden alle Stationen gekennzeichnet, die fehlerhaft arbeiten.

<table>
<tr><th></th><th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>D9228</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td>D9229</td><td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td></tr>
<tr><td>D9230</td><td>48</td><td>47</td><td>46</td><td>45</td><td>44</td><td>43</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td></tr>
<tr><td>D9231</td><td>64</td><td>63</td><td>62</td><td>61</td><td>60</td><td>59</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td></tr>
</table>
Nummer der Remote-E/A- oder lokale Station
Bei einem Fehler wird das Bit, welches der fehlerhaften Station zugeordnet ist, auf „1“ gesetzt. Liegt beispielsweise in der lokalen Stationen 36 und Remote-E/A-Station 14 ein Fehler vor, werden die Bits 2 und 13 in D9228 auf „1“ gesetzt. Der Wert bei Auslesen von D9228 lautet dann: 8196 (2004H).</td></tr>
<tr><td>D9229</td><td>Zustand der Stationen
17 bis 32</td></tr>
<tr><td>D9230*</td><td>Zustand der Stationen
33 bis 48</td></tr>
<tr><td>D9231*</td><td>Zustand der Stationen
49 bis 64</td></tr>
<tr><td>D9232</td><td rowspan="8">Schleifenfehler einer lokalen oder Remote-E/A-Station

Anmerkung:
Nicht für MELSECNET/B</td><td>Zustand der Stationen
1 bis 8</td><td rowspan="8">In den Datenregistern wird die Adresse der Stationen gekennzeichnet, an der ein Fehler in der Vorwärts- oder Rückwärtsschleife erkannt wurde.

<table>
<tr><td></td><td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr>
<tr><td>D9232</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td></tr>
<tr><td></td><td colspan="2">8</td><td colspan="2">7</td><td colspan="2">6</td><td colspan="2">5</td><td colspan="2">4</td><td colspan="2">3</td><td colspan="2">2</td><td colspan="2">1</td></tr>
<tr><td>D9233</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td></tr>
<tr><td></td><td colspan="2">16</td><td colspan="2">15</td><td colspan="2">14</td><td colspan="2">13</td><td colspan="2">12</td><td colspan="2">11</td><td colspan="2">10</td><td colspan="2">9</td></tr>
<tr><td>D9234</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td></tr>
<tr><td></td><td colspan="2">24</td><td colspan="2">23</td><td colspan="2">22</td><td colspan="2">21</td><td colspan="2">20</td><td colspan="2">19</td><td colspan="2">18</td><td colspan="2">17</td></tr>
<tr><td>D9235</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td></tr>
<tr><td></td><td colspan="2">32</td><td colspan="2">31</td><td colspan="2">30</td><td colspan="2">29</td><td colspan="2">28</td><td colspan="2">27</td><td colspan="2">26</td><td colspan="2">25</td></tr>
<tr><td>D9136</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td></tr>
<tr><td></td><td colspan="2">40</td><td colspan="2">39</td><td colspan="2">38</td><td colspan="2">37</td><td colspan="2">36</td><td colspan="2">35</td><td colspan="2">34</td><td colspan="2">33</td></tr>
<tr><td>D9237</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td></tr>
<tr><td></td><td colspan="2">48</td><td colspan="2">47</td><td colspan="2">46</td><td colspan="2">45</td><td colspan="2">44</td><td colspan="2">43</td><td colspan="2">42</td><td colspan="2">41</td></tr>
<tr><td>D9238</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td></tr>
<tr><td></td><td colspan="2">56</td><td colspan="2">55</td><td colspan="2">54</td><td colspan="2">53</td><td colspan="2">52</td><td colspan="2">51</td><td colspan="2">50</td><td colspan="2">49</td></tr>
<tr><td>D9239</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td><td colspan="2">R V</td></tr>
<tr><td></td><td colspan="2">64</td><td colspan="2">63</td><td colspan="2">62</td><td colspan="2">61</td><td colspan="2">60</td><td colspan="2">59</td><td colspan="2">58</td><td colspan="2">57</td></tr>
</table>
Stationsnummern
In der Tabelle steht der Buchstabe V für Vorwärtsschleife und R für die Rückwärtsschleife. Sobald ein Fehler erkannt wird, wird das zugehörige Bit auf „1“ gesetzt. Liegt beispielsweise in der Vorwärtsschleife von Station 5 ein Fehler vor, wird Bit 8 von D9232 auf „1“ gesetzt. Der Wert bei Auslesen von D9232 lautet dann: 256 (100H).</td></tr>
<tr><td>D9233</td><td>Zustand der Stationen
9 bis 16</td></tr>
<tr><td>D9234</td><td>Zustand der Stationen
17 bis 24</td></tr>
<tr><td>D9235</td><td>Zustand der Stationen
25 bis 32</td></tr>
<tr><td>D9236</td><td>Zustand der Stationen
33 bis 40</td></tr>
<tr><td>D9237</td><td>Zustand der Stationen
41 bis 48</td></tr>
<tr><td>D9238</td><td>Zustand der Stationen
49 bis 56</td></tr>
<tr><td>D9239</td><td>Zustand der Stationen
57 bis 64</td></tr>
<tr><td>D9240</td><td>Zähler für Übertragungsfehler</td><td>Zähler für Übertragungsfehler</td><td>Sobald einer der Übertragungsfehler CRC, OVER oder AB.IF erkannt wird, wird der Wert in D9240 um „1“ erhöht. Der Zähler stoppt am Maximalwert FFFFH. Das Rücksetzen des Registerinhalts erfolgt mittels CPU-RESET.</td></tr>
<tr><td>D9241*</td><td rowspan="2">Link-Typ einer lokalen Station

