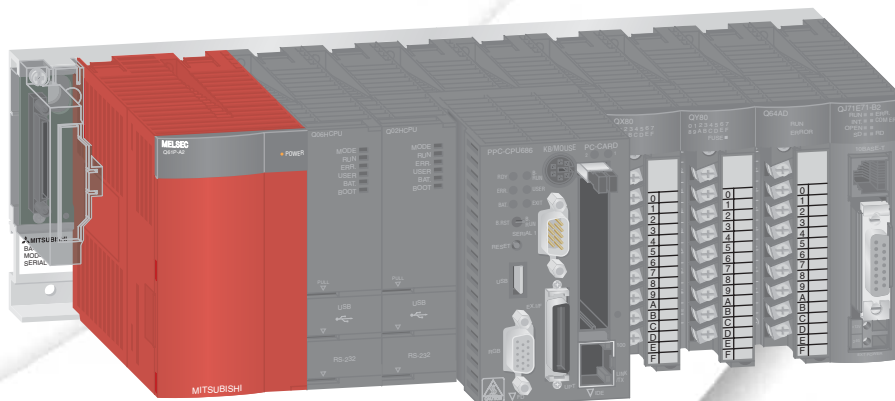


MELSEC SYSTEM Q

Speicherprogrammierbare Steuerungen

Schulungshandbuch



GX IEC Developer

Zu diesem Handbuch

Die in diesem Handbuch vorliegenden Texte, Abbildungen und Beispiele dienen ausschließlich der Erläuterung zur Installation, Bedienung und Betrieb der Programmier-Software GX IEC Developer.

Sollten sich Fragen zur Programmierung und Betrieb der in diesem Handbuch erwähnten speicherprogrammierbaren Steuerungen ergeben, zögern Sie nicht, Ihr zuständiges Verkaufsbüro oder einen Ihrer Vertriebspartner (siehe Umschlagrückseite) zu kontaktieren. Aktuelle Informationen sowie Antworten auf häufig gestellte Fragen erhalten Sie über die Mitsubishi-Homepage unter www.mitsubishi-automation.de.

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

**Schulungshandbuch
Programmier-Software GX IEC Developer
Artikel-Nr.: 170297**

Version	Änderungen / Ergänzungen / Korrekturen
A 07/2006 pdp-dk	Erste Ausgabe

Sicherheitshinweise

Zielgruppe

Dieses Handbuch richtet sich ausschließlich an anerkannt ausgebildete Elektrofachkräfte, die mit den Sicherheitsstandards der Automatisierungstechnik vertraut sind. Projektierung, Installation, Inbetriebnahme, Wartung und Prüfung der Geräte dürfen nur von einer anerkannt ausgebildeten Elektrofachkraft, die mit den Sicherheitsstandards der Automatisierungstechnik vertraut ist, ausgeführt werden. Eingriffe in die Hard- und Software unserer Produkte, soweit sie nicht in diesem Handbuch beschrieben sind, dürfen nur durch unser Fachpersonal vorgenommen werden.

Bestimmungsgemäßer Gebrauch

Die Speicherprogrammierbaren Steuerungen des MELSEC System Q sind nur für die Einsatzbereiche vorgesehen, die in der vorliegenden Bedienungsanleitung beschrieben sind. Achten Sie auf die Einhaltung aller im Handbuch angegebenen Kenndaten. Die Produkte wurden unter Beachtung der Sicherheitsnormen entwickelt, gefertigt, geprüft und dokumentiert. Bei Beachtung der für Projektierung, Montage und ordnungsgemäßen Betrieb beschriebenen Handhabungsvorschriften und Sicherheitshinweise gehen vom Produkt im Normalfall keine Gefahren für Personen oder Sachen aus. Unqualifizierte Eingriffe in die Hard- oder Software bzw. Nichtbeachtung der in diesem Handbuch angegebenen oder am Produkt angebrachten Warnhinweise können zu schweren Personen- oder Sachschäden führen. Es dürfen nur von MITSUBISHI ELECTRIC empfohlene Zusatz- bzw. Erweiterungsgeräte in Verbindung mit den speicherprogrammierbaren Steuerungen des MELSEC System Q benutzt werden. Jede andere darüber hinausgehende Verwendung oder Benutzung gilt als nicht bestimmungsgemäß.

Sicherheitsrelevante Vorschriften

Bei der Projektierung, Installation, Inbetriebnahme, Wartung und Prüfung der Geräte müssen die für den spezifischen Einsatzfall gültigen Sicherheits- und Unfallverhütungsvorschriften beachtet werden. Es müssen besonders folgende Vorschriften (ohne Anspruch auf Vollständigkeit) beachtet werden:

- VDE-Vorschriften
 - VDE 0100
Bestimmungen für das Errichten von Starkstromanlagen mit einer Nennspannung bis 1000 V
 - VDE 0105
Betrieb von Starkstromanlagen
 - VDE 0113
Elektrische Anlagen mit elektronischen Betriebsmitteln
 - VDE 0160
Ausrüstung von Starkstromanlagen und elektrischen Betriebsmitteln
 - VDE 0550/0551
Bestimmungen für Transformatoren
 - VDE 0700
Sicherheit elektrischer Geräte für den Hausgebrauch und ähnliche Zwecke
 - VDE 0860
Sicherheitsbestimmungen für netzbetriebene elektronische Geräte und deren Zubehör für den Hausgebrauch und ähnliche Zwecke
- Brandverhütungsvorschriften)

-
- Unfallverhütungsvorschriften
 - VBG Nr.4
Elektrische Anlagen und Betriebsmittel

Gefahrenhinweise

Die einzelnen Hinweise haben folgende Bedeutung:



GEFAHR:

Bedeutet, dass eine Gefahr für das Leben und die Gesundheit des Anwenders besteht, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.



ACHTUNG:

Bedeutet eine Warnung vor möglichen Beschädigungen des Gerätes oder anderen Sachwerten, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

Allgemeine Gefahrenhinweise und Sicherheitsvorkehrungen

Die folgenden Gefahrenhinweise sind als generelle Richtlinie für Servoantriebe in Verbindung mit anderen Geräten zu verstehen. Diese Hinweise müssen bei Projektierung, Installation und Betrieb der elektrotechnischen Anlage unbedingt beachtet werden.



GEFAHR:

- *Die im spezifischen Einsatzfall geltenden Sicherheits- und Unfallverhütungsvorschriften sind zu beachten. Der Einbau, die Verdrahtung und das Öffnen der Baugruppen, Bauteile und Geräte müssen im spannungslosen Zustand erfolgen.*
- *Baugruppen, Bauteile und Geräte müssen in einem berührungssicheren Gehäuse mit einer bestimmungsgemäßen Abdeckung und Schutzeinrichtung installiert werden.*
- *Bei Geräten mit einem ortsfesten Netzanschluss müssen ein allpoliger Netztrennschalter und eine Sicherung in die Gebäudeinstallation eingebaut werden.*
- *Überprüfen Sie spannungsführende Kabel und Leitungen, mit denen die Geräte verbunden sind, regelmäßig auf Isolationsfehler oder Bruchstellen. Bei Feststellung eines Fehlers in der Verkabelung müssen Sie die Geräte und die Verkabelung sofort spannungslos schalten und die defekte Verkabelung ersetzen.*
- *Überprüfen Sie vor der Inbetriebnahme, ob der zulässige Netzspannungsbereich mit der örtlichen Netzspannung übereinstimmt.*
- *Damit ein Leitungs- oder Aderbruch auf der Signalseite nicht zu undefinierten Zuständen führen kann, sind entsprechende Sicherheitsvorkehrungen zu treffen.*
- *Treffen Sie die erforderlichen Vorkehrungen, um nach Spannungseinbrüchen und -ausfällen ein unterbrochenes Programm ordnungsgemäß wieder aufnehmen zu können. Dabei dürfen auch kurzzeitig keine gefährlichen Betriebszustände auftreten.*
- *Fehlerstrom-Schutzeinrichtungen nach DIN VDE 0641 Teil 1-3 sind als alleiniger Schutz bei indirekten Berührungen in Verbindung mit speicherprogrammierbaren Steuerungen nicht ausreichend. Hierfür sind zusätzliche bzw. andere Schutzmaßnahmen zu ergreifen.*
- *NOT-AUS-Einrichtungen gemäß EN60204/IEC 204 VDE 0113 müssen in allen Betriebsarten der SPS wirksam bleiben. Ein Entriegeln der NOT-AUS-Einrichtung darf keinen unkontrollierten oder undefinierten Wiederanlauf bewirken.*
- *Damit ein Leitungs- oder Aderbruch auf der Signalseite nicht zu undefinierten Zuständen in der Steuerung führen kann, sind hard- und softwareseitig entsprechende Sicherheitsvorkehrungen zu treffen.*
- *Beim Einsatz der Module muss stets auf die strikte Einhaltung der Kenndaten für elektrische und physikalische Größen geachtet werden.*

Inhalt

1	Kursübersicht und Anforderungen	
1.1	Die Schulungs-Hardware	1-1
2	Die SPS-Hardware	
2.1	Speicherprogrammierbare Steuerungen	2-1
2.1.1	Geschichte und Entwicklung	2-1
2.1.2	Grundsätzliche Anforderungen an eine SPS	2-1
2.1.3	Vergleich zwischen SPS und festverdrahteter Steuerung	2-1
2.1.4	Programmierung in Kontaktplan	2-2
2.1.5	SCADA und MMI	2-2
2.2	Aufbau einer SPS	2-3
2.2.1	Spezifikationen für ein SPS-System	2-3
2.3	Das MELSEC System Q	2-4
2.3.1	Systemkonfiguration	2-4
2.3.2	Baugruppenträger	2-6
2.3.3	Zuordnung der E/A-Adressen beim Hauptbaugruppenträger	2-8
2.3.4	Zuordnung der E/A-Adressen bei Erweiterungsbaugruppenträgern	2-9
2.4	Erweiterungskabel	2-10
2.5	Netzteile	2-10
2.5.1	Auswahl eines geeigneten Netzteils	2-11
2.6	CPU-Module	2-12
2.6.1	Technische Daten	2-13
2.7	Anschluss externer Signale	2-20
2.7.1	Verdrahtung von Ein- und Ausgängen	2-20
2.8	Digitale Ein- und Ausgangsmodule	2-21
2.8.1	Digital-Eingangsmodule	2-22
2.8.2	Digitale Ausgangsmodule	2-30
2.9	Sondermodule	2-38
2.9.1	Analog-Eingangsmodule	2-38
2.9.2	Analog-Ausgangsmodule	2-38
2.9.3	Temperaturregelmodule mit PID-Algorithmus	2-39
2.9.4	High-Speed-Zählermodule	2-39
2.9.5	Positioniermodule	2-40
2.9.6	Schnittstellenmodule zur seriellen Übertragung	2-40
2.9.7	BASIC-programmierbare Schnittstellenmodule	2-41
2.9.8	ETHERNET-Module	2-41
2.9.9	MELSECNET-Module	2-42

2.9.10	Master-Modul/Lokales Modul für CC-Link	2-42
2.9.11	PROFIBUS/DP-Modul	2-43
2.9.12	DeviceNet-Master-Modul QJ71DN91	2-43
2.9.13	Web-Server-Modul	2-44
2.10	SPS-Grundlagen	2-45
2.10.1	Programmier-Software	2-45
2.10.2	Programmverarbeitung in der SPS	2-46
2.10.3	Operanden einer SPS	2-48

3 Programmierung

3.1	Die Norm IEC 61131-3	3-1
3.2	Software-Struktur und Festlegung von Begriffen	3-2
3.2.1	Definition von Begriffen in der Norm IEC61131-3	3-2
3.2.2	Systemvariablen	3-9
3.2.3	System-Label	3-10
3.3	Programmiersprachen	3-11
3.3.1	Text-Editoren	3-11
3.3.2	Grafik-Editoren	3-13
3.4	Datentypen	3-15
3.4.1	Einfache Datentypen	3-15
3.4.2	Komplexe Datentypen	3-15
3.4.3	MELSEC Timer und Counter	3-20

4 Anlegen eines Projekts

4.1	GX IEC Developer starten	4-2
4.2	Programmierung	4-4
4.2.1	Anwendungsbeispiel: Drehtisch	4-4
4.2.2	Anlegen eines neuen Projekts	4-6
4.2.3	Anlegen einer neuen POE	4-8
4.2.4	Deklaration der Globalen Variablen	4-9
4.2.5	Body der POE programmieren	4-14
4.2.6	Anlegen einer neuen Task	4-30
4.2.7	Dokumentation eines Programms	4-34
4.2.8	Projekt prüfen und in den Maschinencode wandeln	4-36
4.2.9	Demonstration des Geführter Editors	4-37
4.3	Programm in die SPS übertragen	4-38
4.3.1	Anschluss des Programmiergeräts an die SPS	4-38
4.3.2	Übertragungseinstellungen (Ports)	4-39
4.3.3	Speicher der SPS formatieren	4-42
4.3.4	Transfer des Projekts in die SPS	4-43

4.4	Ausführung des Programms beobachten	4-45
4.4.1	Beobachten in geteilten Fenstern	4-46
4.4.2	Erkennbarkeit des Monitormodus erhöhen	4-48
4.5	Querverweisliste	4-49
4.6	SPS-Diagnose	4-52
4.7	Dokumentation des Projekts	4-53
5 Programmbeispiel		
5.1	QUIZMASTER	5-1
5.1.1	Vorgehensweise	5-2
5.1.2	Schritte zum Testen des Beispielprogramms „Quizmaster“	5-6
5.1.3	Erläuterung des Beispielprogramms „Quizmaster“	5-6
6 Funktion und Funktionsbaustein		
6.1	Funktionen	6-1
6.1.1	Beispiel: Programmierung einer Funktion	6-1
6.1.2	Verarbeitung von Gleitkommazahlen	6-11
6.2	Programmierung eines Funktionsbausteins	6-15
6.3	Ausführungsarten für Funktionsbausteine	6-23
6.3.1	Makro-Code-Abarbeitung	6-24
6.3.2	EN-Eingang und ENO-Ausgang	6-24
7 Erweiterte Monitorfunktionen		
7.1	Entry-Data-Monitor	7-1
7.1.1	Anpassung des Entry-Data-Monitor	7-3
7.1.2	Zustände von Bit-Operanden in der EDM verändern	7-7
7.2	Header-Monitor	7-8
7.3	Gleichzeitige Anzeige mehrerer Fenster	7-9
7.4	Anzeige von Bit-Gruppen.	7-11
7.5	Werte von Variablen im Body der POE ändern	7-12
7.6	Instanzen von Funktionsbausteinen monitoren	7-13
8 Ein- und Ausgänge steuern		
8	Ein- und Ausgänge steuern	8-1

9	Operanden editieren	
9	Operanden editieren	9-1
10	Online-Modus	
10.1	Online-Change-Modus	10-1
10.2	Online-Programmänderung	10-4
11	Strukturierter Datentyp (SDT)	
11.1	Beispiel für die Anwendung einer SDT	11-2
11.2	Automatisches Füllen von Variablen	11-6
11.3	SDT als Variable eines Funktionsbausteins	11-9
12	Arrays	
12.1	Übersicht	12-1
12.2	Beispiel für ein eindimensionales Array.	12-3
13	Der Umgang mit Bibliotheken	
13.1	Anwenderbibliotheken	13-1
13.1.1	Beispiel: Anlegen einer neuen Bibliothek	13-1
13.1.2	Öffnen einer Bibliothek	13-3
13.1.3	Speichern eines Funktionsbausteins in eine geöffnete Bibliothek	13-4
13.2	Hinweise zu Bibliotheken.	13-7
13.3	Bibliothek in ein Projekt importieren	13-8
13.3.1	Import einer Anwenderbibliothek	13-8
13.3.2	Import eines Funktionsbausteins aus einer Mitsubishi-Bibliothek	13-11
13.3.3	Hilfefunktion der Bibliotheken	13-14
14	Schutz der Programme	
14.1	Vergabe eines Passworts	14-1
14.1.1	Einrichten eines Passworts.	14-1
14.1.2	Sicherheitsstufe ändern	14-2
14.1.3	Sicherheitsstufe einer POE ändern.	14-3

15 Programmieren in Ablaufsprache (AS)

15.1	Was ist Ablaufsprache?	15-1
15.2	Elemente von AS-Programmen	15-2
15.2.1	Transitionen	15-2
15.2.2	Initialisierungs-Schritt	15-2
15.2.3	Abschluss-Schritt	15-2
15.3	Verzweigungen und Sprünge	15-4
15.4	Aktionen	15-5
15.5	Komplexe Transitionen	15-7
15.6	Anzeige von AS-Programmen im Monitormodus	15-8

16 IEC-Anweisungsliste

16.1	Beispiel für IEC-Anweisungsliste (IL)	16-1
16.1.1	Einige nützliche Tipps	16-1
16.2	Mischen von IEC- und MELSEC-Anweisungsliste	16-2

17 Strukturierter Text (ST)

17.1	Elemente für Strukturierten Text	17-1
17.2	Beispiel für Strukturierten Text	17-2

18 Kommunikation über ETHERNET

18.1	Parametrierung eines ETHERNET-Moduls	18-1
18.1.1	Einstellung der Netzwerk-Parameter	18-2
18.2	Einstellungen am PC für ein ETHERNET-Netzwerk	18-8
18.3	Einstellungen zum Zugriff auf die SPS über das ETHERNET	18-9
18.4	Einstellung des Bediengeräts	18-13
18.5	Kommunikation über MX Component	18-16

A Anhang

A.1	Diagnosemerker (SM)	A-1
A.2	Übereinstimmungen zwischen Sonder- und Diagnosemerkern	A-7
A.3	Diagnoseregister (SD)	A-13
A.3.1	Programmzyklusinformationen	A-31

1 Kursübersicht und Anforderungen

Dieses Schulungshandbuch soll eine Einführung zu den speicherprogrammierbaren Steuerungen des MELSEC System Q von Mitsubishi Electric geben und Ihnen als Um- oder Einsteiger die ersten Schritte mit der Programmier-Software GX-IEC Developer (Version 7) erleichtern.

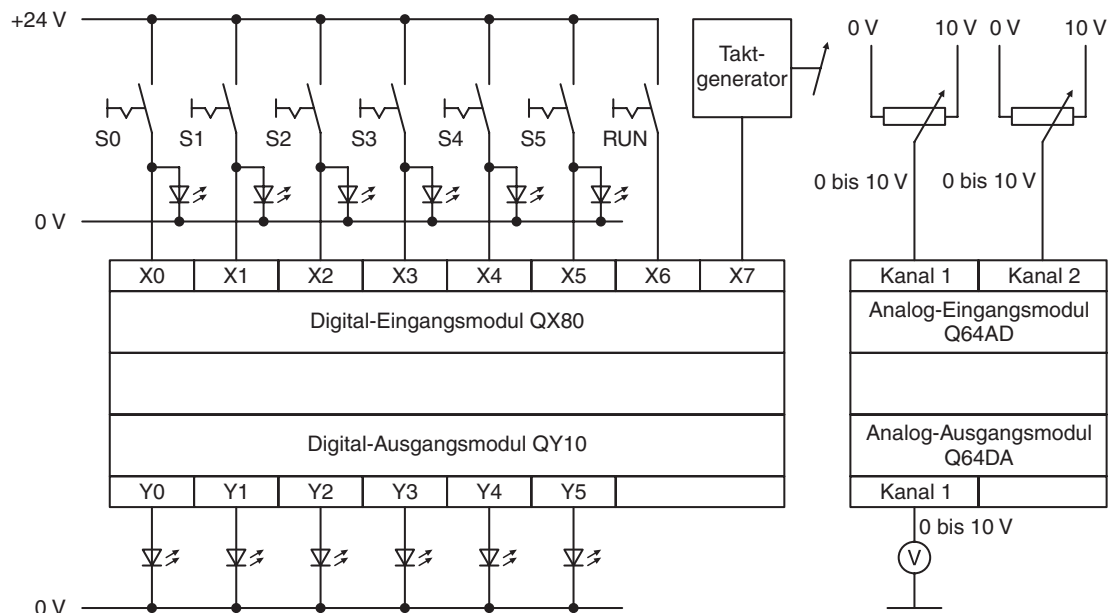
Nach einer Übersicht über die Komponenten des MELSEC System Q in Kapitel 2 wird in den restlichen Kapiteln dieses Handbuchs auf die Programmierung nach IEC61131-3 eingegangen. Anhand konkreter Beispiele wird die Hardware-Konfiguration und die Handhabung des GX-IEC Developers, bis hin zur Fehlerdiagnose und ETHERNET-Anbindung, demonstriert.

Der Umgang mit einem Personal-Computer und einem Microsoft Windows® Betriebssystem sollte Ihnen vertraut sein.

1.1 Die Schulungs-Hardware

Für Schulungen werden verschiedene Schulungs-Racks mit unterschiedlicher Bestückung verwendet. Für die Beispiele in diesem Handbuch wird ein Schulungs-Rack mit der folgenden Konfiguration verwendet:

- 6 Schalter zur Eingabe von digitalen Signalen: X0-X5
- Einstellbarer Takt-Eingang (1–100 Hz und 0,1– 10 kHz): X7
- 6 LEDs zur Anzeige der Zustände von digitaler Ausgängen: Y0-Y5
- 4 Analog-Eingangskanäle: Modul Q64AD mit der Kopfadresse 30H
- 4 Analog-Ausgabekanal: Q64DA mit der Kopfadresse 40H.



Falls Schulungsracks mit anderen Konfigurationen oder Adresszuordnungen verwendet werden, müssen die Programmbeispiele dieses Handbuchs entsprechend angepasst werden.

2 Die SPS-Hardware

2.1 Speicherprogrammierbare Steuerungen

2.1.1 Geschichte und Entwicklung

Die erste speicherprogrammierbare Steuerung (SPS) wurde 1968 von der Firma Bedford Associates, die von Richard Morley gegründet wurde, entwickelt. Von der Bezeichnung für diese erste SPS – *Modular Digital Controller* – leitet sich der Name der Firma MODICON ab.

Speicherprogrammierbare Steuerungen wurden als Ersatz für umfangreiche Schütz-Steuerungen entwickelt. Bei diesen Anlagen sind Änderungen im Steuerungsablauf meist nur mit einem großen Verdrahtungsaufwand oder dem Austausch von Bauteilen zu realisieren. Bei einer SPS dagegen genügt häufig eine Änderung des Programms im Speicher der Steuerung.

Die Entwicklung der Mikroprozessoren ab ca. 1970 und die immer weiter zunehmende Verarbeitungsgeschwindigkeit ermöglichte den Einsatz von speicherprogrammierbaren Steuerungen auch in komplexen Anwendungen und die Übernahme weiterer Funktionen. Heute ist es durchaus selbstverständlich, dass eine SPS sozusagen das Herz der Automatisierung darstellt und dieses System mit einer Leitebene (SCADA, Supervisory Control And Data Acquisition), Bediengeräten (MMI, Mensch-Maschine-Interface) oder Expertensystemen verbunden ist. Die Anforderungen an eine SPS umfassen neben der Steuerung auch die Datenverarbeitung und die Prozessleittechnik.

2.1.2 Grundsätzliche Anforderungen an eine SPS

- Eine SPS muss sich leicht programmieren lassen. Programmänderungen müssen vor Ort ebenso einfach möglich sein.
- Wartungs- und reparaturfreundlich - vorzugsweise durch modularen Aufbau
- Eine SPS muss den rauen Einsatzbedingungen in einer industriellen Umgebung mechanisch und elektrisch gewachsen sein.
- Eine SPS muss kleiner als vergleichbare Schütz- oder konventionelle Steuerungen sein.
- Eine SPS muss preiswerter als vergleichbare Schütz- oder konventionelle Steuerungen sein.

2.1.3 Vergleich zwischen SPS und festverdrahteter Steuerung

Merkmal	SPS	Festverdrahtete Steuerung mit Schützen
Kosten pro Funktion	Niedrig	Niedrig - wenn in der entsprechenden Steuerung mehr als 10 Schütze verwendet werden.
Abmessungen	Sehr kompakt	Sperrig
Verarbeitungsgeschwindigkeit	Schnell	Langsam
Widerstand gegen elektromagnetische Störungen	Gut	Hervorragend
Aufbau	Einfache Programmierung	zeitaufwändige Verdrahtung
Komplexe Funktionen	Möglich	Nicht möglich
Änderung des Funktionsablaufs	Sehr einfach	Sehr schwierig (Verdrahtungsänderung)
Wartungsfreundlichkeit	Hervorragend (Eine SPS fällt selten aus.)	Schlecht - Schütze erfordern eine ständige Wartung

2.1.4 Programmierung in Kontaktplan

Eine SPS muss von Technikern und Betriebselektrikern gewartet werden können. Aus diesem Grund wurde die Kontaktplanprogrammierung entwickelt. Die Elemente dieser Programmiersprache erinnern an die Schaltzeichen, die auch bei Schützsteuerungen verwendet werden und die jeder kennengelernt hat, der eine elektrotechnische Ausbildung absolviert hat.

Bei den frühen SPS-Programmen gab es entweder keine oder nur sehr eingeschränkte Möglichkeiten zur Dokumentation der Programme. Dadurch, dass meist nur Adressen oder einfache Kommentare angegeben werden konnten, waren umfangreiche Programme schwer verständlich. Mit der Entwicklung fortschrittlicher Programmierwerkzeuge, wie den **GX Developer** von Mitsubishi, wurden die Dokumentationsmöglichkeiten drastisch verbessert.

Lange Zeit gab es keinen einheitlichen Standard bei der Programmierung von speicherprogrammierbaren Steuerungen. Dieser Standard wurde 1998 mit der Einführung der Norm **IEC 61131-3** geschaffen. Die Programmier-Software **GX-IEC Developer** von Mitsubishi Electric ermöglicht die strukturierte Programmierung nach IEC61131-3.

2.1.5 SCADA und MMI

Bei den ersten SPS erfolgten Eingaben durch den Bediener genauso wie bei konventionellen Steuerungen mit Tastern oder Schaltern. Zur Anzeige wurden Meldeleuchten verwendet.

Die Einführung des Personal Computers (PC) in den achtziger Jahren des letzten Jahrhunderts ermöglichte die Entwicklung von PC-basierenden Ein-/Ausgabegeräten. Wird ein PC mit einer speziellen Software eingesetzt, spricht man auch von SCADA (Supervisory Control And Data Acquisition), was ein System zur Bedienung und Datenerfassung bezeichnet.

Spezielle Bediengeräte werden als MMI (Mensch-Maschine-Interface) bezeichnet, weil sie die Schnittstelle zwischen dem zu steuernden Prozess und dem Bediener bilden. Heute haben sich SCADA und MMI zur Bedienung durchgesetzt und erhöhen in Verbindung mit einer SPS die Bedienerfreundlichkeit.

Mitsubishi bietet eine große Auswahl an MMI-Produkten und SCADA-Lösungen, passend für alle Anwendungen.



2.2 Aufbau einer SPS

Im Gegensatz zu einer Steuerung, deren Funktion nur durch die Verdrahtung bestimmt wird, wird bei einer speicherprogrammierbaren Steuerung (SPS) die Funktion durch ein Programm festgelegt. Zwar benötigt auch eine SPS zur Verbindung mit der Außenwelt eine Verdrahtung, der Inhalt des Programmspeichers kann aber jederzeit geändert und das Programm an verschiedene Steuerungsaufgaben angepasst werden.

Bei speicherprogrammierbaren Steuerungen werden Daten eingegeben, verarbeitet und die Verarbeitungsergebnisse wieder ausgegeben. Dieser Prozess gliedert sich in:

- eine Eingabeebene,
- eine Verarbeitungsebene und
- eine Ausgabeebene.

Eingabeebene

Die Eingabeebene dient zur Übergabe von Steuersignalen, die von Schaltern, Tastern oder Sensoren stammen, an die Verarbeitungsebene.

Die Signale dieser Bauelemente entstehen im Steuerungsprozess und werden als logischer Zustand den Eingängen zugeführt. Die Eingabeebene übergibt die Signale in aufbereiteter Form der Verarbeitungsebene.

Verarbeitungsebene (CPU)

Die von der Eingabeebene erfassten und aufbereiteten Signale werden in der Verarbeitungsebene durch ein gespeichertes Programm verarbeitet und logisch verknüpft. Der Programmspeicher der Verarbeitungsebene ist frei programmierbar. Eine Änderung des Verarbeitungsablaufs ist jederzeit durch Änderung oder Austausch des gespeicherten Programms möglich.

Ausgabeebene

Die Resultate, die aus der Verarbeitung der Eingangssignale im Programm entstanden sind, beeinflussen in der Ausgangsebene die an den Ausgängen angeschlossenen Schaltglieder wie z. B. Schütze, Meldeleuchten, Magnetventile usw..

2.2.1 Spezifikationen für ein SPS-System

Im folgenden sind stichpunktartig einige Überlegungen aufgeführt, die bei der Konfiguration einer SPS berücksichtigt werden müssen.

Externe Geräte, Ein- und Ausgänge

- Anforderungen an Ein- und Ausgänge
- Signalspannung: 24V Gleichspannung oder 110V/240 V Wechselspannung?
- Bei 24 V DC Gleichspannung: werden plus- oder minusschaltende Sensoren an die Eingänge angeschlossen?
- Ausgangstyp: Transistor (plus- oder minusschaltend), Triac, Relais oder potentialfreier Relaiskontakt?

Versorgungsspannung

- 24V Gleichspannung oder 110V/240 V Wechselspannung?

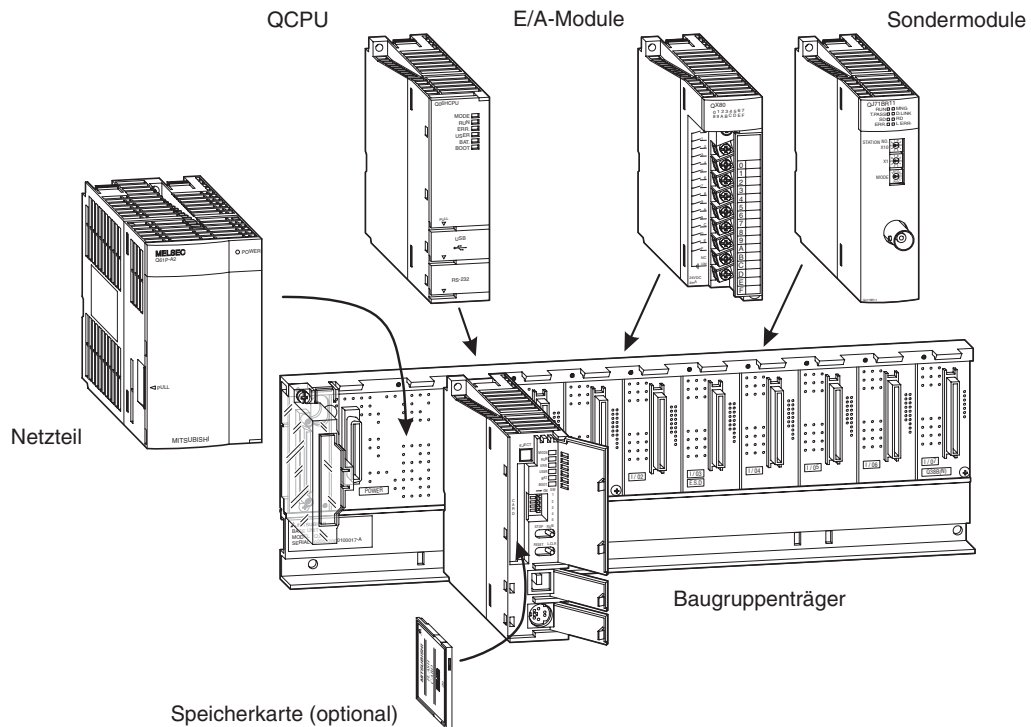
Sondermodule

- Anzahl der Sondermodule (z. B. Analog-, Netzwerk oder Schnittstellenmodule) im System
- Ist für Sondermodule eine externe Spannungsversorgung erforderlich?

2.3 Das MELSEC System Q

Der folgende Abschnitt gibt einen Überblick über den Aufbau einer speicherprogrammierbaren Steuerung des MELSEC System Q.

2.3.1 Systemkonfiguration



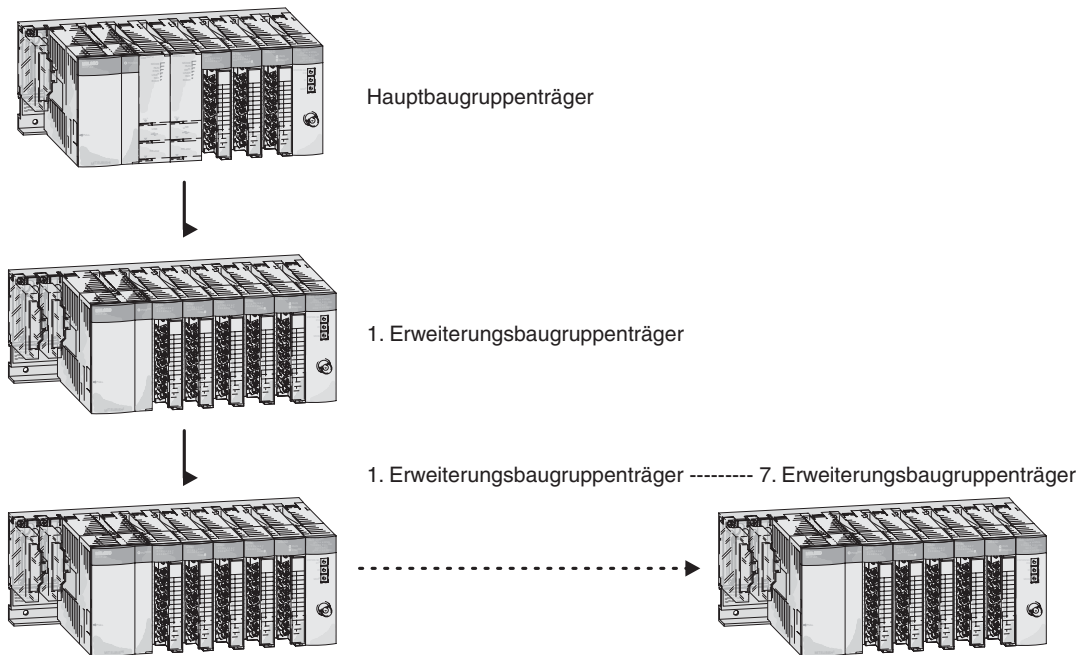
Die CPU und die Module werden auf einem Hauptbaugruppenträger montiert. Über die Rückwand des Baugruppenträgers können die einzelnen Module miteinander kommunizieren. Die Stromversorgung des gesamten Systems übernimmt ein Netzteil, das ebenfalls auf dem Baugruppenträger installiert wird.

Hauptbaugruppenträger sind in verschiedenen Ausführungen mit 3 bis 12 Steckplätzen für E/A- oder Sondermodule erhältlich. Durch den Anschluss von Erweiterungsbaugruppenträgern mit zusätzlichen Steckplätzen kann ein System ausgebaut werden.

Freie Steckplätze auf einem Baugruppenträger können durch Leermodule vor Verschmutzung oder mechanischen Beschädigungen geschützt werden. Zusätzlich können mit einem Leermodule E/A-Adressen für einen späteren Ausbau des Systems reserviert werden.

Bei der Verdrahtung von umfangreichen Anlagen oder bei Maschinen mit modularem Aufbau bieten dezentrale Ein- und Ausgänge (E/A-Stationen) Vorteile, die unmittelbar vor Ort angeordnet sind. Dabei können die Verbindungen zwischen den Ein- oder Ausgängen und den Sensoren bzw. den Schaltgliedern kurz gehalten werden. Zur Verbindung zwischen einer dezentralen E/A-Station und dem System mit der SPS-CPU wird nur ein Netzkabel benötigt.

Hauptbaugruppenträger und Erweiterungsbaugruppenträger



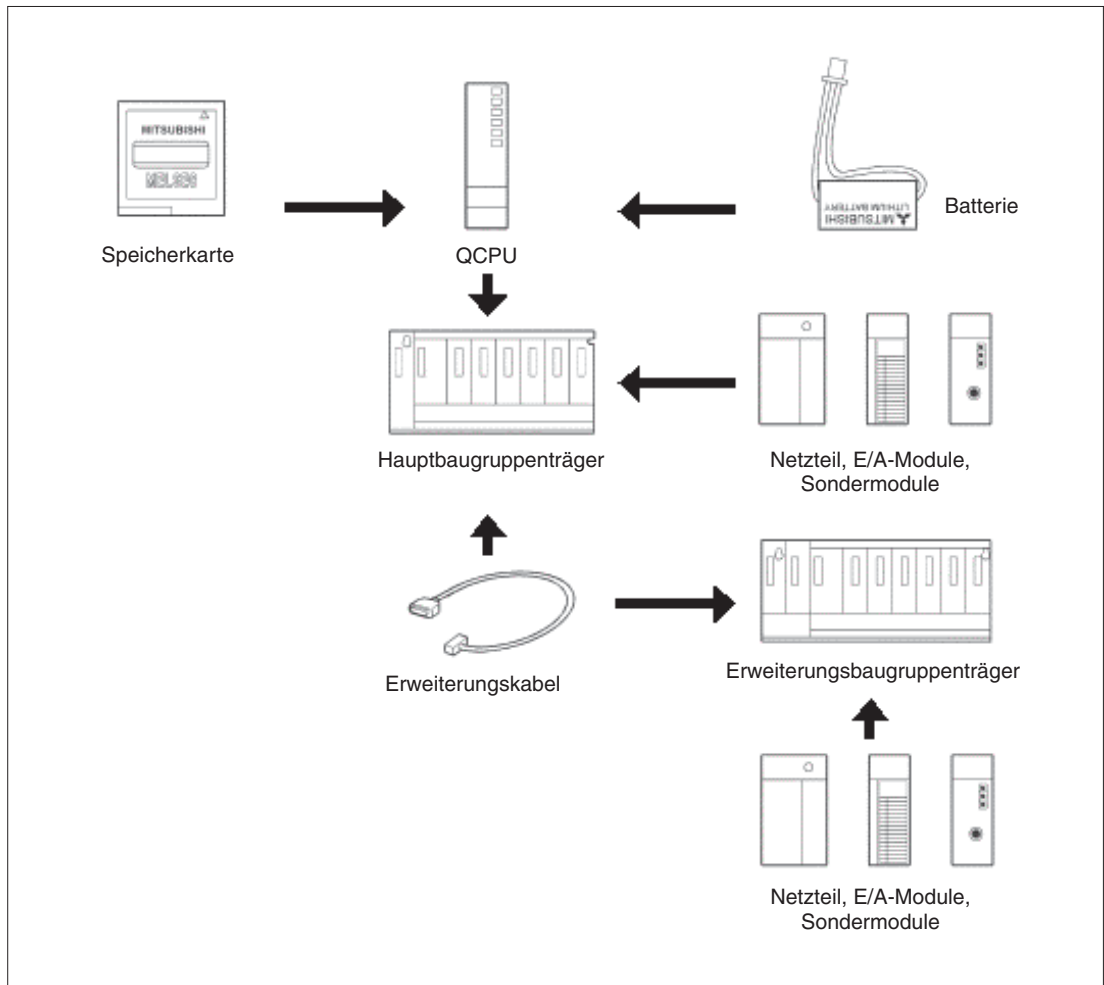
Der Hauptbaugruppenträger und die Erweiterungsbaugruppenträger werden einfach durch ein Kabel miteinander verbunden. Dieses Erweiterungskabel versorgt auch den Erweiterungsbaugruppenträger mit Spannung, falls dieser kein eigenes Netzteil besitzt.

An einem Hauptbaugruppenträger des MELSEC System Q können bis zu sieben Erweiterungsbaugruppenträger mit bis zu 64 Modulen angeschlossen werden. Die Länge aller Erweiterungskabel darf 13,2 m nicht überschreiten.

Beim Auswahl des Netzteils muss die Stromaufnahme der Ein- und Ausgangsmodule, der Sondermodule und der peripheren Geräte berücksichtigt werden. Falls erforderlich, muss ein Erweiterungsbaugruppenträger mit einem weiteren Netzteil verwendet werden.

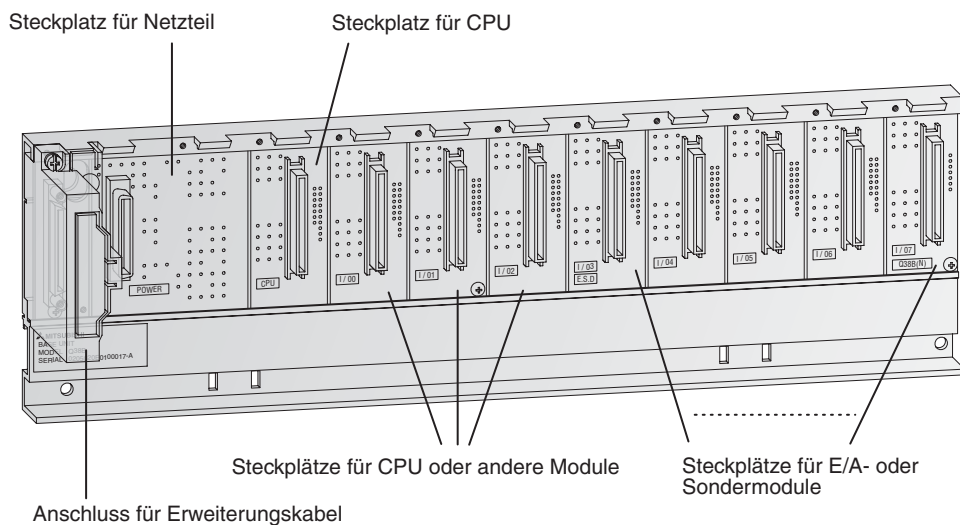
Anzahl der anschließbaren Erweiterungsbaugruppenträger

- An einem Hauptbaugruppenträger mit einer Q00CPU oder Q01CPU können bis zu 4 Erweiterungsbaugruppenträger mit bis zu 24 E/A-Modulen angeschlossen werden.
- Eine SPS des System Q mit einer Q02-, Q02H-, Q06H-, Q12H- oder Q25HCPU kann mit bis zu 7 Erweiterungsbaugruppenträgern und 64 E/A-Modulen ausgebaut werden.



2.3.2 Baugruppenträger

Die Hauptbaugruppenträger nehmen ein Netzteil, ein oder mehrere CPU-Module und E/A- oder Sondermodule auf. In den Erweiterungsbaugruppenträgern können E/A- und Sondermodule installiert werden. Die Baugruppen werden entweder direkt, z. B. im Schaltschrank, oder mit Hilfe von Adaptoren auf einer DIN-Schiene installiert.



In den folgenden Tabellen sind alle erhältlichen Baugruppenträger aufgeführt.

Merkmal	Hauptbaugruppenträger				
	Q33B	Q35B	Q38B	Q38RB	Q312B
Anzahl der Steckplätze für Netzteile	1	1	1	2*	1
Anzahl der Steckplätze für E/A- oder Sondermodule	3	5	8	8	12

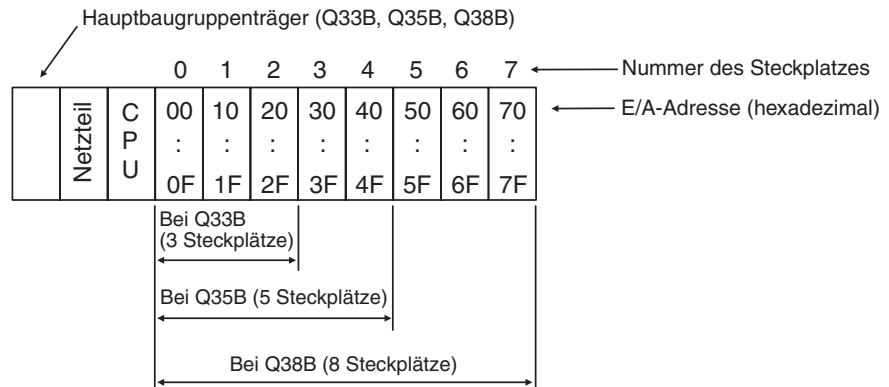
* In diesem Hauptbaugruppenträger können redundante Netzteile verwendet werden.

Merkmal	Erweiterungsbau­grup­pen­trä­ger						
	Q52B	Q55B	Q63B	Q65B	Q68B	Q68RB	Q612B
Anzahl der Steckplätze für Netzteile	—	—	1	1	1	2*	1
Anzahl der Steckplätze für E/A- oder Sondermodule	2	5	3	5	8	8	12

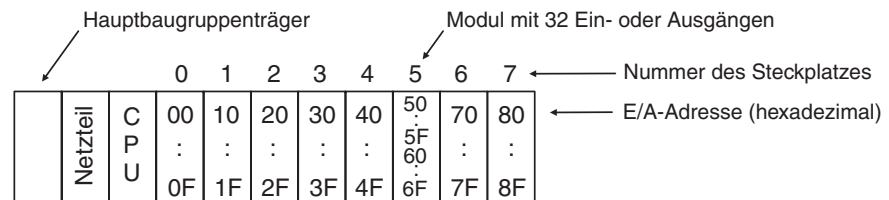
* In diesem Erweiterungsbau­grup­pen­trä­ger können redundante Netzteile verwendet werden.

2.3.3 Zuordnung der E/A-Adressen beim Hauptbaugruppenträger

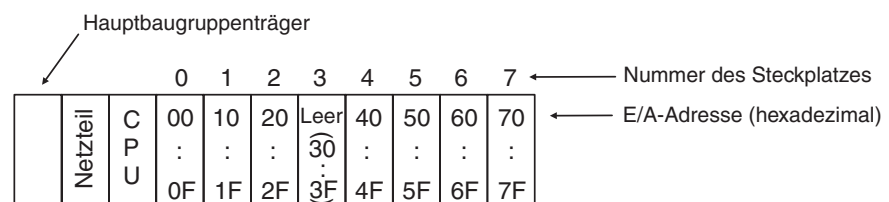
Den Ein- und Ausgängen einer SPS müssen Adressen zugewiesen werden, damit sie im Programm angesprochen werden können. Die Adressen der Ein- und Ausgänge der auf dem Hauptbaugruppenträger installierten E/A-Module und die Kopfadressen der Sondermodule werden den Steckplätzen automatisch zugewiesen. Die Zuordnung kann jedoch auch vom Anwender vorgenommen werden.



Bei der Zuordnung der E/A-Adressen wird vom System vorausgesetzt, dass auf allen Steckplätzen Module mit 16 Ein- oder Ausgängen installiert sind. Die E/A-Adressen erhöhen sich daher mit jedem Steckplatz um den Wert 16 (0 bis F Hexadezimal). Falls ein Steckplatz ein Modul mit zum Beispiel 32 Ein- oder Ausgängen enthält (wie in der folgenden Abbildung Steckplatz 5), wird dies aber auch berücksichtigt und die Adressen der folgenden Steckplätze werden entsprechend verschoben.



16 E/A-Adressen werden auch einem leeren Steckplatz zugewiesen. Die folgende Abbildung zeigt eine Konfiguration, bei der auf dem Steckplatz 3 kein E/A-Modul installiert wurde.

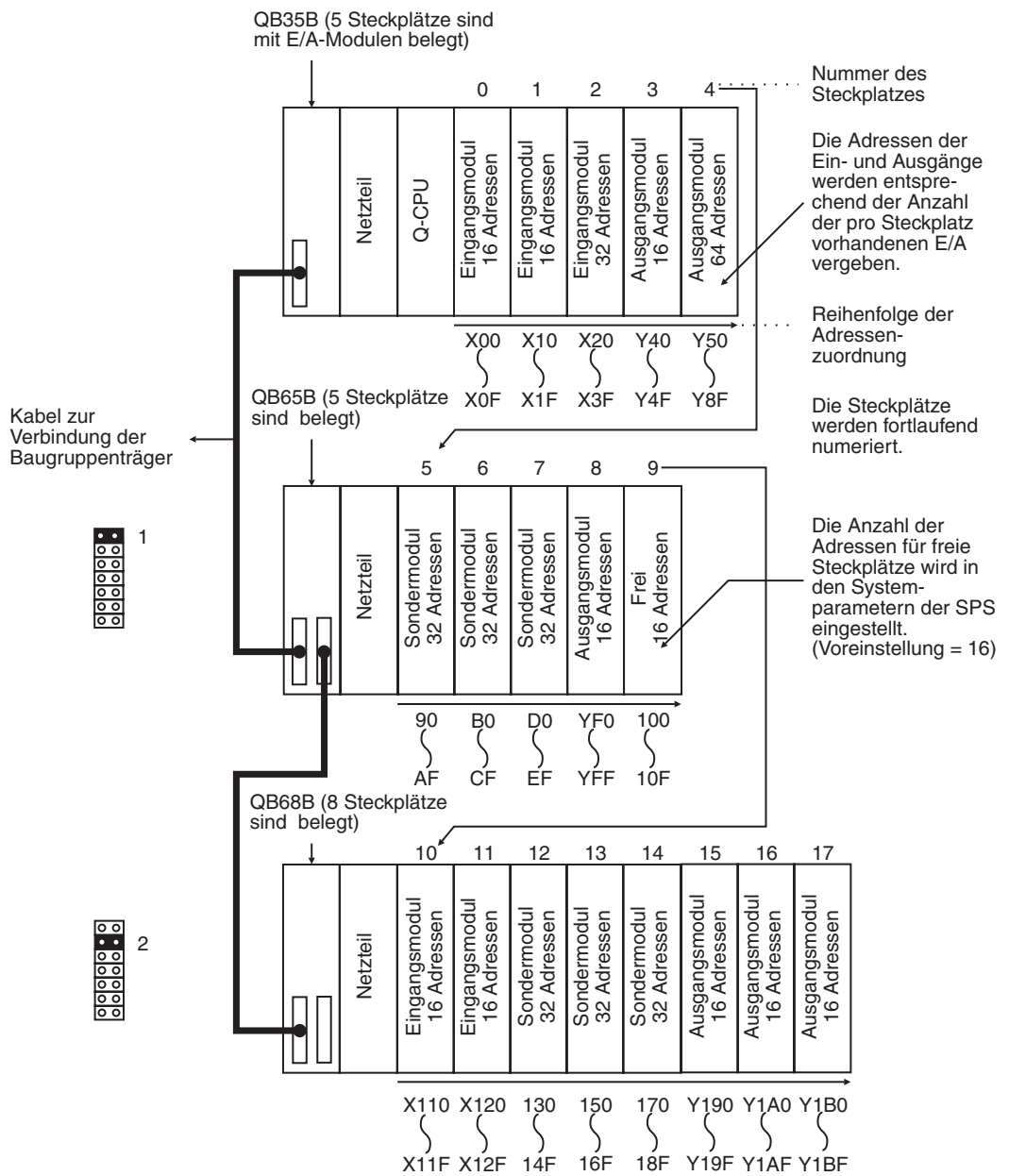


2.3.4 Zuordnung der E/A-Adressen bei Erweiterungsbaugruppenträgern

Falls zusätzlich zu den Steckplätzen auf dem Hauptbaugruppenträger weitere Steckplätze benötigt werden, können Erweiterungsbaugruppenträger angeschlossen werden. Die Zuordnung der E/A-Adressen erfolgt nach den folgenden Regeln:

- Die E/A-Adressen der Steckplätze der Erweiterungsbaugruppenträger werden in aufsteigender Reihenfolge hexadezimal vergeben.
- Die Adressierung des Hauptbaugruppenträgers wird mit dem ersten Steckplatz des ersten Erweiterungsbaugruppenträgers nach dem Hauptbaugruppenträger fortgesetzt.

Die folgende Abbildung soll die Adressierung verdeutlichen:



2.4 Erweiterungskabel

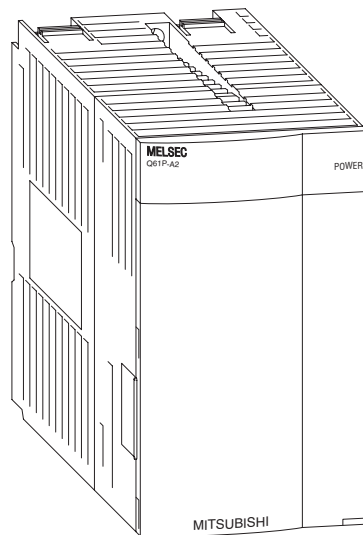
Mit den Erweiterungskabeln werden Haupt- und Erweiterungsbaugruppenträger verbunden.

Erweiterungskabel	QC05B	QC06B	QC12B	QC30B	QC50B	QC100B
Länge	0,45 m	0,50 m	1,2 m	3,0 m	5,0 m	10,0 m

Die maximale Länge aller Verbindungskabel darf 13,2 m nicht überschreiten.

Zum Anschluss der Erweiterungsbaugruppenträger ohne eigenes Netzteil (Q52B, Q55B) wird das Kabel QC05B empfohlen.

2.5 Netzteile



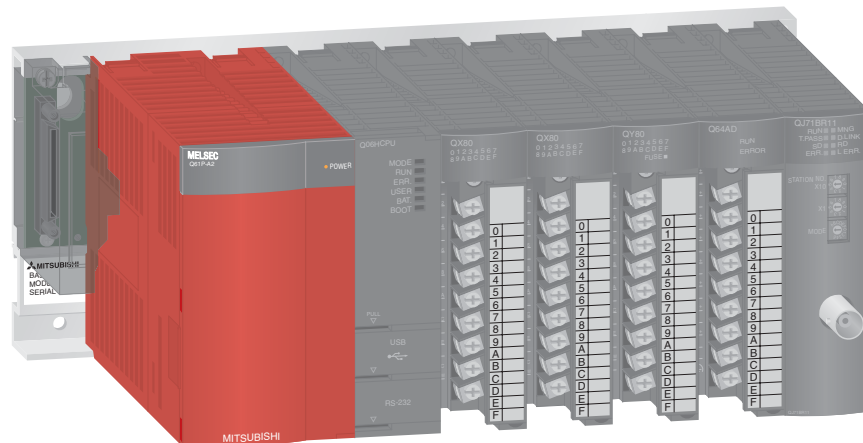
Das System Q wird mit einer Gleichspannung von 5 Volt betrieben. Es stehen Netzteile mit Eingangsspannungen von 24 V DC oder 100 bis 240 V AC zur Verfügung.

Merkmal	Q63P	Q61P-A1	Q61P-A2	Q62P	Q64P
Eingangsspannung	24 V DC	100 – 120 V AC	200 – 220 V AC	100 – 240 V AC	100 – 120 V AC 200 – 240 V AC
Leistungsaufnahme	45 W	105 VA	105 VA	105 VA	105 VA
Ausgangsspannung	5 V DC	5 V DC	5 V DC	5 V DC, 3 A	5 V DC
Ausgangsstrom	6 A	6 A	6 A	24 V DC, 0,6 A	8,5 A

2.5.1 Auswahl eines geeigneten Netzteils

Die Stromaufnahme der auf den Baugruppenträger installierten Module darf den Nennstrom, den das Netzteil liefern kann, nicht überschreiten. Falls dies der Fall ist, muss die Anzahl der Module im Baugruppenträger reduziert werden.

Beispiel zur Berechnung der Stromaufnahme:

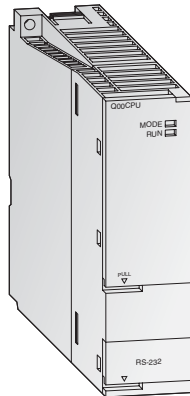


Modul	Art des Moduls	Stromaufnahme
Q06HCPU	CPU-Modul	0,64 A
QX80	Digital-Eingangsmodul	0,16 A
QX80	Digital-Eingangsmodul	0,16 A
QY80	Digital-Ausgangsmodul	0,008 A
Q64AD	Analog-Eingangsmodul	0,63 A
QJ71BR11	MELSECNET/H-Modul	0,75 A
Gesamtstromaufnahme		2,42 A

Die Summe der Stromaufnahmen beträgt 2,42 A und liegt damit unter dem Nennstrom von 6 A, den das Netzteil liefern kann. Beim Betrieb der SPS werden daher keine Probleme auftreten.

2.6 CPU-Module

Basis-SPS-CPU



Die CPU-Module der MELSEC System Q sind als Single-CPU und als Multiprozessor-CPU verfügbar, wodurch eine große Einsatzvielfalt erreicht wird. Die Leistungsfähigkeit der Steuerung wächst dabei mit der Applikation durch einfaches Auswechseln der CPU (nicht bei Q00JCPU).

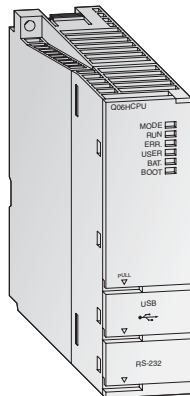
Während Q00CPU und Q01CPU klassische Modul-CPU sind, bildet die Q00JCPU eine untrennbare Einheit aus CPU, Netzteil und Baugruppenträger und ermöglicht so den preiswerten Einstieg in die modulare SPS-Technik.

Die Standard-CPU wurden speziell für Applikationen entwickelt, bei denen ein einfach zu realisierender und kompakter Systemaufbau im Vordergrund steht.

Besondere Merkmale:

- Jede CPU ist mit einer RS232C-Schnittstelle zur einfachen Programmierung und Überwachung durch einen PC oder ein Bediengerät ausgestattet.
- Integrierte Flash-ROMs für Speicherbetrieb ohne zusätzlichen Speicherkartensteckplatz
- Verarbeitung der Ein- und Ausgänge als Prozessabbild

Hochleistungs-SPS-CPU



Bei den Hochleistungs-CPU stehen hohe Verarbeitungsgeschwindigkeit und Erweiterungsfähigkeit im Vordergrund. Sie verfügen über eine Vielfalt von Funktionen und eine nochmals optimierte Programmier- und Debugging-Umgebung, um eine flexible Reaktion auf alle Systeme sicherzustellen.

Die beiden Prozess-CPU Q12PHCPU und Q25PHCPU verfügen über erweiterte Regelungsfunktionen mit 2 Freiheitsgraden, kaskadierter PID und Auto-Tuning-Funktion. Zusätzlich stehen hier 52 verschiedene Prozess-Befehlsfunktionen zur Verfügung. Die Anzahl der PID-Regelkreise ist nicht limitiert.

Besondere Merkmale:

- Jede Multi-Prozessor-H-CPU ist mit einer USB-Schnittstelle zur einfachen und schnellen Programmierung und Überwachung durch einen PC ausgestattet.
- Verarbeitung der Ein- und Ausgänge als Prozessabbild
- Gleitkommaarithmetik in Übereinstimmung mit IEEE 754
- Direktes Ansprechen und Bearbeiten von PID-Regelkreisen
- Mathematische Funktionen, wie z. B. trigonometrische, Exponential- und Logarithmusfunktionen
- Modulaustausch im RUN-Betrieb (mit Prozess-CPU)
- Multiprozessorbetrieb ist mit bis zu 4 CPU-Modulen möglich.

2.6.1 Technische Daten

Merkmal		Q00CPU	Q01CPU	Q02CPU	Q02HCPU	Q06HCPU	Q12HCPU	Q25HCPU
Art der Steuerung		Zyklische Bearbeitung des gespeicherten Programms						
E/A-Steuerung		Auffrischung des Prozessabbildes						
Programmiersprache		IEC Kontaktplan (KOP), Anweisungsliste (AWL), Funktionsbausteinsprache (FUB), Strukturierter Text (ST), Ablaufsprache (AS)						
Verarbeitungs- geschwindigkeit	LD	160 ns	100 ns	79 ns	34 ns			
	MOV	560 ns	350 ns	237 ns	102 ns			
	Gemischte Anweisungen pro µs	2,0	2,7	4,4	10,3			
	Gleitkomma-addition	27 µs*		1,8 µs	0,78 µs			
Anzahl der Anweisungen(ohne Anweisungen für intelligente Sondermodule)		249		363				
Rechenanweisungen für Fließkommazahlen		Möglich*		Möglich				
Anweisungen zur Verarbeitung von Zeichenfolgen		nur \$MOV ist möglich		Möglich				
Anweisungen zur PID-Regelung		Möglich*		Möglich				
Anweisungen für Sonderfunktionen (Trigonometrische Funktionen, Wurzel- und Logarithmusberechnung etc.)		Möglich*		Möglich				

* Nur bei einer Q00/Q01 CPU ab der Funktionsversion B (Die ersten 5 Stellen der Seriennummer lauten in diesem Fall mindestens „04122“.)

Merkmal		Q00CPU	Q01CPU	Q02CPU	Q02HCPU	Q06HCPU	Q12HCPU	Q25HCPU
Konstante Zykluszeit (Start des Programms in festen Intervallen)		1 bis 2000 ms (parametrierbar in Schritten von 1 ms)		0,5 bis 2000 ms (parametrierbar in Schritten von 0.5 ms)				
Programmspeicher (Anzahl der Schritte)		8 k	14 k	28 k		60 k	124 k	252 k
Speicher- kapazität	Integrierter Programmspeicher (Laufwerk 0)	94 kByte		112 kByte		240 kByte	496 kByte	1 MB
	RAM-Speicherkarte (Laufwerk 1)	—		Abhängig von der installierten Speicherkarte (max. 1 MB)				
	RAM-Speicherkarte (Laufwerk 2)	—		Abhängig von der installierten Speicherkarte (max. 4 MB bei Flash-ROM, max. 32 MB bei ATA-Speicherkarten)				
	Integriertes RAM (Laufwerk 3)	128 kByte*		64 kByte			256 kByte	
	Integriertes ROM (Laufwerk 4)	94 kByte		112 kByte		240 kByte	496 kByte	1 MB
	Gemeinsamer Speicherbereich für Multi-Prozessorbetrieb	1 kByte**		8 kByte				
E/A- Adressen	Gesamt (inkl. dezentralen E/A)	2048		8192				
	Lokale E/A	1024		4096				

* 64 kByte bei Funktionsversion A

** Nur bei einer Q00/Q01 CPU ab der Funktionsversion B (Die ersten 5 Stellen der Seriennummer lauten in diesem Fall mindestens „04122“.)

Anzahl der Operanden

Operand (Symbol)	Q00CPU	Q01CPU	Q02CPU	Q02HCPU	Q06HCPU	Q12HCPU	Q25HCPU
Merker (M)	8192		8192				
Latch-Merker (L)	2048		8192				
Link-Merker (B)	2048		8195				
Timer (T)	512		2048				
Remanente Timer (ST)	0		0				
Counter (C)	512		1024				
Datenregister (D)	11136		12288				
Link-Register (W)	2048		8196				
Fehlermerker (F)	1024		2048				
Flankenmerker (V)	1024		2048				

Die vorstehende Tabelle zeigt die voreingestellten Operanden. Die Anzahl der Operanden kann in den Parametern verändert werden.

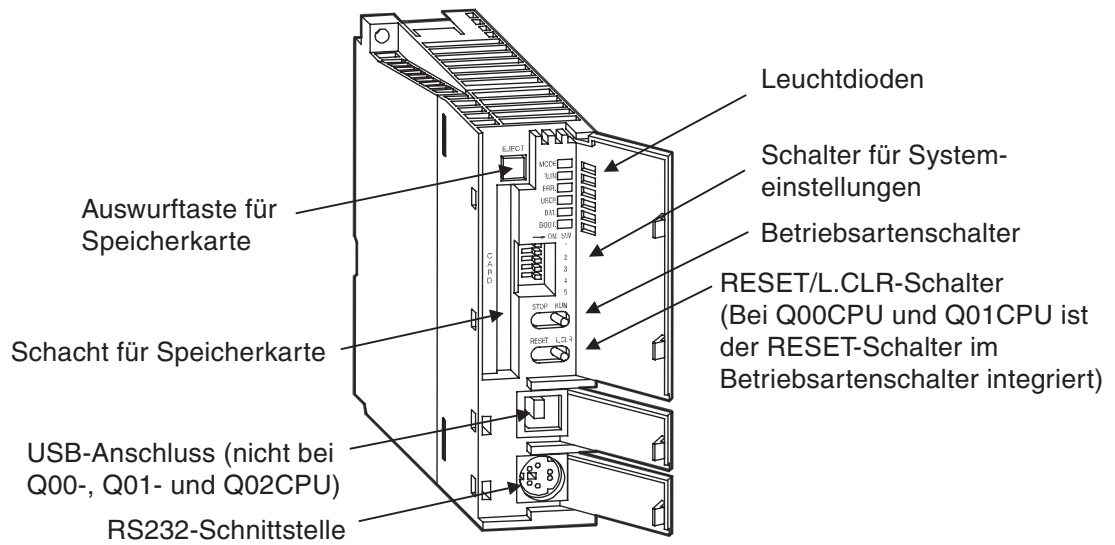
Operand (Symbol)	Q00CPU	Q01CPU	Q02CPU	Q02HCPU	Q06HCPU	Q12HCPU	Q25HCPU
File-Register (R)	32768		32768 (bei Verwendung des integrierten Speichers)			131072 (integrierter Speicher)	
Link-Sondermerker (SB)	1024		2048				
Link-Sonderegister (SW)	1024		2048				
Schrittmerker (S)	2048 (S0 bis 127 / Block)		8192				
Index-Register (Z)	10		16				
Pointer (P)	300		4096				
Interrupt-Pointer (D)	128		256				
Sondermerker (SM)	1024		2048				
Sonderegister (SD)	1024		2048				
Funktions-Eingänge	16		16				
Funktions-Ausgänge	16		16				
Funktions-Register	5		5				

Die Anzahl der File-Register kann bei den CPU-Typen Q02, Q02H, Q06H, Q12H und Q25H durch Verwendung einer Speicherkarte auf bis zu 1.041.408 Adressen erhöht werden.

Bedienelemente, Schnittstellen und Stromaufnahme der CPU-Module

Merkmal	Q00CPU	Q01CPU	Q02CPU	Q02HCPU	Q06HCPU	Q12HCPU	Q25HCPU
Funktionen des Betriebsarschalters	RUN, STOP, RESET		RUN, STOP, RESET, L.CLR (Löschen des Latch-Bereiches)				
Schnittstellen	RS232		RS232	RS232, USB			
Steckplatz für Speicherkarte	—		1 Steckplatz				
LED zur Anzeige des Betriebszustandes	RUN, ERR.		MODE, RUN, ERR., USER, BAT., BOOT, POWER				
Stromaufnahme bei 5V DC	0,25 A	0,27 A	0,60 A	0,64 A			

Bedienelemente der CPU-Module



Leuchtdioden

- MODE- und RUN-LED

<p>Q06HCPU</p> <p>MODE <input checked="" type="checkbox"/></p> <p>RUN <input checked="" type="checkbox"/></p> <p>ERR. <input type="checkbox"/></p> <p>USER <input type="checkbox"/></p> <p>BAT. <input type="checkbox"/></p> <p>BOOT <input type="checkbox"/></p>	<p>Grün: Q-Modus</p> <p>EIN: CPU ist in der Betriebsart RUN</p> <p>AUS: CPU ist in der Betriebsart STOP oder ein Fehler, der die Programmbearbeitung unterbricht, ist aufgetreten</p> <p>BLINKT: Nach einer Programm- oder Parameteränderung wurde der Betriebsartenschalter auf RUN geschaltet, die CPU ist aber noch nicht in der Betriebsart RUN</p>
---	---

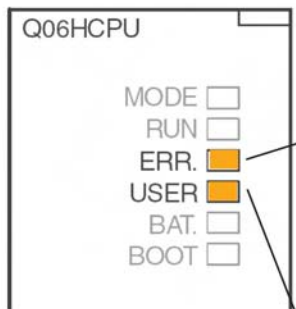
CPU nach einer Programm- oder Parameteränderung, die in der Betriebsart STOP ausgeführt wurde, in „RUN“ schalten:

- ① 1. RESET/L.CLR-Schalter in Stellung „RESET“ schalten.
- ② 2. RUN/STOP-Schalter in Stellung „RUN“ schalten.

oder, falls kein Reset ausgeführt werden soll:

- ① RUN/STOP-Schalter von „STOP“ in Stellung „RUN“ schalten
- ② RUN/STOP-Schalter dann wieder auf „STOP“ schalten
- ③ RUN/STOP-Schalter auf „RUN“ schalten.

● ERR.- und USER-LED



EIN: Ein Fehler, der nicht zum Programmabbruch führt, wurde bei der Selbstdiagnose erkannt

AUS: Fehlerfreier Betrieb der CPU

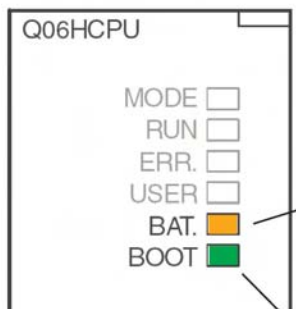
BLINKT: Bei der Selbstdiagnose wurde ein Fehler erkannt, der zum Programmabbruch führt

EIN: Durch die CHK-Anweisung wurde ein Fehler erkannt oder es wurde ein Fehlermerker (F) gesetzt

AUS: Fehlerfreier Betrieb der CPU

BLINKT: Der Latch-Bereich wird gelöscht

● BAT- und BOOT-LED



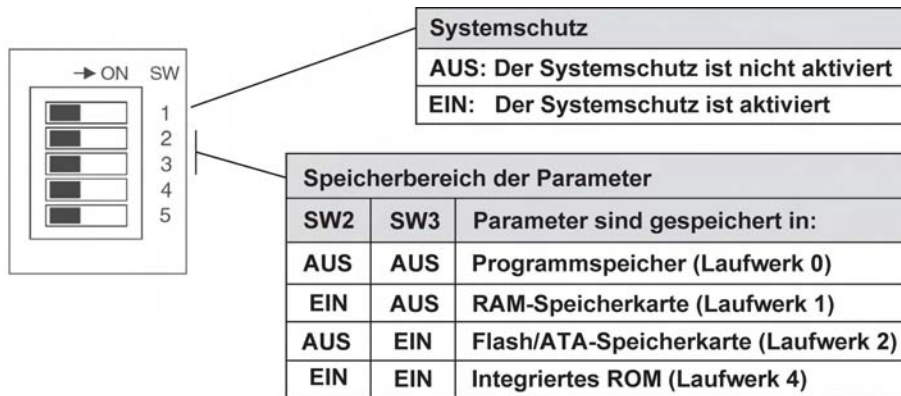
EIN: Die Spannung der Pufferbatterie der CPU oder der Speicherkarte ist zu niedrig

AUS: Die Batteriespannungen sind normal

EIN: Ein Programm wird geladen

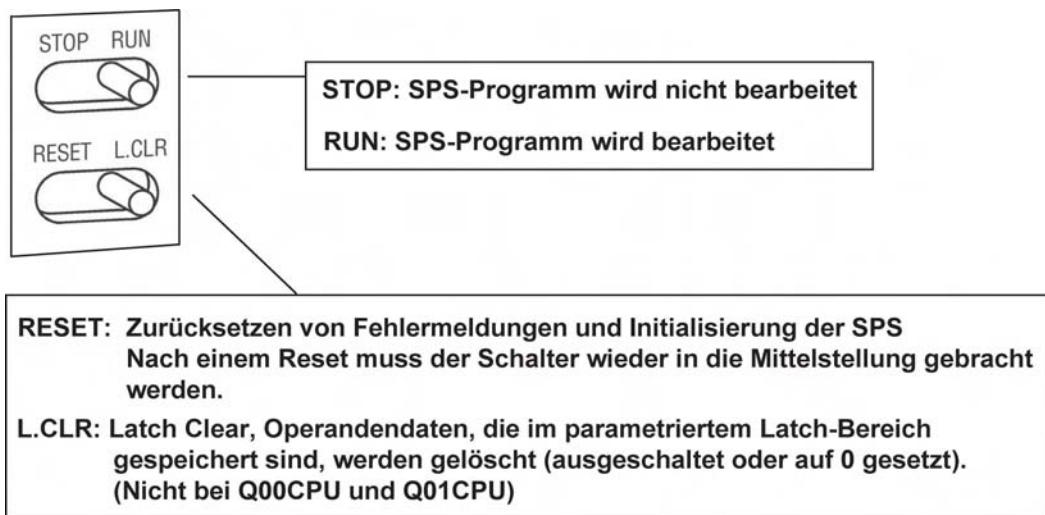
AUS: Es wird kein Boot-Vorgang ausgeführt

Systemschalter

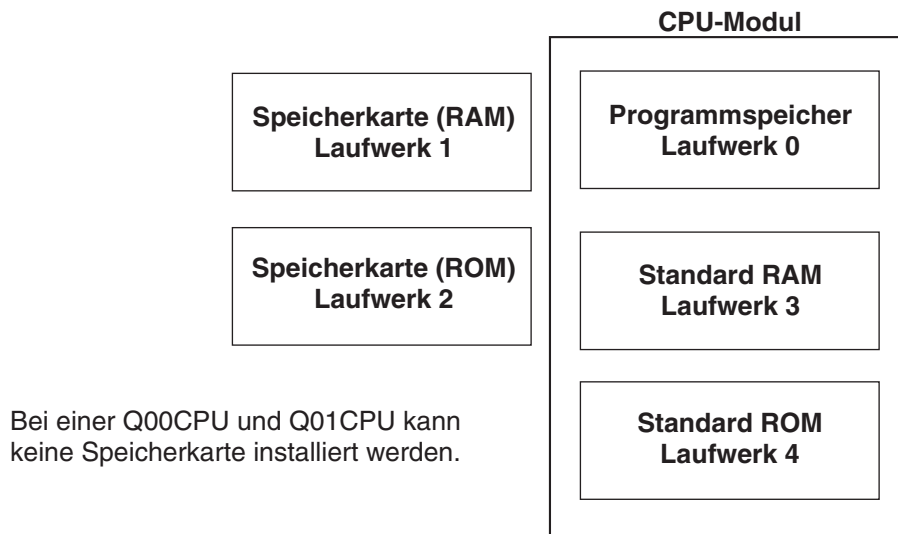


Im integrierten RAM (Laufwerk 3) können keine Parameter gespeichert werden.
 Bei der Auslieferung des CPU-Moduls sind alle Schalter in der Stellung „AUS“.

RUN/STOP-Schalter, RESET/L.CLR-Schalter



Konfiguration des Speichers



Was kann wo gespeichert werden?

Q00CPU und Q01CPU

Daten	Integrierter Speicher		
	Programmspeicher (Laufwerk 0)	RAM (Laufwerk 3)	ROM (Laufwerk 4)
Programm	●	○	●
Parameter	●	○	●
Parameter für Sondermodule	●	○	●
Operandenkommentare	●	○	●
File-Register	○	●	○

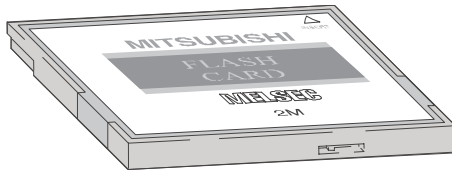
- = Speicherung ist möglich
- = Speicherung ist nicht möglich

Q02CPU, Q02HCPU, Q06HCPU, Q12HCPU und Q25HCPU:

Daten	Integrierter Speicher			Speicherkarten		
	Programmspeicher (Laufwerk 0)	RAM (Laufwerk 3)	ROM (Laufwerk 4)	RAM (Laufwerk 1)	Flash ROM (Laufwerk 2)	ATA ROM (Laufwerk 2)
Programm	●	○	●	●	●	●
Parameter	●	○	●	●	●	●
Parameter für Sondermodule	●	○	●	●	●	●
Operandenkommentare	●	○	●	●	●	●
Initialisierungswerte	●	○	●	●	●	●
File-Register	○	●	○	●	●	○
Lokale Operanden	○	●	○	●	○	○
TRACE-Daten	○	○	○	●	○	○
Fehlerhistorie	○	○	○	●	○	○
Daten, die mit einer FWRITE-Anweisung eingetragen wurden	○	○	○	○	○	●

- = Speicherung ist möglich
- = Speicherung ist nicht möglich

Speicherkarten

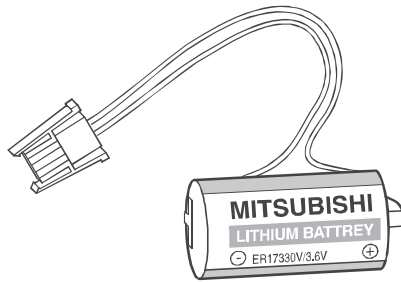


Die gespeicherten Daten können durch einen Schreibschutz gegen unbeabsichtigtes Löschen geschützt werden. In der SRAM-Speicherkarte puffert eine integrierte Batterie die gespeicherten Daten bei einem Spannungsausfall.

Erhältliche Speicherkarten

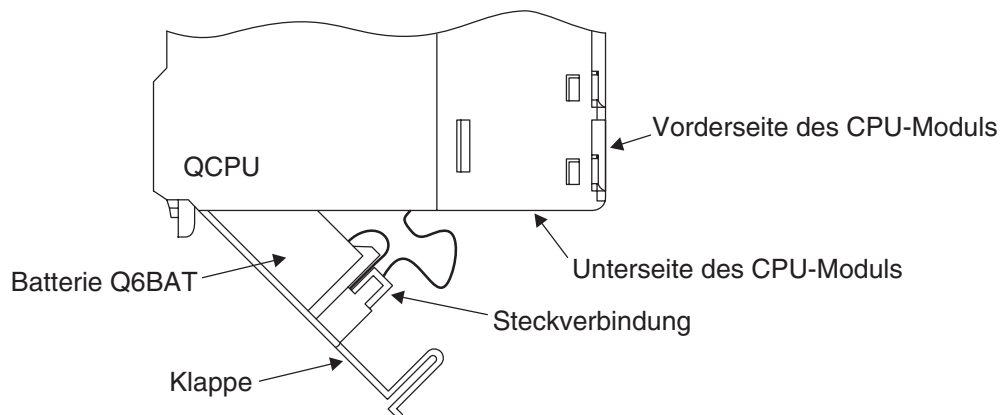
Bezeichnung	Art des Speichers	Speicherkapazität [Byte]	Speicherkapazität [Dateien]	Anzahl der Schreibvorgänge
Q2MEM-1MBS	SRAM	1011 k	256	Keine Beschränkung
Q2MEM-2MBS		2034 k	288	
Q2MEM-2MBF	Flash ROM	2035 k	288	100 000
Q2MEM-4MBF		4079 k		
Q2MEM-8MBA	ATA ROM	7940 k	512	1 000 000
Q2MEM-16MBA		15932 k		
Q2MEM-32MBA		31854 k		

Installation der Pufferbatterie in das CPU-Modul



Die Batterie ist an der Unterseite des CPU-Moduls eingebaut. Bei einem Spannungsausfall kann sie den Programmspeicher, das integrierte RAM und die Uhr der CPU mehrere tausend Stunden (abhängig vom Typ der CPU) puffern.

Bei der Auslieferung eines CPU-Moduls ist die Batterie zwar im CPU-Modul eingebaut, zum Schutz vor Kurzschlüssen und um eine Entladung zu vermeiden, ist die Steckverbindung zwischen Batterie und CPU aber getrennt. Vor der Inbetriebnahme der CPU muss die Batterie angeschlossen werden.



Die Batterie sollte alle 10 Jahre gewechselt werden.

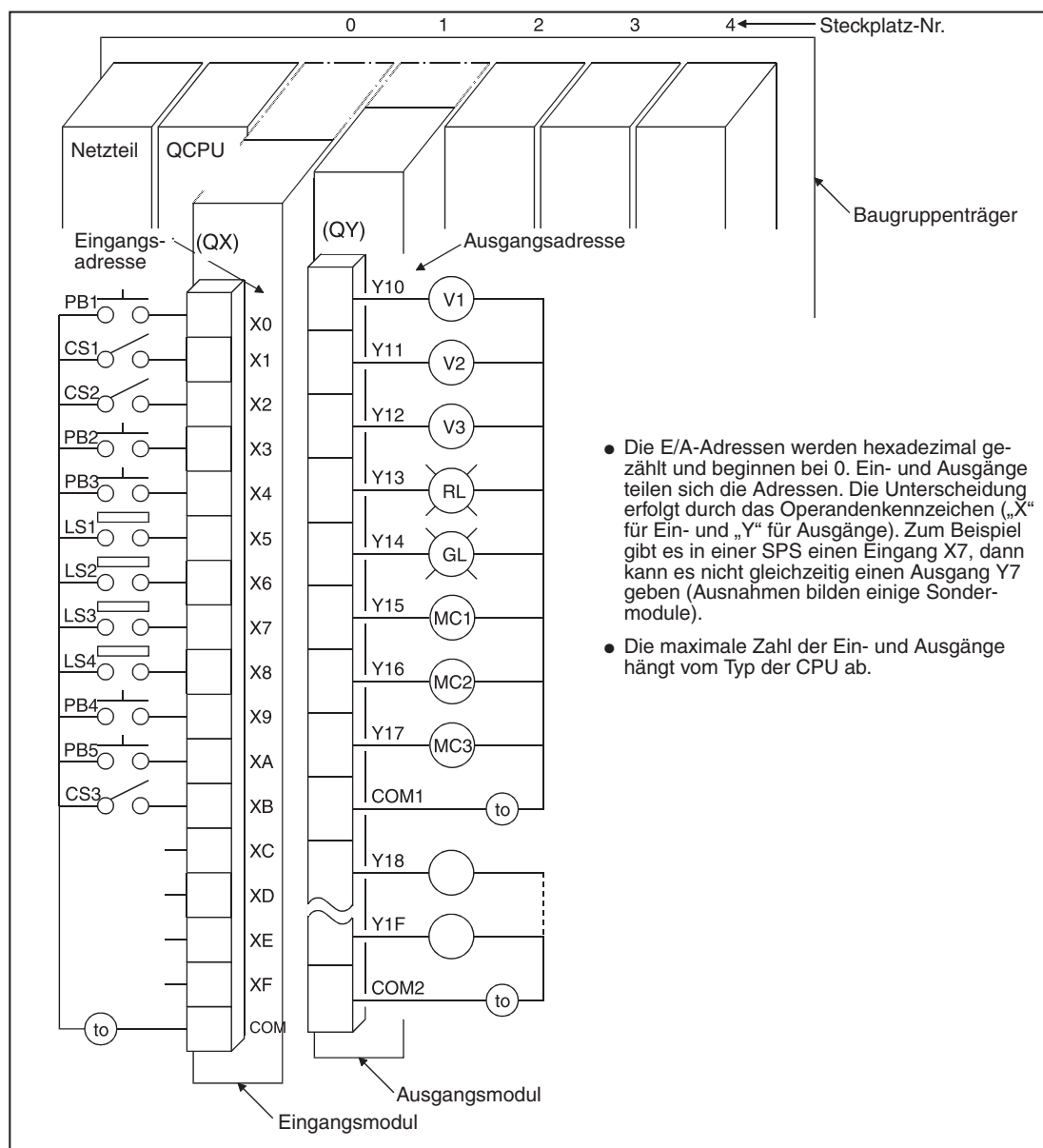
2.7 Anschluss externer Signale

2.7.1 Verdrahtung von Ein- und Ausgängen

Die Signale, die externe Geräte an die Eingänge der SPS liefern, werden für die Programmierung in Eingangsadressen umgewandelt. Die Adresse eines SPS-Eingangs wird dadurch bestimmt, auf welchem Steckplatz des Baugruppenträgers das Eingangsmodul installiert ist (siehe Abschnitt 2.3.3) und an welchem Eingang eines Moduls ein Signalsangeschlossen ist.

Die Adressen der durch das Programm gesteuerten Ausgänge werden ebenfalls durch den Steckplatz und dem Anschluss am Modul bestimmt. Um ein externes Gerät zu schalten, muss dessen Anschluss mit dem entsprechenden SPS-Ausgang verbunden werden.

Die Ein- und Ausgänge werden hexadezimal (0, 1, 2 ...9, A, B, C, D, E, F) adressiert. Dadurch ergeben sich Gruppen zu 16 Ein- oder Ausgängen.



2.8 Digitale Ein- und Ausgangsmodule

Ein- und Ausgangsmodule verbinden die CPU einer SPS mit dem zu steuernden Prozess. Während digitale Eingangsmodule die Signale externer Geräte in eine EIN/AUS-Information für die CPU umwandeln, können externe Schaltglieder durch digitale Ausgangsmodule ein- oder ausgeschaltet werden.

Eingangssignale können von einer Vielzahl von Sensoren oder Geräten stammen:

- Drucktaster
- Drehschalter mit mehreren Stellungen
- Schlüsselschalter
- Endschalter
- Niveauschalter
- Sensoren zur Durchflussüberwachung
- Lichtschranken oder Lichttaster
- Näherungsschalter (Induktiv oder kapazitiv), Näherungsschalter sind in der Regel mit einem Transistorausgangs ausgestattet, der entweder plus- oder minusschaltend ausgelegt ist.

Mit **Ausgangssignalen** werden zum Beispiel angesteuert:

- Schütze
- Meldeleuchten
- Magnetventile
- Eingänge externer Geräte wie z. B. Frequenzumrichter

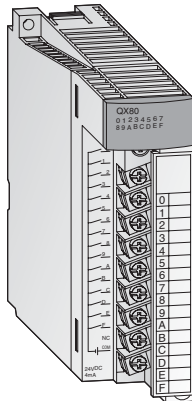
Übersicht der digitalen E/A-Module

Modultyp		Anzahl der Ein- und Ausgänge			
		8	16	32	64
Eingangsmodule	120 V AC	○	●	○	○
	240 V AC	●	○	○	○
	24 V DC	○	●	●	●
	24 V DC (schnelle Eingänge)	●	○	○	○
	5 V DC / 12 V DC	○	●	●	●
Ausgangsmodule	Relais	●	●	○	○
	Relais mit getrennten Kontakten	●	○	○	○
	Triac-Ausgänge	○	●	○	○
	Transistor-Ausgänge (minusschaltend)	●	●	●	●
	Transistor-Ausgänge (plusschaltend)	○	●	●	○
Kombinierte Ein-/Ausgangsmodule		●	○	●	○

- = Es ist ein Modul verfügbar
- = Es steht kein Modul zur Verfügung

2.8.1 Digital-Eingangsmodule

Es stehen Digital-Eingangsmodule für verschiedene Eingangsspannungen zur Verfügung:



Eingangsspannung	Anzahl der Eingänge			
	8	16	32	64
5 – 12 V DC		QX70	QX71	QX72
24 V DC		QX80	QX81	QX82
24 V DC (Interrupt-Modul)		QI60		
100 – 120 V AC		QX10		
100 – 240 V AC	QX28			

Bei den Eingangs-Modulen mit 8 oder 16 Eingängen erfolgt der Anschluss der externen Signale über abnehmbare Klemmleisten mit Schraubklemmen. Module mit 32 oder 64 Eingängen werden über Stecker angeschlossen.

Grundsätzliches über digitale Eingangsmodule

Alle Eingänge sind durch Optokoppler isoliert. Dadurch wird die empfindliche Elektronik der SPS nicht durch elektromagnetische Störungen beeinflusst, die durch externe Geräte verursacht werden.

Ein anderes oft anzutreffendes Problem ist das Prellen der Kontakte von mechanischen Schaltern. Damit sich diese Störungen nicht auf die SPS auswirken, werden die Eingangssignale gefiltert. Ein geänderter Signalzustand wird nur erfasst, wenn er für eine bestimmte Zeit am Eingang anliegt. Kurzzeitige Störsignale werden dadurch von der SPS nicht als Eingangssignale interpretiert.

HINWEIS

A-Serie: Die Filterzeit ist für Standard-Eingangsmodule auf 10 ms voreingestellt.

System Q: Für Standard-Eingangsmodule ist die Filterzeit auf 10 ms voreingestellt. Diese Voreinstellung kann aber für jedes installierte Modul in den Parametern im Bereich von 1 ms bis 70 ms geändert werden. Bitte beachten Sie hierzu die technischen Daten der Module.

Die eingestellte Filterzeit beeinflusst auch die Reaktionszeit der SPS und sollte daher bei der Programmierung berücksichtigt werden. Bei einer kurzen Filterzeit wird zwar die Reaktionszeit der SPS reduziert, gleichzeitig steigt aber die Empfindlichkeit gegenüber externen Störeinflüssen. Die Eingangssignale sollten in diesem Fall über abgeschirmte Leitungen geführt und diese Signalleitungen sollten getrennt von Leitungen verlegt werden, die potentielle Störquellen darstellen. Falls sehr kurze Reaktionszeiten gefordert sind, sollten spezielle Module wie das Interrupt-Modul QI60 eingesetzt werden.

Damit die SPS einen eingeschalteten Eingang erkennt, muss ein Mindeststrom in diesen Eingang (oder aus dem Eingang heraus) fließen. Dieser Strom hängt vom Typ des Eingangsmoduls ab und beträgt in den meisten Fällen 3 mA. Wird – auch bei vermeintlich eingeschaltetem Eingang – dieser Strom nicht erreicht, bleibt der Eingang für die CPU weiterhin ausgeschaltet. Der Eingangsstrom wird durch den Innenwiderstand des Eingangsmoduls begrenzt. Fließt durch eine zu hohe Eingangsspannung ein zu großer Eingangsstrom, wird das Eingangsmodul beschädigt. Eingangsströme von bis zu 7 mA sind zulässig.

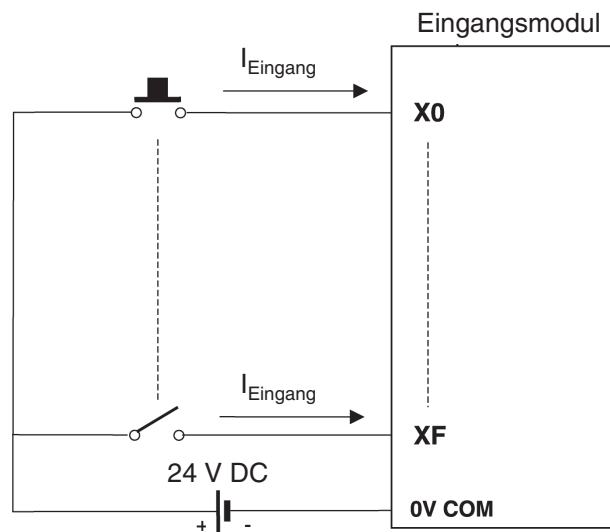
Die SPS-CPU erfasst den Zustand der Eingänge am Anfang der zyklischen Programmbearbeitung und speichert sie. Im Programm werden nur die gespeicherten Zustände verarbeitet. Erst vor der erneuten Bearbeitung des Programms werden die Eingangszustände wieder aktualisiert.

Plus- und minusschaltende Eingänge

Im MELSEC System Q stehen Gleichspannungs-Eingangsmodule für plus- oder minusschaltende Sensoren zur Verfügung. An einigen Modulen wie z. B. dem QX71 können aber auch wahlweise plus- oder minusschaltende Sensoren angeschlossen werden. Im angelsächsischen Sprachraum wird bei plus- und minusschaltenden Gebern von „Source“ (Stromquelle) bzw. „Sink“ (Stromsenke) gesprochen. Diese Bezeichnungen beziehen sich auf die Richtung, in die der Strom bei eingeschaltetem Eingang fließt.

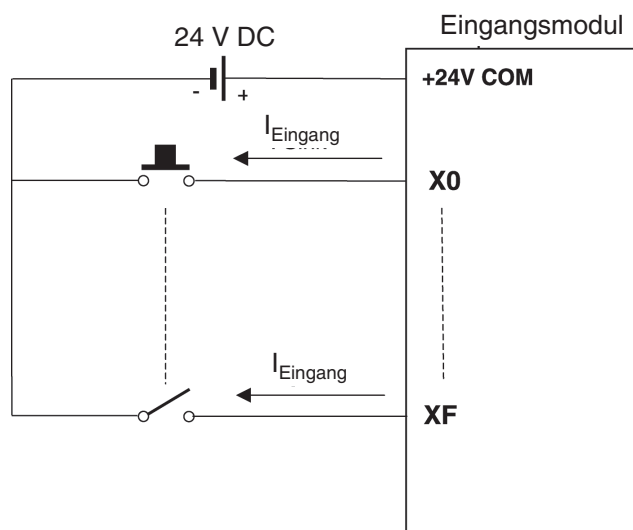
Anschluss plusschaltender Geber („Source“)

Ein plusschaltender Geber verbindet den Pluspol einer Spannungsquelle mit einem SPS-Eingang. Der Minuspol der Spannungsquelle bildet das gemeinsame Bezugspotential aller Eingänge einer Gruppe. Bei eingeschaltetem Geber fließt ein Strom in das Eingangsmodul, daher die englische Bezeichnung „Source“ – der Geber arbeitet als Stromquelle.



Anschluss minusschaltender Geber („Sink“)

Ein minusschaltender Geber verbindet den Minuspol einer Spannungsquelle mit einem SPS-Eingang. Das gemeinsame Bezugspotential aller Eingänge einer Gruppe ist der Pluspol der Spannungsquelle. Bei eingeschaltetem Geber fließt ein Strom aus dem Eingangsmodul, der Geber wirkt als Stromsenke, daher die englische Bezeichnung „Sink“.



Näherungsschalter und optische Sensoren

Näherungsschalter sind berührungslose Schalter. Sie geben ein Signal an die SPS, wenn sich ein Objekt dem Schalter mit einem geringen Abstand nähert. Der Schalter muss vom zu erfassenden Objekt nicht berührt werden. Dadurch ergeben sich viele Einsatzmöglichkeiten in der Anlagenautomatisierung. Näherungsschalter können induktiv oder kapazitiv arbeiten.

Weit verbreitet in industriellen Steuerungen sind auch **optische Sensoren** in Form von Lichtschranken oder Lichttaster. (Lichtschranken benötigen einen Spiegel, der den Lichtstrahl reflektiert. Bei Lichttastern wird das ausgesendete Licht vom Objekt zurückgestrahlt.)