(siehe auch D9202, D9203)</td><td>Zustand der Stationen
33 bis 48</td><td rowspan="2">Die Datenregister speichern die Kompabilität der Slave-Statioen zum MELSECNET bzw. MESECNET II.
Ist eine Station kompatibel zum MELSECNET II, wird in das der Station zugeordnete Bit des Sonderregisters eine „1“ geschrieben. Ist sie kompatibel zu MELSECNET, wird eine „0“ gesetzt.</td></tr>
<tr><td>D9242*</td><td>Zustand der Stationen
49 bis 64</td></tr>
<tr><td>D9243</td><td>Kontrolle der Stationsnummer</td><td>Nummer einer Station (0 bis 64)</td><td>Das Datenregister dient der Kontrolle der eigenen Adresse einer lokalen Station.</td></tr>
<tr><td>D9244</td><td>Anzahl der angeschlossenen Slave-Station</td><td>Summe der Slave-Stationen</td><td>Das Datenregister speichert die Summe der Slave-Stationen, die sich innerhalb einer Leitung befinden.</td></tr>
<tr><td>D9245</td><td>Zähler für Übertragungsfehler</td><td>Summe der Übertragungsfehler</td><td>Sobald einer der Übertragungsfehler CRC, OVER oder AB.IF erkannt wird, wird der Wert in D9245 um „1“ erhöht. Der Zähler stoppt am Maximalwert FFFFH. Das Rücksetzen des Registerinhalts erfolgt mittels CPU-RESET.</td></tr>
</table>

Sonderregister im Link-Betrieb in der Master-Station

<table>
<tr><th>Adresse</th><th>Name</th><th>Bedeutung</th><th>Beschreibung</th></tr>
<tr><td>D9248</td><td rowspan="4">Betriebszustand der lokalen Statioen</td><td>Zustand der Stationen 1 bis 16</td><td rowspan="4">Die Datenregister kennzeichnen alle lokalen Stationen, die sich im PAUSE- oder STOP-Zustand befinden.

<table>
<tr><th></th><th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>D9196</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td>D9197</td><td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td></tr>
<tr><td>D9198</td><td>48</td><td>47</td><td>46</td><td>45</td><td>44</td><td>43</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td></tr>
<tr><td>D9199</td><td>64</td><td>63</td><td>62</td><td>61</td><td>60</td><td>59</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td></tr>
</table>
Stationsnummern

Befindet sich eine lokale Station im PAUSE- oder STOP-Zustand, wird das zugehörige Bit auf „1“ gesetzt. Schaltet beispielsweise Station 7 und 15 in den STOP-Zustand, wird Bit 6 und 14 in D9248 auf „1“ gesetzt. Der Wert bei Auslesen von D9248 lautet dann: 16448 (4040_H).</td></tr>
<tr><td>D9249</td><td>Zustand der Stationen 17 bis 32</td></tr>
<tr><td>D9250*</td><td>Zustand der Stationen 33 bis 48</td></tr>
<tr><td>D9251*</td><td>Zustand der Stationen 49 bis 64</td></tr>
<tr><td>D9252</td><td rowspan="4">Fehler in einer lokalen Station</td><td>Zustand der Stationen 1 bis 16</td><td rowspan="4">Die Datenregister kennzeichnen die lokale Station außerhalb der Host-Station, in der ein Fehler vorliegt.