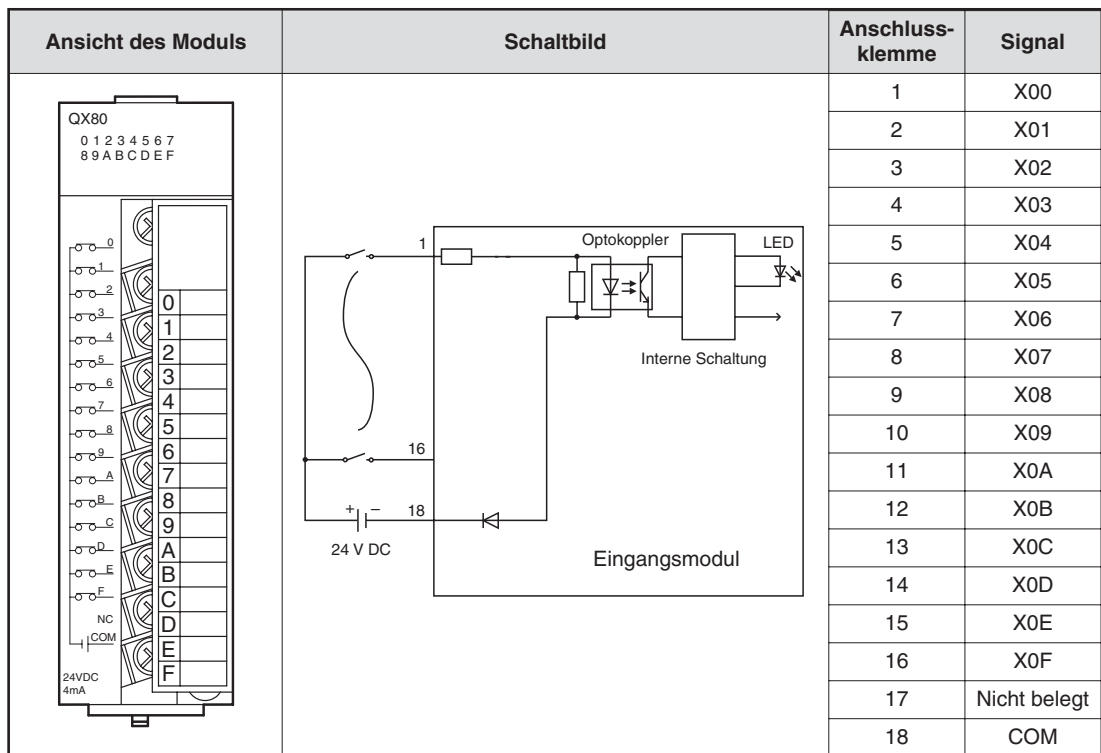
Näherungsschalter und Lichtschranken oder -taster sind mit einer internen Elektronik ausgestattet, die in den meisten Fällen eine Versorgungsspannung von 24 V DC benötigt. Die Ausgänge dieser elektronischen Schalter sind in der Regel als Transistorausgänge ausgelegt und schalten entweder plus oder minus:

- PNP-Transistorausgang: plusschaltend (source)
- NPN-Transistorausgang: minusschaltend (sink)

Beispiel für ein Eingangsmodul für plusschaltende Geber

Merkmal		Technische Daten
Bezeichnung des Moduls		QX80
Eingänge		16
Isolation		durch Optokoppler
Eingangsnennspannung		24 V DC (+20/-15%, Welligkeit bis 5%)
Eingangsstrom		ca. 4 mA
Gleichzeitig schaltbare Eingänge		100 % (Alle Eingänge können gleichzeitig eingeschaltet sein.)
Einschaltstromspitze		Max. 200 mA für 1 ms (bei 132 V AC)
Spannung und -strom für EIN		≥ 19 V DC / ≥ 3 mA
Spannung und -strom für AUS		≤ 11 V DC / ≤ 1,7 mA
Eingangswiderstand		ca. 5,6 kΩ
Ansprechzeit	AUS → EIN	1, 5, 10, 20, 70 ms (parametrierbar, Voreinstellung: 10 ms)*
	EIN → AUS	1, 5, 10, 20, 70 ms (parametrierbar, Voreinstellung: 10 ms)*
Durchschlagfestigkeit		560 V AC Effektivwert für 3 Zyklen (Einsatzhöhe 2000 m)
Isolationswiderstand		≥ 10 MΩ (Messung mit Isolationsprüfgerät)
Störfestigkeit		Geprüft mit Störsimulator (Spitzenwert der Störspannung: 500 V, Einschalt-dauer der Störspannung: 1 μs, Frequenz der Störspannung: 25 bis 60 Hz)
		Hochfrequente, nicht periodische Störspannung (IEC61000-4-4): 1kV
Eingangsgruppen		1 Gruppe mit 16 Eingängen, Bezugspotential: Anschlussklemme 18
Statusanzeige der Eingänge		Eine LED pro Eingang
Anschluss der Verdrahtung		Klemmblock mit 18 Schraubklemmen (M3 x 6)
Empfohlener Leitungsquerschnitt		0,3 bis 0,75 mm ² , Drahtdurchmesser: 2,8 mm
Interne Stromaufnahme (5 V DC)		50 mA (wenn alle Eingänge eingeschaltet sind)
Gewicht		0,16 kg

* Die Ansprechzeiten von AUS nach EIN und von EIN nach AUS können nicht separat eingestellt werden.



Funktion eines Eingangsmoduls mit plusschaltenden Gebern

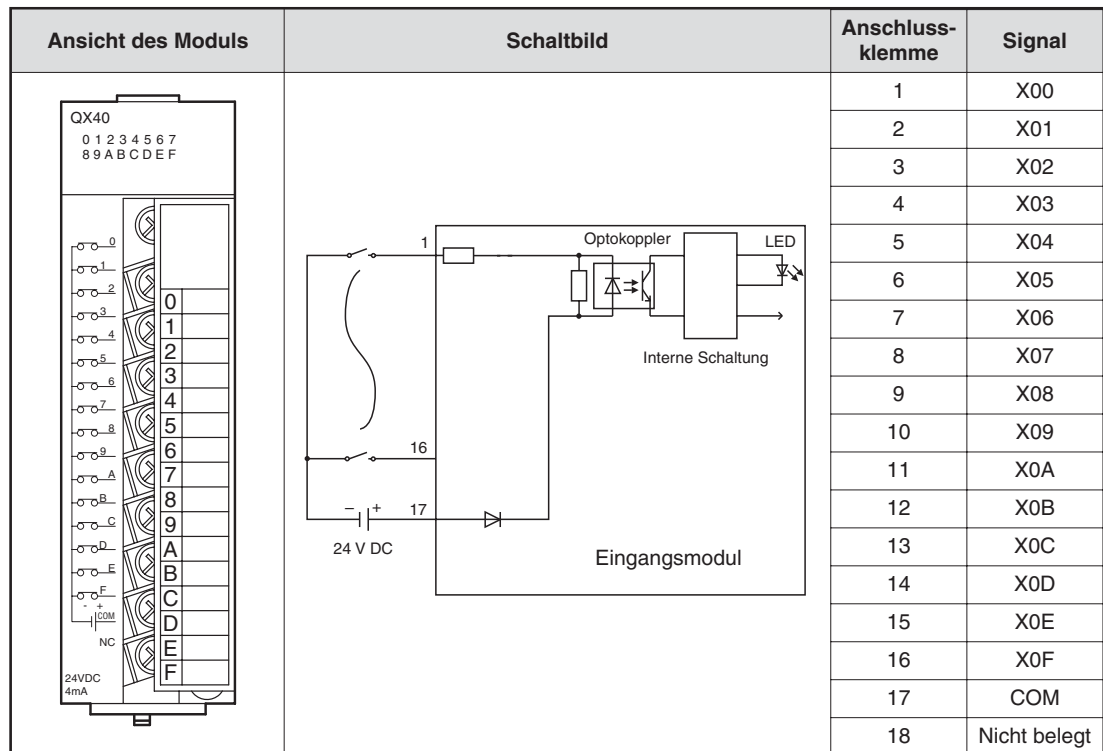
Wird ein an einem Eingangsmodul angeschlossener Geber, wie z. B. ein Drucktaster mit Schließerfunktion, betätigt, wird der SPS-Eingang eingeschaltet. Dabei laufen die folgenden Vorgänge ab, die sich auf das Schaltbild auf der vorherigen Seite beziehen:

- Bei betätigtem Taster wird der Pluspol der externen 24-Volt-Spannungsquelle mit Anschluss 1 des Eingangsmoduls verbunden.
- Anschluss 1 ist über einen Widerstand und die Leuchtdiode des Optokopplers mit dem Minuspol der externen Spannungsquelle (Anschluss 18) verbunden. Dadurch fließt ein Strom durch die LED des Optokopplers.
- Der fließende Strom bringt die LED zum Leuchten. Dadurch wird der Fototransistor des Optokopplers durchgesteuert.
- Durch den Optokoppler wird die externe Eingangsspannung von der Versorgungsspannung der SPS getrennt. Dadurch werden Störungen, die in industriellen Umgebungen oft diese externen Gleichspannungen überlagern, nicht auf die Versorgungsspannung der SPS übertragen. Durch den Optokoppler wird der Eingang zudem unempfindlicher gegen Störungen.
- Wenn der Fototransistor des Optokopplers durchsteuert, wird an die Eingangslogik des Moduls ein Signal übermittelt. In diesem Beispiel registriert die Elektronik, dass der Eingang X0 eingeschaltet ist. Die Leuchtdiode an der Vorderseite des Eingangsmoduls leuchtet in diesem Fall auf und signalisiert diesen Signalzustand.

Beispiel für ein Eingangsmodul für minusschaltende Geber

Merkmal		Technische Daten
Bezeichnung des Moduls		QX40
Eingänge		16
Isolation		durch Optokoppler
Eingangsnennspannung		24 V DC (+20/-15%, Welligkeit bis 5%)
Eingangsstrom		ca. 4 mA
Gleichzeitig schaltbare Eingänge		100 % (Alle Eingänge können gleichzeitig eingeschaltet sein.)
Einschaltstromspitze		Max. 200 mA für 1 ms (bei 132 V AC)
Spannung und -strom für EIN		≥ 19 V DC / ≥ 3 mA
Spannung und -strom für AUS		≤ 11 V DC / ≤ 1,7 mA
Eingangswiderstand		ca. 5,6 kΩ
Ansprechzeit	AUS → EIN	1, 5, 10, 20, 70 ms (parametrierbar, Voreinstellung: 10 ms)*
	EIN → AUS	1, 5, 10, 20, 70 ms (parametrierbar, Voreinstellung: 10 ms)*
Durchschlagfestigkeit		560 V AC Effektivwert für 3 Zyklen (Einsatzhöhe 2000 m)
Isolationswiderstand		≥ 10 MΩ (Messung mit Isolationsprüfgerät)
Störfestigkeit		Geprüft mit Störsimulator (Spitzenwert der Störspannung: 500 V, Einschalt-dauer der Störspannung: 1 μs, Frequenz der Störspannung: 25 bis 60 Hz) Hochfrequente, nicht periodische Störspannung (IEC61000-4-4): 1kV
Eingangsgruppen		1 Gruppe mit 16 Eingängen, Bezugspotential: Anschlussklemme 17
Statusanzeige der Eingänge		Eine LED pro Eingang
Anschluss der Verdrahtung		Klemmblock mit 18 Schraubklemmen (M3 x 6)
Empfohlener Leitungsquerschnitt		0,3 bis 0,75 mm ² , Drahtdurchmesser: 2,8 mm
Interne Stromaufnahme (5 V DC)		50 mA (wenn alle Eingänge eingeschaltet sind)
Gewicht		0,16 kg

* Die Ansprechzeiten von AUS nach EIN und von EIN nach AUS können nicht separat eingestellt werden.



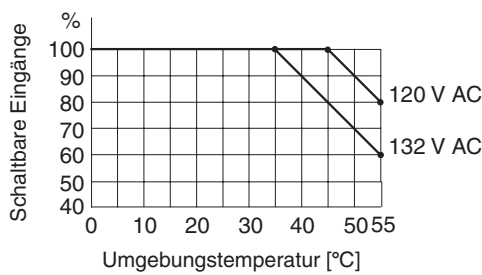
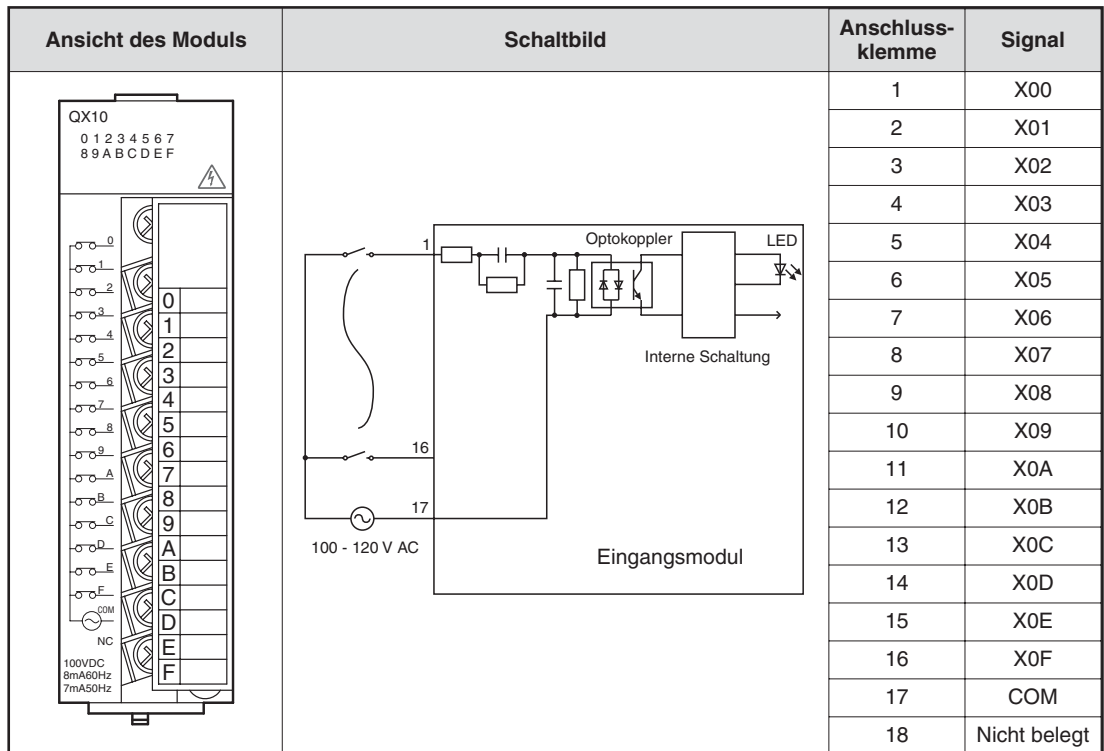
Funktion eines Eingangsmoduls mit minusschaltenden Gebern

Wenn der im Schaltbild auf der vorherigen Seite an der Klemme 1 angeschlossene Schalter betätigt wird, fließt der Strom wie folgt:

- Aus dem Pluspol der externen 24-Volt-Spannungsquelle in den Anschluss für das Bezugspotential (Klemme 17).
- Durch die Leuchtdiode des Optokopplers und den Vorwiderstand zur Klemme 1 (Anschluss für Eingangs X0) des Eingangsmoduls.
- Der Strom durch die LED des Optokopplers bringt diese zum Leuchten. Dadurch wird der Fototransistor des Optokopplers eingeschaltet.
- Wenn der Fototransistor des Optokopplers durchsteuert, wird an die Eingangslogik des Moduls ein Signal übermittelt. In diesem Beispiel registriert die Elektronik, dass der Eingang X0 eingeschaltet ist. Die Leuchtdiode an der Vorderseite des Eingangsmoduls leuchtet in diesem Fall auf und signalisiert diesen Signalzustand.
- Aus dem Anschluss für X0 fließt der Strom über den betätigten Schalter zum Minuspol der externen Spannungsquelle.

Beispiel für ein Wechselspannungs-Eingangsmodul

Merkmal		Technische Daten
Bezeichnung des Moduls		QX10
Eingänge		16
Isolation		Durch Optokoppler
Eingangsnennspannung		100 — 120 V AC (+10/-15 %) 50/60 Hz (± 3 Hz) (Verzerrungen bis 5 %)
Eingangsstrom		ca. 8 mA bei 100 V AC, 60 Hz; ca. 7 mA bei 100 V AC, 50 Hz
Gleichzeitig schaltbare Eingänge		siehe Diagramm
Einschaltstromspitze		Max. 200 mA für 1 ms (bei 132 V AC)
Spannung und -strom für EIN		≥ 80 V AC / ≥ 5 mA (50 Hz, 60 Hz)
Spannung und -strom für AUS		≤ 30 V AC / ≤ 1 mA (50 Hz, 60 Hz)
Eingangswiderstand		ca. 15 k Ω bei 60 Hz, ca. 18 k Ω bei 50 Hz
Ansprechzeit	AUS \rightarrow EIN	≤ 15 ms (100 V AC, 50 Hz, 60 Hz)
	EIN \rightarrow AUS	≤ 20 ms (100 V AC, 50 Hz, 60 Hz)
Durchschlagfestigkeit		1780 V AC Effektivwert für 3 Zyklen (Einsatzhöhe 2000 m)
Isolationswiderstand		≥ 10 M Ω (Messung mit Isolationsprüfgerät)
Störfestigkeit		Geprüft mit Störsimulator (Spitzenwert der Störspannung: 1500 V, Einschalt-dauer der Störspannung: 1 μ s, Frequenz der Störspannung: 25 bis 60 Hz) Hochfrequente, nicht periodische Störspannung (IEC61000-4-4): 1kV
Eingangsgruppen		1 Gruppe mit 16 Eingängen, Bezugspotential: Anschlussklemme 17
Statusanzeige der Eingänge		Eine LED pro Eingang
Anschluss der Verdrahtung		Klemmblock mit 18 Schraubklemmen (M3 x 6)
Empfohlener Leitungsquerschnitt		0,3 bis 0,75 mm ² , Drahtdurchmesser: 2,8 mm
Interne Stromaufnahme (5 V DC)		50 mA
Gewicht		0,17 kg

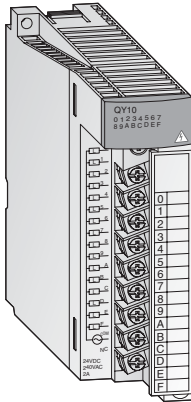


Die Anzahl der gleichzeitig schaltbare Eingänge hängt beim Modul QX10 von der Umgebungstemperatur ab.

Bei Eingangsmodulen für Wechselspannungen sollte dieselbe Spannung (100 - 120V AC), die auch die SPS versorgt, zum Schalten der Eingänge verwendet werden. Dadurch wird verhindert, dass an den Eingängen eine falsche Spannung angeschlossen wird.

2.8.2 Digitale Ausgangsmodule

Durch unterschiedliche Schaltelemente bieten die Ausgangsmodule eine Lösung für jede Steuerungsaufgabe:



Ausgangstyp	Nennspannung	Anzahl der Ausgänge		
		8	16	32
Relais	24 V DC / 240 V AC	QY18A	QY10	
Triac	100 – 240 V AC		QY22	
Transistor	5 / 12 V DC		QY70	QY71
	12 / 24 V DC		QY80	QY81P
	5 – 24 V DC	QY68A		

Die Module mit 8 oder 16 Ausgängen besitzen abnehmbare Klemmleisten mit Schraubklemmen zum Anschluss der Ausgangssignale. Module mit 32 Ausgängen werden über Stecker angeschlossen.

Ausgangstypen

Die digitalen Ausgangsmodule des MELSEC System Q sind in vier verschiedenen Ausgangstypen erhältlich.

- Relais
- Triac
- Transistor (plusschaltend)
- Transistor (minusschaltend)

Typ	Vorteile	Nachteile
Relais	<ul style="list-style-type: none"> ● Ein Modul kann verschiedenen Spannungen schalten. ● Potentialfreie Kontakte ● Schalten hoher Ströme ist möglich 	<ul style="list-style-type: none"> ● Langsam (max. 1 Hz) ● Begrenzte Lebensdauer (elektromechanisch) ● Gefahr von verbrannten Schaltkontakten ● Laut (das Schalten ist hörbar)
Triac	<ul style="list-style-type: none"> ● Zuverlässig ● Hohe Schaltgeschwindigkeit ● Geeignet für hohe Anforderungen 	<ul style="list-style-type: none"> ● Schaltet nur Wechselspannung ● max. Schaltstrom 0,6 A pro Ausgang ● Benötigt 10 ms Schaltzeit bei 50 Hz AC
Transistor	<ul style="list-style-type: none"> ● Sehr zuverlässig ● Sehr hohe Schaltgeschwindigkeit ● Besonders geeignet für hohe Anforderungen 	<ul style="list-style-type: none"> ● Schaltet nur niedrige Gleichspannungen ● max. Schaltstrom 0,1 A pro Ausgang

Relais-Ausgangsmodule

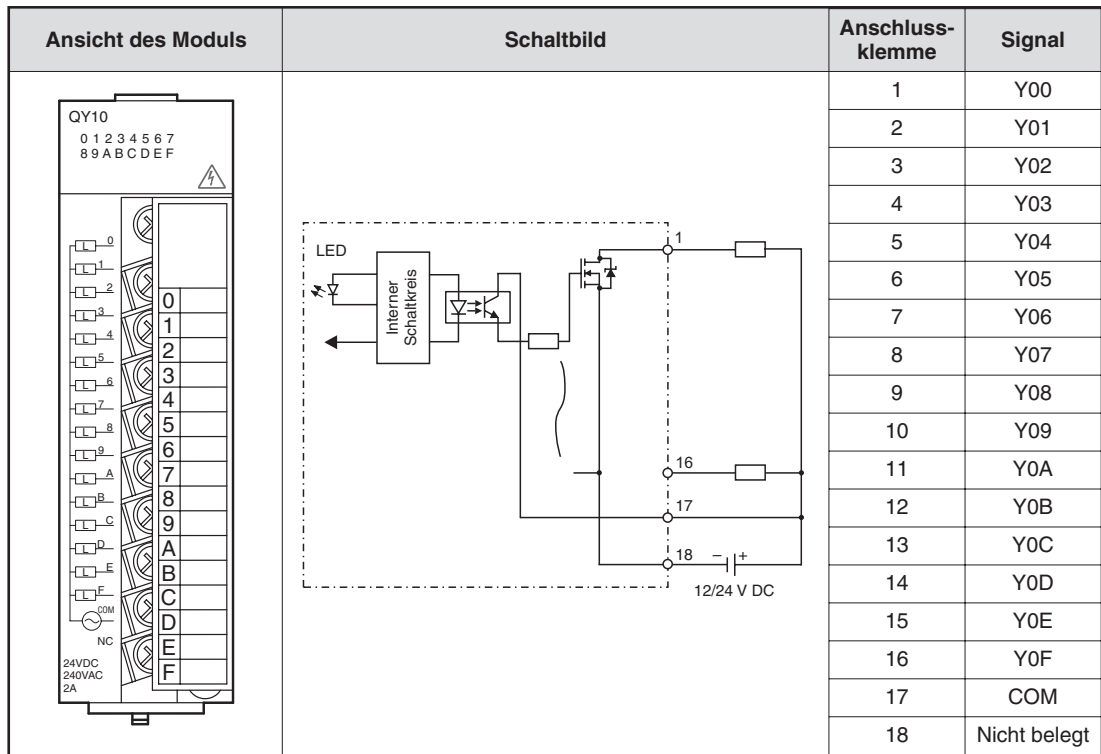
Relais-Ausgangsmodule enthalten pro Ausgang ein Relais, dessen Schaltkontakt wiederum die angeschlossene Lastspannung schaltet. Dadurch wird Trennung zwischen interner Spannung der SPS und externen Lasten erreicht.

Es stehen Relais-Ausgangsmodule mit einem gemeinsamen Bezugspotential und Module mit unabhängigen, potentialfreien Relaiskontakten zur Verfügung.

Wie auch bei den anderen Ausgangsmodulen wird der Ausgang durch das SPS-Programm gesteuert. Am Ende des Programms werden die SPS-Ausgänge aktualisiert. Das heißt, dass zu diesem Zeitpunkt alle logischen Ausgangszustände, die sich durch das Programm ergeben haben, an die physischen Ausgänge übertragen werden. Ein eingeschalteter Ausgang wird durch eine leuchtende LED angezeigt. Dadurch ist auch eine Kontrolle direkt an der SPS möglich. Ein Relais-Ausgangsmodul hat eine Reaktionszeit von ca. 10 ms.

Beispiel für ein Relais-Ausgangsmodul

Merkmal		Technische Daten
Bezeichnung des Moduls		QY10
Ausgänge		16
Isolation		Durch Relais
Ausgangsnennspannung/-strom		24 V DC 2 A (Ohmsche Last) pro Ausgang 240 V AC 2 A ($\cos\varphi = 1$) pro Ausgang; Max. 8 A je Gruppe
Minimale Schaltlast		5 V DC, 1 mA
Max. Schaltspannung		125 V DC / 264 V AC
Ansprechzeit	AUS → EIN	≤10 ms
	EIN → AUS	≤12 ms
Lebensdauer der Kontakte	Mechanisch	≥20 Mio. Schaltungen
	Elektrisch	≥100.000 Schaltungen bei Ausgangsnennspannung/-strom
		≥100.000 Schaltungen bei 200 V AC, 1,5 A; 240 V AC 1 A ($\cos\varphi = 0,7$)
		≥300.000 Schaltungen bei 200 V AC, 0,4 A; 240 V AC 0,3 A ($\cos\varphi = 0,7$)
		≥100.000 Schaltungen bei 200 V AC, 1 A; 240 V AC 0,5 A ($\cos\varphi = 0,35$) ≥300.000 Schaltungen bei 200 V AC, 0,3 A; 240 V AC 0,15 A ($\cos\varphi = 0,35$)
≥100.000 Schaltungen bei 24 V DC 1 A; 100 V DC 0,1 A (L/R = 0,7 ms) ≥300.000 Schaltungen bei 24 V DC 0,3 A; 100 V DC 0,03 A (L/R = 0,7ms)		
Max. Schaltfrequenz		3600 Schaltungen/Stunde
Netzfilter		—
Sicherung		—
Durchschlagfestigkeit		2830 V AC Effektivwert für 3 Zyklen (Einsatzhöhe 2000 m)
Isolationswiderstand		≥10 MΩ (Messung mit Isolationsprüfgerät)
Störfestigkeit		Geprüft mit Störsimulator (Spitzenwert der Störspannung: 1500 V, Einschalt-dauer der Störspannung: 1 μs, Frequenz der Störspannung: 25 bis 60 Hz)
		Hochfrequente, nicht periodische Störspannung (IEC61000-4-4): 1kV
Ausgangsgruppen		1 Gruppe mit 16 Ausgängen, Bezugspotential: Anschlussklemme 17
Statusanzeige der Ausgänge		Eine LED pro Ausgang
Anschluss der Verdrahtung		Klemmblock mit 18 Schraubklemmen (M3 x 6)
Empfohlener Leitungsquerschnitt		0,3 bis 0,75 mm ² , Max. Durchmesser der Drähte: 2,8 mm
Interne Stromaufnahme (5 V DC)		430 mA
Gewicht		0,22 kg



Triac-Ausgangsmodule

Digitale Triac-Ausgangsmodule schalten Wechselspannungen von 100 bis 240 V. Die Schaltspannung ist durch Optokoppler von der SPS-Versorgungsspannung getrennt. Die Reaktionszeit von Triac-Ausgangsmodulen ist kürzer als die von Relais-Ausgangsmodulen. Zum Einschalten wird 1 ms und zum Ausschalten werden 10 ms benötigt.

Ein Triac kann einen maximalen Strom von 0,6 A schalten. Eine Anlage mit Triac-Ausgangsmodulen muss so ausgelegt werden, dass dieser maximaler Schaltstrom nicht überschritten wird.

Auch bei ausgeschaltetem Ausgang fließt durch den Triac ein Leckstrom von max. 3 mA. Durch diesen geringen Strom können Meldeleuchten auch bei ausgeschaltetem Ausgang weiter leuchten oder kleine Relais angezogen bleiben.



GEFAHR:

Durch den Leckstrom besteht die Gefahr von elektrischen Schlägen auch bei ausgeschaltetem Triac-Ausgang.

Schalten Sie vor Arbeiten an einer elektrischen Anlage immer die Spannung komplett aus.

Beispiel für ein Triac-Ausgangsmodul

Merkmal		Technische Daten
Bezeichnung des Moduls		QY22
Ausgänge		16
Isolation		durch Optokoppler
Ausgangsnennspannung/-strom		100 – 240 V AC (+20/-15 %), 0,6 A pro Ausgang, 4,8 A pro Modul
Minimale Schaltlast		24 V AC, 100 mA; 100 V AC, 25 mA, 240 V AC, 25 mA
Max. Einschaltstromspitze		20 A
Leckstrom bei ausgeschaltetem Ausgang		≤ 3 mA bei 120 V AC, 60 Hz ≤ 1,5 mA bei 240 V AC, 60 Hz
Max. Spannungsabfall bei eingeschaltetem Ausgang		1,5 V
Ansprechzeit	AUS → EIN	0,5 x Periodendauer + max. 1 ms
	EIN → AUS	0,5 x Periodendauer + max. 1 ms
Netzfilter		RC-Element
Sicherung		—
Durchschlagfestigkeit		2830 V AC Effektivwert für 3 Zyklen (Einsatzhöhe 2000 m)
Isolationswiderstand		≥10 MΩ (Messung mit Isolationsprüfgerät)
Störfestigkeit		Geprüft mit Störsimulator (Spitzenwert der Störspannung: 1500 V, Einschalt-dauer der Störspannung: 1 μs, Frequenz der Störspannung: 25 bis 60 Hz)
		Hochfrequente, nicht periodische Störspannung (IEC61000-4-4): 1kV
Ausgangsgruppen		1 Gruppe mit 16 Ausgängen, Bezugspotential: Anschlussklemme 17
Statusanzeige der Ausgänge		Eine LED pro Ausgang
Anschluss der Verdrahtung		Klemmblock mit 18 Schraubklemmen (M3 x 6)
Empfohlener Leitungsquerschnitt		0,3 bis 0,75 mm ² , Max. Durchmesser der Drähte: 2,8 mm
Interne Stromaufnahme (5 V DC)		250 mA (Alle Ausgänge sind eingeschaltet.)
Gewicht		0,40 kg

Ansicht des Moduls	Schaltbild	Anschluss-klemme	Signal
		1	Y00
		2	Y01
		3	Y02
		4	Y03
		5	Y04
		6	Y05
		7	Y06
		8	Y07
		9	Y08
		10	Y09
		11	Y0A
		12	Y0B
		13	Y0C
		14	Y0D
		15	Y0E
		16	Y0F
		17	COM
		18	Nicht belegt

Transistor-Ausgangsmodule

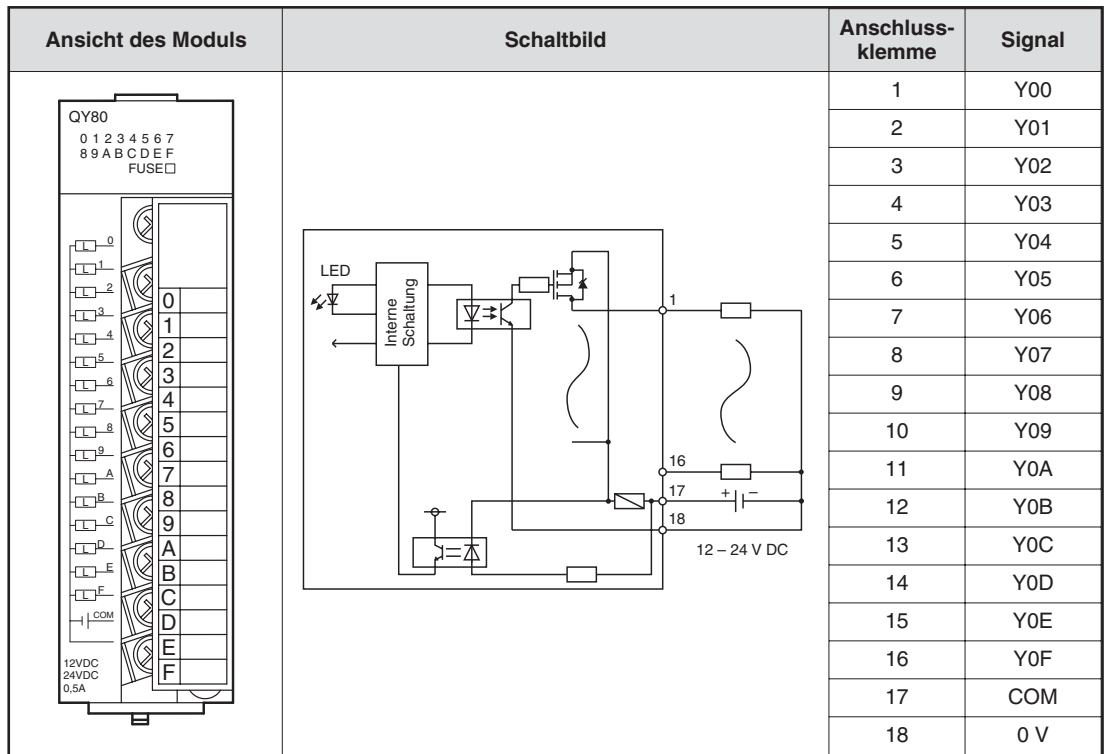
Auch bei Transistor-Ausgangsmodulen sind die Schaltspannung und die Versorgungsspannung der SPS durch Optokoppler isoliert.

Ein Transistor-Ausgangsmodul benötigt nur 1 ms, um einen Ausgang zu schalten. Die technischen Daten wie zum Beispiel die Schaltströme können den Handbüchern der Module oder der Installationsanleitung zu den Ein- und Ausgabemodulen (Art.-Nr. 141758) entnommen werden.

Im MELSEC System Q sind plus- oder minusschaltende Ausgangsmodule erhältlich.

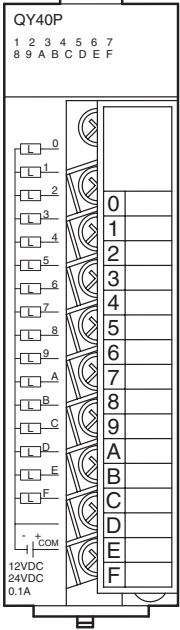
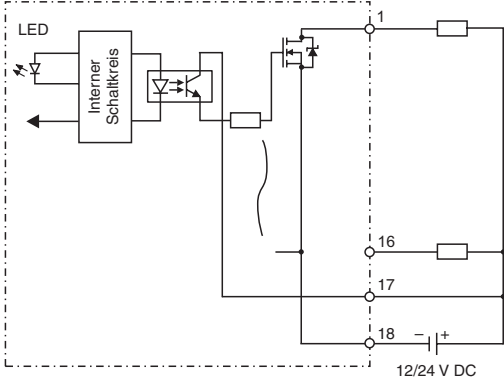
Beispiel für ein plusschaltendes Ausgangsmodul

Merkmal		Technische Daten
Bezeichnung des Moduls		QY80
Ausgänge		16
Isolation		durch Optokoppler
Ausgangsnennspannung		12 bis 24 V DC (+20/-15%)
Ausgangsspannungsbereich		10,2 bis 28,8 V DC
Max. Schaltbedingung		0,5 A pro Ausgang, 4 A pro Gruppe
Max. Einschaltstromspitze		4 A für 10 ms
Leckstrom bei ausgeschaltetem Ausgang		≤0,1 mA
Spannungsabfall bei eingeschaltetem Ausgang		Typisch 0,2 V DC bei 0,5 A, max. 0,3 V bei 0,5 A
Ansprechzeit	AUS → EIN	≤1 ms
	EIN → AUS	≤1 ms (bei Nennschaltbedingungen und ohmscher Belastung)
Netzfilter		Z-Diode
Sicherung		6,7 A; nicht austauschbar
Anzeige einer defekten Sicherung		Durch Einschalten einer LED und Signal an die CPU
Versorgung des Moduls	Spannung	12 bis 24 V DC (+20/-15%, Welligkeit 5%)
	Strom	20 mA (bei 24 V DC und wenn alle Ausgänge geschaltet sind)
Durchschlagfestigkeit		560 V AC Effektivwert für 3 Zyklen (Einsatzhöhe 2000 m)
Isolationswiderstand		≥10 MΩ (Messung mit Isolationsprüfgerät)
Störfestigkeit		Geprüft mit Störsimulator (Spitzenwert der Störspannung: 500 V, Einschalt-dauer der Störspannung: 1 μs, Frequenz der Störspannung: 25 bis 60 Hz)
		Hochfrequente, nicht periodische Störspannung (IEC61000-4-4): 1kV
Ausgangsgruppen		1 Gruppe mit 16 Ausgängen, Bezugspotential: Anschlussklemme 17
Statusanzeige der Ausgänge		Eine LED pro Ausgang
Anschluss der Verdrahtung		Klemmblock mit 18 Schraubklemmen (M3 x 6)
Empfohlener Leitungsquerschnitt		0,3 bis 0,75 mm ² , Max. Durchmesser der Drähte: 2,8 mm
Interne Stromaufnahme (5 V DC)		80 mA
Gewicht		0,17 kg



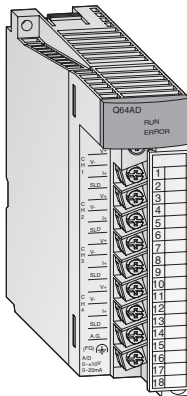
Beispiel für ein minusschaltendes Ausgangsmodul

Merkmal		Technische Daten
Bezeichnung des Moduls		QY40P
Ausgänge		16
Isolation		durch Optokoppler
Ausgangsnennspannung		12 bis 24 V DC (+20/-15%)
Ausgangsspannungsbereich		10,2 bis 28,8 V DC
Max. Schaltbedingung		0,1 A pro Ausgang, 1,6 A pro Gruppe
Max. Einschaltstromspitze		0,7 A für 10 ms
Leckstrom bei ausgeschaltetem Ausgang		≤0,1 mA
Spannungsabfall bei eingeschaltetem Ausgang		Typisch 0,1 V DC bei 0,1 A, max. 0,2 V bei 0,1 A
Ansprechzeit	AUS → EIN	≤1 ms
	EIN → AUS	≤1 ms (bei Nennschaltbedingungen und ohmscher Belastung)
Netzfilter		Z-Diode
Sicherung		—
Anzeige einer defekten Sicherung		Durch Einschalten einer LED und Signal an die CPU
Versorgung des Moduls	Spannung	12 bis 24 V DC (+20/-15%, Welligkeit 5%)
	Strom	10 mA (bei 24 V DC und wenn alle Ausgänge geschaltet sind)
Durchschlagfestigkeit		560 V AC Effektivwert für 3 Zyklen (Einsatzhöhe 2000 m)
Isolationswiderstand		≥10 MΩ (Messung mit Isolationsprüfgerät)
Störfestigkeit		Geprüft mit Störsimulator (Spitzenwert der Störspannung: 500 V, Einschalt-dauer der Störspannung: 1 μs, Frequenz der Störspannung: 25 bis 60 Hz) Hochfrequente, nicht periodische Störspannung (IEC61000-4-4): 1kV
Ausgangsgruppen		1 Gruppe mit 16 Ausgängen, Bezugspotential: Anschlussklemme 18
Statusanzeige der Ausgänge		Eine LED pro Ausgang
Anschluss der Verdrahtung		Klemmblock mit 18 Schraubklemmen (M3 x 6)
Empfohlener Leitungsquerschnitt		0,3 bis 0,75 mm ² , Max. Durchmesser der Drähte: 2,8 mm
Interne Stromaufnahme (5 V DC)		65 mA
Gewicht		0,16 kg

Ansicht des Moduls	Schaltbild	Anschlussklemme	Signal
 <p>QY40P 1 2 3 4 5 6 7 8 9 A B C D E F</p> <p>0 1 2 3 4 5 6 7 8 9 A B C D E F</p> <p>COM 12VDC 24VDC 0.1A</p>	 <p>LED</p> <p>Interner Schaltkreis</p> <p>1</p> <p>16</p> <p>17</p> <p>18</p> <p>12/24 V DC</p>	1	Y00
		2	Y01
		3	Y02
		4	Y03
		5	Y04
		6	Y05
		7	Y06
		8	Y07
		9	Y08
		10	Y09
		11	Y0A
		12	Y0B
		13	Y0C
		14	Y0D
		15	Y0E
		16	Y0F
		17	12/24 V DC
		18	COM

2.9 Sondermodule

2.9.1 Analog-Eingangsmodule



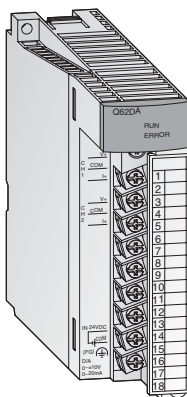
Zur Wandlung analoger Prozesssignale in digitale Werte und damit zur Weiterverarbeitung in der CPU kommen Analog-Eingangsmodule zum Einsatz.

Die Module des System Q vereinen eine hohe Auflösung von bis zu 0,333 mV bzw. 1,33 mA mit einer extrem kurzen Wandlungszeit von nur 80 µs pro Eingang.

Der Anschluss der Eingangssignale erfolgt bei allen Modulen über eine abnehmbare Klemmleiste mit Schraubklemmen.

Eingangsart	Nenneingangsbereich	Einstellbarer Eingangsbereich	Anzahl der Eingänge	
			4	8
Spannung	-10 bis +10 V	1 bis 5 V 0 bis 5 V 0 bis 10 V -10 bis +10 V		Q68ADV
Strom	0 bis 20 mA	0 bis 20 mA 4 bis 20 mA		Q68ADI
Spannung oder Strom (für jeden Eingang separat wählbar)	-10 bis +10 V 0 bis 20 mA	Wie bei Q68ADV und Q68ADI	Q64AD	

2.9.2 Analog-Ausgangsmodule



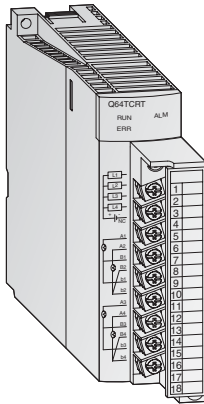
Die Analog-Ausgangsmodule wandeln digitale Werte in ein analoges Strom- oder Spannungssignal. Bei einer extrem kurzen Wandlungszeit von nur 80 µs pro Ausgang wird eine Auflösung von bis zu 0,333 mV bzw. 0,83 µA erreicht. Die kurzschlussfesten Ausgänge sind durch Optokoppler von der Steuerung isoliert.

Bei allen Modulen erfolgt der Anschluss über eine abnehmbare Klemmleiste mit Schraubklemmen.

Ausgangsart	Nennausgangsbereich	Einstellbarer Ausgangsbereich	Anzahl der Ausgänge		
			2	4	8
Spannung oder Strom (für jeden Ausgang separat wählbar)	-10 bis +10 V 0 bis 20 mA	1 bis 5 V -10 bis +10 V 0 bis 20 mA 4 bis 20 mA	Q62DA	Q64DA	
Spannung	-10 bis +10 V	-10 bis +10 V			Q68DAV
Strom	0 bis 20 mA	0 bis 20 mA 4 bis 20 mA			Q68DAI

2.9.3 Temperaturregelmodule mit PID-Algorithmus

Die Temperaturregelmodule ermöglichen die Temperaturregelung, ohne dass für die Regelungsaufgaben die CPU der SPS belastet wird.

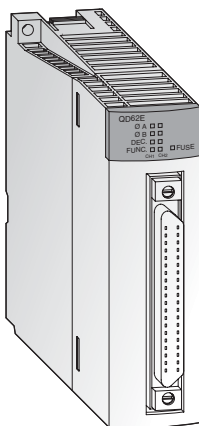


Besondere Merkmale:

- 4 Kanäle zur Temperaturerfassung und 4 PID-Regelkreise pro Modul
- Temperaturmessung entweder mit Pt100-Widerstandsthermometern (Q64TCRT and Q64TCRTBW) oder mit Thermoelementen (Q64TCTT und Q64TCTTBW)
- Integrierte Drahtbruchererkennung für die Heizung bei den Modulen Q64TCRTBW und Q64TCTTBW
- Optimierung der Regelung durch Autotuning
- Transistorausgang für die Ansteuerung des Stellgliedes

2.9.4 High-Speed-Zählermodule

Die Zählermodule QD62E, QD62 und QD62D erfassen Impulse, deren Frequenz für normale Eingangsmodule zu hoch ist.



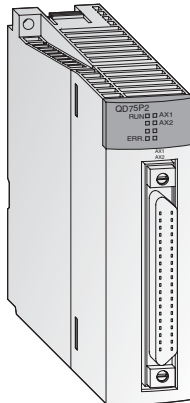
Besondere Merkmale:

- Max. Zählfrequenz bis zu 500 kHz
- Eingang für Inkrementaldrehgeber mit automatischer Vor- und Rückwärtserkennung
- Zählwertvorgabe und Funktionsauswahl über digitale Eingänge
- 32-Bit-Zählbereich mit Vorzeichen (-2 147 483 648 bis +2 147 483 647)
- Als auf- oder abwärtszählender Zähler oder Ringzähler einsetzbar
- Alle Module bieten zwei Zählereingänge.
- Pro Zählkanal stehen 2 digitale Ausgänge zur Verfügung, die abhängig vom Zählwert geschaltet werden

Alle Module werden über eine 40-polige Steckverbindung angeschlossen.

2.9.5 Positioniermodule

In Verbindung mit Schrittmotoren oder Servoverstärkern können die Positioniermodule QD75P1, QD75P2 und QD75P4 zur Positionierung oder Geschwindigkeitssteuerung eingesetzt werden.

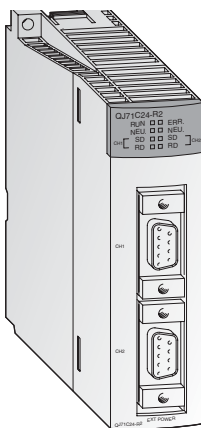


Besondere Merkmale:

- Steuerung von bis zu vier Achsen linearinterpolierend (QD75P4) oder zwei Achsen zirkularinterpolierend (QD75P2 und QD75P4)
- Speicherung von bis zu 600 Positionsdaten im Flash-ROM
- Als Einheiten bei der Positionierung können Impulse, μm , Inch oder Winkelgrade vorgegeben werden.
- Parametrierung und Positionsdatenvorgabe über das SPS-Programm oder mit Hilfe der Programmier-Software GX Configurator QP.

2.9.6 Schnittstellenmodule zur seriellen Übertragung

Die Module QJ71C24 und QJ71C24-R2 dienen zur Kommunikation mit Peripheriegeräten. Dabei werden standardisierte serielle Schnittstellen verwendet.

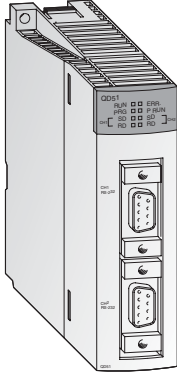


Besondere Merkmale:

- Zwei RS232C-Schnittstellen (bei QJ71C24-R2) oder eine RS422/485- und eine RS232C-Schnittstelle (bei QJ71C24)
- Übertragungsgeschwindigkeiten von bis zu 115200 Baud
- Zugriffsmöglichkeit auf die Daten der SPS durch übergeordnete PCs mit Visualisierungs- oder Monitorsoftware
- Der Anschluss eines Druckers ist möglich.
- Integrierter Speicher zur Ablage von Qualitäts-, Produktions- oder Alarmdaten, die nach Bedarf übertragen werden
- Ein freies Protokoll zum Datenaustausch kann definiert werden
- Die Programmierung der SPS über die Schnittstellenmodule ist möglich.