<table>
<tr><th></th><th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>D9196</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td>D9197</td><td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td></tr>
<tr><td>D9198</td><td>48</td><td>47</td><td>46</td><td>45</td><td>44</td><td>43</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td></tr>
<tr><td>D9199</td><td>64</td><td>63</td><td>62</td><td>61</td><td>60</td><td>59</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td></tr>
</table>
Stationsnummern

Sobald in einer lokalen Station ein Fehler erkannt wird, wird das zugehörige Bit auf „1“ gesetzt. Liegt beispielsweise in Station 6 ein Fehler vor, wird Bit 5 in D9252 auf „1“ gesetzt. Der Wert bei Auslesen von D9252 lautet dann: 2048 (800_H).</td></tr>
<tr><td>D9253</td><td>Zustand der Stationen 17 bis 32</td></tr>
<tr><td>D9254*</td><td>Zustand der Stationen 33 bis 48</td></tr>
<tr><td>D9255*</td><td>Zustand der Stationen 49 bis 64</td></tr>
</table>

* Die gekennzeichneten Sonderregister sind nicht im MELSECNET/B einsetzbar.

Index

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z

Symbole

Ziffern

HEADQUARTERS

MITSUBISHI ELECTRIC EUROPE B.V. — EUROPA
German Branch
Gothaer Straße 8
D-40880 Ratingen
Telefon: 02102 / 486-0
Telefax: 02102 / 486-1120
E-Mail: megfamail@meg.mee.com

MITSUBISHI ELECTRIC EUROPE B.V. — FRANKREICH
French Branch
25, Boulevard des Bouvets
F-92741 Nanterre Cedex
Telefon: +33 1 55 68 55 68
Telefax: +33 1 55 68 56 85
E-Mail: factory.automation@fra.mee.com

MITSUBISHI ELECTRIC EUROPE B.V. — IRLAND
Irish Branch
Westgate Business Park, Ballymount
IRL-Dublin 24
Telefon: +353 (0) 1 / 419 88 00
Fax: +353 (0) 1 / 419 88 90
E-Mail: sales.info@meir.mee.com

MITSUBISHI ELECTRIC EUROPE B.V. — ITALIEN
Italian Branch
Via Paracelso 12
I-20041 Agrate Brianza (MI)
Telefon: +39 039 6053 1
Telefax: +39 039 6053 312
E-Mail: factory.automation@it.mee.com

MITSUBISHI ELECTRIC EUROPE B.V. — SPANIEN
Spanish Branch
Carretera de Rubí 76-80
E-08190 Sant Cugat del Vallés
Telefon: +34 9 3 / 565 3131
Telefax: +34 9 3 / 589 2948
E-Mail: industrial@sp.mee.com

MITSUBISHI ELECTRIC EUROPE B.V. — UK
UK Branch
Travellers Lane
GB-Hatfield Herts. AL10 8 XB
Telefon: +44 (0) 1707 / 27 61 00
Telefax: +44 (0) 1707 / 27 86 95
E-Mail: automation@meuk.mee.com

MITSUBISHI ELECTRIC CORPORATION — JAPAN
Office Tower "Z" 14 F
8-12,1 chome, Harumi Chuo-Ku
Tokyo 104-6212
Telefon: +81 3 6221 6060
Telefax: +81 3 6221 6075

MITSUBISHI ELECTRIC AUTOMATION — USA
500 Corporate Woods Parkway
Vernon Hills, IL 60061
Telefon: +1 847 / 478 21 00
Telefax: +1 847 / 478 22 83

KUNDEN-TECHNOLOGIE-CENTER DEUTSCHLAND

MITSUBISHI ELECTRIC EUROPE B.V.
Kunden-Technologie-Center Nord
Revierstraße 5
D-44379 Dortmund
Telefon: (02 31) 96 70 41-0
Telefax: (02 31) 96 70 41-41

MITSUBISHI ELECTRIC EUROPE B.V.
Kunden-Technologie-Center Süd-West
Kurze Straße 40
D-70794 Filderstadt
Telefon: (07 11) 77 05 98-0
Telefax: (07 11) 77 05 98-79

MITSUBISHI ELECTRIC EUROPE B.V.
Kunden-Technologie-Center Süd-Ost
Am Söldnermoos 8
D-85399 Hallbergmoos
Telefon: (08 11) 99 87 40
Telefax: (08 11) 99 87 410

EUROPÄISCHE VERTRETUNGEN

Getronics b.v. — BELGIEN
Control Systems
Pontbeeklaan 43
BE-1731 Asse-Zellik
Telefon: +32 (0) 2 / 467 17 51
Telefax: +32 (0) 2 / 467 17 45
E-Mail: infoautomation@getronics.com