2.9.7 BASIC-programmierbare Schnittstellenmodule

Die Module QD51S-R24 und QD51 arbeiten unabhängig von der CPU der SPS ein eigenes Programm ab, das in AD51H-Basic geschrieben wird. So können Daten mit Peripheriegeräten ausgetauscht werden, ohne dass die SPS-CPU belastet wird.

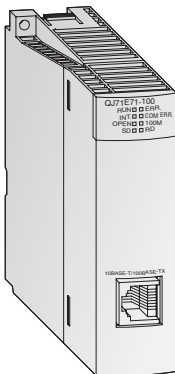


Besondere Merkmale:

- Entweder zwei RS232C-Schnittstellen (bei QD51) oder eine RS422/485- und eine RS232C-Schnittstelle (bei QD51S-R24)
- Übertragungsgeschwindigkeiten von bis zu 38400 Baud
- Auf Operanden innerhalb der SPS-CPU und Pufferspeichern von Sondermodulen kann zugegriffen werden.
- Über die Schnittstellenmodule kann die Betriebsart der SPS-CPU ferngesteuert geändert werden (RUN/STOP-Umschaltung)

2.9.8 ETHERNET-Module

Mit den Modulen QJ71E71 und QD71E71-B2 kann das MELSEC System Q über das ETHERNET mit anderen Geräten, wie z. B. einem Personal Computer, verbunden werden. Neben dem Datenaustausch per TCP/IP- oder UDP/IP-Kommunikation können über das ETHERNET auch SPS-Daten gelesen oder geändert sowie der Betrieb und der Zustand der CPU überwacht werden.

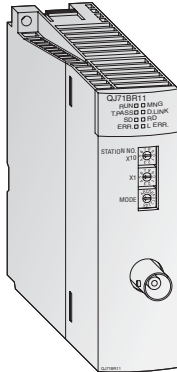


Besondere Merkmale:

- 10BASE5-, 10BASE2- oder 10BASE-T-Schnittstelle
- Übertragungsgeschwindigkeit von 10 bzw. 100 Mbit/s
- FTP-Serverfunktion ist möglich
- Datenaustausch über Sende- und Empfangspuffer mit fester Größe
- Bis zu 16 logische Verbindungen können gleichzeitig aufgebaut werden.
- Mit einem PC, auf dem die Software GX Developer oder GX IEC Developer installiert ist, kann das Programm der SPS über das ETHERNET geändert werden.

2.9.9 MELSECNET-Module

Die Module QJ71BR11 und QJ71LP21 ermöglichen die Anbindung des MELSEC System Q an ein MELSECNET/10 oder MELSECNET/10H und damit die Kommunikation mit Steuerungen der Q-, QnA und QnAS-Serie.

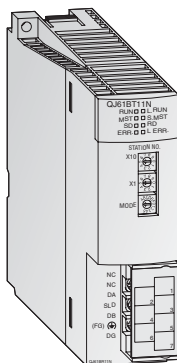


Besondere Merkmale:

- Zwei verschiedene Netzwerk-Topologien können genutzt werden: Koaxialer Bus (QJ71BR11) oder optischer Doppelring (QJ71LP21)
- Hohe Übertragungsgeschwindigkeit: 10 Mbit/s bei koaxialem Bus und wahlweise 10 oder 20 Mbit/s bei optischem Doppelring
- Datenaustausch mit SPS/PC und dezentralen E/A-Stationen ist möglich
- Daten können mit beliebigen Stationen ausgetauscht werden, unabhängig davon, wie viele Netzwerke zwischen den Stationen liegen.
- Ausblendung einer fehlerhaften Station beim koaxialen Bus und Loop-back-Funktion beim optischem Doppelring, wenn eine Station gestört ist.
- Bei Ausfall der Kontrollstation übernimmt eine andere Station automatisch deren Aufgabe

2.9.10 Master-Modul/Lokales Modul für CC-Link

Das QJ61BT11 ist in einem CC-Link-System als Master- oder lokale Station einsetzbar und dient zur Steuerung und Überwachung von dezentralen Ein- und Ausgängen.

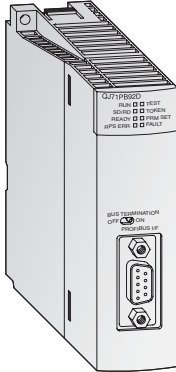


Besondere Merkmale:

- Die Parametrierung aller im Netzwerk vorhandenen Module erfolgt direkt über das Master-Modul.
- Automatische Kommunikation zwischen den dezentralen Geräten und dem Master-Modul. Die Abtastzeit für 2048 E/As beträgt nur 3,3 ms.
- Übertragungsgeschwindigkeiten von bis zu 10 Mbit/s
- Erweiterung eines Systems um bis zu 2048 dezentrale E/As durch ein Master-Modul
- Mit einem zusätzlichen Stand-by-Master kann ein redundantes System aufgebaut werden. Nach Ausfall der Master-Station wird die Kommunikation fortgesetzt.
- Automatischer Start des CC-Link ohne Parametrierung
- Abhängig von Bedingungen im Netzwerk können Interrupt-Programme gestartet werden.

2.9.11 PROFIBUS/DP-Modul

Das PROFIBUS/DP-Master-Modul QJ71PB92D und das PROFIBUS/DP-Slave-Modul QJ71PB93D erlauben den Datenaustausch von Steuerungen im MELSEC System Q mit anderen Geräten an einem PROFIBUS/DP-Netzwerk.

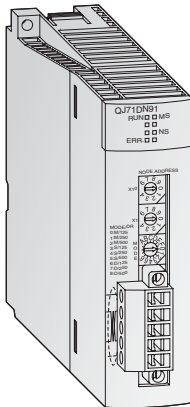


Besondere Merkmale:

- Die Master-Station kann mit bis zu 60 Slave-Stationen Daten austauschen.
- Pro Slave können 244 Eingangs- und 244 Ausgangsbytes verarbeitet werden.
- Globale Dienste wie SYNC und FREEZE sowie Diagnosefunktionen für bestimmte Slaves werden unterstützt.
- Der Datenaustausch kann über automatisch und zusätzlich durch Blockanweisungen stattfinden.

2.9.12 DeviceNet-Master-Modul QJ71DN91

Das QJ71DN91 verbindet eine SPS des MELSEC System Q mit dem DeviceNet. Das DeviceNet ist eine preiswerte Lösung zur Netzwerkanbindung von „Low-Level“-Endgeräten.



Besondere Merkmale:

- Die Positionen der Master-Station und der Slave-Stationen können vom Anwender frei gewählt werden.
- Übertragungsgeschwindigkeiten von 125, 250 oder 500 kBit/s
- Die Leitungslänge kann bis 500 m betragen.
- Kommunikationsmethoden:
 - Polling
 - Bit strobe
 - Zustandsänderung
 - Zyklisch

2.9.13 Web-Server-Modul

Durch das Web-Server-Modul QJ71WS96 wird die Fernüberwachung einer SPS des MELSEC System Q ermöglicht.



Besondere Merkmale:

- Zugriff auf die Steuerung via Internet
- Einfachste Parametrierung
- Nutzer benötigt für Einstellungen und zur Fernüberwachung nur einen Web-Browser.
- RS232-Schnittstelle zum Anschluss eines Modems
- Für die Kommunikation können verschiedene Netzwerkverbindungen genutzt werden: ADSL, Modem, LAN, etc.
- Senden und Empfangen von Daten per E-Mail oder FTP
- Selbstgestaltete Webseiten und Java-Applets integrierbar
- Standard-Verbindung über ETHERNET für den Datenaustausch mit anderen Steuerungen oder PCs
- Erfassung und Speicherung von Ereignissen und CPU-Zuständen

2.10 SPS-Grundlagen

2.10.1 Programmier-Software

Um eine speicherprogrammierbare Steuerung (SPS) mit einem üblichen PC programmieren zu können, ist eine besondere Programmier-Software erforderlich. Sie sollte die folgenden Anforderungen erfüllen:

- Zur Programmierung werden leicht verständliche und wiedererkennbare Symbole oder Abkürzungen verwendet, wie bei der Kontaktplanprogrammierung oder der Programmierung in Form einer Anweisungsliste.
- Die eingegebenen Anweisungen (Syntax) und die Funktionalität des Programms sollten geprüft werden können, bevor das Programm in die SPS übertragen wird.
- Die SPS-Programme müssen dauerhaft auf der Festplatte des PC oder einem anderen Datenträger gespeichert werden können.
- Es muss möglich sein, bereits vorhandene Programme von der Festplatte des PC oder einem anderen Datenträger zu laden.
- Die Programme müssen mit ausführlichem Kommentar versehen werden können.
- Das Programm muss ausgedruckt werden können.
- Über eine serielle Schnittstelle muss das Programm in die SPS übertragen werden können. Umgekehrt muss es auch möglich sein, ein Programm in der SPS in den PC zu übertragen.
- Die Ausführung des Programms und die Operandenzustände müssen in „Echtzeit“ beobachtet werden können.
- Während die SPS das Programm ausführt, müssen Programmänderungen möglich sein.
- Einstellungen und Parameter zum Betrieb der SPS müssen veränderbar sein.
- Operandenzustände der SPS müssen gespeichert und bei Bedarf wieder geladen werden können.
- SPS-Programme sollten ohne angeschlossene SPS simuliert werden können.

Dies sind nur einige der Anforderungen an eine Programmier-Software!

2.10.2 Programmverarbeitung in der SPS

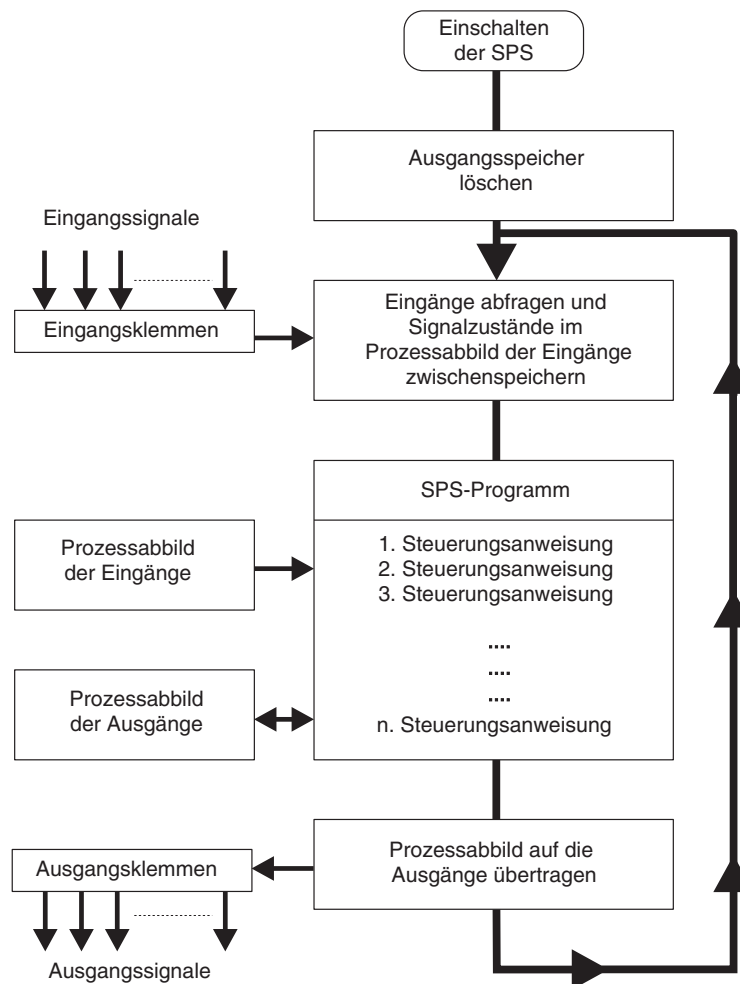
Eine SPS arbeitet nach einem vorgegebenen Programm, das in der Regel außerhalb der Steuerung erstellt, in die Steuerung übertragen und im Programmspeicher abgelegt wird. Für die Programmierung ist es wichtig zu wissen, wie das Programm von der SPS verarbeitet wird.

Das Programm besteht aus einer Folge einzelner Anweisungen, die die Funktion der Steuerung festlegen. Die SPS arbeitet die Steuerungsanweisungen in der programmierten Reihenfolge nacheinander (sequentiell) ab.

Der gesamte Programmdurchlauf wird ständig wiederholt, es findet also ein zyklischer Programmdurchlauf statt. Die für einen Programmdurchlauf benötigte Zeit wird als Programmzykluszeit bezeichnet.

Prozessabbildverfahren

Bei der Programmbearbeitung in der SPS wird nicht direkt auf die Ein- und Ausgänge, sondern auf ihr Prozessabbild zugegriffen:



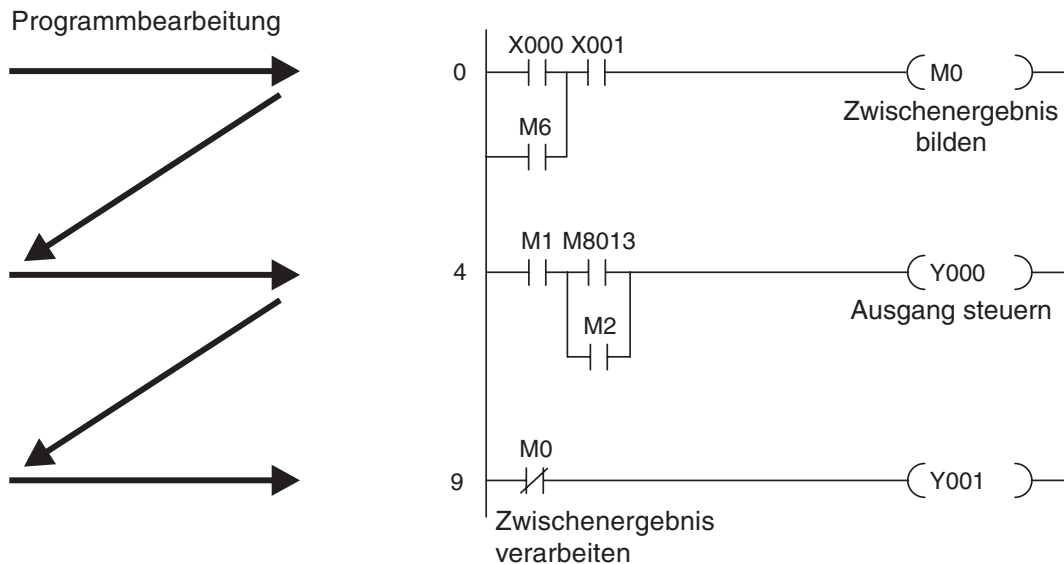
Prozessabbild der Eingänge

Am Anfang eines Programmzyklus werden die Signalzustände der Eingänge abgefragt und zwischengespeichert: Es wird ein sogenanntes Prozessabbild der Eingänge angelegt.

Programmdurchlauf

Während des anschließenden Programmdurchlaufs greift die SPS auf die gespeicherten Eingangszustände im Prozessabbild zu. Signaländerungen an den Eingängen werden daher erst im nächsten Programmzyklus erkannt.

Das Programm wird von oben nach unten, in der Reihenfolge der Eingabe, abgearbeitet. Zwischenergebnisse können noch im selben Programmzyklus verwendet werden.



Prozessabbild der Ausgänge

Verknüpfungsergebnisse, die die Ausgänge betreffen, werden in einem Ausgangszwischenspeicher hinterlegt (Prozessabbild der Ausgänge). Erst am Ende des Programmdurchlaufs werden die Zwischenergebnisse an die Ausgänge übertragen. Im Ausgangszwischenspeicher bleibt das Prozessabbild der Ausgänge bis zum nächsten Überschreiben erhalten. Nach der Wertzuweisung an die Ausgänge wird der Programmzyklus wiederholt.

Signalverarbeitung in der SPS im Vergleich zur verbindungsprogrammierten Steuerung

Bei einer verbindungsprogrammierten Steuerung ist das Programm durch die Art der Funktionsglieder und deren Verbindung (Verdrahtung) vorgegeben. Alle Steuerungsvorgänge werden gleichzeitig (parallel) ausgeführt. Jede Änderung der Eingangssignalzustände bewirkt sofort eine Änderung der Ausgangssignalzustände.


Bei einer SPS kann eine Änderung der Eingangssignalzustände während des Programmdurchlaufs erst wieder beim nächsten Programmzyklus berücksichtigt werden. Dieser Nachteil wird durch kurze Programmzykluszeiten weitgehend wieder ausgeglichen. Die Programmzykluszeit ist abhängig von der Anzahl und der Art der Steuerungsanweisungen.

2.10.3 Operanden einer SPS

Die Operanden einer SPS werden in Steuerungsanweisungen verwendet, das heißt, ihre Signalzustände oder Werte können durch das SPS-Programm abgefragt oder beeinflusst werden. Ein Operand besteht aus

- einem Operandenkennzeichen und
- einer Operandenadresse.

Beispiel für die Angabe eines Operanden (z. B. Eingang 0):



Beispiele für Operandenkennzeichen:

Operandenkennzeichen	Typ	Bedeutung
X	Eingang	Eingangsklemme der SPS (z. B. Schalter)
Y	Ausgang	Ausgangsklemme der SPS (z. B. Schütz oder Lampe)
M	Merker	Zwischenspeicher in der SPS, der zwei Zustände („Ein“ oder „Aus“) annehmen kann
T	Timer	„Zeitrelais“ zur Realisierung von zeitabhängigen Funktionen
C	Counter	Zähler
D	Datenregister	Datenspeicher in der SPS, in dem z. B. Messwerte oder Rechenergebnisse abgelegt werden können.

3 Programmierung

3.1 Die Norm IEC 61131-3

Die Norm IEC 61131-3 ist der internationale Standard für SPS-Programme. (IEC: *International Electromechanical Commission*). IEC 61131-3 umfasst nicht nur die SPS-Programmiersprachen, sondern bietet auch umfassende Konzepte und Richtlinien zum Aufbau eines SPS-Projekts.

Durch den Einsatz der Norm IEC 61131-3 und die Anwendung von modularen Anweisungen können strukturierte Programme erstellt werden. Dadurch steigt die Effektivität, weil bereits geprüfte Programmteile wiederverwendet werden können. Gleichzeitig wird die Wahrscheinlichkeit von Programmfehlern verringert.

Durch die konsequente Anwendung der Norm IEC 61131-3 wird die Fehlersuche erheblich erleichtert, weil einzelne Programmelemente unabhängig voneinander geprüft werden können.

Ein anderer wichtiger Vorteil der Norm IEC 61131-3 ist deren Unterstützung bei der Verwaltung eines Projekts und der Qualitätssicherung. Insbesondere unterstützt die strukturierte Arbeitsweise der IEC61131-3 den Ablauf von Prozessen, die durch eine SPS gesteuert werden. In einigen Industriezweigen wird die strukturierte Programmierung sogar vorgeschrieben. So zum Beispiel in der pharmazeutischen oder petrochemischen Industrie, wo hohe Sicherheitsanforderungen erfüllt werden müssen.

Auf den ersten Blick erscheint die Programmierung nach der Norm IEC 61131-3 aufwendig und arbeitsintensiv. Die Vorteile der strukturierten Programmierung gegenüber unstrukturierten Programmen sprechen aber für die IEC61131-3.

PLCopen



PLCopen ist eine hersteller- und produktunabhängige Organisation. Sie wurde gegründet, um die Programmierung nach IEC61131-3 in Industriesteuerungen weiter zu fördern. Diese Organisation hat drei Ebenen für den Entwurf und die Implementierung von Systemen definiert, die die Übereinstimmung mit der Norm IEC61131-3 angeben.

PLCopen hat festgelegt:

- ein Zulassungsverfahren
- zugelassene Test-Institute
- Die Anforderungen an eine Test-Software, die gemeinsam von den Mitgliedern entwickelt wurde.
- ein definiertes Zertifizierungsverfahren
- Mitglieder mit zertifizierten Produkten

Dadurch wird die Übereinstimmung mit der Norm IEC 61131-3 sichergestellt – jetzt und in der Zukunft.

PLCopen Certification



Die Software GX IEC Developer von Mitsubishi ist voll kompatibel mit „**Base Level IL**“ (Anweisungsliste) und „**Base Level ST**“ (Strukturierter Text) und wurde für diese Standards komplett zertifiziert.

3.2 Software-Struktur und Festlegung von Begriffen

In diesem Abschnitt werden die folgenden Begriffe behandelt, die im Umgang mit der Programmier-Software verwendet werden:

- POE
- Globale Variablen
- Lokale Variablen
- Anwenderdefinierte Funktionen und Funktionsbausteine
- Task-Pool
- Programmiersprachen
 - Anweisungsliste (AWL)
 - Kontaktplan (KOP)
 - Funktionsbausteinsprache (FBS)
 - Ablaufsprache (AS)
 - Strukturierter Text (ST)
 - MELSEC-Anweisungsliste

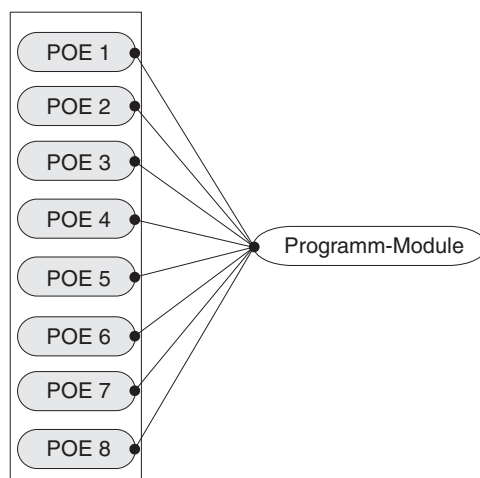
3.2.1 Definition von Begriffen in der Norm IEC61131-3

Projekt

Ein Projekt umfasst die Programme, die Dokumentation und die Parameter, die für eine Anwendung benötigt werden.

POE - Programm-Organisationseinheit

Ein Gesamtprogramm wird in der IEC (6)1131-3 in Teilprogramme, den Programm-Organisationseinheiten (POE) aufgeteilt. Eine POE ist die kleinste unabhängige Software-Einheit eines Ablauf-Programmes.



Ein SPS-Programm wird in einzelne Programm-Organisationseinheiten (POE) aufgeteilt.

Man unterscheidet drei Arten von POEs, die auf Grund ihrer Funktionalität eingeteilt werden:

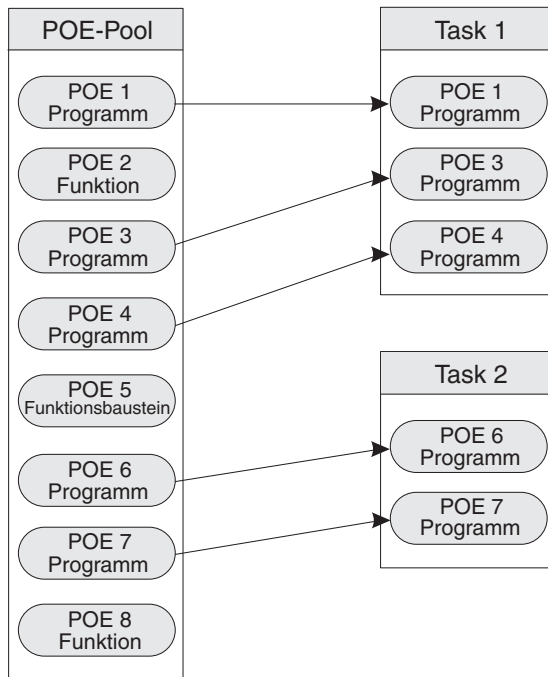
- Programmbausteine (PRG)

- Funktionen (FUN)
- Funktionsbausteine (FB)

Funktionen und Funktionsbausteine sind Unterprogramme, die in anderen POEs aufgerufen werden können.

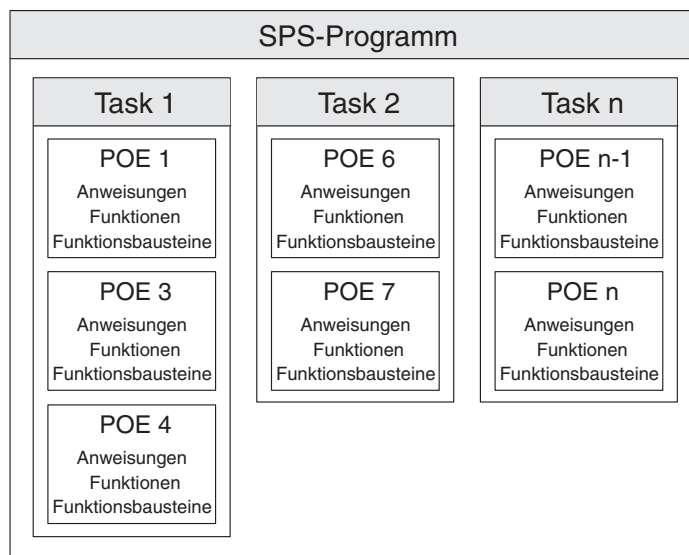
Das endgültige Programm wird von den Programm-POEs gebildet, die in sogenannten **Task** zusammengefasst sind.

Tasks

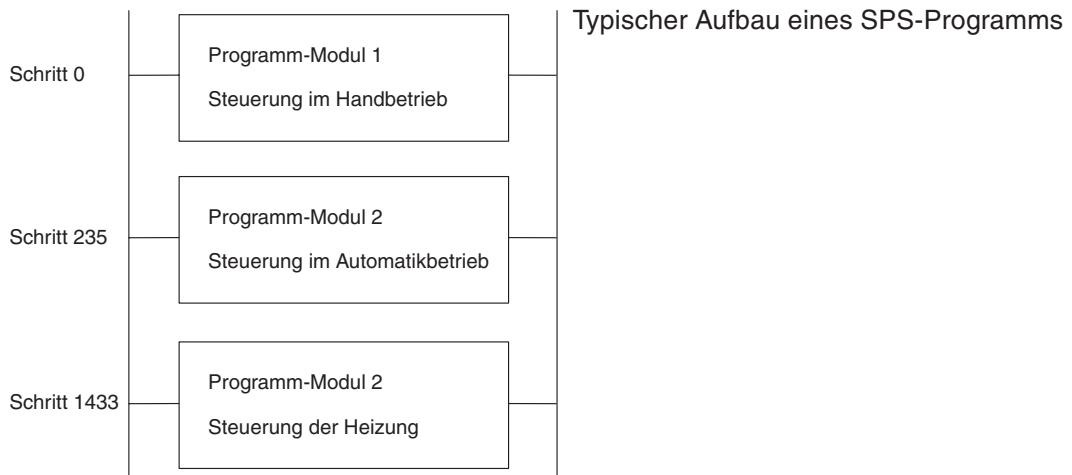


Die POEs werden im POE-Pool abgelegt. Die Programm-POEs werden zu einer Gruppe (Task) zusammengefasst.

Die einzelnen Task wiederum bilden das Gesamtprogramm:

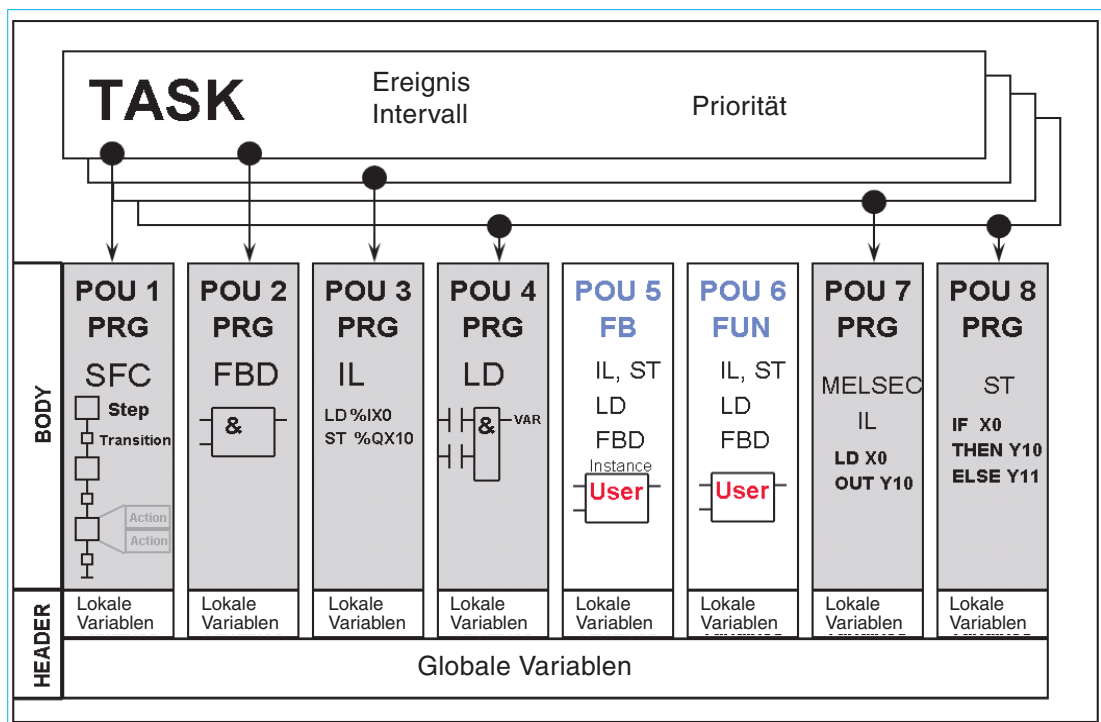


Die meisten SPS-Programme sind so geschrieben, dass bestimmte Programmteile für bestimmte Aufgaben zuständig sind. Sie können aus einem großen Programm oder aus mehreren Unterprogrammen bestehen. Die Unterprogramme werden im Hauptprogramm zum Beispiel durch CALL- oder CJ-Anweisungen aufgerufen.



Bei dem oben abgebildeten Programm bietet es sich an, für jedes Programmmodul, das eine spezielle Aufgabe erfüllt, eine separate Task zu verwenden.

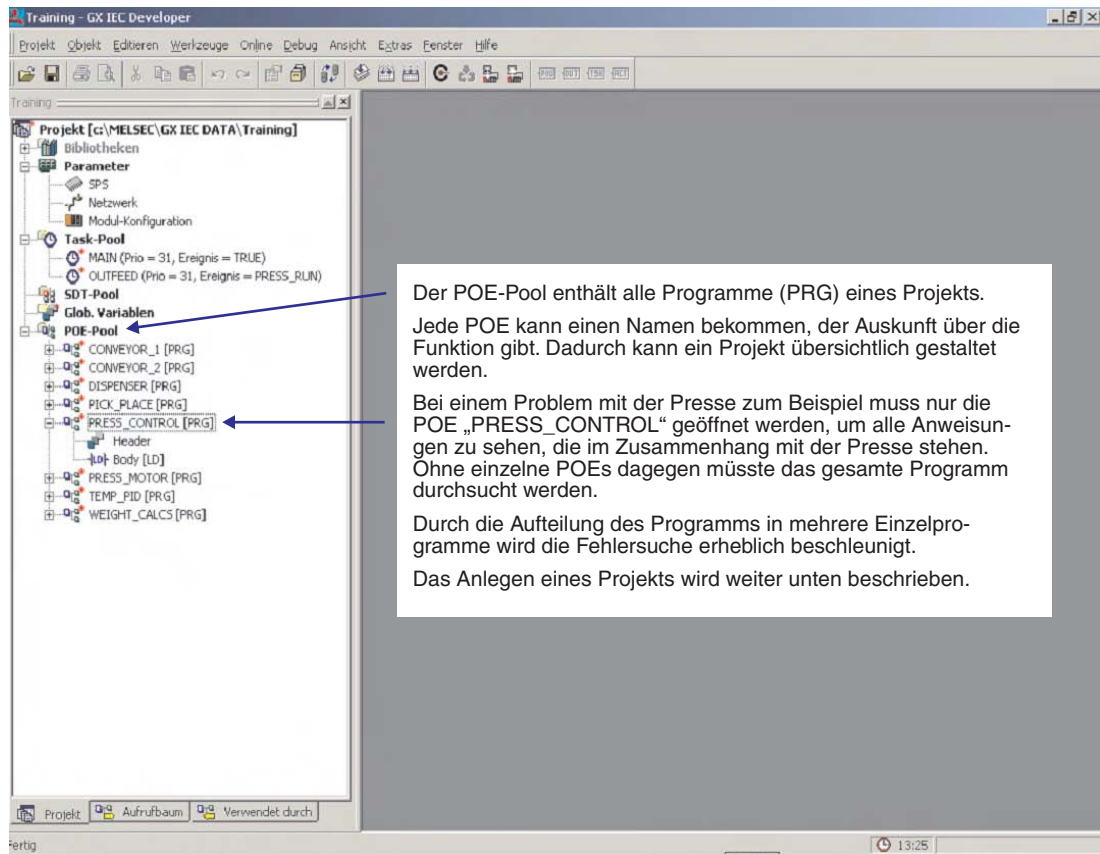
Eine POE kann in jeder der zur Verfügung stehenden Programmiersprachen (KOP, AWL, FBS, AS oder ST) programmiert werden. Die folgende Abbildung zeigt ein Beispiel, bei der POEs unterschiedlicher Programmiersprachen in einem Projekt zusammengefasst sind.



POE-Pool

Ein Projekt besteht meist aus mehreren Programmorganisations-Einheiten (POE), von denen jede eine bestimmte Steuerungsaufgabe erfüllt. Alle POEs sind im POE-Pool zusammengefasst. Eine POE kann in einer beliebigen IEC-Programmiersprache erstellt worden sein. Dadurch kann für eine Anwendung die Programmiersprache gewählt werden, die am besten zur Lösung einer Aufgabe geeignet ist. Zum Beispiel können aufwendige Verriegelungsfunktionen mit einem grafischen Editor, wie etwa Kontaktplan, und aufwendige Berechnungen mit einem Texteditor oder der Funktionsbausteinsprache programmiert werden. Die Programmier-Software setzt das Programm in einen Code um, den die SPS verarbeiten kann. Für den Programmierer ändert sich nach dieser Konvertierung das Aussehen des Programms aber nicht.

Die folgende Abbildung zeigt ein Beispiel für einen POE-Pool.



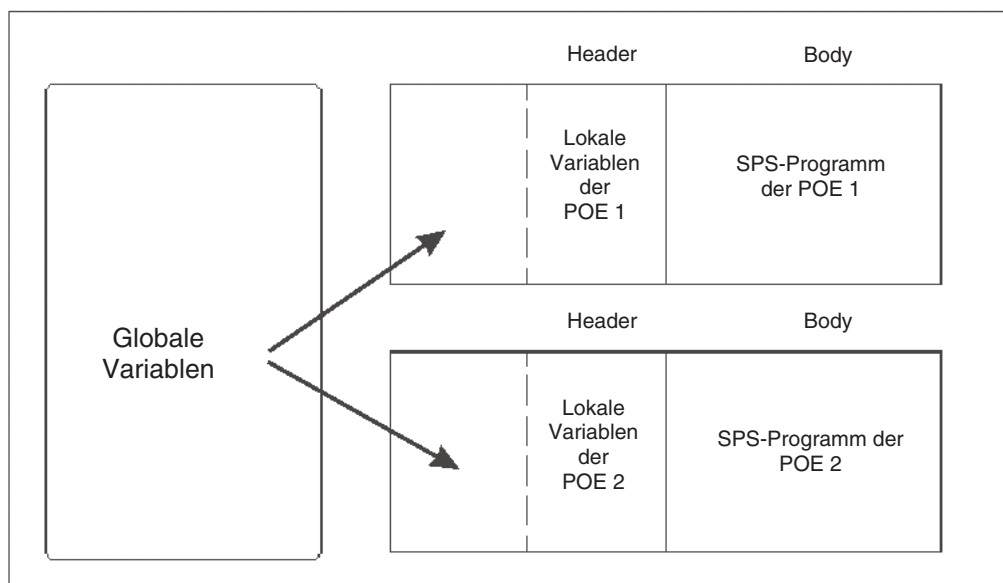
Elemente einer POE

Jede Programm-Organisationseinheit setzt sich zusammen aus:

- dem Header und
- dem Body

Im Header werden die Variablen festgelegt, die in dieser POE verwendet werden.

Der Body enthält das eigentliche SPS-Programm in den verschiedenen Sprachen.



Variable – lokal und global

- Variablen

Variablen enthalten die Werte von Eingängen, Ausgängen oder internen Operanden der SPS. Schon vor Beginn der Programmierung muss festgelegt werden, welche Variablen in den einzelnen Programmblöcken benötigt werden. Zu jeder POE gehört eine Liste mit Variablen, die ausschließlich für diese POE festgelegt wurden und die nur innerhalb dieser POE verwendet werden: die Lokalen Variablen.

Variablen, auf die alle POEs eines Programms zugreifen können, werden in einer separaten Liste eingetragen, dies sind die Globalen Variablen.

- Lokale Variablen

Wenn Operanden in eine Liste mit Lokalen Variablen eingetragen werden, reserviert die Software GX IEC Developer automatisch System Variablen für die Speicherung von Daten innerhalb einer bestimmten POE. Auf diese Variablen können andere Programmteile des Projekts nicht zugreifen.

- Global Variablen

Globale Variablen können als gemeinsame Variablen angesehen werden. Sie sind die Schnittstelle zu den physischen SPS-Operanden wie zum Beispiel Ein- oder Ausgängen. Globale Variablen gelten für das gesamte Programm und können in allen Programm-POEs und Funktionsbausteinen verwendet werden. Sie verweisen entweder auf vorhandene Ein- und Ausgänge der SPS oder auf interne SPS-Operanden. Globale Variablen ermöglichen einen Datenaustausch zwischen den POEs, aber auch mit externen Geräten wie MMI oder SCADA.

IEC61131-3 Variablen im Vergleich mit MELSEC Variablen

Mit dem GX IEC Developer können bei der Programmierung entweder „Bezeichner“ (Symbolischer Name einer Variablen) oder absolute Adressen in der Mitsubishi-Schreibweise (X0, M0 etc.) verwendet werden.

Die Verwendung symbolischer Namen entspricht der Norm IEC 61131-3.

Falls symbolische Namen verwendet werden, müssen diese absoluten SPS-Adressen zugeordnet werden.

Liste der Lokalen Variablen

Damit in einer POE auf eine Globale Variable zugegriffen werden kann, muss diese Variable im Header der POE eingetragen werden. Im Header können Lokale und Globale Variablen eingetragen werden.

Eine Lokale Variable kann als Speicher für ein Zwischenergebnis angesehen werden. Um zum Beispiel im Programm eine Berechnung mit drei Werten in fünf Schritten auszuführen, müssen mehrere Zwischenergebnisse in Datenregister gespeichert werden, bevor in einem Register das Endergebnis eingetragen werden kann. Da die Zwischenergebnisse nur kurzzeitig in einem Programmteil gespeichert werden müssen, können sie in Lokale Variablen abgelegt werden. Auf die drei Anfangswerte und Endergebnis dagegen müssen auch andere POEs zugreifen können. Daher werden Sie in Globale Variablen gespeichert.

Liste der Globalen Variablen

In der Liste der Globalen Variablen werden alle Bezeichner für tatsächliche SPS-Operanden wie zum Beispiel E/A-Adressen eingetragen.

Globale Variablen gelten für das gesamte Programm und können in allen Programm-POEs und Funktionsbausteinen verwendet werden.

Task-Pool und Task Manager

Die POEs, die als Programme deklariert wurden, werden in Gruppen, den Task zusammengefasst. Sämtliche Tasks bilden das Gesamtprogramm.

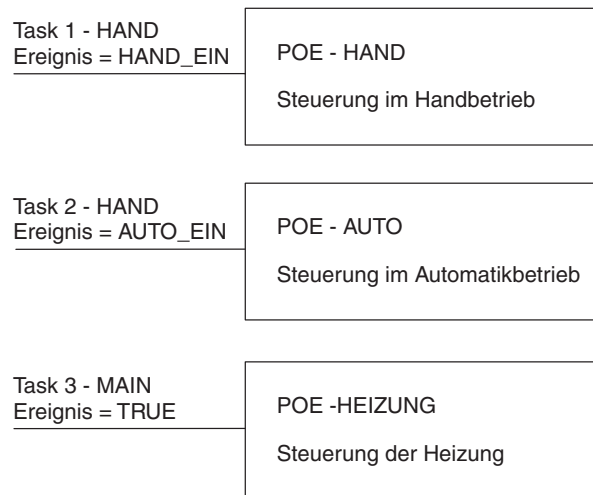
Für die einzelnen Tasks können Ausführungsbedingungen vergeben werden.

- Ereignis: Aufruf der Task z. B. bei einem Interrupt oder zyklisch (TRUE)
- Intervall: Ausführung in definierten Zeitintervallen
- Priorität: Festlegung der Reihenfolge der Abarbeitung der Task

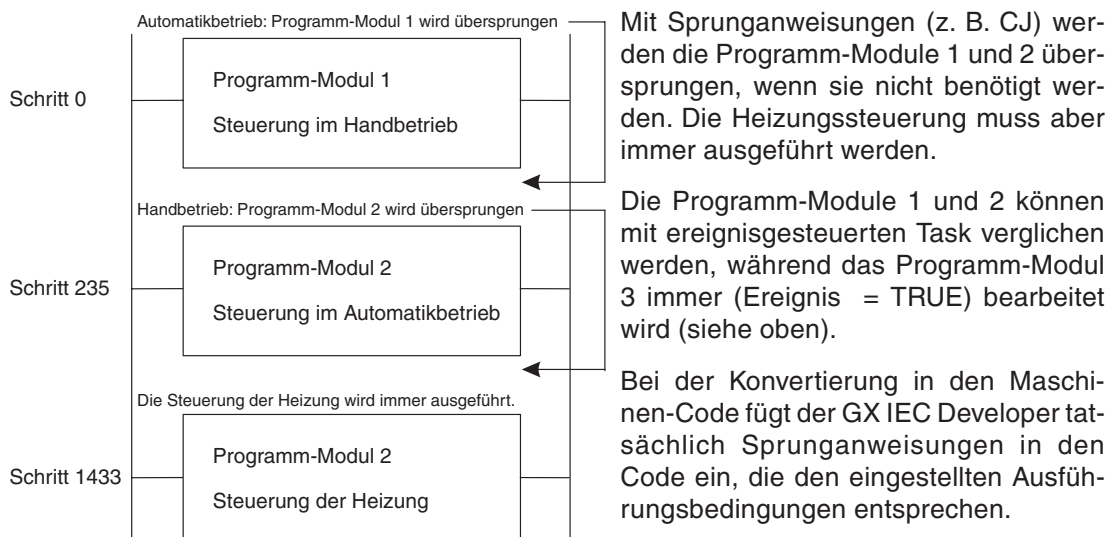
In der folgenden Abbildung wurden für die drei Task verschiedene Ausführungsbedingungen eingestellt:

- Task 1 wird nur ausgeführt, wenn der Merker HAND_EIN gesetzt (1) ist.
- Task 2 wird nur ausgeführt, wenn der Merker AUTO_EIN gesetzt (1) ist.
- Task 3 wird zyklisch (ständig) ausgeführt (Wegen Ereignis = TRUE)

Die Bezeichner der Merker müssen als Globale Variablen definiert und SPS-Operanden (Bit-Operanden wie z. B. den Eingängen X0 und X1) zugewiesen werden.



Zur Veranschaulichung der Ausführungsbedingungen kann das oben abgebildete Programm mit einem Programm verglichen werden, in denen alle Anweisungen untereinander stehen, bestimmte Programmteile aber übersprungen werden.



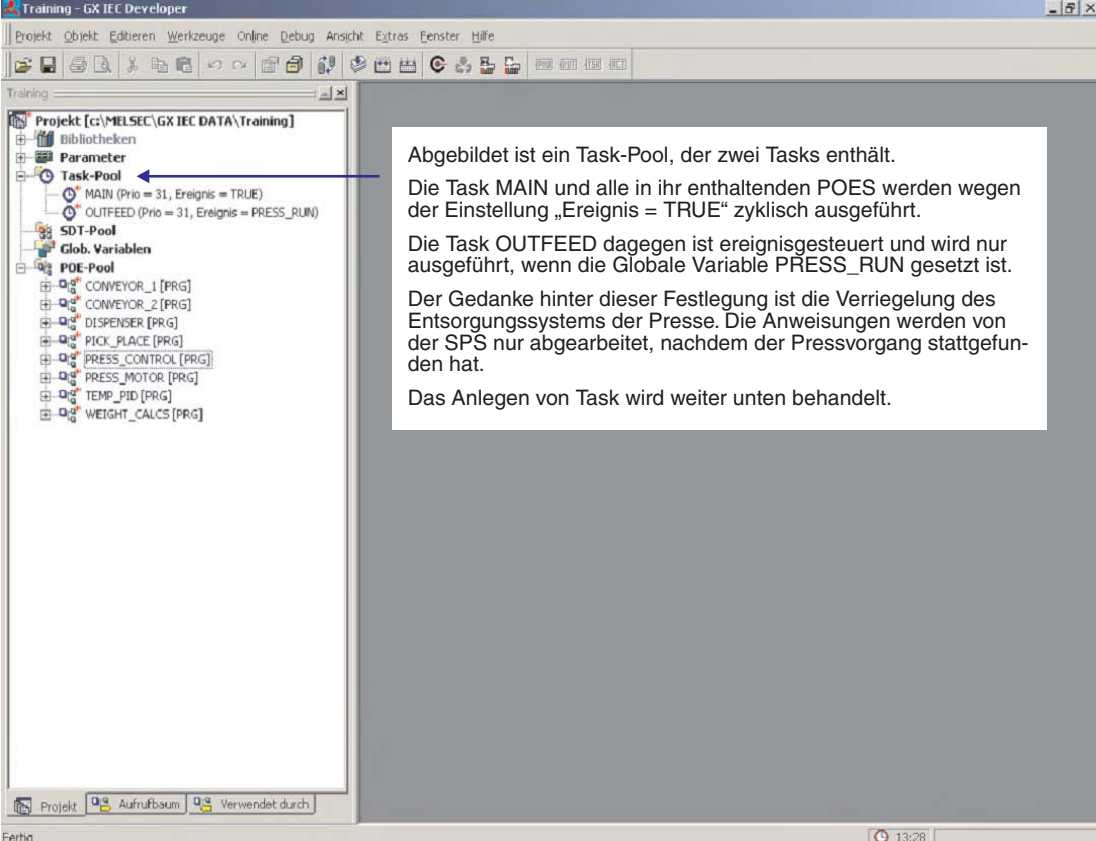
Einer Task können mehrere POEs zugewiesen werden. Eine Task, deren Ausführungsbedingung „Ereignis = TRUE“ lautet, wird zum Beispiel alle POEs enthalten, die zyklisch ausgeführt werden müssen. Eine POE kann aber nicht mehreren Task zugewiesen werden.

Task können mit verschiedenen Prioritäten ausgeführt werden. Die Priorität kann durch ein Zeitintervall oder durch Interrupts vorgegeben werden.

HINWEIS

Eine POE, die keiner Task zugewiesen wurde, wird beim Transfer des Projekts **nicht** in die SPS übertragen.

Der **Task Pool** enthält alle Task eines Projekts.



Abgebildet ist ein Task-Pool, der zwei Tasks enthält.

Die Task MAIN und alle in ihr enthaltenden POES werden wegen der Einstellung „Ereignis = TRUE“ zyklisch ausgeführt.

Die Task OUTFEED dagegen ist ereignisgesteuert und wird nur ausgeführt, wenn die Globale Variable PRESS_RUN gesetzt ist.

Der Gedanke hinter dieser Festlegung ist die Verriegelung des Entsorgungssystems der Presse. Die Anweisungen werden von der SPS nur abgearbeitet, nachdem der Pressvorgang stattgefunden hat.

Das Anlegen von Task wird weiter unten behandelt.

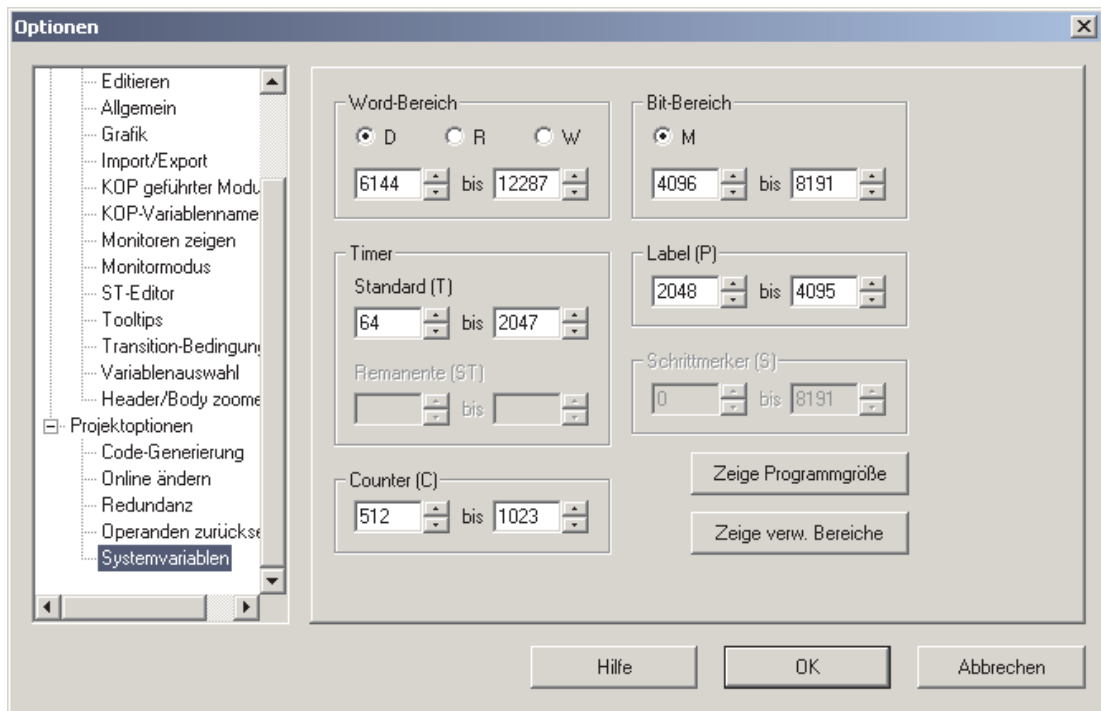
Mit dem **Task-Manager** kann der SPS-Zyklus effizient genutzt werden, indem nur die Programmteile ausgeführt werden, die tatsächlich gebraucht werden. Zusätzlich bietet er eine einfache Möglichkeit zur Zuordnung von Programmteilen zu bestimmten Ereignissen und Prioritäten.

Der Programmierer kann sich auf den Inhalt der Programme konzentrieren und muss sich nicht darum kümmern, ob Sprunganweisungen korrekt sind und die Regeln eingehalten werden.

Das Verwalten von Programmteilen im Task-Manager bietet auch Vorteile bei Serienmaschinen. Steuerungen für Anlagen oder Prozesse, die aus vielen Standard-Funktionsgruppen bestehen, können POEs für jede einzelne Funktionsgruppe enthalten. Das gesamte Programm besteht in diesem Fall aus vielen POEs. Werden verschiedene Varianten einer Maschine oder einer Anlage ausgeliefert, müssen im Task Manager nur die POEs für die spezielle Maschine aktiviert werden. Beim Transfer des Programms in die SPS werden nur die tatsächlich benötigten POEs übertragen.

3.2.2 Systemvariablen

Die Operandenbereiche, die der GX IEC Developer Systemvariablen zuweist, können im Menü **Extras** unter dem Menüpunkt **Optionen** eingestellt werden.



Die gewählten Systemvariablen gelten für das aktuelle Projekt. (Bei anderen Steuerungen als dem MELSEC System Q können eventuell nicht alle hier beschriebenen Einstellungen vorgenommen werden.)

- **Wort-Bereich**
 D: Als Wort-Systemvariable werden Datenregister verwendet.
 R: Als Wort-Systemvariable werden File-Register verwendet.
 W: Als Wort-Systemvariable werden Link-Register verwendet.
 von/bis: Der einstellbare Bereich hängt von der verwendeten SPS-CPU ab.
- **Timer**
 Standard (T) – Bereich der normalen Timer
 Remanente (ST) – Bereich der remanenten Timer
 von/bis: Der einstellbare Bereich hängt von der verwendeten SPS-CPU ab.
- **Counter (C)**
 von/bis: Der einstellbare Bereich hängt von der verwendeten SPS-CPU ab.
- **Bit-Bereich**
 M: Als Bit-Systemvariable werden Merker verwendet.
 von/bis: Der einstellbare Bereich hängt von der verwendeten SPS-CPU ab.
- **Label (P)**
 Label, die als System-Label verwendet werden (siehe Abschnitt 3.2.3)
 von/bis: Der einstellbare Bereich hängt von der verwendeten SPS-CPU ab.

- Schrittmerker (S)

von/bis: Der einstellbare Bereich hängt von der verwendeten SPS-CPU ab.

- Zeige Programmgröße

In einem separaten Fenster wird angezeigt, wie viele Schritte das Hauptprogramm (Hpt.) sowie die Unterprogramme (Sub) und Programme in Ablaufsprache (AS) enthalten. Falls Programme noch nicht in Maschinensprache konvertiert wurden, wird an Stelle der Programmgröße ein Fragezeichen angezeigt. Falls in der verwendeten CPU keine Ablauf- oder Unterprogramme ausgeführt werden können, wird die entsprechende Zeile grau dargestellt.

- Zeige verw. Bereiche

Nach einem Klick auf dieses Schaltfeld werden die eingestellten Operandenbereiche für die Systemvariablen angezeigt.



3.2.3 System-Label

Die System-Label, die innerhalb der Systemvariablen eingestellt werden können (siehe vorherigen Abschnitt 3.2.2) werden vom GX IEC Developer für die interne Verwaltung des Projekts verwendet. GX IEC Developer braucht System-Label für:

- die Kennzeichnung von Netzwerken in Programmen (Netzwerk-Label)
- Ereignisgesteuerte Task (außer bei der Einstellung „Ereignis = TRUE“)
- Anwenderdefinierte Funktionsbausteine (ein Label pro Funktionsbaustein, außer bei Makro-Programmierung)
- System-Timer (Diese werden vom Task-Manager für intervallgesteuerte Task und als lokale Timer verwendet.)

3.3 Programmiersprachen

Die Software GX IEC Developer bietet verschiedene Möglichkeiten zur Programmierung. Die folgenden Programmiersprachen können genutzt werden:

Text-Editoren

- Anweisungsliste (IEC und MELSEC)
- Strukturierter Text

Grafik-Editoren

- Kontaktplan
- Funktionsbausteinsprache
- Ablaufsprache

Mit Ausnahme der Ablaufsprache kann ein Programm bei allen Programmiersprachen in einzelne Abschnitte, sogenannten Netzwerken, unterteilt werden. Diese Netzwerke können mit „Label“ gekennzeichnet werden. Ein Label kann aus bis zu 8 Zeichen bestehen, die mit einem Doppelpunkt abgeschlossen werden (z. B. START:). Netzwerke werden fortlaufend nummeriert und können in Sprunganweisungen als Ziele verwendet werden.

3.3.1 Text-Editoren

Anweisungsliste (AWL)

Bei der Programmierung in Form einer Anweisungsliste werden die Steuerungsanweisungen direkt als Text eingegeben.

Eine Anweisungsliste besteht aus einer Folge von Steuerungsanweisungen. Eine Steuerungsanweisung besteht aus einer Anweisung (Befehl) und einem oder – bei Applikationsanweisungen – auch mehreren Operanden. Einige Steueranweisungen kommen auch ohne Operanden aus. Diese Anweisungen steuern zum Beispiel die Programmverarbeitung in der SPS.

Jede Steuerungsanweisung muss in einer separaten Zeile eingetragen werden. Jeder Anweisung kann ein zusätzliches Label oder Kommentar zugewiesen werden.

Es werden zwei verschiedenen Arten von Anweisungslisten verwendet:

- IEC-Anweisungsliste
 - IEC-Anweisungslisten werden genauso eingegeben oder bearbeitet wie MELSEC-Anweisungslisten (siehe unten). Die folgenden Unterschiede müssen jedoch bei der Programmierung beachtet werden:
 - MELSEC-Netzwerke in einer IEC-Anweisungsliste
 - In einer IEC-Anweisungsliste können MELSEC-Netzwerke eingefügt werden. Dadurch können MELSEC-Systemanweisungen verwendet werden.
 - Der Akkumulator
 - Im Akkumulator werden ähnlich wie bei Hochsprachen Ergebnisse gespeichert. Das Ergebnis jeder Operation wird direkt nach der Ausführung der Anweisung im Akkumulator abgelegt. Dadurch enthält der Akkumulator immer das Ergebnis der zuletzt ausgeführten Anweisung. Der Anwender muss keine Eingangsbedingungen (Ausführungsbedingungen) programmieren, die Ausführung hängt immer vom Inhalt des Akkumulators ab.

Ein ausführlichere Beschreibung der IEC-Anweisungsliste finden Sie in Kapitel 16.

- MELSEC-Anweisungsliste

MELSEC-Anweisungslisten werden genauso eingegeben oder bearbeitet wie IEC-Anweisungslisten (siehe oben). Es können jedoch nur MELSEC-Anweisungen verwendet werden, die Programmierung nach dem IEC-Standard ist nicht möglich.

MELSEC	LD	X0
	CJ	P_20
	LD	X1
	POU	Y0
P_20 MELSEC	LD	X2
	OUT	Y1

Beispiel für ein MELSEC-Netzwerk

Strukturierter Text (ST)

Strukturierter Text ist ein nützliches Werkzeug. Besonders Programmierer, die mit Hochsprachen vertraut sind, werden dieses Werkzeug zu schätzen wissen. Wenn bei der Programmierung die Wirkungsweise einer SPS berücksichtigt und das Programm sorgfältig vorbereitet wird, ist die Programmierung mit Strukturierem Text sehr komfortabel.

Der Editor für strukturierten Text ist kompatibel zur IEC 61131-3, es werden alle Anforderungen erfüllt.

(*Beispiel für Strukturierem Text*)

Beispiel für strukturierem Text

```
Y20:=X10;
Y21:=X11 AND X12 OR X13;
M0:=(M1 AND (M2 OR M3)) OR X14;
```

In Kapitel 17 ist die Programmierung mit Strukturierem Text detaillierter beschrieben.

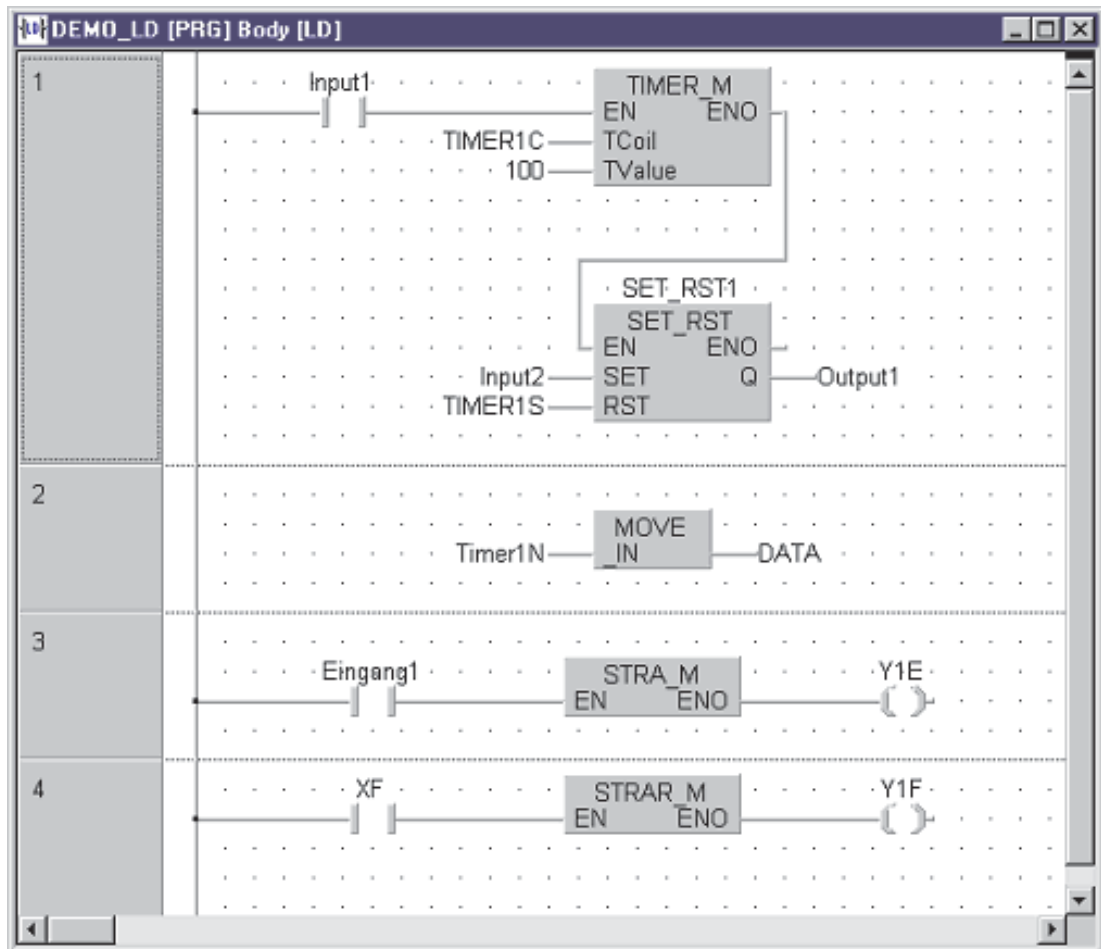
3.3.2 Grafik-Editoren

Kontaktplan (KOP)

Die Programmierung im Kontaktplan ist ähnlich dem Zeichnen von Schaltplänen für konventionelle Schützsteuerungen. Ein Kontaktplan besteht aus Eingangskontakten (Schließer und Öffner), Ausgängen (Spulen), aber auch Funktionen und Funktionsbausteinen. Diese Elemente werden durch horizontale Linien verbunden. Untereinander angeordnete Elemente können durch vertikale Linien verbunden werden. So entsteht grafisch das Programm. Ein Netzwerk beginnt im Kontaktplan immer an der linken Sammelschiene.

Funktionen und Funktionsbausteinen werden in einem Kontaktplan-Programm als Kästen dargestellt. Zusätzlich zu den Ein- und Ausgängen, die für die Funktion erforderlich sind, besitzen Funktionen und Funktionsbausteinen einen EN-Eingang und einen ENO-Ausgang. Über den EN-Eingang (ENable = Freigabe) kann die Ausführung der Funktion oder des Funktionsbausteins gesteuert werden. Der ENO-Ausgang (ENO = ENable Out) hat immer den selben Zustand wie der EN-Eingang.

Beispiel für Kontaktplan:



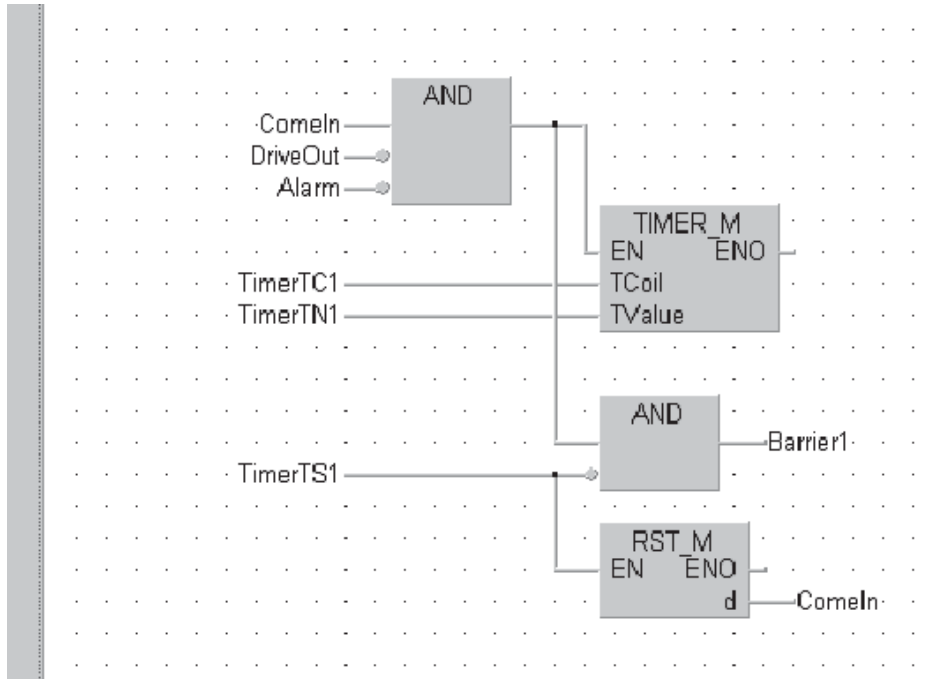
Funktionsbausteinsprache (FBS)

Bei der Funktionsbausteinsprache werden alle Elemente als Blöcke angegeben. Diese Blöcke werden mit horizontalen und vertikalen Linien verbunden. Eine Sammelschiene ist nicht vorhanden.

Zusätzlich zu den Ein- und Ausgängen, die für die Funktion erforderlich sind, besitzen einige Blöcke einen EN-Eingang und einen ENO-Ausgang. Über den EN-Eingang (ENable = Frei-

gabe) kann die Ausführung der Funktion oder des Funktionsbausteins gesteuert werden. Der ENO-Ausgang (ENO = ENable Out) hat immer denselben Zustand wie der EN-Eingang.

Beispiel für die Programmierung in der Funktionsbausteinsprache:

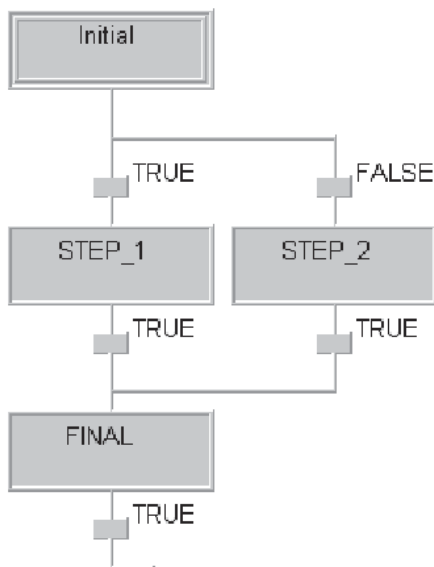


Ablaufsprache (AS)

Die Ablaufsprache ist eine grafische Strukturierungssprache, mit der Prozessabläufe übersichtlich dargestellt werden können.

Die einzig mögliche Programmklasse für ein AS-Programm ist „PRG“ (Programm). Das heißt, ein AS-Programm kann keine Funktion oder Funktionsbaustein sein.

Die Ablaufsprache besteht im wesentlichen aus zwei Grundelementen: Schritte und Transitionen. Ein Prozess besteht aus einer Reihe von Schritten, die durch Transitionen voneinander getrennt sind. Zur selben Zeit kann nur ein Schritt des Programms aktiviert sein. Ein Schritt wird nur aktiviert, wenn der vorherige Schritt vollständig abgearbeitet wurde und die Bedingung für die Transition erfüllt ist.



Beispiel für ein Programm in Ablaufsprache

3.4 Datentypen

Die Programmier-Software GX IEC Developer unterstützt die in diesem Abschnitt beschriebenen Datentypen.

3.4.1 Einfache Datentypen

Datentyp		Wertebereich		Größe	Zuweisbare SPS-Operanden / gültige CPU-Typen
BOOL	Boolean	Bit-Operand	0 (FALSE), 1 (TRUE)	1 Bit	X, Y, M, B
INT	Ganze Zahl	Register	-32768 bis +32767	16 Bit	D, W, R
DINT	Doppelte ganze Zahl		-2.147.483.648 bis 2.147.483.647	32 Bit	
WORD	Bit-Folge	K4M0*	0 bis 65.535	16 Bit	X, Y, M, B
DWORD		K8M0*	0 bis 4.294.967.295	32 Bit	
REAL	Gleitkommazahl	7 Stellen		32 Bit	Alle CPUs des MELSEC System Q *
STRING	Zeichenfolge	20 Zeichen (Voreinstellung)		32 Bit	
TIME	Zeitwert	-T#24d0h31m23s64800ms bis T#24d20h31m23s64700ms		32 Bit	

* Frühe Versionen der Q00JCP unterstützen diese Datentypen nicht.

3.4.2 Komplexe Datentypen

ARRAYS

Ein Array (Feld) besteht aus Variablen des gleichen Typs. Zum Beispiel ist ein **ARRAY [0..2] OF INT** ein eindimensionales Array mit drei Integer-Elements (0,1,2). Falls als Startadresse des Arrays das Register D0 angegeben wird, besteht das Array aus D0, D1 and D2.

Bezeichner	Adresse	Typ	Größe
Motor_Volt	D0	ARRAY	[0...2] OF INT

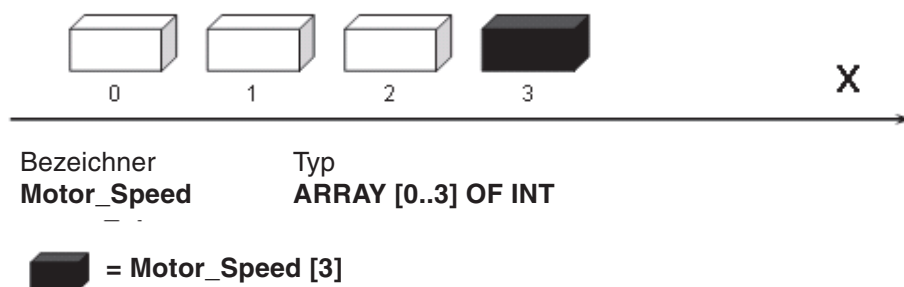
Im Programm können einzelne Array-Elemente angegeben werden. Zum Beispiel wird bei der Angabe von „Motor_Volts[1]“ und „ Motor_Volts[2]“ auf die Register D1 und D2 zugegriffen.

Arrays können bis zu 3 Dimensionen haben. Zum Beispiel hat ARRAY [0...2, 0...4] drei Elemente in der ersten und fünf Elemente in der zweiten Dimension.

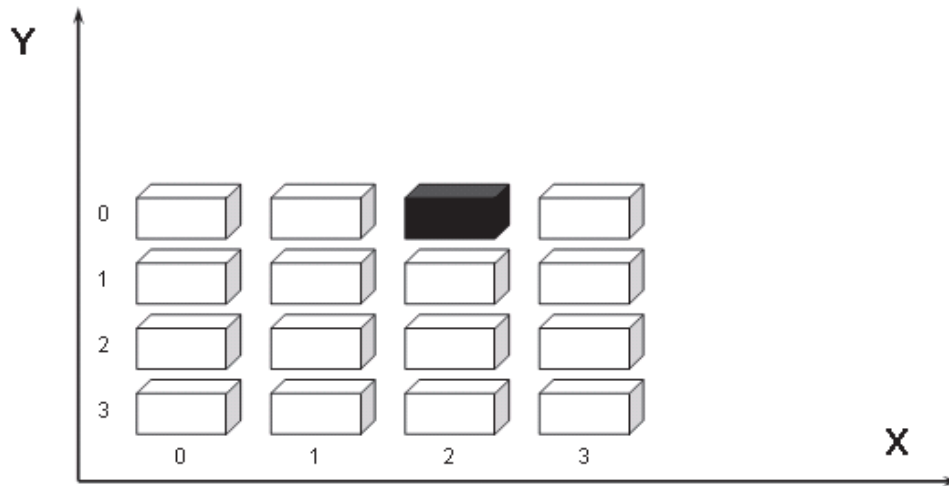
Arrays bieten eine bequeme Möglichkeit, Bezeichner zu vergeben. Durch nur eine Definition in der Liste der Lokalen oder der Globalen Variablen kann auf viele Elemente zugegriffen werden.

Zum besseren Verständnis werden die drei Array-Typen im folgenden grafisch dargestellt.

Eindimensionales Array



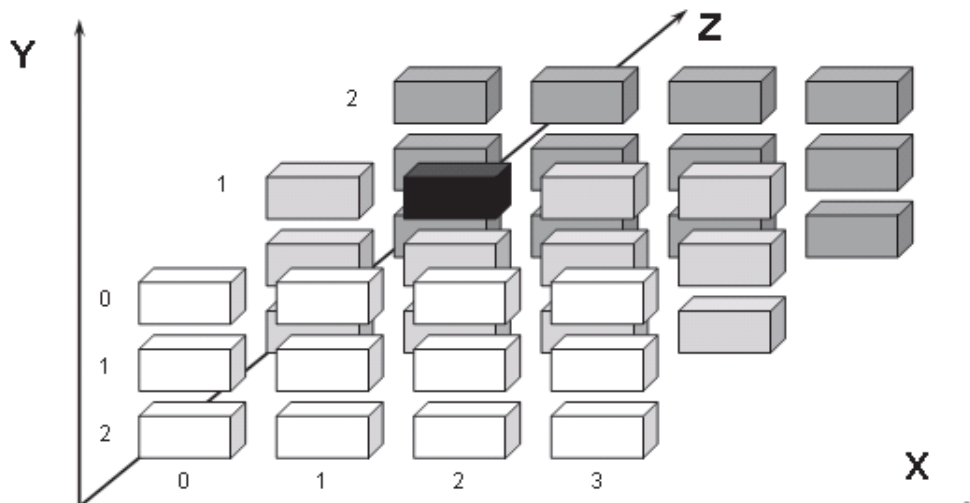
Zweidimensionales Array



Bezeichner **Motor_Volt** Typ **ARRAY [0..3, 0..3] OF INT**

= Motor_Volt [2, 0]

Dreidimensionales Array



Identifier **Motor_Strom** Type **ARRAY [0..3, 0..2, 0..2] OF INT**
 Bezeichner **Motor_Strom** Typ **ARRAY [0..3, 0..2, 0..2] OF INT**
 = Motor_Volt [1, 0, 1]


Strukturierter Datentyp (SDT)

Der Strukturierte Datentyp (SDT) ist eine Sammlung von Variablen, denen unterschiedliche Datentypen zugeordnet werden können. Der Anwender kann selbst Strukturierte Datentypen zusammenstellen. Dies ist hilfreich bei Programmen, die gemeinsame Elemente enthalten, wie zum Beispiel die Steuerung von sechs identischen Silos. Für dieses Programm kann eine SDT mit dem Namen „Silo“ definiert werden, die verschiedene Elemente, wie z. B. INT, BOOL usw., enthält.

Bei der Deklaration der Globalen Variablenliste können Bezeichner vom Datentyp „Silo“ angegeben werden. Das bedeutet, dass eine vorher definierte Sammlung mit dem Namen „Silo“ und den zur Steuerung jedes Silos benötigten Elementen verwendet wird. Durch die Wiederverwendbarkeit der SDT wird die Zeit für die Programmierung reduziert.

Beispiel für die Anwendung einer SDT

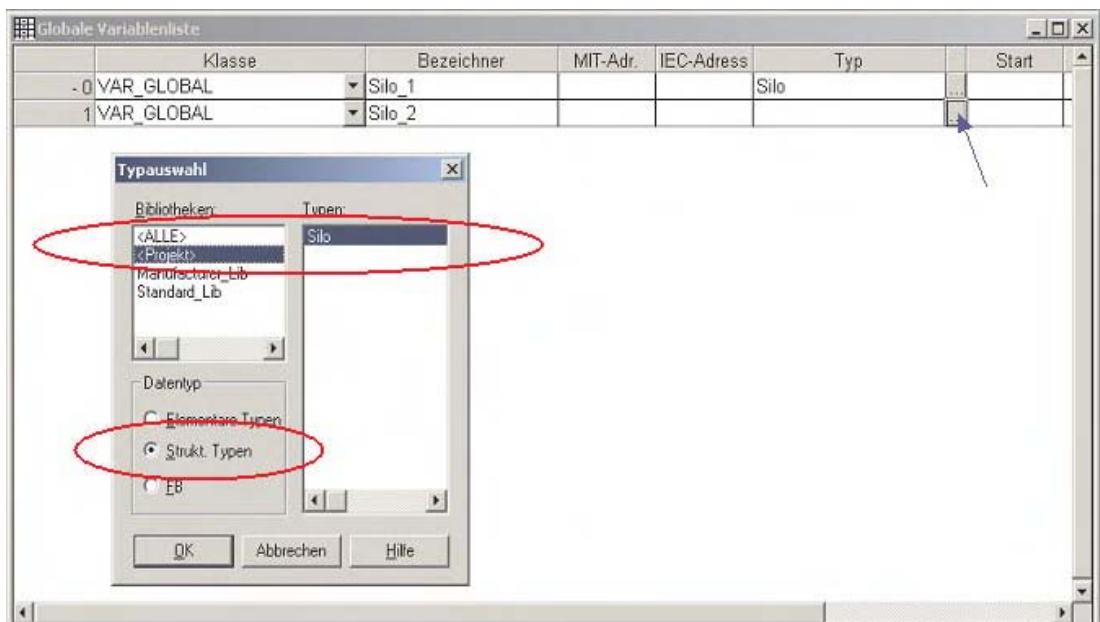
Die Abbildung unten zeigt die Deklaration eines Datentyps mit dem Namen „Silo“. Die Sammlung für „Silo“ enthält zwei Variablen vom Typ INT und eine Variable vom Typ BOOL.



	Bezeichner	Typ	Start	Kommentar
0	temperature	INT	0	
1	volume	INT	0	
2	Emergency_OFF	BOOL	FALSE	

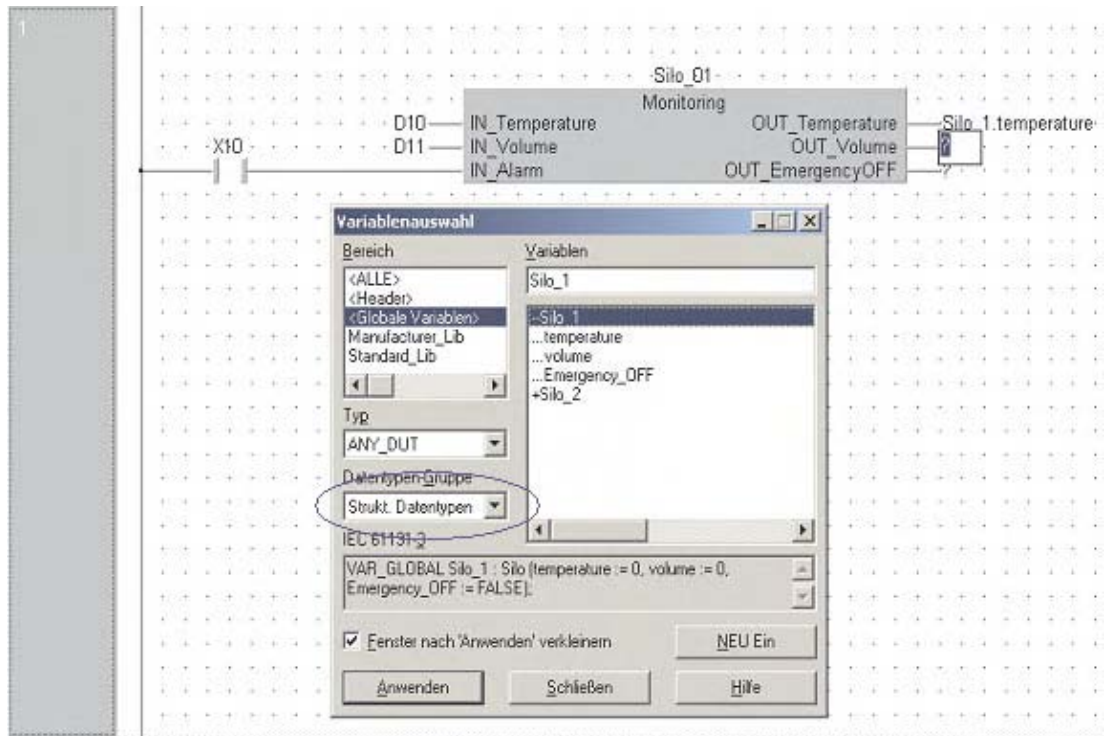
Verwendung einer SDT als Globale Variable

Klicken Sie im Projekt-Navigator doppelt auf **Glob. Variablen** und geben Sie in der Liste der Globalen Variablen die folgenden Zeilen ein. Die SDT „Silo“ wird zweimal verwendet.

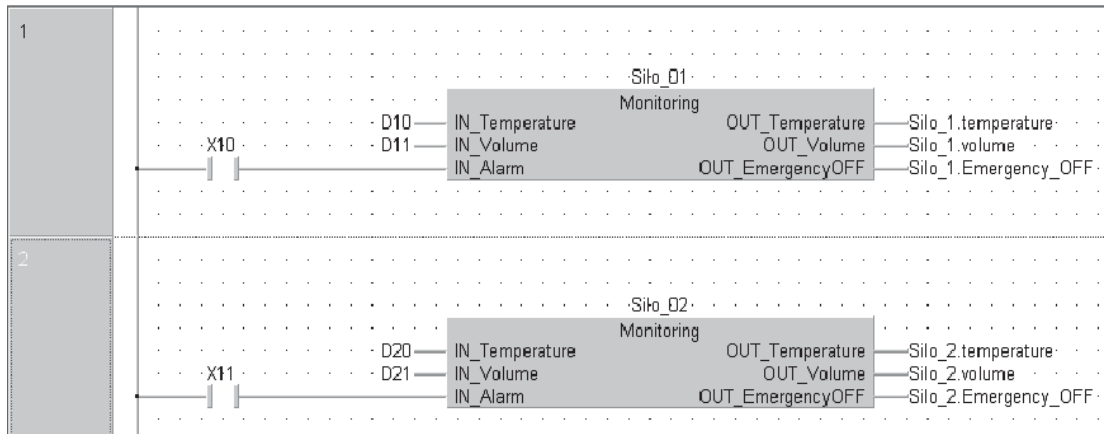


	Klasse	Bezeichner	MIT-Adr.	IEC-Adress	Typ	Start
0	VAR_GLOBAL	Silo_1			Silo	
1	VAR_GLOBAL	Silo_2				

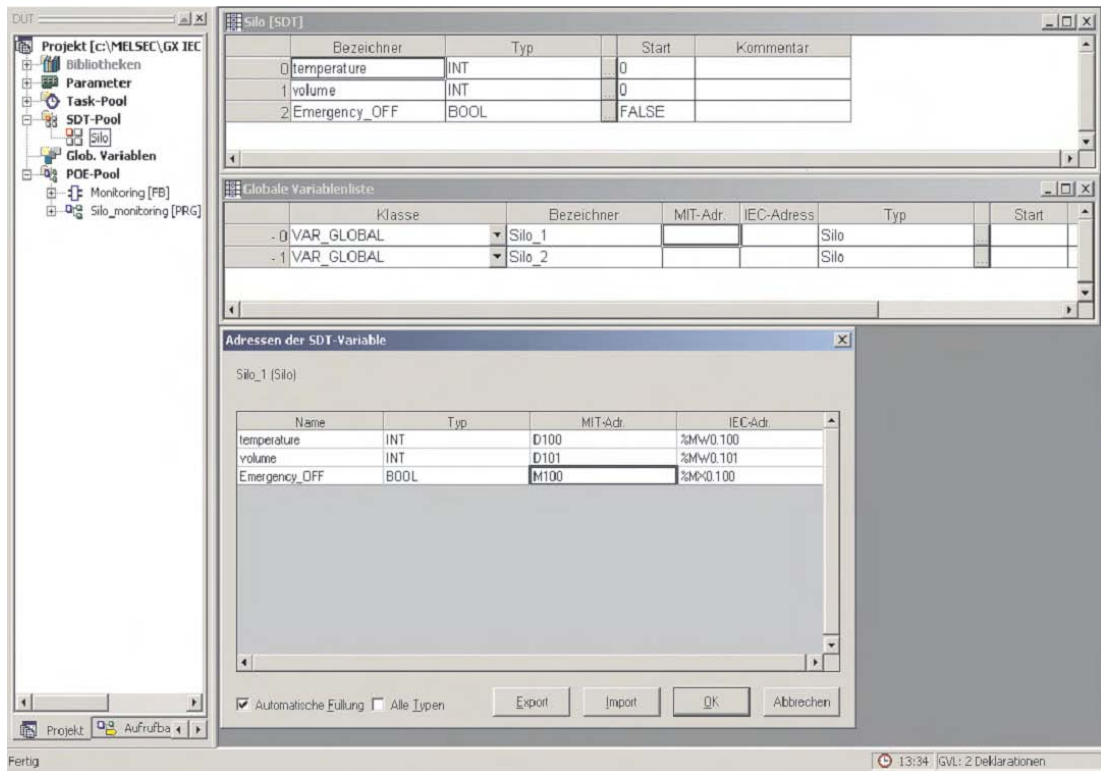
Die Variablen werden in der Globalen Variablen Liste gespeichert. Die Struktur beider Variablen, Silo_1 und Silo_2, ist identisch. Zur Angabe von Variablen einer bestimmten SDT im Programm wird dem Variablennamen einfach die Bezeichnung der Globalen Variablen vorangestellt.



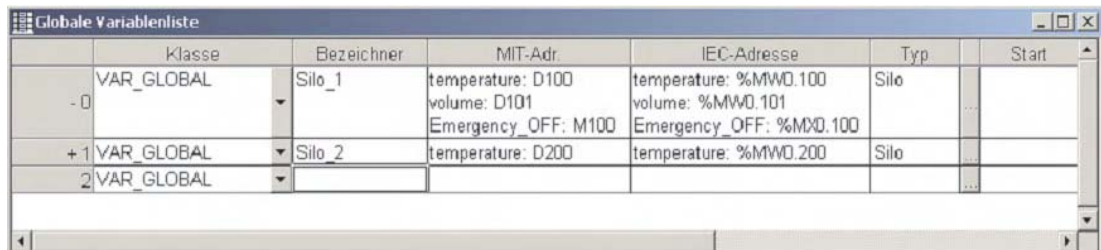
Für dieses Beispiel wurde ein Funktionsbaustein zur Erfassung und Speicherung der Silodaten programmiert. Dieser Funktionsbaustein wird für jedes Silo einmal aufgerufen. Um die Variablen eindeutig zuzuordnen, werden die Elemente der SDT mit „Silo_01“ und „Silo_02“ gekennzeichnet.



Die Globale Variablenliste wurde erweitert, um Adressen für alle Elemente von SDT-Variablen definieren zu können. Nicht definierte Adressen werden vom System eingesetzt.



Um alle Definitionen angezeigt zu bekommen (wenn mehr als eine Definition gespeichert ist), kann die Anzeige der SDT-Einträge in der Globalen Variablenliste durch einen Doppelklick auf auf die Tabellenzelle mit der Teilenummer erweitert werden.



Ein weiteres Beispiel für die Anwendung einer SDT enthält Kapitel 11.

3.4.3 MELSEC Timer und Counter

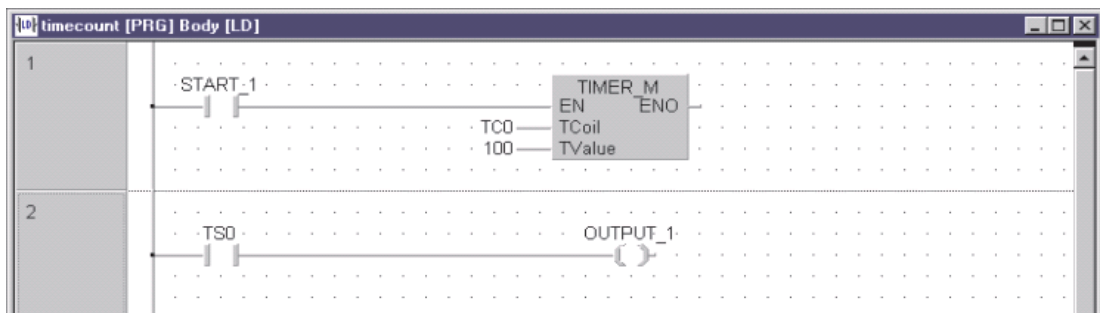
Bei der Programmierung von normalen Timern und Countern muss eine IEC-Konvention beachtet werden:

Programmierung der „Spule“ des Timers/Counters: **TCn / CCn**

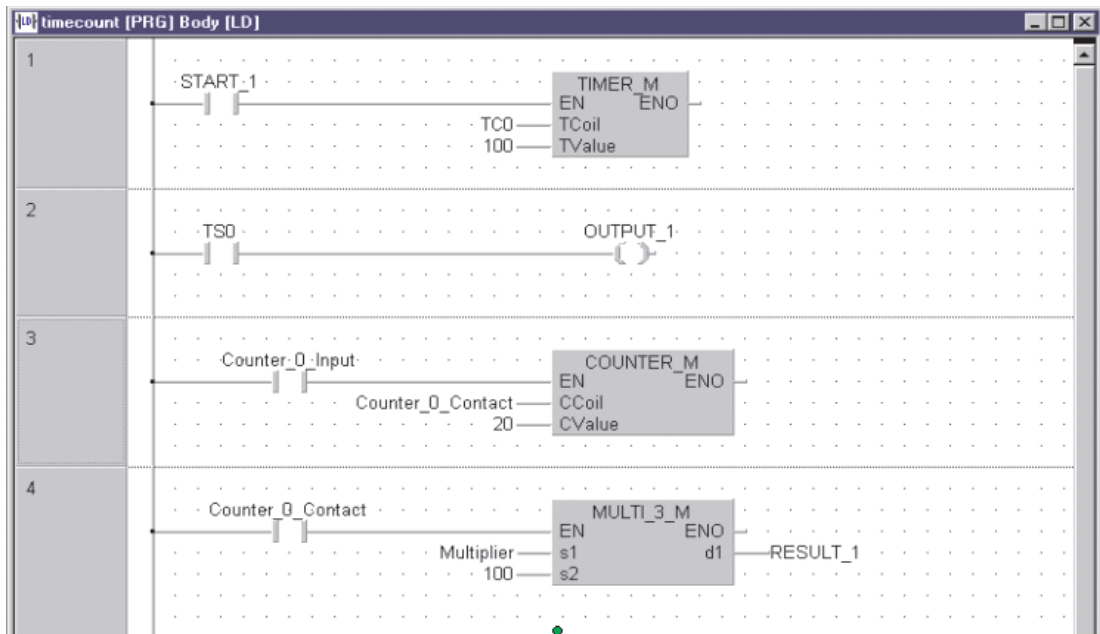
Programmierung der Timers/Counters-Ausgangs („Kontakt“): **TSn / CSn**

Programmierung des Istwerts des Timers/Counters: **TNn / CNn**

Im folgenden Beispiel wird der Timer T0 mit TC0 und TS0 angesprochen. In diesem Fall werden Mitsubishi-Adressen verwendet, deshalb ist es wichtig, die Zuordnung der Timer und Counter zu den Systemvariablen zu prüfen (Abschnitt 3.2.2).



Im folgenden Beispiel wurden zur Programmierung des Counters Variablen verwendet, die vorher als Globale und Lokale Variablen deklariert wurden.



4 Anlegen eines Projekts

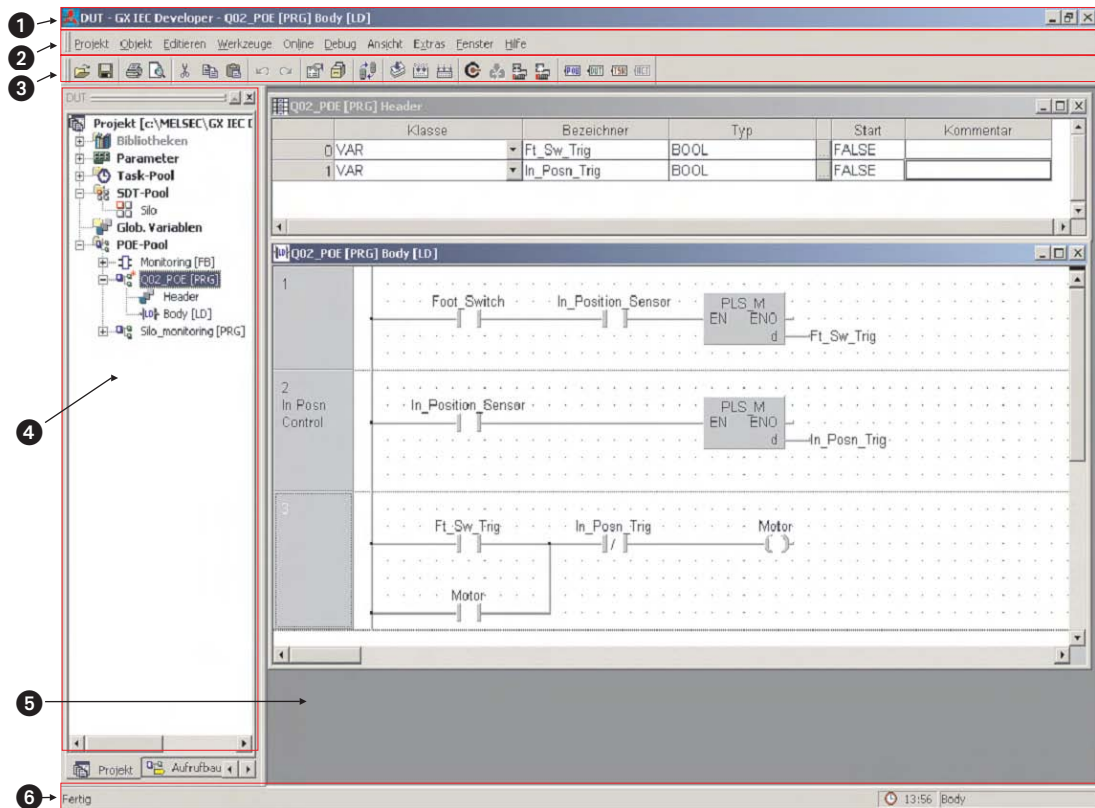
In diesem Kapitel werden alle Schritte beschrieben, die zur Erzeugung und Handhabung eines Projekts im GX IEC Developer erforderlich sind. Die Programmierung erfolgt in Kontaktplan.

Behandelte Tätigkeiten

- Projekt Navigator verwenden
- Globale Variablenliste mit Bezeichnern definieren und nutzen
- Variablen im Header des Programms deklarieren
- Programmieren mit dem IEC-Kontaktplaneditor
- Programmieren von Timern und Countern
- Programme kommentieren und dokumentieren
- Übertragen des Programms in die SPS und Prüfen des Programms

4.1 GX IEC Developer starten

Nach dem Start des GX IEC Developer unter Windows sehen Sie die Bedienoberfläche* der Software.