TELECON CO. — BULGARIEN
4, A. Ljapchev Blvd.
BG-1756 Sofia
Telefon: +359 (0) 2 / 97 44 05 8
Telefax: +359 (0) 2 / 97 44 06 1
E-Mail: —

louis poulsen — DÄNEMARK
industri & automation
Geminivej 32
DK-2670 Greve
Telefon: +45 (0) 70 / 10 15 35
Telefax: +45 (0) 43 / 95 95 91
E-Mail: lpia@lpmail.com

UTU Elektrotehnika AS — ESTLAND
Pärnu mnt.160i
EE-11317 Tallinn
Telefon: +372 (0) 6 / 51 72 80
Telefax: +372 (0) 6 / 51 72 88
E-Mail: utu@utu.ee

Beijer Electronics OY — FINNLAND
Ansatie 6a
FI-01740 Vantaa
Telefon: +358 (0) 9 / 886 77 500
Telefax: +358 (0) 9 / 886 77 555
E-Mail: info@beijer.fi

UTECO A.B.E.E. — GRIECHENLAND
5, Mavrogenous Str.
GR-18542 Piraeus
Telefon: +302 (0) 10 / 42 10 050
Telefax: +302 (0) 10 / 42 12 033
E-Mail: sales@uteco.gr

INEA CR d.o.o. — KROATIEN
Losinjska 4 a
HR-10000 Zagreb
Telefon: +385 (0)1 / 36 940-01
Telefax: +385 (0)1 / 36 940-03
E-Mail: inea@inea.hr

SIA POWEL — LETTLAND
Lienes iela 28
LV-1009 Riga
Telefon: +371 784 / 2280
Telefax: +371 784 / 2281
E-Mail: utu@utu.lv

UAB UTU POWEL — LITAUEN
Savanoriu pr. 187
LT-2053 Vilnius
Telefon: +370 (0) 52323-101
Telefax: +370 (0) 52322-980
E-Mail: powel@utu.lt

Intehsis srl — MOLDAWIEN
Cuza-Voda 36/1-81
MD-2061 Chisinau
Telefon: +373 (0)2 / 562263
Telefax: +373 (0)2 / 562263
E-Mail: intehsis@mdl.net

Getronics b.v. — NIEDERLANDE
Control Systems
Donauweg 2 B
NL-1043 AJ Amsterdam
Telefon: +31 (0) 20 / 587 67 00
Telefax: +31 (0) 20 / 587 68 39
E-Mail: info.gia@getronics.com

Beijer Electronics AS — NORWEGEN
Teglverksveien 1
N-3002 Drammen
Telefon: +47 (0) 32 / 24 30 00
Telefax: +47 (0) 32 / 84 85 77
E-Mail: info@beijer.no

GEVA — ÖSTERREICH
Wiener Straße 89
AT-2500 Baden
Telefon: +43 (0) 2252 / 85 55 20
Telefax: +43 (0) 2252 / 488 60
E-Mail: office@geva.at

EUROPÄISCHE VERTRETUNGEN

MPL Technology Sp. z o.o. — POLEN
ul. Sliczna 36
PL-31-444 Kraków
Telefon: +48 (0) 12 / 632 28 85
Telefax: +48 (0) 12 / 632 47 82
E-Mail: krakow@mpl.pl

Sirius Trading & Services srl — RUMÄNIEN
Str. Biharia Nr. 67-77
RO-013981 Bucuresti 1
Telefon: +40 (0) 21 / 201 1146
Telefax: +40 (0) 21 / 201 1148
E-Mail: sirius@siriustrading.ro

Beijer Electronics AB — SCHWEDEN
Box 426
S-20124 Malmö
Telefon: +46 (0) 40 / 35 86 00
Telefax: +46 (0) 40 / 35 86 02
E-Mail: info@beijer.se

ECONOTEC AG — SCHWEIZ
Postfach 282
CH-8309 Nürensdorf
Telefon: +41 (0) 1 / 838 48 11
Telefax: +41 (0) 1 / 838 48 12
E-Mail: info@econotec.ch

INEA d.o.o. — SLOWENIEN
Stegne 11
SI-1000 Ljubljana
Telefon: +386 (0) 1-513 8100
Telefax: +386 (0) 1-513 8170
E-Mail: inea@inea.si

AutoCont Control Systems s.r.o. — TSCHECHISCHE REPUBLIK
Nemocnicni 12
CZ-702 00 Ostrava 2
Telefon: +420 59 / 6152 111
Telefax: +420 59 / 6152 562
E-Mail: consys@autocont.cz