* In dieser Abbildung ist zur besseren Veranschaulichung bereits ein Projekt geöffnet. Nach dem Start des GX IEC Developer muss erst ein vorhandenes Projekt geöffnet oder ein neues Projekt angelegt werden.

1 Titelleiste

In der Titelleiste wird der Pfad und der Name des aktuellen Projekts angezeigt. Rechts in der Titelleiste finden Sie die üblichen Schaltfelder zum Minimieren, Verkleinern und Vergrößern der Darstellung und zum Beenden des GX IEC Developer.

2 Menüleiste

Die Menüleiste zeigt die einzelnen Menüs des GX Developers. Wenn Sie auf einen Eintrag klicken, wird ein Drop-Down-Menü angezeigt, in dem Sie die Menüpunkte wählen können.

Wenn ein Menüpunkt durch ein „▶“ gekennzeichnet ist, erscheinen nach einem Mausklick auf diesen Menüpunkt die Alternativen, die zur Auswahl stehen. Wird ein Menüpunkt von drei Punkten gefolgt, wird nach einem Klick auf diesen Menüpunkt ein Dialogfenster angezeigt. Viele Funktionen aus den Menüs können auch mit Hilfe der Werkzeugleisten aufgerufen werden.

Je nachdem, an welcher Stelle Sie sich im Programm befinden, ändert sich die Menüstruktur. Befehle, die grau dargestellt sind, lassen sich nicht ausführen.

3 Werkzeugleisten

Die am häufigsten gebrauchten Funktionen können durch einen Klick auf das entsprechende Schaltfeld in einer Werkzeugleiste direkt aufgerufen werden. Je nachdem, an wel-

cher Stelle Sie sich im Programm befinden, zeigen die Werkzeugleisten unterschiedliche Symbole.

④ **Projekt-Navigator**

Das Programm und dessen Dokumentation wird zusammen mit den Parametern für die SPS und Netzwerke in einem Projekt zusammen gefasst.

Der Projekt-Navigator zeigt die Verzeichnisse des momentan bearbeiteten Projekts. Hier können Sie die Dateien Programme, Dokumentation und Parameter durch einen Doppelklick öffnen. Das Navigator-Fenster erscheint erst, nachdem ein Projekt geöffnet oder ein neues Projekt angelegt wurde.

⑤ **Editor (Body)**

In diesem Bereich können Programm-Organisationseinheiten (POEs) eingegeben oder bearbeitet werden. Jede POE besteht aus einem „Header“ und einem „Body“.

– Header

Im Header werden die Variablen festgelegt, die in dieser POE verwendet werden.

– Body

Der Body enthält das eigentliche SPS-Programm in den verschiedenen Sprachen. Ein POE kann ein Programm, eine Funktion oder ein Funktionsbaustein sein.

⑥ **Statusleiste**

In der Statusleiste neben anderen Informationen der Typ der SPS und der Eingabemodus (Einfügen/Überschreiben) angezeigt. Die Statusleiste kann im Menü **Ansicht** konfiguriert werden.

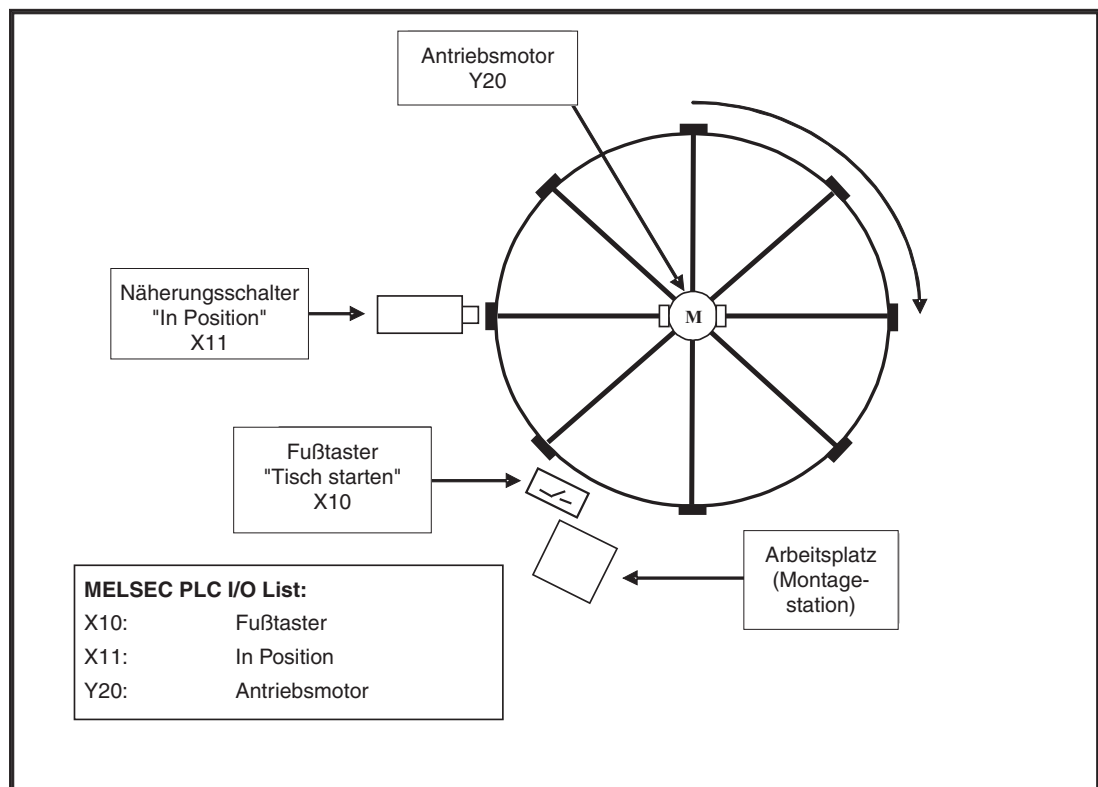
4.2 Programmierung

4.2.1 Anwendungsbeispiel: Drehtisch

Mit diesem einfachen Beispielprogramm für einen Drehtisch, der acht definierte Positionen anfahren kann, soll der Gebrauch der Werkzeuge des GX IEC Developers demonstriert werden.

Funktionsbeschreibung

- ① Nach der Betätigung eines Taster mit dem Fuß beginnt der Tisch mit der Drehung.
- ② Wenn sich der Tisch dreht, wird der Schalter „In Position“ ausgeschaltet.
- ③ Wird die nächste Position erreicht, wird der Schalter „In Position“ wieder eingeschaltet.
- ④ In dieser Position kann zum Beispiel ein Produkt, das sich auf dem Tisch befindet, zusammgebaut werden.
- ⑤ Nach dem Zusammenbau wird der Tisch auf die nächste Position gefahren. Der Zyklus beginnt wieder bei ①.



Vor der Programmierung des Drehtisches müssen einige Besonderheiten berücksichtigt werden. So kann der Antriebsmotor nicht einfach durch den Fußtaster ein- und durch den Positionsschalter ausgeschaltet werden:

- Der Bediener kann vergessen, den Fuß von dem Taster zu nehmen. Dadurch könnte der Tisch die nächste Position überfahren.
- Der Schalter „In-Position“ (Eingang X11) bleibt betätigt, wenn der Tisch die Position angefahren hat. Dadurch wird ein erneutes Einschalten des Motors verhindert.

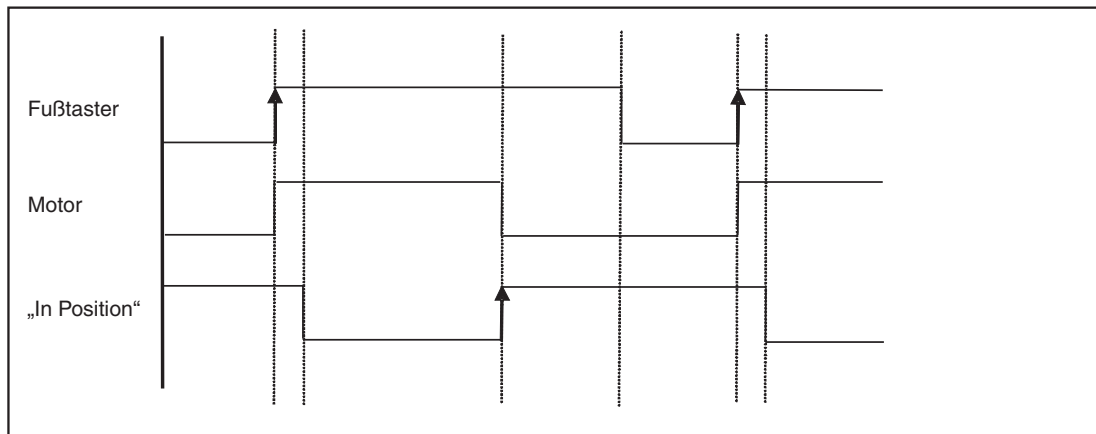
Im Programm müssen daher Verriegelungen zum Schutz vor Fehlbedienungen vorgesehen werden. Eine andere Möglichkeit ist die Anwendung von IEC- oder MELSEC- Anweisungen, die nur auf die Flanken der Eingangssignale reagieren.

Die am häufigsten verbreitete Anweisung aus dem MELSEC-Befehlsvorrat ist die PLS-Anweisung, die auf eine steigende Signalfanke reagiert. Sie wird auch in diesem Beispiel an Stelle der ebenfalls geeigneten IEC-Anweisung R_TRIG eingesetzt.

In der folgenden Abbildung ist der Ablauf der einzelnen Signale zur Steuerung des Tisches dargestellt. Bitte beachten Sie, dass durch die steigende Flanke des Fußtasters der Motor eingeschaltet wird, unabhängig davon, ob der Schalter „In Position“ betätigt ist. Dieser Schalter wird kurz nach dem Start des Motors ausgeschaltet, wenn der Tisch seine Position verlässt.

Der Motor wird ausgeschaltet, wenn der Schalter „In Position“ erneut betätigt wird. Beim Ausschalten kann der Fußtaster weiter betätigt sein. Der Motor wird erst wieder eingeschaltet, wenn der Fuß vom Taster genommen und der Taster danach wieder betätigt wird. Dies wird durch die Flankenauswertung des Eingangssignals erreicht.

Zeitlicher Ablauf der Signale zur Drehtischsteuerung



4.2.2 Anlegen eines neuen Projekts

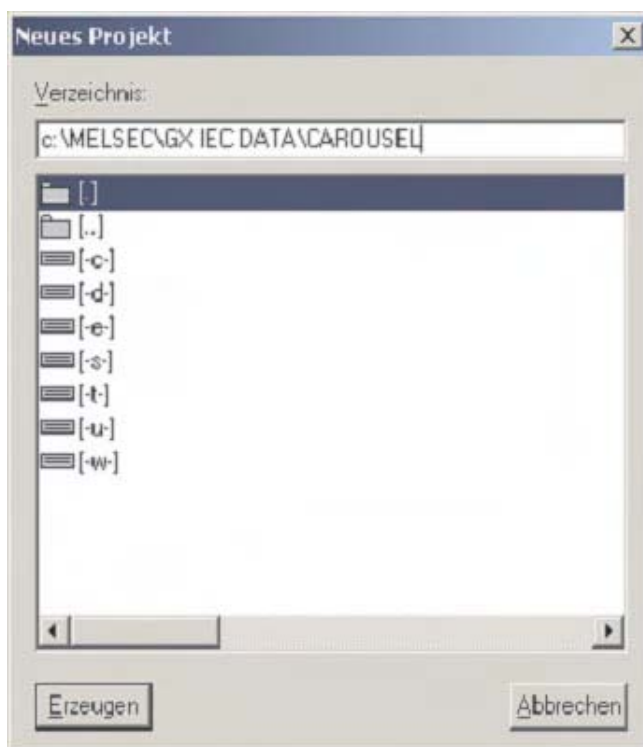
① Klicken Sie im Menü **Projekt** auf **Neu**.



② Wählen Sie die SPS und die verwendete CPU.

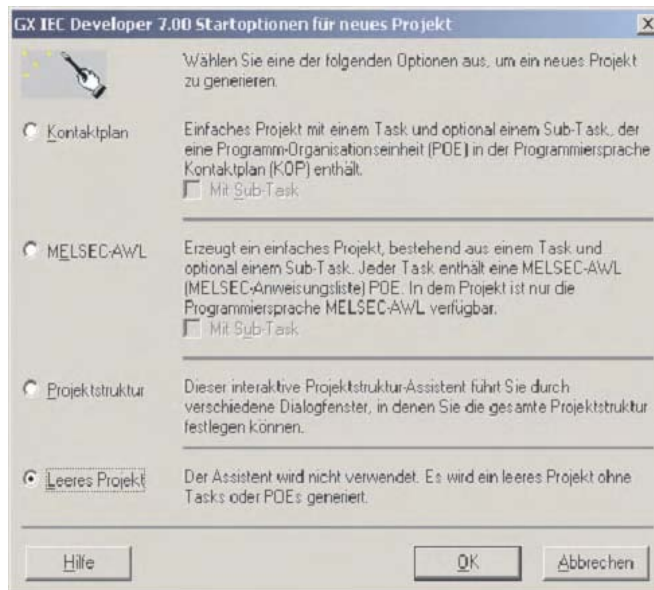


③ Wählen Sie im Dialogfenster **Neues Projekt** einen Namen für das Projekt. Für dieses Beispiel geben Sie bitte „C:\MELSEC\GX-IEC DATA\CAROUSEL“ an und klicken anschließend auf **Erzeugen**.



Der Assistent für ein neues Projekt

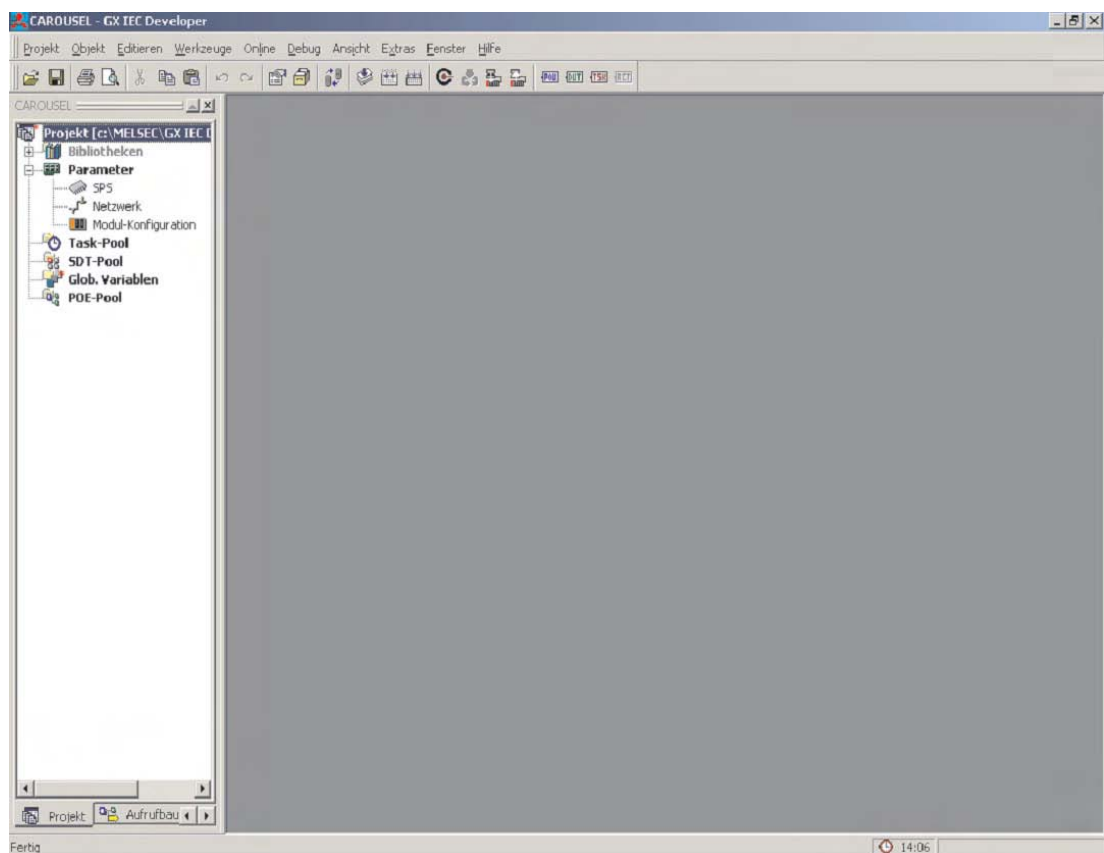
Nach der Angabe eines Projektnamens wird das folgende Dialogfenster angezeigt:



Dieser Assistent hilft Ihnen beim Anlegen eines Projekts, indem er die Grundstruktur für einfache Projekte erzeugt.


Wählen Sie **Leeres Projekt** und klicken Sie anschließend auf **OK**. Dadurch wird verhindert, dass der Assistent Projektelemente erzeugt. Selbstverständlich können Sie auch die Hilfe des Assistenten in Anspruch nehmen, aber zur Demonstration der Grundfunktionen des GX-IEC Developer und zur Schulung wird in diesem Beispiel das Programm manuell aufgebaut.

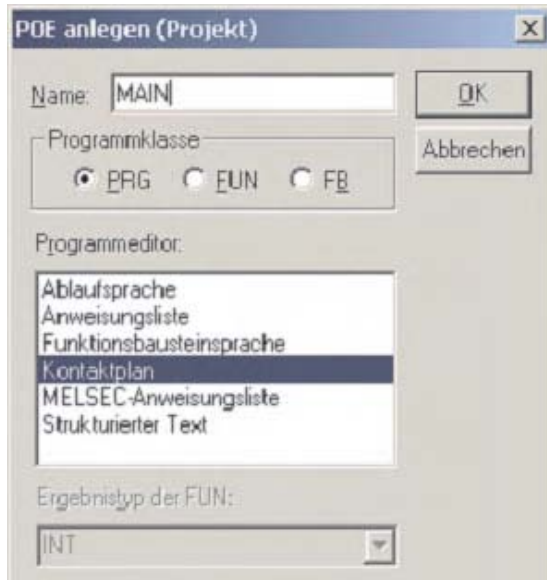
Das neue Projekt wird angezeigt:



Der Arbeitsbereich ist noch leer. Im Projekt Navigator an der linken Seite kann jedes Element des Projekts durch einen Doppelklick ausgewählt werden.

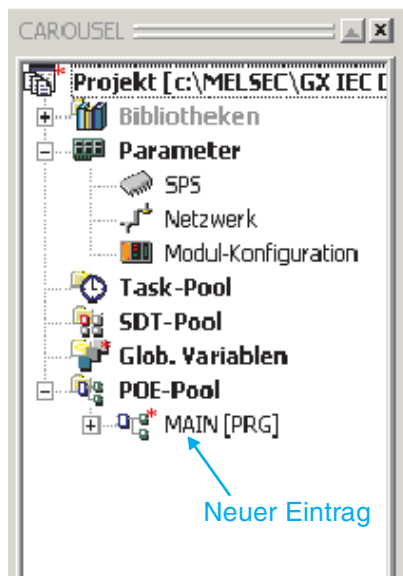
4.2.3 Anlegen einer neuen POE

- 1 Klicken Sie auf das Schaltfeld  („Neue POE“) oder klicken Sie auf **POE-Pool** im **Projekt Navigator** und betätigen Sie anschließend die rechte Maustaste. Klicken Sie im dann auf **Neue POE**. In beiden Fällen öffnet sich das folgende Dialogfenster:

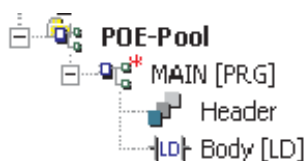


Der Name der POE ist „MAIN“. Als Programmklasse wählen Sie bitte **PRG** (Programm) und als Programmierer **Kontaktplan**.

- 2 Klicken Sie auf **OK**. Die neue POE wird in den POE-Pool im Navigatorfenster eingetragen.



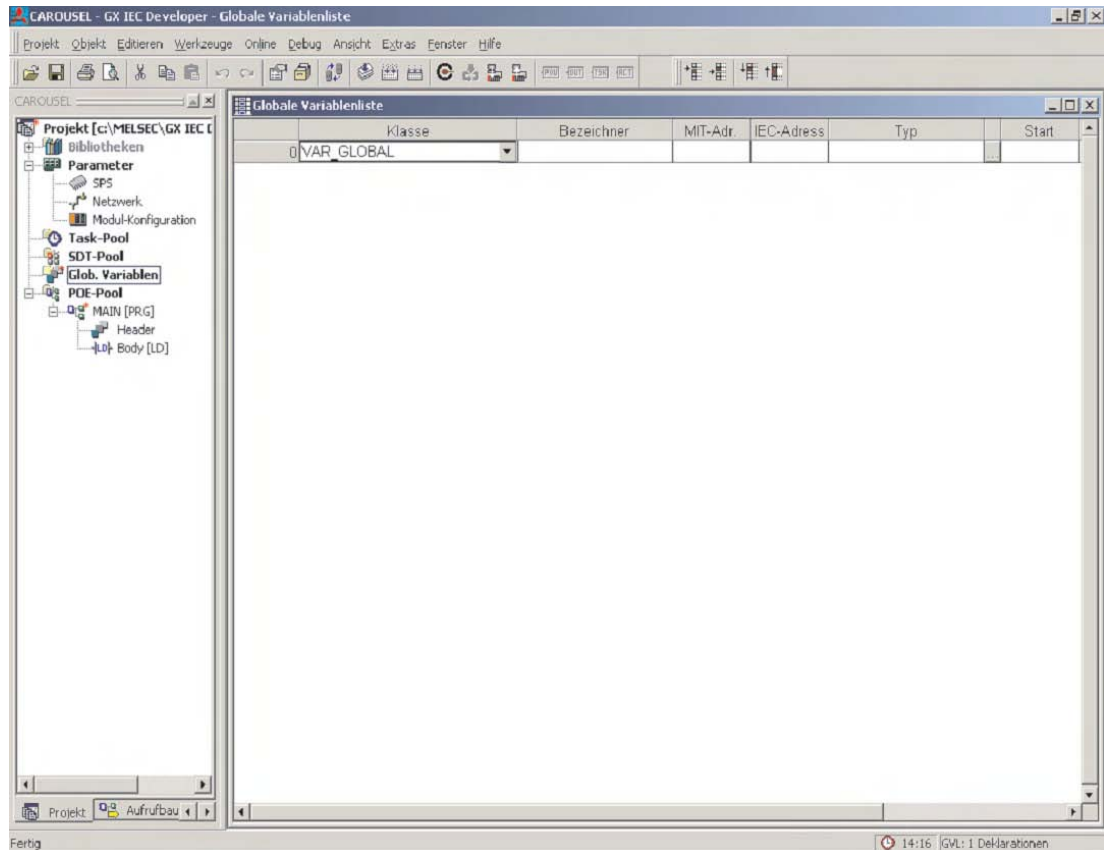
- 3 Klicken Sie im Navigatorfenster doppelt auf **MAIN** oder klicken Sie auf das Pluszeichen (+) vor **MAIN**, um das Unterverzeichnis zu öffnen und die Einträge für den **Header** und den **Body** anzuzeigen.



4.2.4 Deklaration der Globalen Variablen

Vor der Programmierung müssen alle verwendeten Ein- und Ausgänge der SPS und die gemeinsam im Projekt verwendeten Variablen angegeben werden.

Klicken Sie im Navigatorfenster doppelt auf **Glo. Variablen**, um die Liste der Globalen Variablen (GVL) zu öffnen.



Globale Variablen sind die Verbindung zu den physischen SPS-Operanden.

Wie bereits im Kapitel 3 beschrieben, müssen bei der Programmierung nach IEC 61131-3 symbolische Bezeichner (Namen) an Stelle von absoluten Operandenadressen verwendet werden. Daher werden die Operandenadressen vorher als Globale Variablen definiert. In der Liste der Globalen Variablen muss der Bezeichner, die Operandenadresse (in IEC- oder MELSEC-Schreibweise) und der Operandentyp (z. B. Bit- oder Wort-Operand) angegeben werden. Wenn die Liste einmal ausgefüllt wurde, kann sie von allen POEs genutzt werden, die für das Projekt angelegt werden.

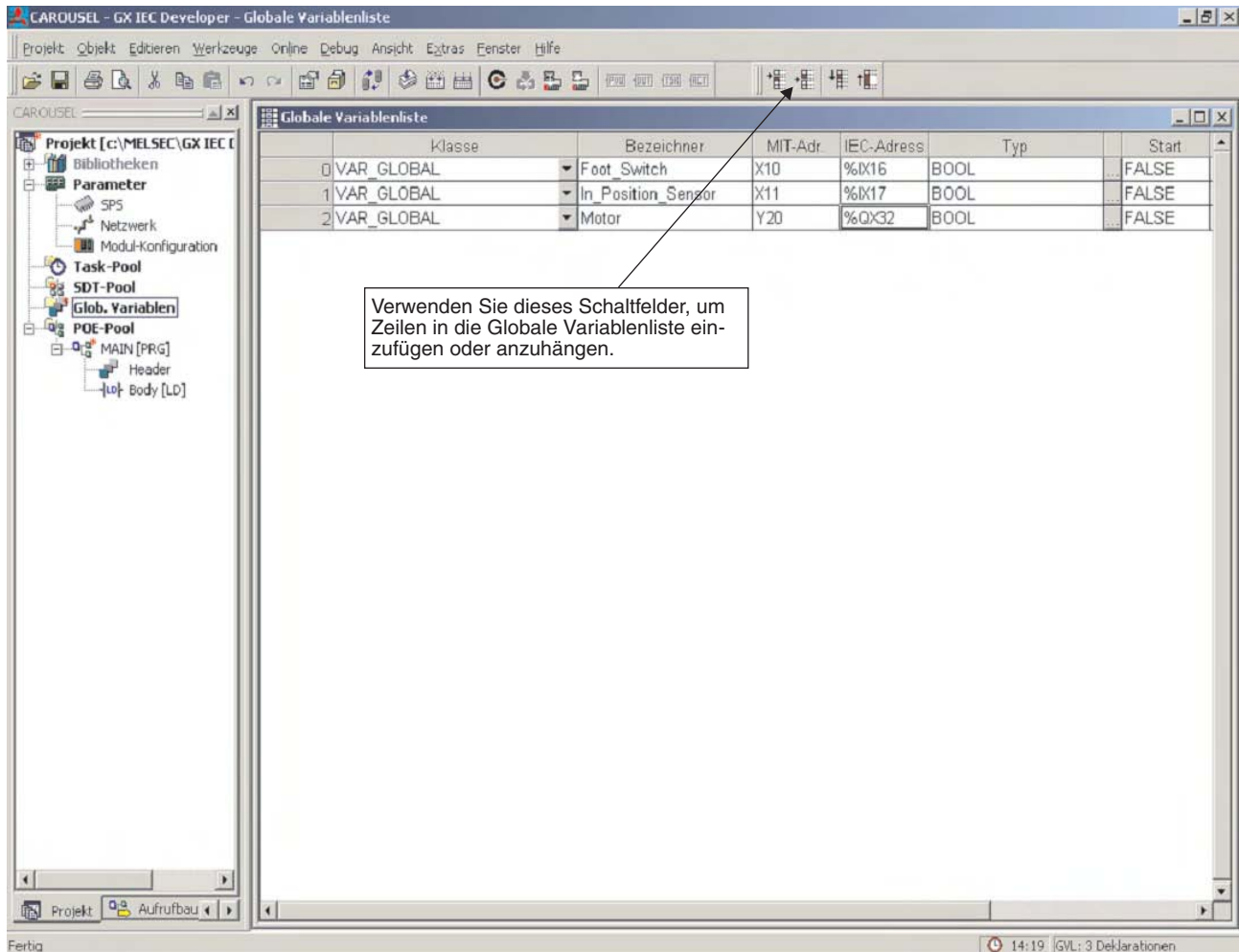
Angaben beim Deklarieren von Variablen

Für jede Variable können in der Variablenliste die folgenden Elemente angegeben werden:

- Klasse
Mit der Angabe der Klasse werden der Variablen bestimmte Eigenschaften zugeordnet, die die Verwendung der Variablen im Projekt kennzeichnen.
- Bezeichner
Jede Variablen erhält eine symbolische Adresse. Dieser individuelle Name (Bezeichner) kann frei gewählt werden, muss aber mit einem Buchstaben oder einem (einzelnen) Unterstrich beginnen (z. B. Motor_M1_EIN oder _Antreib_bereit). Leerzeichen oder mathematische Zeichen (z. B. +, -, *) dürfen im Bezeichner nicht verwendet werden.
- MIT-Adr.
Die absolute Adresse einer Variablen kann sowohl mit der IEC-Syntax (**IEC-Adresse**, siehe unten) als auch mit der MITSUBISHI-Syntax (**MIT-Adr**) eingegeben werden.
- IEC-Adresse
Bei der Angabe der absoluten Adresse einer Variablen kann die IEC-Syntax (**IEC-Adresse**) oder die MITSUBISHI-Syntax (**MIT-Adr**, siehe oben) verwendet werden.
- Typ
Angabe des Datentyps wie z. B. BOOL, INT, REAL, WORD (siehe Abschnitt 3.4)
- Start
Die Anfangswerte werden automatisch durch das System eingetragen und können vom Anwender nicht verändert werden.
- Kommentar
Für jede Variable können bis zu 64 Zeichen Kommentar eingegeben werden.

Die Globale Variablenliste muss nicht ausgefüllt werden, wenn im Programm keine symbolischen Namen, sondern nur Mitsubishi-Adressen verwendet werden. Allerdings entspricht das Programm dann nicht mehr der IEC 6113-3.

Füllen Sie die Globale Variablenliste so aus, wie in der folgenden Abbildung. Der Typ einer Variablen wird nach der Eingabe der Operandenadresse vom GX IEC Developer automatisch erkannt und eingetragen. Der Typ kann aber auch manuell eingegeben oder verändert werden, nachdem auf das Feld rechts neben dem Typ-Eingabefeld geklickt wurde. Falls eine Adresse im Feld **MIT-Adr.** eingegeben wurde, wandelt der GX IEC Developer diese Adresse automatisch in die IEC-Syntax und trägt sie in das Feld IEC-Adresse ein.



Ungenutzte Variablen suchen

Mit der Funktion **Ungenutzte Variablen suchen** im Menü **Extras** können alle Globalen und Lokalen Variablen eines Projekts gesucht werden, die deklariert, aber nicht verwendet werden. Bis auf Variablen in anwenderdefinierten Bibliotheken können ungenutzte Variablen anschließend gelöscht werden.




HINWEIS

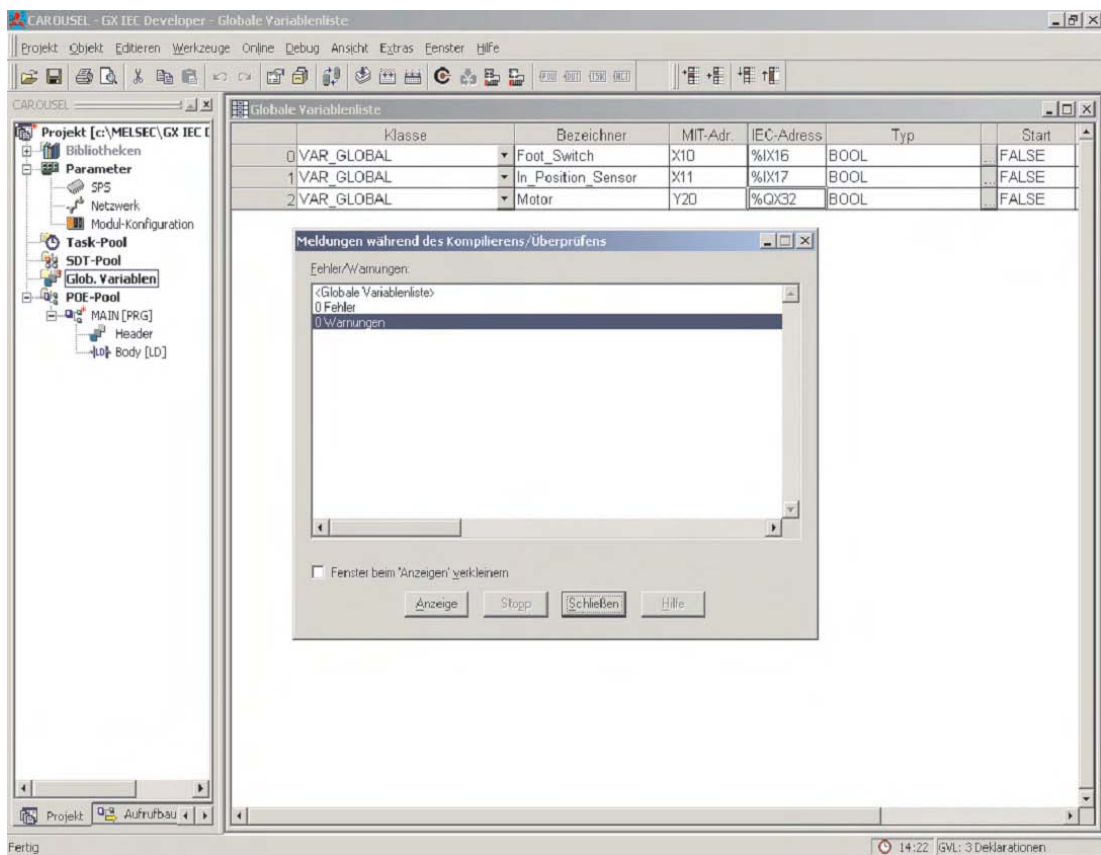
Ungenutzte Variablen können nur gesucht werden, wenn das Projekt in den Maschinencode gewandelt und danach nicht verändert wurde. Falls ungenutzte Variablen nicht gesucht werden können, erscheint eine Warnmeldung.

HINWEISE

Die Globale Variablenliste (GVL) bietet eine Eingabehilfe, indem der zuletzt eingegebene Bezeichner und dessen Operandenadresse beim Einfügen einer neuen Zeile fortgesetzt werden. Wenn z. B. in der GVL eine Anzahl Ventile deklariert werden sollen (Ventil_1 bis Ventil_n), wird in einer Zeile als Bezeichner „Ventil_1“ eingetragen. Wenn unter diese Zeile weitere Zeilen mit Hilfe des Icons in der Werkzeugleiste oder der Tastenkombination „Umschalttaste + Übernahme“ eingefügt werden, werden Bezeichner und Operandenadresse um jeweils 1 erhöht und in die neue Zeile eingetragen (z. B. Ventil_2, Ventil_3 etc.). Falls diese Eingabehilfe nicht gewünscht wird, kann sie im Menü **Extras (Optionen → Allgemeine Optionen → Editieren)** ausgeschaltet werden.

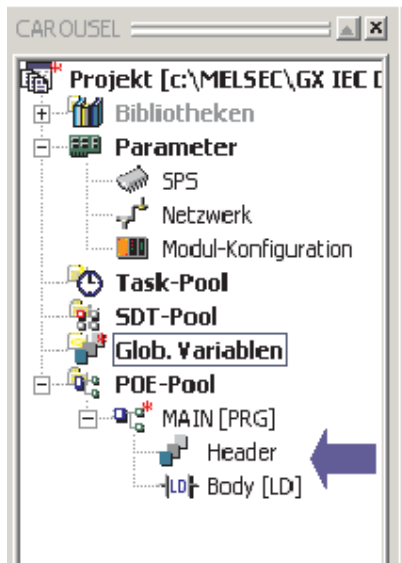
Bei Steuerungen der FX2N-, der FX3U-, und der AnA(S)-Serie sowie des MELSEC System Q können Gleitkommazahlen (IEC-Datentyp REAL) als Variable angegeben werden.

Nach der Eingabe der Daten in die GVL klicken Sie auf das Schaltfeld  in der Werkzeugleiste, um die Deklarationen zu prüfen:

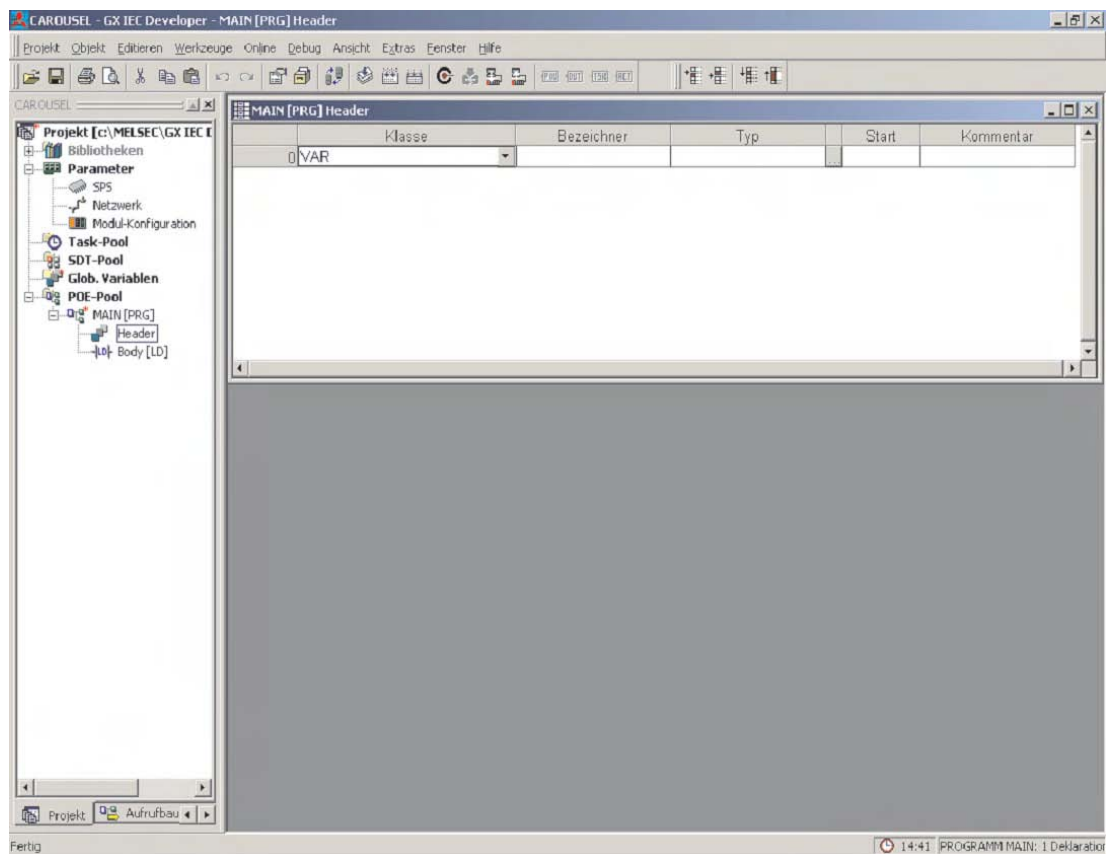


Header der POE definieren

Klicken Sie im Navigatorfenster doppelt auf den Eintrag **Header** der POE **MAIN**.



Das folgende Dialogfenster wird angezeigt. Es sind keine lokalen Variablen definiert.

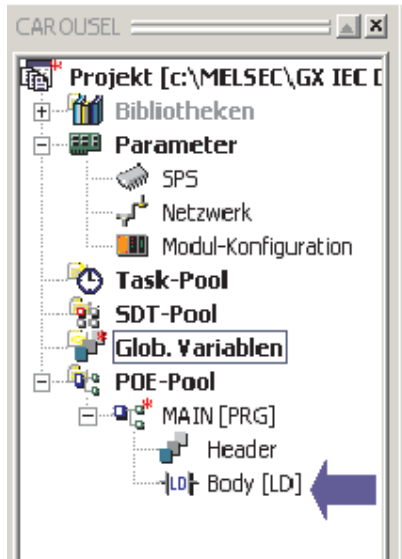


Schließen Sie dieses Dialogfenster.

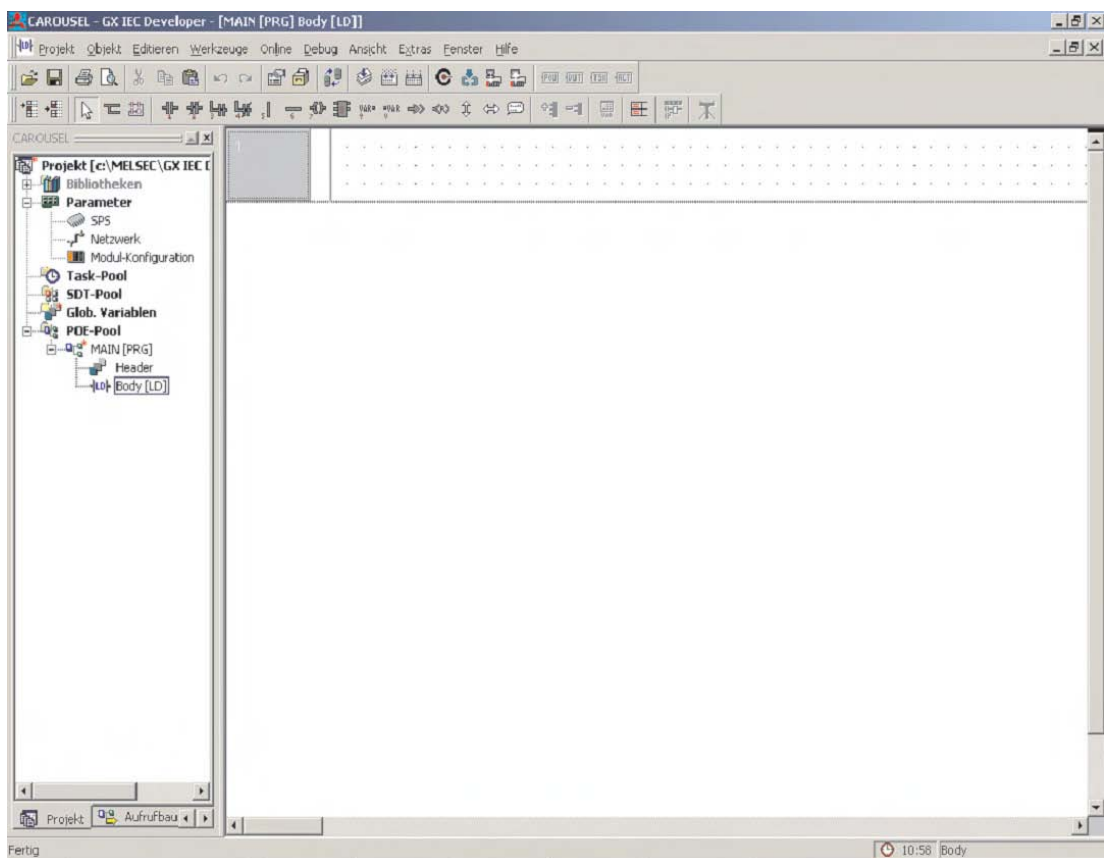
4.2.5 Body der POE programmieren

Der Body einer POE enthält das eigentliche SPS-Programm.

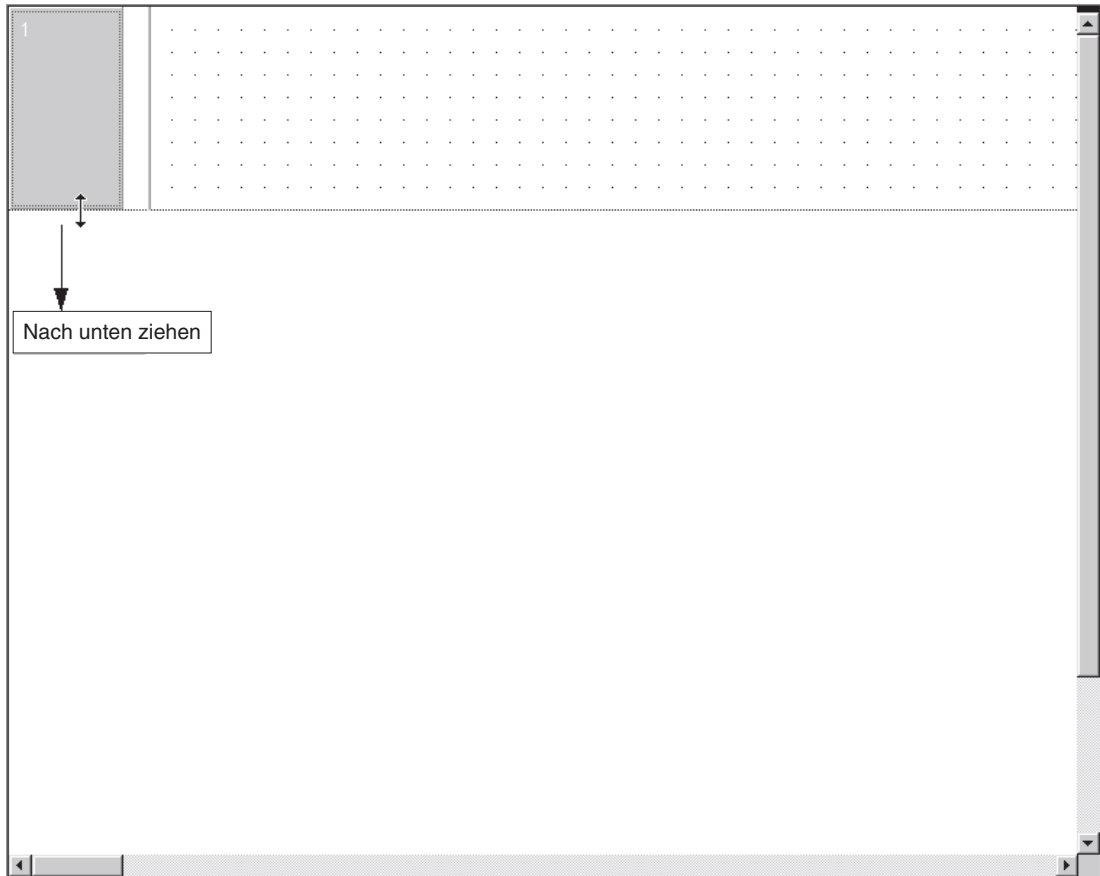
- ① Zum Öffnen des Kontaktplan-Editors klicken Sie im Navigationsfenster doppelt auf den Eintrag **Body** der POE **MAIN**. Der Klammereintrag hinter dem Begriff **Body** zeigt an, in welche Programmiersprache für diese POE gewählt wurde (KOP = Kontaktplan).