GTS — TÜRKEI
Darülaceze Cad. No. 43 Kat. 2
TR-80270 Okmeydani-Istanbul
Telefon: +90 (0) 212 / 320 1640
Telefax: +90 (0) 212 / 320 1649
E-Mail: gts@turk.net

CSC Automation Ltd. — UKRAINE
15, M. Raskova St., Fl. 10, Office 1010
UA-02002 Kiev
Telefon: +380 (0) 44 / 238-83-16
Telefax: +380 (0) 44 / 238-83-17
E-Mail: csc-a@csc-a.kiev.ua

Meltrade Automatika Kft. — UNGARN
55, Harmat St.
HU-1105 Budapest
Telefon: +36 (0)1 / 2605 602
Telefax: +36 (0)1 / 2605 602
E-Mail: office@meltrade.hu

Tehnikon — WEISSRUSSLAND
Oktjabrskaya 16/5, Ap 704
BY-220030 Minsk
Telefon: +375 (0) 17 / 22 75 704
Telefax: +375 (0) 17 / 22 76 669
E-Mail: tehnikon@belsonet.net

VERTRETUNG MITTLERER OSTEN

Texel Electronics Ltd. — ISRAEL
Box 6272
IL-42160 Netanya
Telefon: +972 (0) 9 / 863 08 91
Telefax: +972 (0) 9 / 885 24 30
E-Mail: texel_me@netvision.net.il

VERTRETUNGEN EURASIEN

Avtomatika Sever Ltd. — RUSSLAND
Lva Tolstogo St. 7, Off. 311
RU-197376 St Petersburg
Telefon: +7 812 / 11 83 238
Telefax: +7 812 / 11 83 239
E-Mail: as@avtsev.spb.ru

CONSYS — RUSSLAND
Promyshlennaya St. 42
RU-198099 St Petersburg
Telefon: +7 812 / 325 36 53
Telefax: +7 812 / 147 20 55
E-Mail: consys@consys.spb.ru

Electrotechnical Systems Siberia — RUSSLAND
Partizanskaya St. 27, Office 306
RU-121355 Moscow
Telefon: +7 095 / 416-4321
Telefax: +7 095 / 416-4321
E-Mail: info@eltechsystems.ru

Electrotechnical Systems Siberia — RUSSLAND
Shetinkina St. 33, Office 116
RU-630088 Novosibirsk
Telefon: +7 3832 / 22-03-05
Telefax: +7 3832 / 22-03-05
E-Mail: info@eltechsystems.ru

Elektrostyle — RUSSLAND
ul. Garschina 11
RU-140070 Moscow Oblast
Telefon: +7 095 / 514 9316
Telefax: +7 095 / 514 9317
E-Mail: info@estl.ru

Elektrostyle — RUSSLAND
Krasnij Prospekt 220-1
Office No. 312
RU-630049 Novosibirsk
Telefon: +7 3832 / 10 66 18
Telefax: +7 3832 / 10 66 26
E-Mail: info@estl.ru

ICOS — RUSSLAND
Industrial Computer Systems Zao
Ryazanskij Prospekt 8a, Office 100
RU-109428 Moscow
Telefon: +7 095 / 232 - 0207
Telefax: +7 095 / 232 - 0327
E-Mail: mail@icos.ru

NPP Uralelektra — RUSSLAND
ul. Sverdlova 11a
RU-620027 Ekaterinburg
Telefon: +7 34 32 / 53 27 45
Telefax: +7 34 32 / 53 27 45
E-Mail: elektra@etel.ru

SSMP Rosgidromontazh Ltd. — RUSSLAND
23, Lesoparkovaya Str.
RU-344041 Rostov On Don
Telefon: +7 8632 / 36 00 22
Telefax: +7 8632 / 36 00 26
E-Mail: —

STC Drive Technique — RUSSLAND
ul. Bajkalskaja 239, Office 2 - 23
RU-664075 Irkutsk
Telefon: +7 3952 / 24 38 16
Telefax: +7 3952 / 23 02 98
E-Mail: privod@irk.ru

STC Drive Technique — RUSSLAND
Poslannikov Per. 9, str.1
RU-107005 Moscow
Telefon: +7 095 / 790-72-10
Telefax: +7 095 / 790-72-12
E-Mail: info@privod.ru

VERTRETUNG AFRIKA

CBI Ltd. — SÜDAFRIKA
Private Bag 2016
ZA-1600 Isando
Telefon: +27 (0) 11/ 928 2000
Telefax: +27 (0) 11/ 392 2354
E-Mail: cbi@cbi.co.z