Das folgende Dialogfenster wird angezeigt:

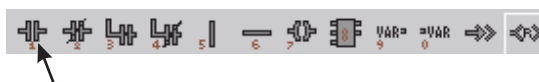


- ② Positionieren Sie den Cursor auf den unteren Rand des Netzwerks, bis aus dem Cursor ein Doppelpfeil wird. Bewegen Sie dann bei betätigter linker Maustaste den Cursor nach unten, um das Netzwerk zu vergrößern:



Auswahl von Kontaktplansymbolen aus der Werkzeugleiste


- ③ Klicken Sie im Auswahlmodus des Editors in der Werkzeugleiste auf das Symbol für einen Schließer-Kontakt:

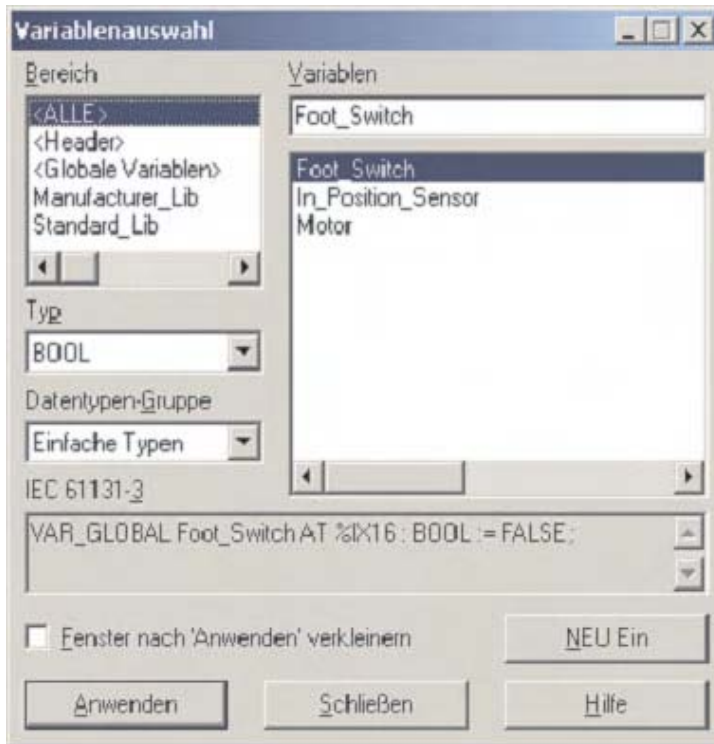


- ④ Bewegen Sie den Cursor in das Netzwerk und betätigen Sie an der Stelle, an der das Symbol angeordnet werden soll, die linke Maustaste:



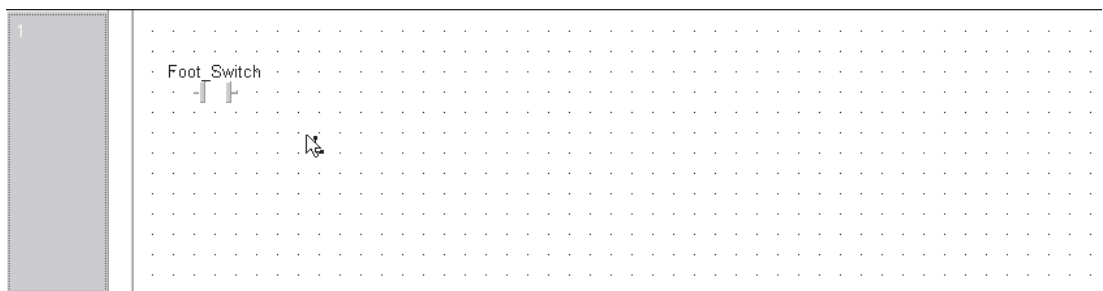
Auswahl von Variablen für eine POE

- ① Betätigen Sie die Taste F2 der PC-Tastatur oder klicken Sie auf das Symbol  in der Werkzeugleiste. Dadurch wird das Dialogfenster **Variablenauswahl** geöffnet:



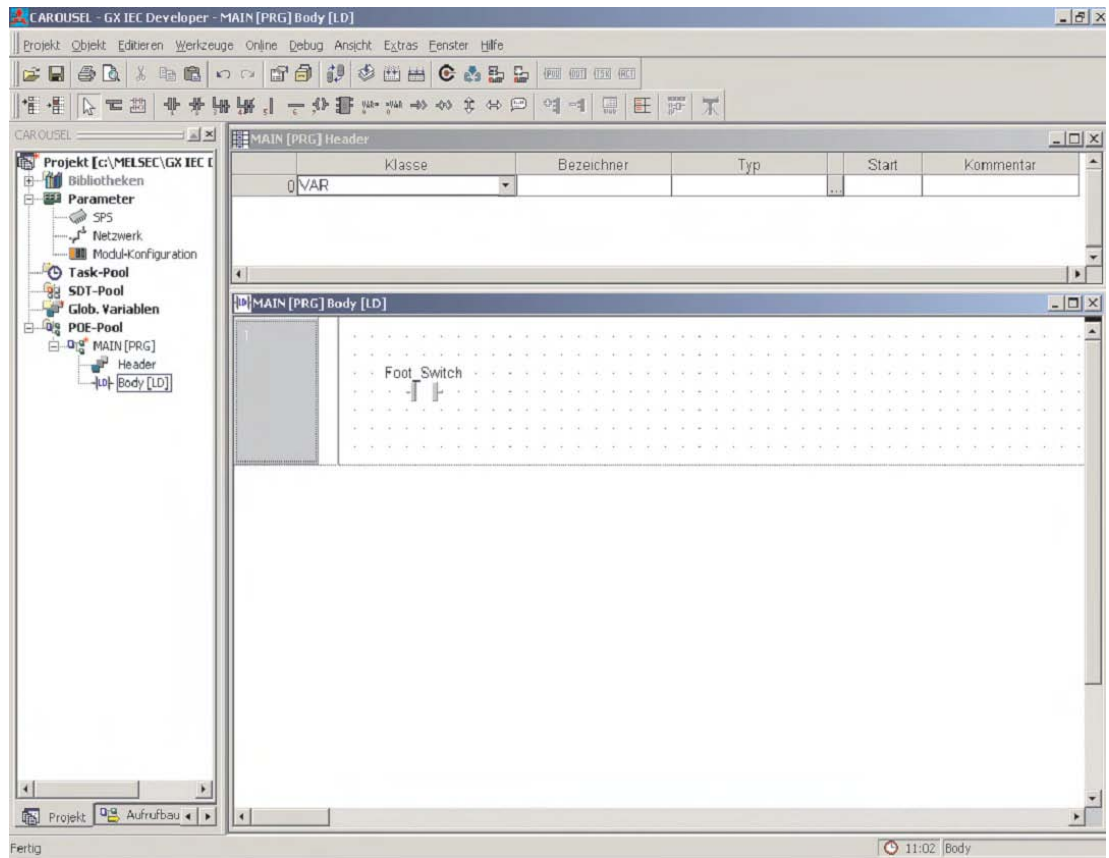
Bitte achten Sie darauf, dass im Auswahlfeld **Bereich** die korrekte Variablenliste angewählt ist.

- ② Klicken Sie auf die Variable „Fußschalter“. Nach der Auswahl wird die Variable blau hinterlegt. Klicken Sie anschließend auf **Anwenden**. Schließen Sie dann das Dialogfenster **Variablenauswahl**.



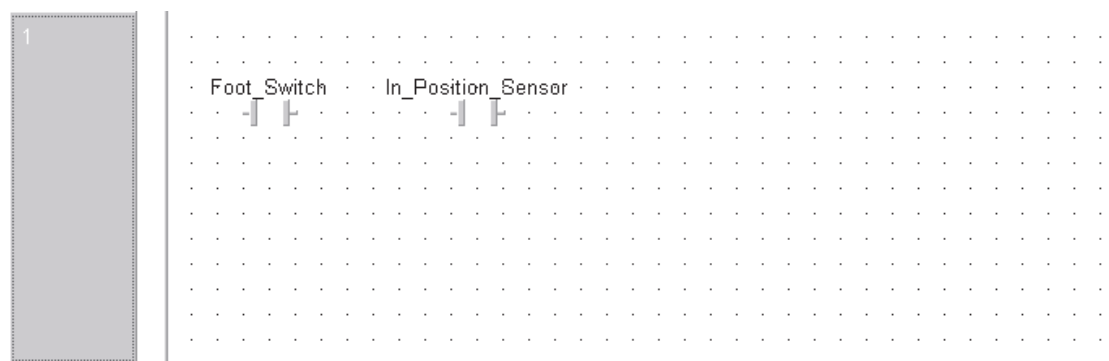
Alternative Methode zur Variablenauswahl: Bearbeiten bei geteilten Fenstern

Der Body und der Header einer POE können gleichzeitig angezeigt werden, wenn beide Einträge geöffnet sind und im Menü **Fenster** die Option **Untereinander** gewählt wird.



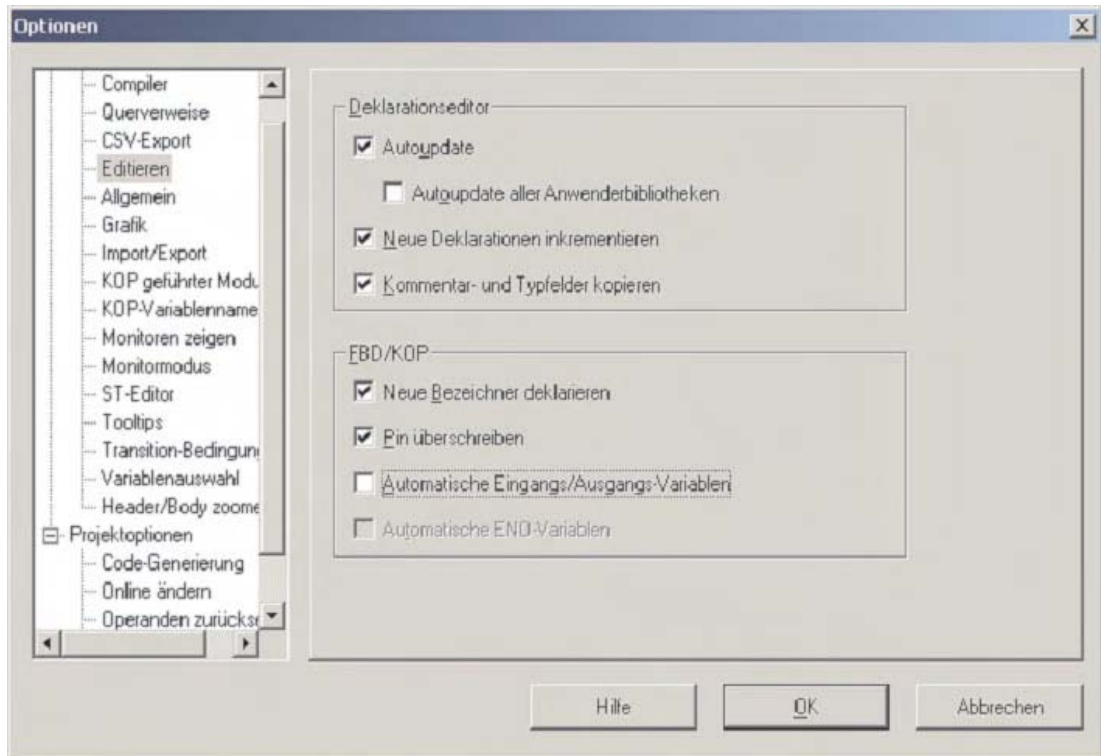
Fortsetzung der Programmierung für das Projekt „Drehtisch“

Programmieren Sie den Schließerkontakt für den Schalter „In Position“ ebenso wie den Fußschalter:




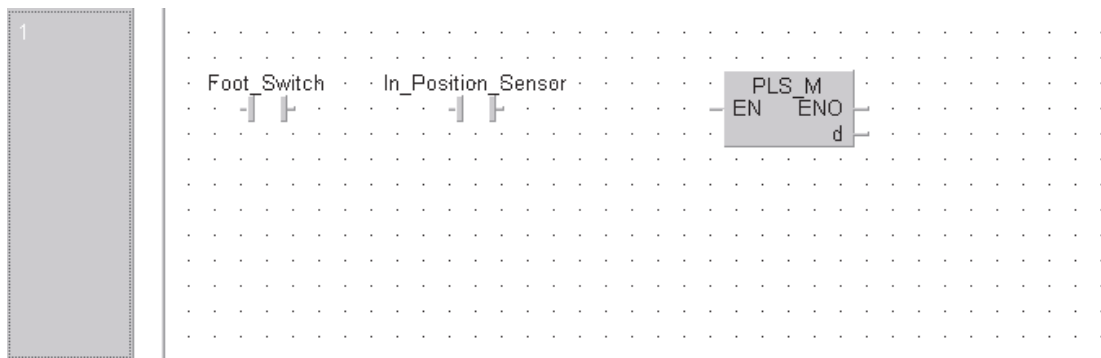
Aufruf einer Funktion im Kontaktplanprogramm

Bevor die Programmierung fortgesetzt wird, sollte die Option **Automatische Eingangs-/Ausgangs-Variablen** deaktiviert werden. Die wird im Menü **Extras (Optionen → Allgemeine Optionen → Editieren)** vorgenommen.




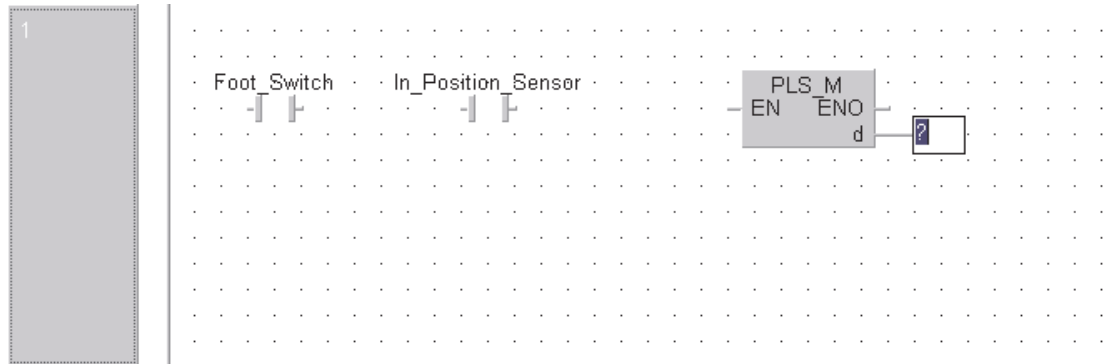
Das Programm wird anschließend um die MELSEC-Funktion „PLS_M“ ergänzt.

- ① Klicken Sie in der Werkzeugleiste auf das Schaltfeld  , um eine Funktion oder einen Funktionsbaustein auszuwählen. Im Auswahlfenster wählen Sie als Operatorengruppe **Funktionen** und geben in dem Eingabefeld **Operatoren** „PLS_M“ ein. Klicken Sie wieder auf **Anwenden** und platzieren Sie die Funktion in das Netzwerk:



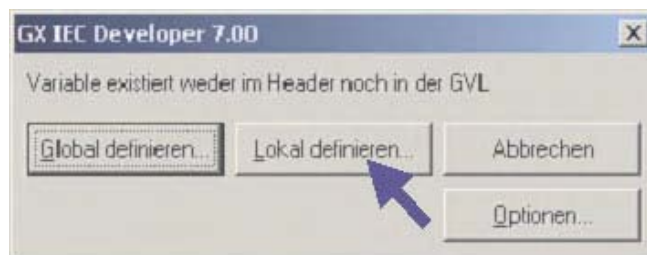
Variablen einer Anweisung zuweisen

- ② Klicken Sie in der Werkzeugleiste auf das Schaltfeld Ausgangsvariable . Klicken Sie anschließend auf den Ausgang der PLS_M-Funktion. Dadurch kann an dieser Stelle eine Ausgangsvariable eingegeben werden.

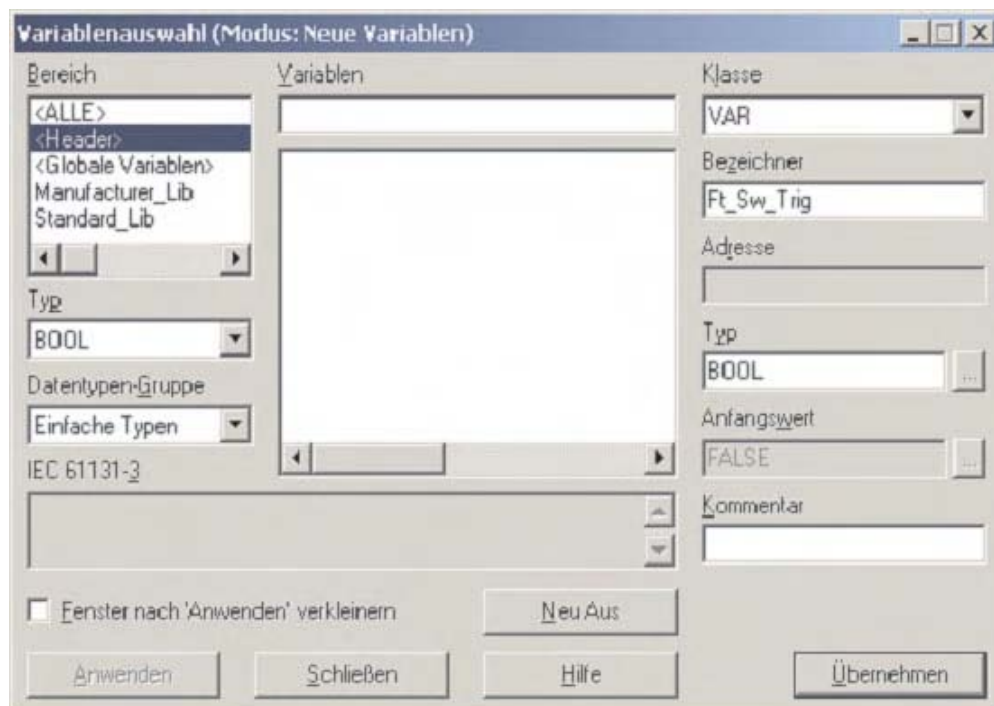


- ③ Tragen Sie in das leere Feld den Namen der Variablen ein: Ft_Sw_Trig.

Da diese Variable noch nicht im deklariert wurde, wird das folgende Dialogfenster angezeigt:



- ④ Klicken Sie auf **Lokal definieren**. Danach wird das unten abgebildete Fenster zur Eingabe einer neuen Variablen geöffnet.

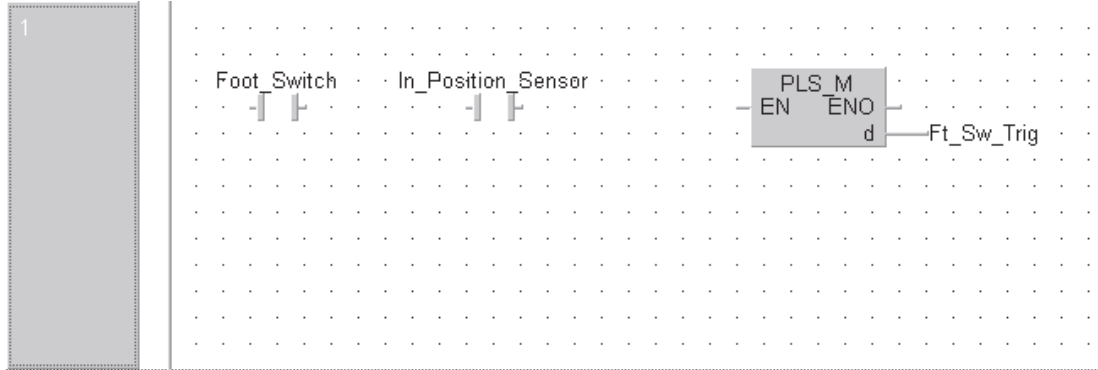


- ⑤ Klicken Sie auf **Übernehmen**, um die neue Variable in die Liste der Lokalen Variablen (Header der POE) einzutragen.

Hinweis

Prüfen Sie, ob die Variable übernommen wurde, indem Sie den Header der POE öffnen.

Nun sollte das Netzwerk so aussehen:

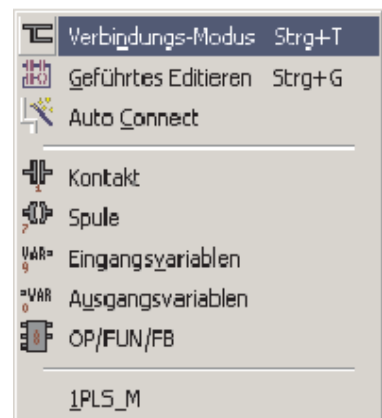


Nun kann das Netzwerk fertiggestellt werden. Dazu müssen die einzelnen Elemente verbunden werden.

- ⑥ Positionieren Sie den Cursor auf eine beliebige Stelle des Netzwerks und betätigen Sie die rechte Maustaste. Deaktivieren Sie die Funktion **Auto Connect**.

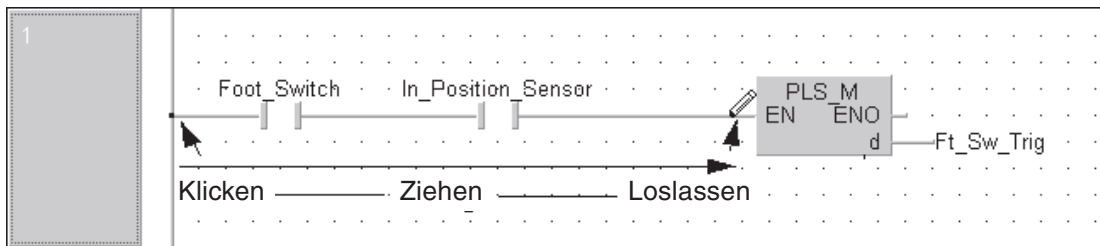


- ⑦ Aktivieren Sie im selben Fenster den **Verbindungs-Modus**.



Bitte beachten Sie, dass der Cursor nun die Form eines Stiftes angenommen hat.

- ⑧ Klicken Sie links auf die Schiene und ziehen Sie den „Stift“ bei betätigter linker Maustaste bis zum EN-Eingang der Funktion PLS_M. Lassen Sie dort die linke Maustaste los:



Das Netzwerk ist nun fertig.

Cursor-Modus ändern

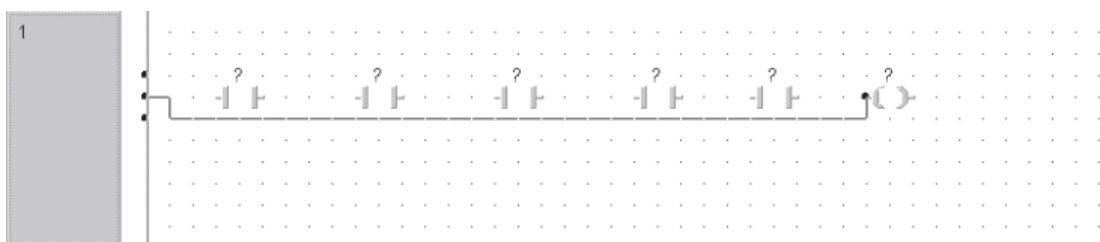
Bevor die Programmierung des Beispiels fortgesetzt wird, soll kurz auf die verschiedenen Modi des Cursors und die zur Verfügung stehenden Editiermöglichkeiten eingegangen werden.

Bei der Programmierung im Kontaktplan kann mit der rechten Maustaste ein Auswahlfenster geöffnet werden (siehe unten). Durch Klicken auf **Auto Connect** kann diese Funktion ein- oder ausgeschaltet werden. In diesem Fenster kann auch der **Verbindungs-Modus** aktiviert oder deaktiviert werden. Der Cursor wird dadurch entweder als Pfeil oder als Stift dargestellt. Diese Umschaltung kann aber auch in der Werkzeugeiste vorgenommen werden.

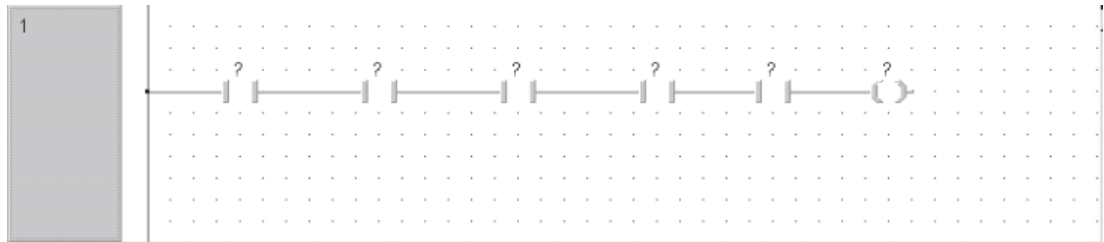


Hinweise zum Gebrauch des Kontaktplan-Editors


In der folgenden Abbildung wird deutlich, dass eine Reihe von Kontakten mit **Auto Connect** nicht verbunden werden kann, weil **Auto Connect** nur zwei Punkte verbindet. Wenn **Auto Connect** aktiviert ist, können diese Kontakte nur verbunden werden, indem immer zwei benachbarte Elemente separat verbunden werden.



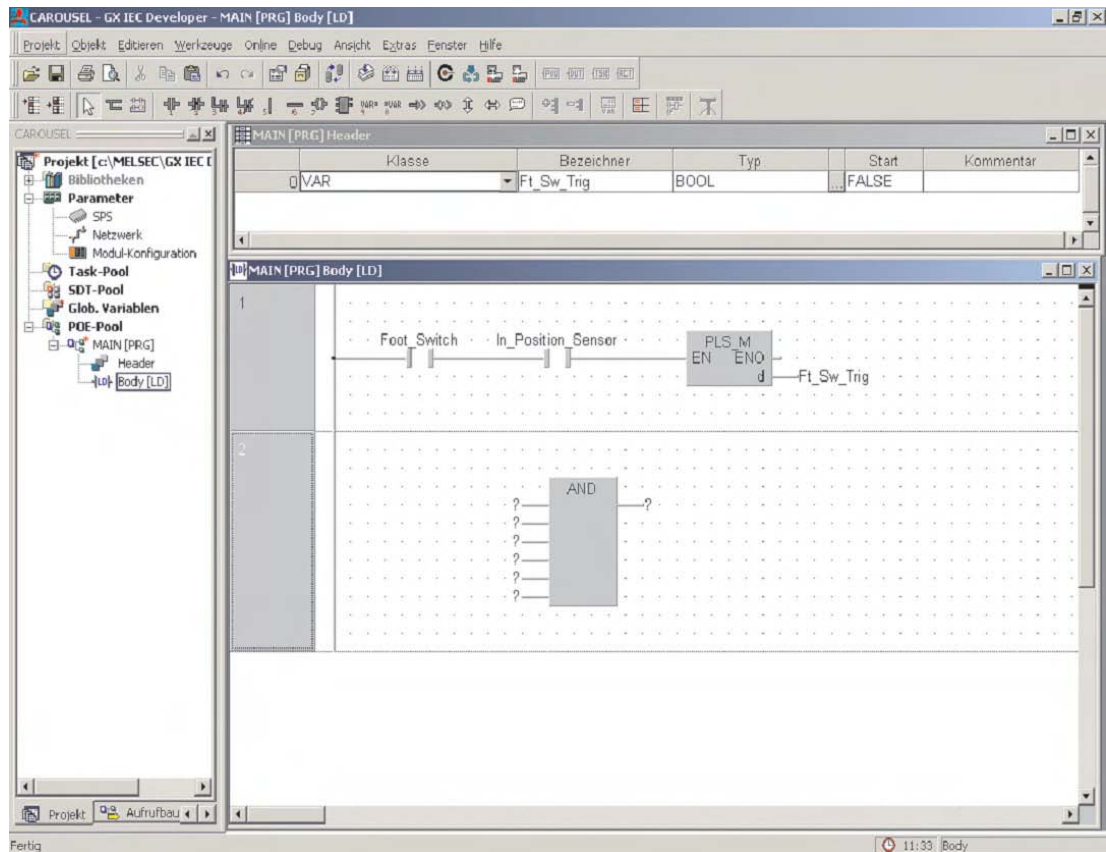
Ohne **Auto Connect** kann der Stift in einem Arbeitsschritt von der Sammelschiene links durch alle Kontakte bis zur „Spule“ ganz rechts geführt werden.




Zum Verbinden mehrerer Elemente oder beim Einfügen vom Elementen in bestehende Netzwerke sollte **Auto Connect** daher ausgeschaltet werden.

Die Anzahl der Eingangsvariablen von Funktionen, wie beispielsweise der MUL-Anweisung (Multiplikation) kann verringert oder vergrößert werden. Klicken Sie dazu auf die Funktion und betätigen Sie anschließend das entsprechende Schaltfeld in der Werkzeugleiste. 

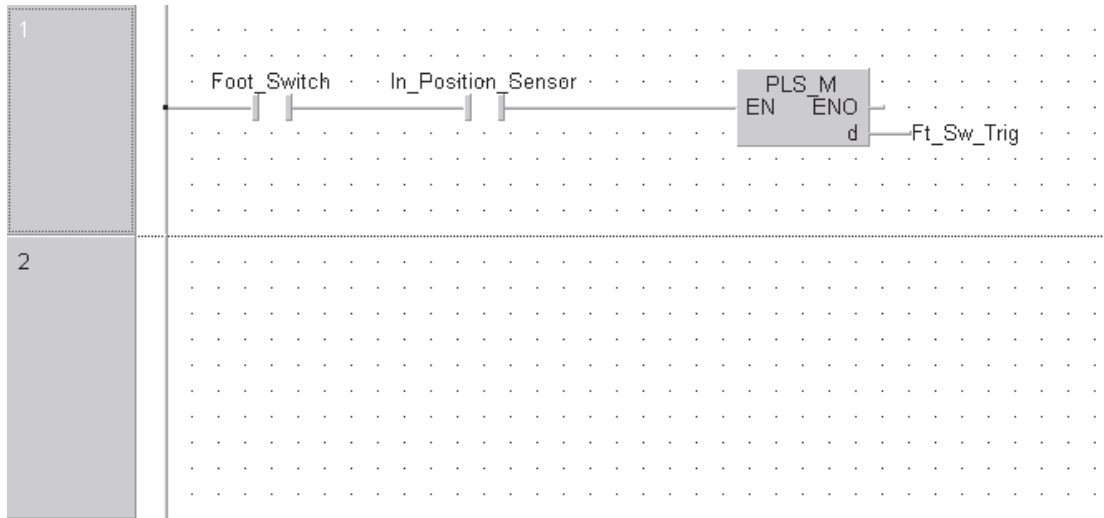
Sie können aber auch den Cursor auf den unteren Rand der Funktion positionieren, bis er als Doppelpfeil dargestellt wird. Ziehen Sie dann den Cursor bei betätigter linker Maustaste nach unten, bis die gewünschte Anzahl Eingänge angezeigt wird. Zur Verringerung der Eingänge bewegen Sie den Cursor bei betätigter linker Maustaste nach oben:



Einfügen eines neuen Programmnetzwerks

- ① Um unter dem momentan bearbeiteten Netzwerk ein neues Netzwerk einzufügen, klicken Sie bitte auf dieses Schaltfeld in der Werkzeugleiste:  .

Es erscheint ein leeres Netzwerk:




- ② Geben Sie in diese Netzwerk die folgenden Programmelemente ein:

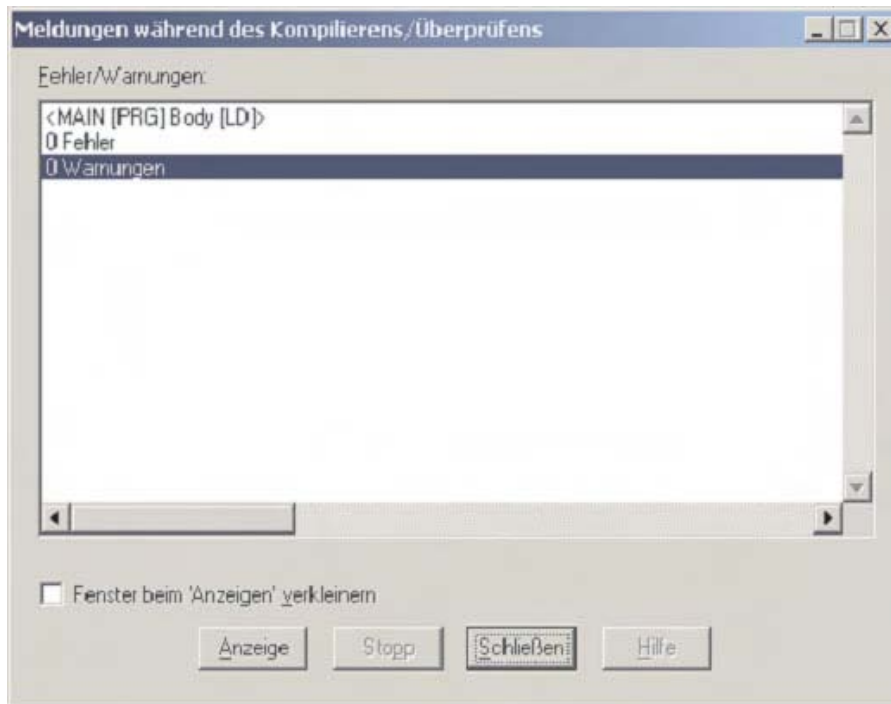


- ③ Programmieren Sie dann das dritte und letzte Netzwerk:



Eingegebenes Programm prüfen

Nach der Eingabe der drei Netzwerke klicken Sie bitte auf das Schaltfeld  , um das Programm zu prüfen.



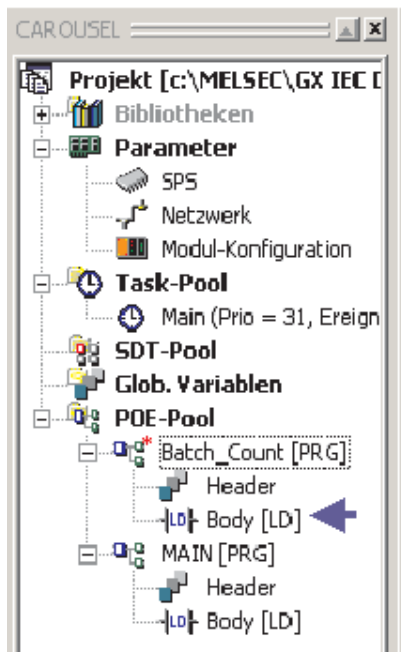
Programmieren von Countern und Timern

Die am Anfang des Kapitels beschriebenen Funktionen zum Bewegen des Drehtisches sind nun programmiert. Anhand von zusätzlichen Funktionen soll die Verwendung von Countern und Timern gezeigt werden:

- Zählung der produzierten Stückzahl
Dieser Zähler erfasst jedes Einschalten des Antriebsmotors. Da damit auch jedesmal ein neues Werkstück zum Arbeitsplatz befördert wird, entspricht dieser Zahlwert der Anzahl der hergestellten Produkte.
- Anzeige von jeweils 10 hergestellten Produkten.
Nachdem 10 Produkte zum Arbeitsplatz transportiert wurden, blinkt eine Leuchte solange im Sekundentakt, bis der Zähler durch einen Taster zurückgesetzt wird.

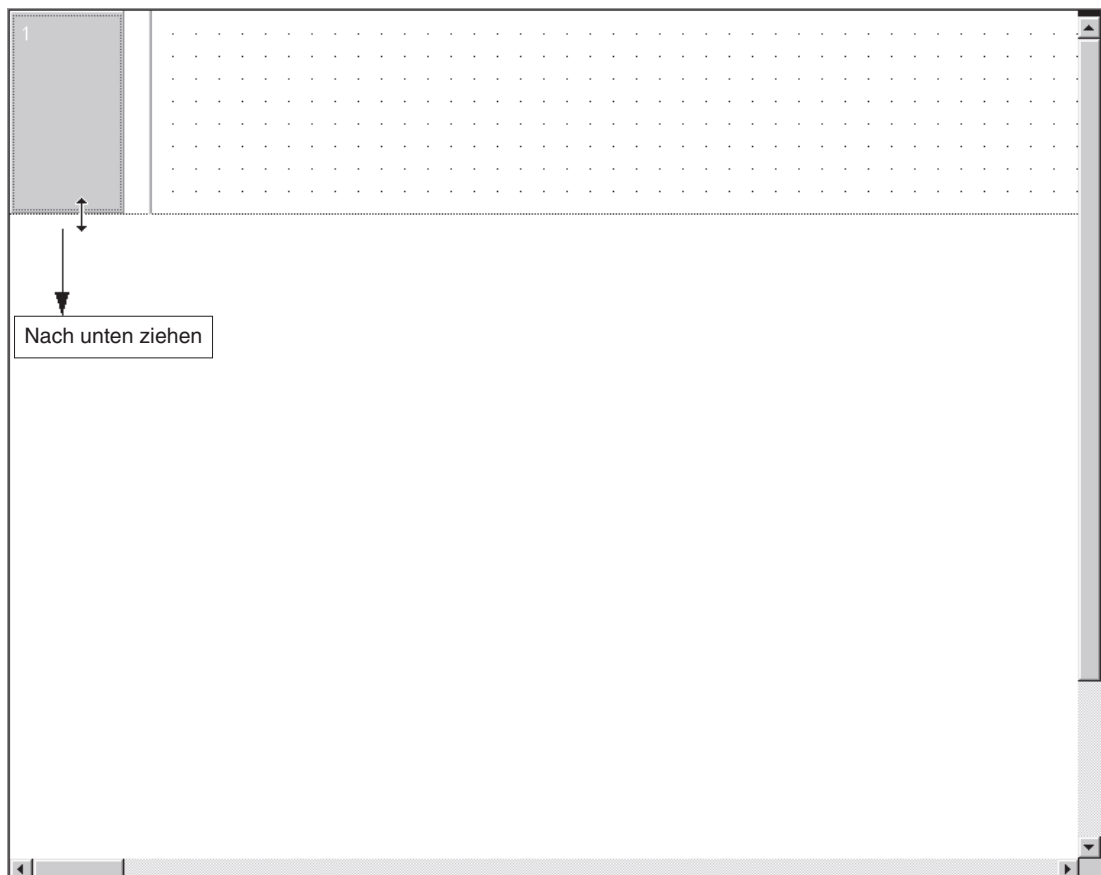
Diese Zähl- und Anzeigefunktionen werden in einer neuen POE programmiert:

① Erzeugen Sie eine neue POE, indem Sie auf das Schaltfeld  klicken.



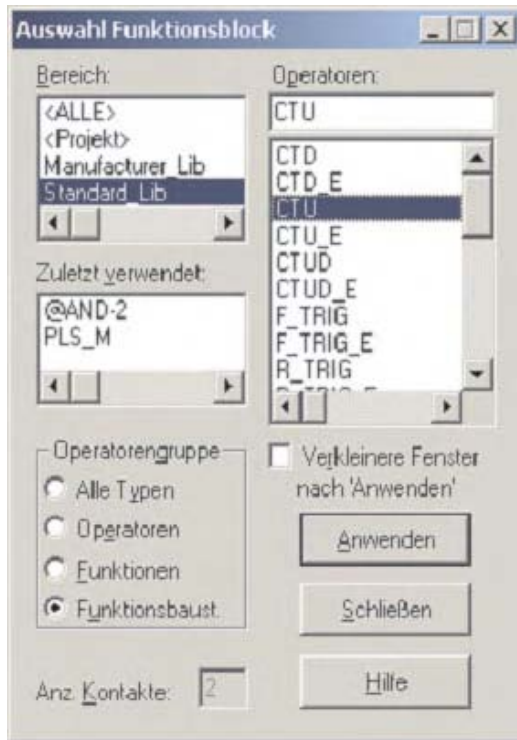
② Öffnen Sie den Body der neuen POE.

Sie können das Netzwerk zur Eingabe des Kontaktplan vergrößern, wenn Sie den Cursor auf den unteren Rand des Netzwerks positionieren, bis aus dem Cursorpfeil ein Doppelpfeil wird und anschließend den Cursor bei betätigter linker Maustaste nach unten bewegen.

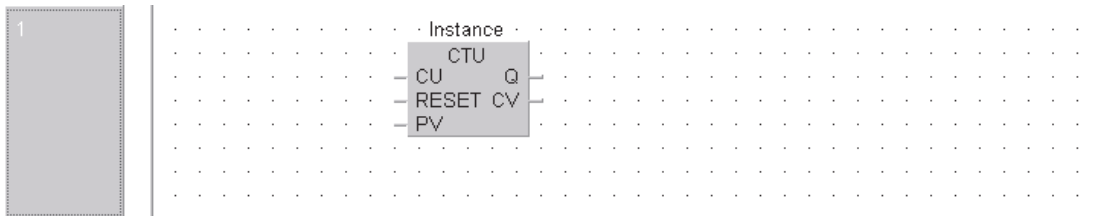


Zählfunktion eingeben

Klicken Sie im Auswahlmodus des Editors auf die Funktionsauswahl und wählen Sie den Funktionsbaustein CTU (Count Up).



Klicken Sie auf Anwenden, positionieren Sie den Cursor an die gewünschte Position im Netzwerk und betätigen Sie die linke Maustaste, um den Funktionsbaustein abzulegen.

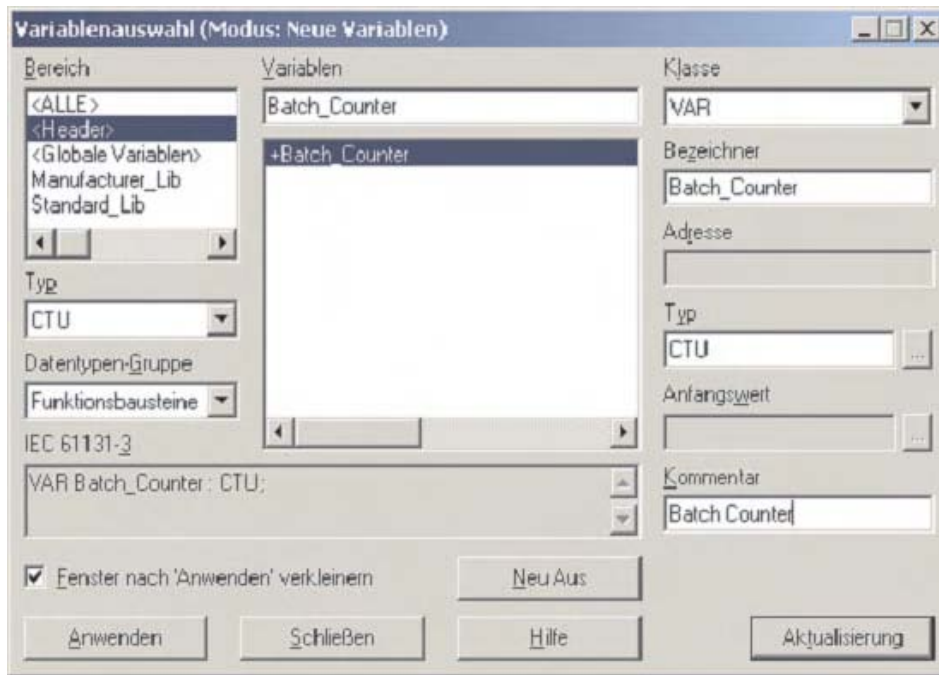


Instanzen von Funktionsbausteinen

Funktionsbausteine können nur in Form von **Instanzen** aufgerufen werden. Bei der Instanziierung eines Funktionsbausteins wird in der Deklarationstabelle des Headers der betreffenden POE eine Kopie des Funktionsbausteins angelegt. Die Instanz ist als Variable VAR mit einem beliebigen Bezeichner zu deklarieren. Für „Typ“ ist der Name des Funktionsbausteins einzutragen. Für ein und demselben Funktionsbaustein können in einer POE mehrere Instanzen mit unterschiedlichen Namen gebildet werden. Die Instanzen werden im Body der POE aufgerufen und die **aktuellen** Parameter werden an die **formalen** Parameter übergeben. Jede Instanz kann mehrmals verwendet werden.

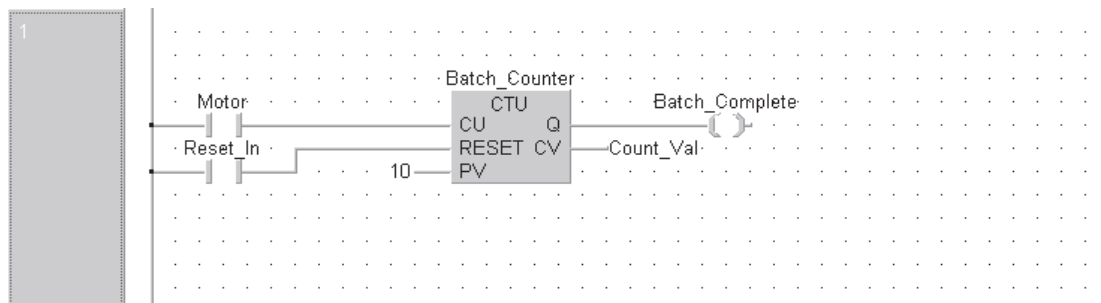
Einfügen des IEC-Funktionsbausteins CTU


- ① Um in dieser POE der Instanz des Funktionsbausteins CTU einen neuen Namen zu geben, klicken Sie über dem Funktionsbaustein auf den Text **Instanz**. Betätigen Sie dann die Taste F2 der PC-Tastatur, um das Dialogfenster **Variablenauswahl** zu öffnen. Nehmen Sie dort die folgenden Einträge vor. (Eventuell muss das Schaltfeld **NEU Ein** betätigt werden, um alle Optionen anzuzeigen.)



- Batch_Counter ·
 - CTU ·
 - CU Q ·
 - RESET CV ·
 - PV ·
- ② Klicken Sie auf **Anwenden** und anschließend auf **Übernehmen**. Dadurch wird der eingegebene Bezeichner (Variablenname) übernommen.

③ Geben Sie nun die übrigen Programmelemente für den Zähler ein:





Die Ein- und Ausgangsvariablen PV und CV können mit Hilfe der Schaltfelder  eingegeben werden.

Globale Variablenliste erweitern.

Beachten Sie besonders, dass die Funktion „Reset_In“ zum Zurücksetzen des Zählers ein neuer Eingang mit der MELSEC Adresse X12 oder der IEC-Adresse %IX18 ist. Er erfordert einen neuen Eintrag in der Globalen Variablenliste:

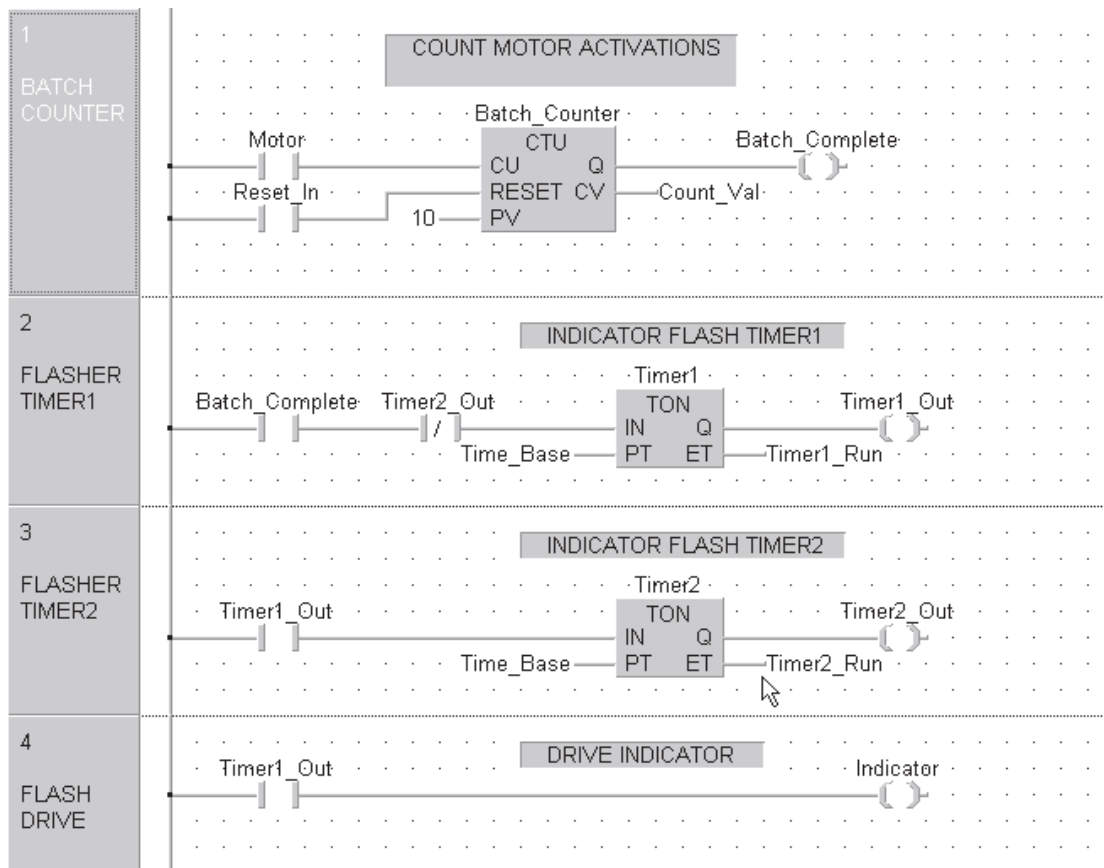
	Klasse	Bezeichner	MIT-Adr.	IEC-Adress	Typ	Start
0	VAR_GLOBAL	Foot_Switch	X10	%IX16	BOOL	FALSE
1	VAR_GLOBAL	In_Position_Sensor	X11	%IX17	BOOL	FALSE
2	VAR_GLOBAL	Reset_In	X12	%IX18	BOOL	FALSE
3	VAR_GLOBAL	Motor	Y20	%QX32	BOOL	FALSE

	Klasse	Bezeichner	Typ	Start	Kommentar
0	VAR	Batch_Counter	CTU	...	Batch Counter
1	VAR	Batch_Complete	BOOL	FALSE	Batch Complete
2	VAR	Count_Val	INT	0	

Nach der Eingabe aller neuen Variablen klicken Sie auf das Schaltfeld , um die GVL zu prüfen und anschließend auf das Schaltfeld , um das Projekt in den Maschinencode zu wandeln.

Programmieren von Timern

Geben Sie nach dem Netzwerk mit dem Zähler die unten abgebildeten Netzwerke ein:





Nach der Programmierung sollte die GVL so aussehen:

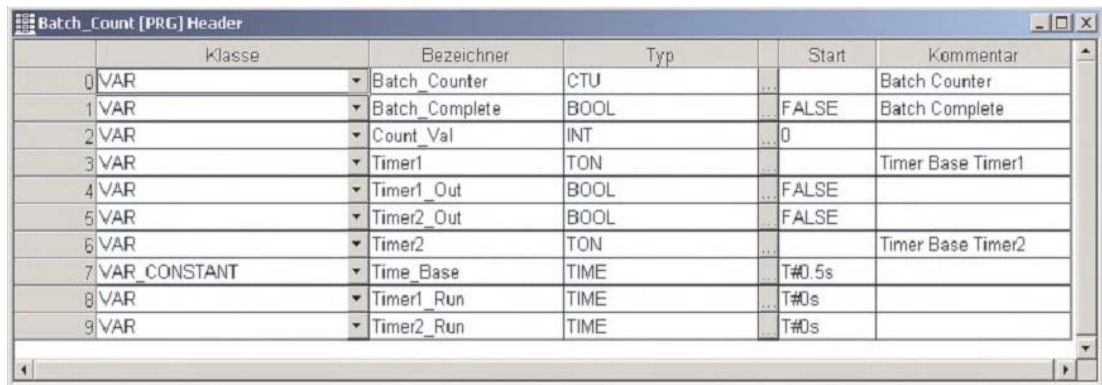
	Klasse	Bezeichner	MIT-Adr.	IEC-Adress	Typ	Start	Korr
0	VAR_GLOBAL	Foot_Switch	X10	%IX16	BOOL	FALSE	
1	VAR_GLOBAL	In_Position_Sensor	X11	%IX17	BOOL	FALSE	
2	VAR_GLOBAL	Reset_In	X12	%IX18	BOOL	FALSE	
3	VAR_GLOBAL	Motor	Y20	%QX32	BOOL	FALSE	
4	VAR_GLOBAL	Indicator	Y21	%QX33	BOOL	FALSE	

Der Header (Lokale Variablenliste) dieser POE sollte die folgenden Einträge enthalten:

	Klasse	Bezeichner	Typ	Start	Kommentar
0	VAR	Batch_Counter	CTU	...	Batch Counter
1	VAR	Batch_Complete	BOOL	... FALSE	Batch Complete
2	VAR	Count_Val	INT	... 0	
3	VAR	Timer1	TON	...	Timer Base Timer1
4	VAR	Timer1_Out	BOOL	... FALSE	
5	VAR	Timer2_Out	BOOL	... FALSE	
6	VAR	Timer2	TON	...	Timer Base Timer2
7	VAR_CONSTANT	Time_Base	TIME	... T#0.5s	
8	VAR	Timer1_Run	TIME	... T#0s	
9	VAR	Timer2_Run	TIME	... T#0s	

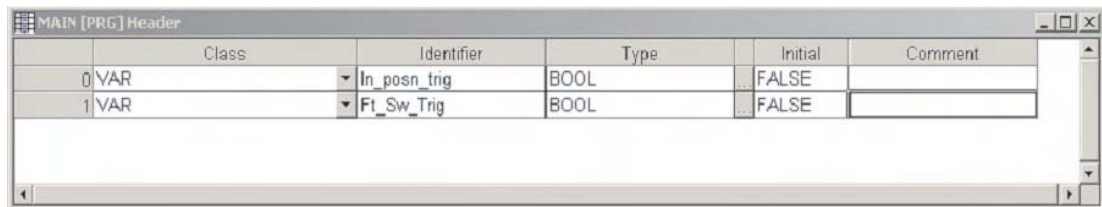
Wenn alle Eingaben vorgenommen wurden, klicken Sie auf das Schaltfeld  und anschließend auf das Schaltfeld , um das Projekt zu prüfen und in den Maschinencode zu wandeln.

Header der POE „Batch_Count“



	Klasse	Bezeichner	Typ	Start	Kommentar
0	VAR	Batch_Counter	CTU	...	Batch Counter
1	VAR	Batch_Complete	BOOL	... FALSE	Batch Complete
2	VAR	Count_Val	INT	... 0	
3	VAR	Timer1	TON	...	Timer Base Timer1
4	VAR	Timer1_Out	BOOL	... FALSE	
5	VAR	Timer2_Out	BOOL	... FALSE	
6	VAR	Timer2	TON	...	Timer Base Timer2
7	VAR_CONSTANT	Time_Base	TIME	... T#0.5s	
8	VAR	Timer1_Run	TIME	... T#0s	
9	VAR	Timer2_Run	TIME	... T#0s	

Header der POE „MAIN“:

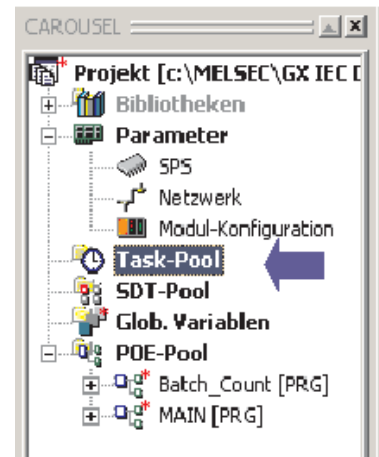



	Class	Identifier	Type	Initial	Comment
0	VAR	In_posn_trig	BOOL	... FALSE	
1	VAR	Ft_Sw_Trig	BOOL	... FALSE	

4.2.6 Anlegen einer neuen Task

Damit die POEs „MAIN“ und „Batch_Count“ in der SPS ausgeführt werden, müssen sie als gültige Task im **Task Pool** eingetragen werden.

- 1 Klicken Sie im Navigatorfenster auf den Eintrag **Task-Pool**, um ihn zu markieren.

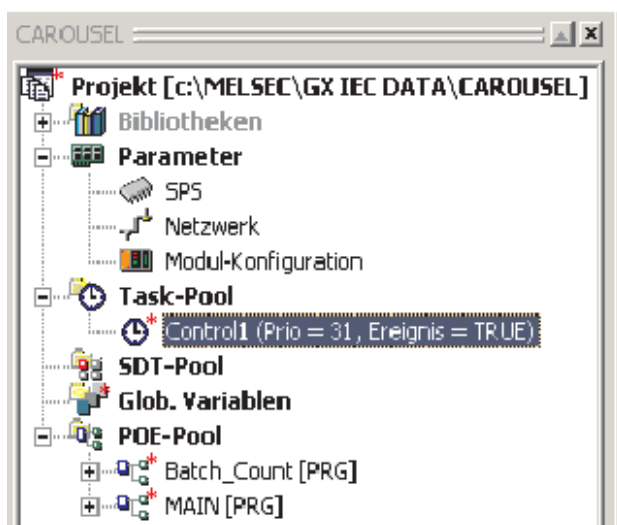


- 2 Klicken Sie dann auf das Schaltfeld  (Neue Task) in der Werkzeugleiste. Sie können aber auch nach dem Markieren des Eintrags **Task-Pool** die rechte Maustaste betätigen und aus dem Menü, das dann angezeigt wird, **Neue Task** wählen.

- 3 Geben Sie in das Dialogfenster, das dann angezeigt wird, den Namen der neuen Task („Control1“) ein .



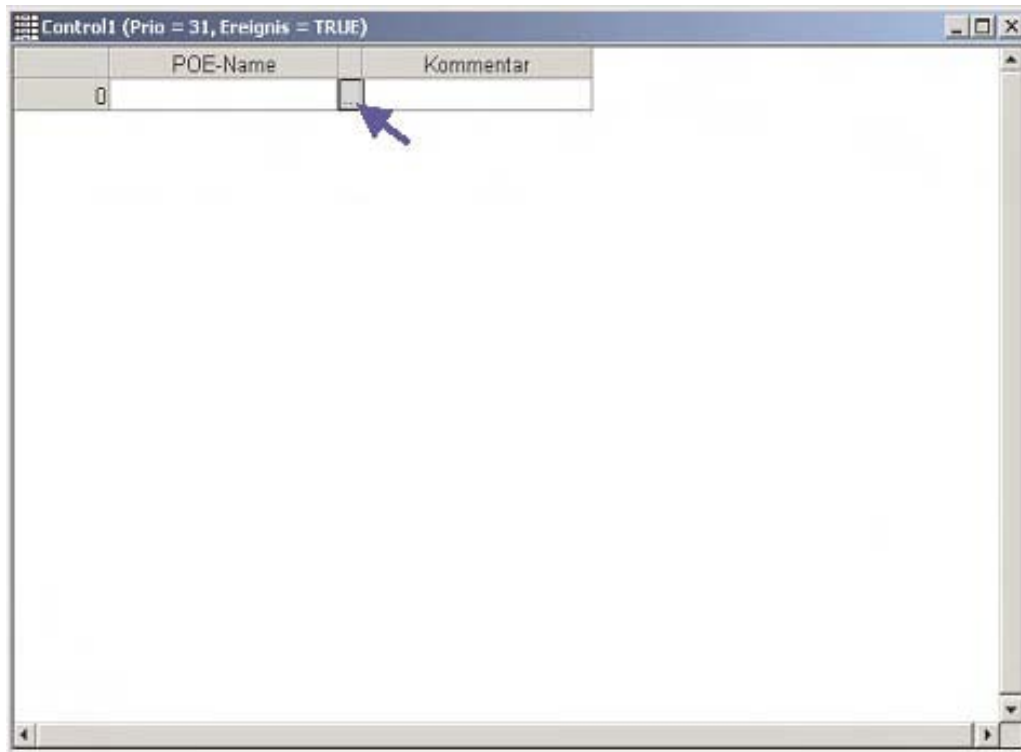
- 4 Klicken Sie nach der Eingabe auf **OK**. Im Navigatorfenster wird nun die neue Task „Control1“ angezeigt:



POE einer Task zuweisen

Der neuen Task „Control1“ müssen nun POEs zugewiesen werden.

- ① Klicken Sie doppelt auf den Eintrag „Control1“ im Navigatorfenster. Dadurch wird die Liste mit den Ereignissen für diese Task angezeigt.



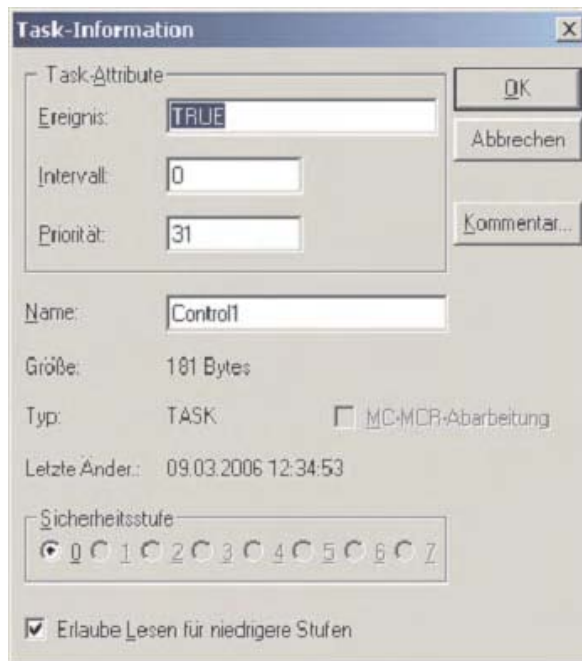
- ② Klicken Sie, wie oben gezeigt, auf das Schaltfeld in der Mitte zwischen den beiden Eingabefeldern. Danach wird das **Programmauswahl**-Fenster angezeigt:



- ③ Wählen Sie MAIN und klicken Sie danach auf **OK**. Dadurch wird die Zuweisung abgeschlossen.


Task-Eigenschaften

Sie können sich die Eigenschaften einer Task anzeigen lassen, indem Sie die gewünschte Task im Navigatorfenster durch einen Mausklick markieren und anschließend die rechte Maustaste betätigen. Wählen Sie aus dem dann angezeigten Menü **Eigenschaften**. Dadurch wird das unten abgebildete Dialogfenster angezeigt.




- Task-Attribute

- Ereignis = TRUE: Die Task wird immer (zyklisch) ausgeführt.
- Intervall = 0: Hier ist der Wert „0“ eingetragen, weil die Task zyklisch ausgeführt wird (Ereignis = TRUE).
- Priorität = 31: 31 ist die niedrigste Priorität. Das bedeutet, dass diese Task zuletzt ausgeführt wird.

Bevor weiter programmiert wird, sollten Sie das Projekt speichern. Klicken Sie dazu auf das Schaltfeld  in der Werkzeugleiste.

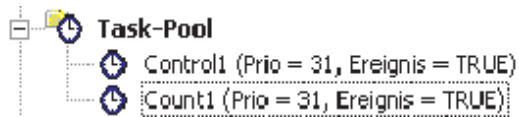
Neue Task für die POE „Batch-Count“ anlegen

Die POE „Batch-Count“ muss ebenfalls einer Task zugewiesen werden.

- ① Zum Anlegen einer neuen Task markieren Sie den Eintrag **Task-Pool** im Navigatorfenster und betätigen dann die rechte Maustaste. Wählen Sie aus dem Menü, das dann angezeigt wird, den Eintrag **Neue Task**. Sie können aber auch auf das Schaltfeld  (Neue Task) in der Werkzeugleiste klicken.
- ② Geben Sie, wie unten gezeigt, den Namen „Count1“ ein.





Die neue Task erscheint im Task-Pool unter den bereits vorhandenen Task:



③ Klicken Sie im Navigatorfenster doppelt auf den Namen der neuen Task „Count1“.

④ Weisen Sie dieser Task die POE mit dem Zähler zu:

	POE-Name	Kommentar
<input type="checkbox"/>	Batch_Count	...


Nach der Eingabe klicken Sie bitte nacheinander auf die Schaltfelder  und , um das Projekt zu prüfen und in den Maschinencode zu wandeln.

Sicher Sie das Projekt, indem Sie das Schaltfeld  betätigen. Das Projekt ist nun vollständig programmiert und kann in die SPS übertragen werden.

4.2.7 Dokumentation eines Programms

Netzwerk-Label

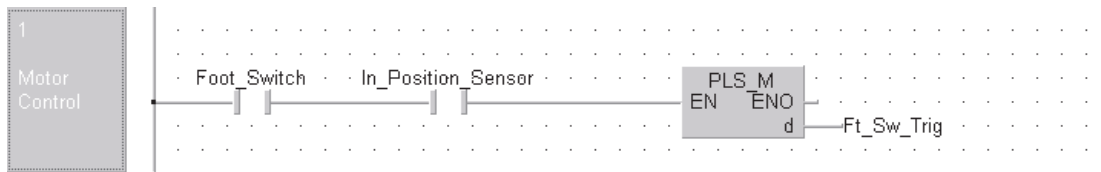
Zu jedem Netzwerk im Programm kann ein Text (Titel) eingegeben werden, der aus bis zu 22 Zeichen bestehen kann. Diese Netzwerk-Titel dienen zur Erläuterung und Gliederung des Programms und sollen zum besseren und schnelleren Verständnis besonders von großen Programmen (mit vielen Netzwerken) beitragen.

- 1 Markieren Sie das erste Netzwerk der POE MAIN, indem Sie einmal in das Netzwerk klicken. Klicken Sie dann in der Werkzeugleiste auf das Schaltfeld  (Netzwerk-Label). Sie können aber auch zweimal in den linken grauen Bereich des Netzwerks klicken.



Geben Sie im Feld **Titel** einen Kommentar ein. Geben Sie bitte nichts in das Feld **Label** ein! Dieses Feld hat eine andere Funktion und darf daher jetzt **nicht** ausgefüllt werden.


- 2 Nachdem Sie auf **OK** geklickt haben, wird der eingegebene Titel an der linken Seite des Netzwerks angezeigt:




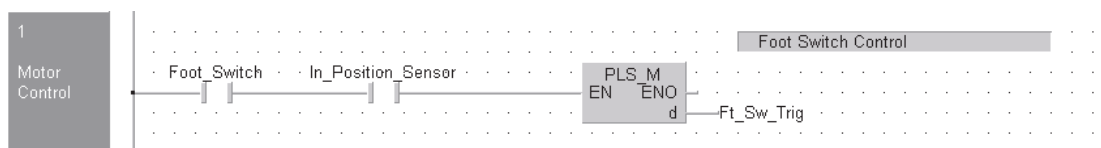
Bitte beachten Sie, dass der Netzwerk-Titel eventuell durch Einfügen von Leerzeichen umformatiert werden muss. Dies hängt von der Bildschirmauflösung ab. Der eingegebene Text wird automatisch umbrochen, damit er in den zur Verfügung stehenden Platz passt. Maximal können 22 Zeichen eingegeben werden.

Netzwerk-Kommentar

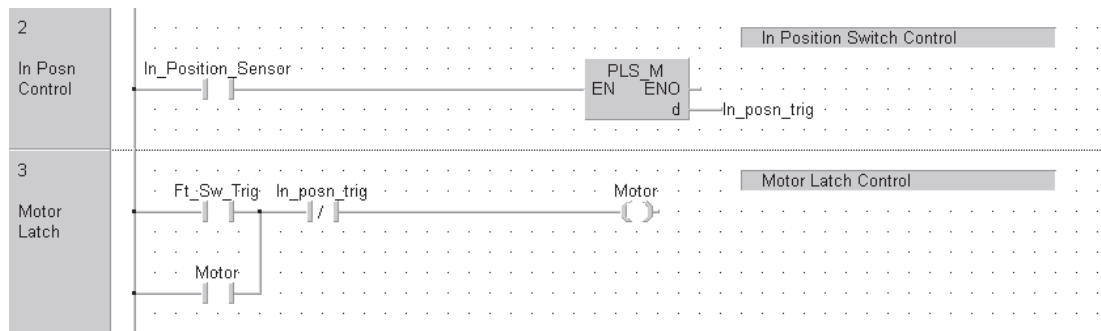
Als Netzwerk-Kommentare können beliebige Texte an jeder Stelle eines Kontaktplan-Netzwerks eingegeben werden. Dadurch kann das Verständnis des Programms erheblich erleichtert werden.

- 1 Zur Eingabe eines Kommentars klicken Sie bitte das Schaltfeld  in der Werkzeugleiste.

- 2 Am Cursor wird dadurch eine Sprechblase angefügt: . Positionieren Sie den Cursor an der Stelle im Netzwerk, an der der Kommentar erscheinen soll, und betätigen Sie dort die linke Maustaste. Geben Sie den Kommentar ein und betätigen Sie danach die Übernahmetaste der PC-Tastatur.



Dokumentieren Sie so auch das andere Netzwerk der POE.



Position eines Kommentars verändern

Um einen bereits eingegebenen Kommentar zu verschieben, klicken Sie im Auswahlmodus (der Cursor wird in diesem Fall als einfacher Pfeil dargestellt) auf den linken Bereich des Kommentars. Lassen Sie die linke Maustaste betätigt und bewegen Sie den Kommentar. An der gewünschten Position lassen Sie dann die linke Maustaste los.


Löschen eines Kommentars

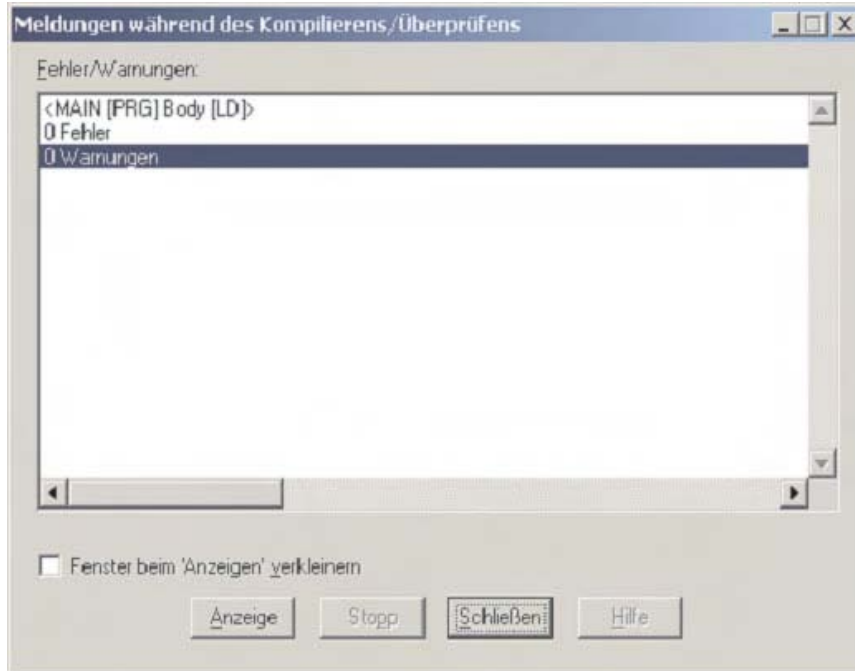
Um nur den Text eines Netzwerk-Kommentars zu löschen, klicken Sie einmal auf den Text und betätigen dann die **Entf**-Taste Ihrer PC-Tastatur. Wenn auch das Kommentarfeld gelöscht werden soll, klicken Sie links neben dem Text in dieses Feld und betätigen dann die **Entf**-Taste.



Kopieren eines Kommentars

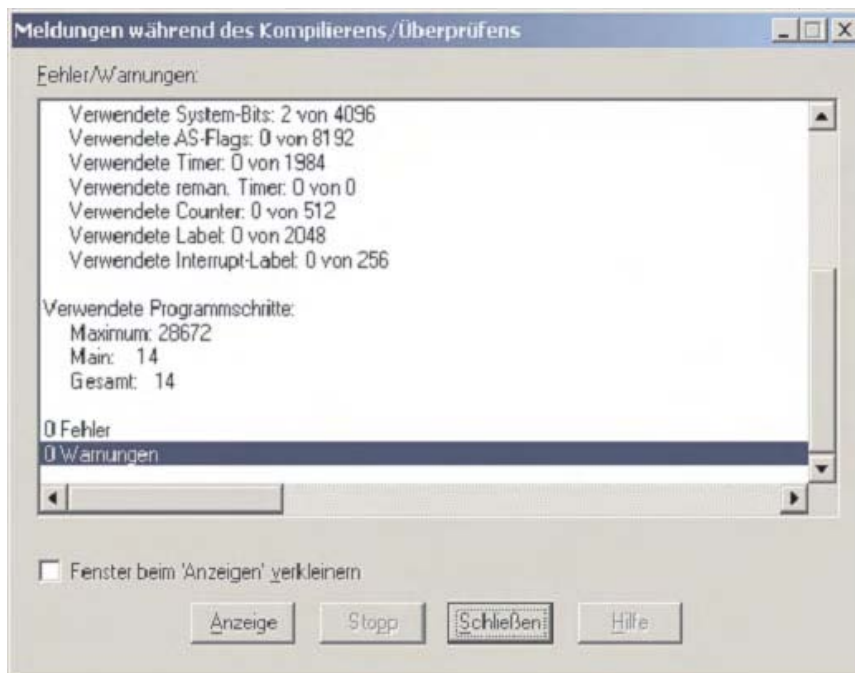
Einen Kommentar können Sie kopieren, indem links neben dem Text in das Kommentarfeld klicken und den Kommentar dadurch markieren. Mit den üblichen Windows-Funktionen für Kopieren und Einfügen kann der Kommentar dann dupliziert werden. Der Kommentar kann auch in ein anderes Netzwerk eingefügt werden.

4.2.8 Projekt prüfen und in den Maschinencode wandeln

- ① Nach der Programmierung und Zuweisung der Task sollte ein Programm geprüft werden. Klicken Sie dazu auf das Schaltfeld  („Prüfen“) in der Werkzeugleiste. Das folgende Dialogfenster wird angezeigt:




- ② Klicken Sie entweder auf das Schaltfeld  („Erstellen“) oder das Schaltfeld  („Alles neu erstellen“). Dadurch wird das Projekt in den für die SPS verständlichen Maschinencode gewandelt. Dieser Vorgang wird auch „Kompilieren“ genannt. Wenn das Projekt fehlerfrei kompiliert gewandelt werden konnte, wird dieses Meldung angezeigt:



- ③ Klicken Sie auf **Schließen**, um das Fenster auszublenden.

4.2.9 Demonstration des Geführten Editors

Zusätzlich zu den bisher beschriebenen Methoden zur Eingabe von Kontaktplan-Programmen bietet der GX IEC Developer ab der Version 6 den **Geführten Editor**. Der Geführte Editor ist ein zusätzlicher Modus zur Erstellung von Programmen im Kontaktplan, bei dem hauptsächlich die Tastatur verwendet und Eingaben mit Hilfe der Maus reduziert werden. Er hilft auch beim Umstieg von der Programmier-Software MELSEC MEDOC oder GX IEC Developer zum GX IEC Developer.

- ① Aktivieren Sie den Geführten Editor durch einen Klick auf das Schaltfeld  in der Werkzeugleiste. Die Arbeitsfläche wird dadurch mit einem Raster versehen:




- ② Verwenden Sie die unten abgebildeten Schaltflächen in der Werkzeugleiste zur Auswahl der Kontaktelemente. Zur Auswahl dieser Elemente kann mit der Tastatur die Ziffer eingegeben werden, die unter dem Symbol angegeben ist. Dadurch wird zur Eingabe keine Maus benötigt:



- ③ Tippen Sie zur Eingabe eines Schließerkontakt eine „1“ ein. Danach ändert sich die Anzeige:



Wenn das Fragezeichen (?) über dem Programmsymbol blau hinterlegt ist, kann zur Eingabe eines Operanden die Taste F2 der PC-Tastatur betätigt werden. Falls Sie eine Maus verwenden, können Sie auch auf das Schaltfeld  in der Werkzeugleiste klicken.

4.3 Programm in die SPS übertragen

4.3.1 Anschluss des Programmiergeräts an die SPS

Voraussetzungen für den Transfer des Programms in die SPS sind, dass die SPS mit dem Programmiergerät verbunden und die Versorgungsspannung der Steuerung eingeschaltet ist.

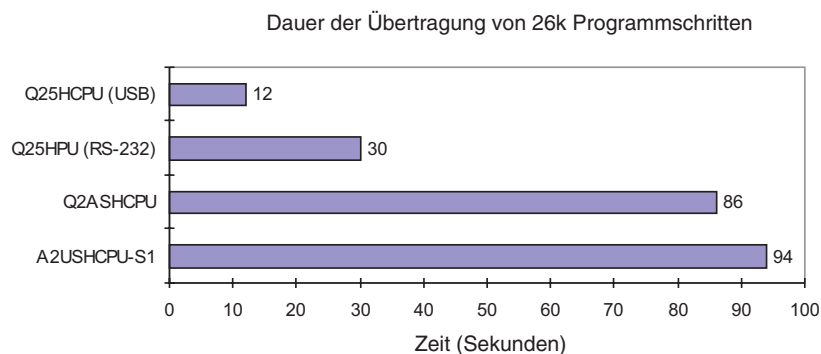
Zum Verbinden eines PC mit der Programmier-Software GX IEC Developer und einer Mitsubishi SPS gibt es mehrere Möglichkeiten:

- **Programmiergeräteschnittstelle der MELSEC FX-, A- oder QnA-Serie**
 Zum Anschluss an die Programmiergeräteschnittstelle wird das Kabel SC 09 verwendet. In das Kabel ist ein RS232/RS422-Konverter integriert, der die Signale des PC an die der SPS anpasst und umgekehrt.
- **Programmiergeräteschnittstelle des MELSEC System Q**
 Zum Anschluss eines PC an die Programmiergeräteschnittstelle der Steuerungen des MELSEC System Q wird ein besonderes RS232-Kabel verwendet.
- **USB-Schnittstelle des MELSEC System Q**
 Die Verbindung zwischen PC und CPU wird mit einem Standard USB-Kabel hergestellt. Der Anschluss an die USB-Schnittstelle wird wegen der hohen Übertragungsgeschwindigkeit besonders empfohlen.

Verbinden Sie Ihren Computer mit der SPS des Trainings-Racks so wie hier dargestellt:



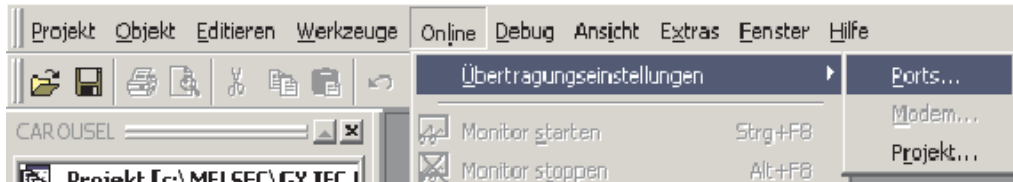
Im folgenden Diagramm sind die Programmübertragungszeiten der schnellsten CPU der A-Serie denen der QnA- Serie und des MELSEC System Q gegenübergestellt. Beachten Sie besonders die kurzen Übertragungszeiten des System Q im Vergleich zur A-Serie.



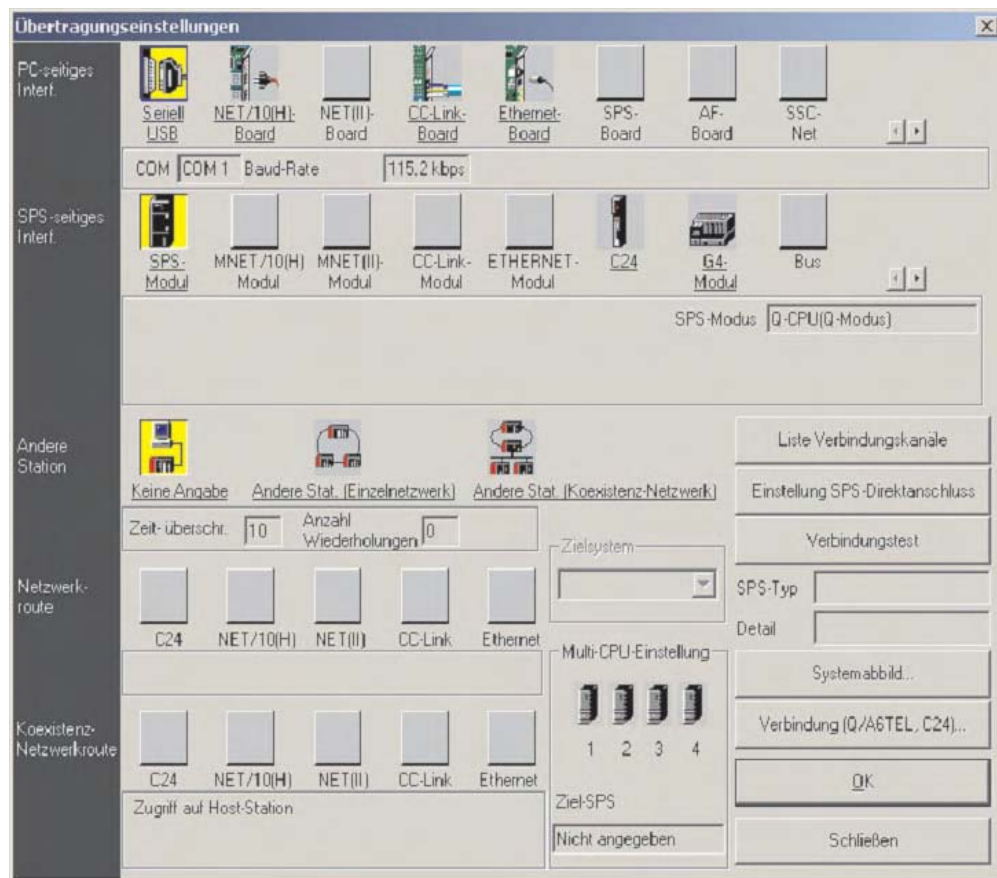
4.3.2 Übertragungseinstellungen (Ports)

Bevor zum ersten Mal ein Projekt in die SPS übertragen werden kann, müssen Einstellungen für die Übertragung vorgenommen werden.

- ① Wählen Sie im Menü **Online** den Eintrag **Übertragungseinstellungen** und dann **Ports**:



Dadurch wird das Dialogfenster **Übertragungseinstellungen** geöffnet:

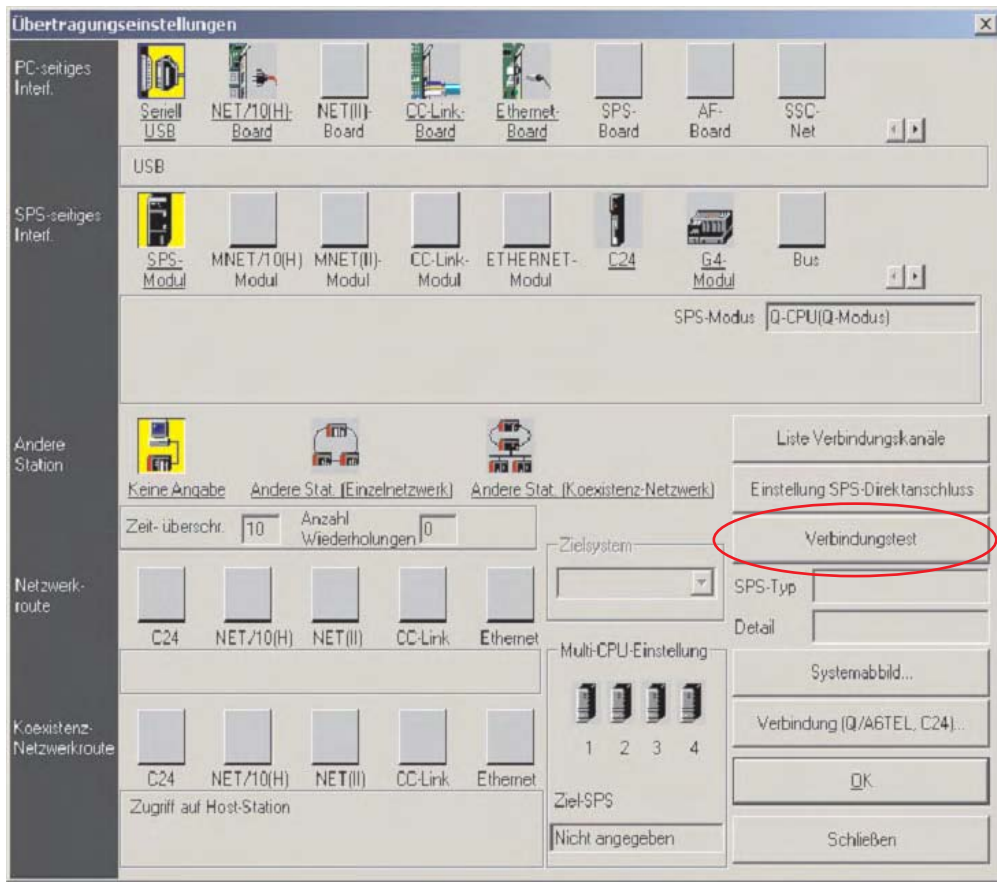


- ② Wählen Sie die Schnittstelle am PC durch einen Doppelklick auf **Seriell/USB** in der Zeile **PC-seitiges I/F** (Schnittstelle am PC). Dadurch wird das rechts abgebildete Dialogfenster angezeigt.



- ③ Wählen Sie, wie oben gezeigt, **USB** und klicken Sie anschließend auf **OK**.

④ Prüfen Sie anschließend die Verbindung, indem Sie auf **Verbindungstest** klicken.



Wenn die beiden Geräte Daten miteinander austauschen können, erscheint diese Meldung.

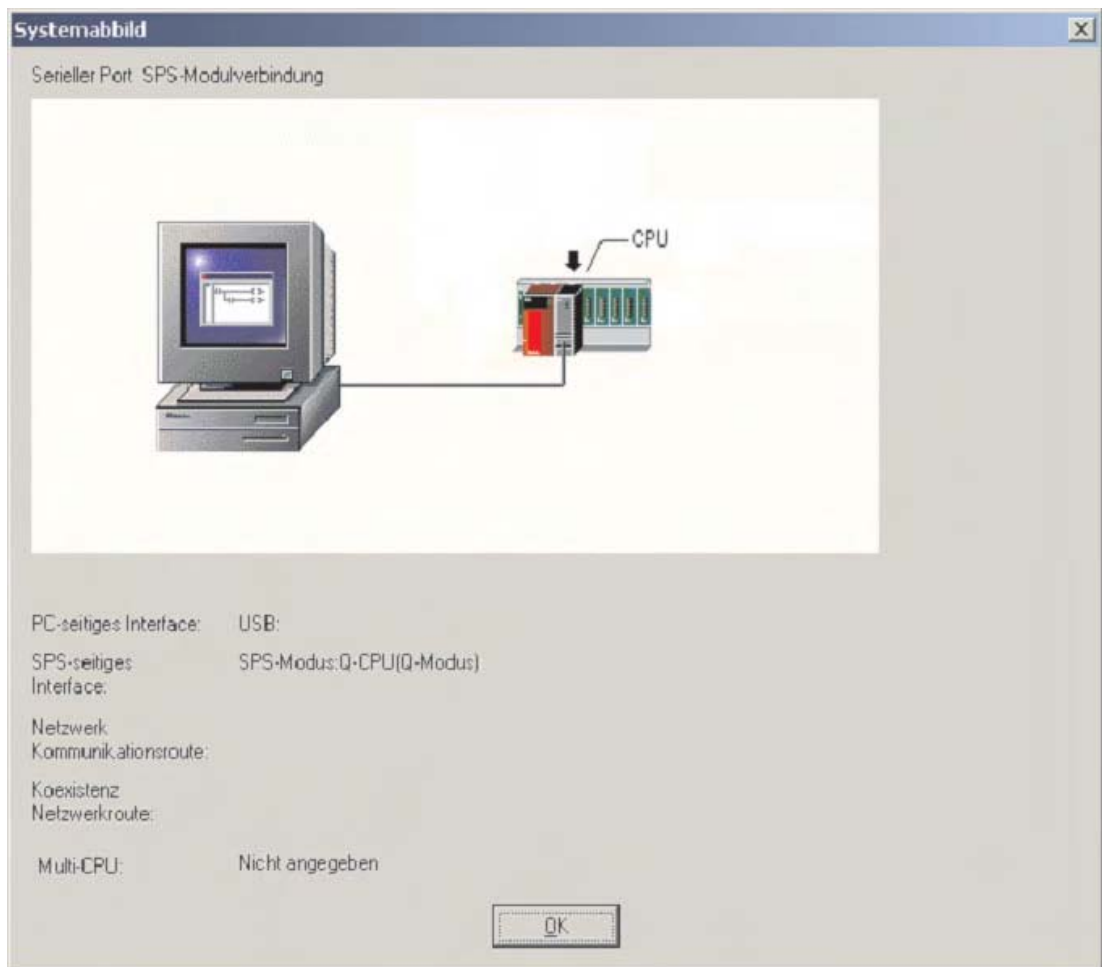


⑤ Klicken Sie auf **OK**, um diese Meldung auszublenden.

Falls eine Fehlermeldung angezeigt wird, prüfen Sie bitte die Leitungsverbindung und die Übertragungseinstellungen.

Systemabbild

- ① Für eine grafische Übersicht klicken Sie bitte auf das Schaltfeld **Systemabbild** im Dialogfenster **Übertragungseinstellungen**.



Der Übersicht kann entnommen werden, dass zur Verbindung zwischen PC und SPS die USB-Schnittstelle genutzt wird.

- ② Schließen Sie dieses Fenster durch einen Klick auf **OK**.

HINWEIS

Wenn zur Verbindung zwischen PC und SPS eine RS232-Schnittstelle genutzt wird, kann der COM-Port ebenfalls frei gewählt werden (z. B. COM1, COM2 usw.). Wählen Sie zu Kommunikation mit der SPS einen freien COM-Port.

- ③ Schließen Sie das Dialogfenster für die Übertragungseinstellungen durch einen Klick auf **OK**, um die Einstellungen zu speichern. Wenn das Dialogfenster **Übertragungseinstellungen** mit dem Schaltfeld **Schließen** geschlossen wird, werden die Einstellungen nicht übernommen.

4.3.3 Speicher der SPS formatieren

Bevor zum ersten Mal Parameter oder ein Programm in eine SPS des MELSEC System Q oder der QnA-Serie übertragen werden, muss der Speicher der CPU formatiert werden. Vor der ersten Verwendung einer Speicherkarte muss diese ebenfalls formatiert werden.

Der Speicher einer CPU des MELSEC System Q ist in unterschiedliche Laufwerke eingeteilt (siehe Abschnitt 2.6.1).

Wählen Sie im Menü **Online** den Eintrag **Laufwerk formatieren**. Es wird das Dialogfenster **Formatieren / Defragmentieren** angezeigt:



- 0 ... 4: Laufwerk, das formatiert oder defragmentiert werden soll
- Erzeuge einen Systembereich, um das Monitoren von anderen Systemen zu beschleunigen

Wenn eine SPS des MELSEC System Q oder der QnA-Serie mit dem GX IEC Developer verbunden ist und es werden z. B. Operandenzustände einer anderen SPS des MELSEC System Q oder der QnA-Serie beobachtet, muss in beiden Steuerungen mindestens ein Systembereich von 1k-Schritten eingerichtet sein. Ist in einer der Steuerungen dieser Systembereich nicht eingerichtet, ist das Monitoren der entfernten SPS nicht möglich.

- Erzeuge Systembereich, um Multi-Block-Online-Change zu aktivieren (MBOC)

In den CPUs, die die MBOC-Funktion unterstützen, können 1024 Schritte im Online-Change-Modus geändert werden. Diese 1024 Schritte müssen sich nicht unbedingt in einem zusammenhängenden Block befinden. Es können auch mehrere Änderungen in verschiedenen kleineren Blöcken vorgenommen werden. Die maximale Anzahl von zu bearbeitenden Blöcken beträgt 64. Die Anzahl von 1024 Schritten darf nicht überschritten

werden. (Die betrifft nur CPUs des MELSEC System Q, mit Ausnahme der Q00(J)- und Q01CPU)


- **Formatieren**

Mit diesem Schaltfeld wird der Formatiervorgang gestartet.

- **Defragment.**

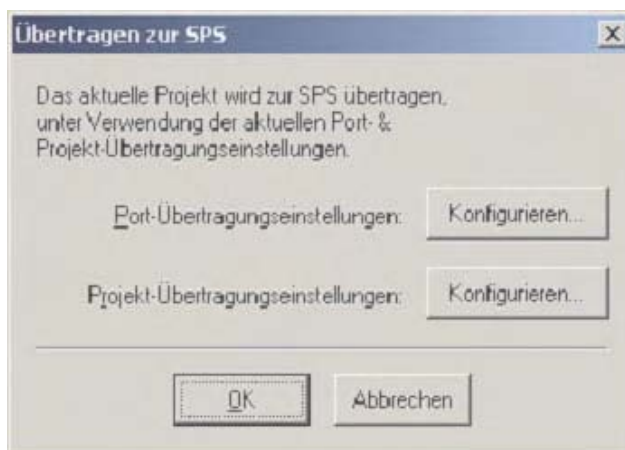
Bedingt durch die Speicherstruktur der CPUs des MELSEC System Q und der QnA-Serie können die Laufwerke nach häufigem Übertragen von Daten in die CPU fragmentiert werden. Dasselbe passiert auch mit der Festplatte eines PCs. Zur Wiederherstellung der ursprünglichen Leistung können die Speicher durch Betätigung des Schaltfelds **Defragment.** defragmentiert werden.

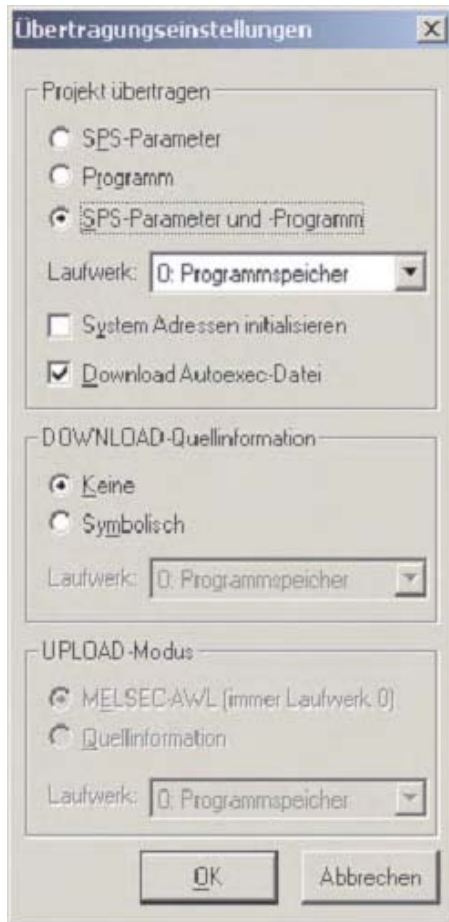
4.3.4 Transfer des Projekts in die SPS

- ① Zur Übertragung eines Projekts in die SPS klicken Sie in der Werkzeugleiste auf das Symbol  („Projekt übertragen“).

Projekt-Übertragungseinstellungen

- ② Das folgende Dialogfenster wird angezeigt. Klicken Sie das Schaltfeld **Konfigurieren** in der Zeile **Projekt-Übertragungseinstellungen**.

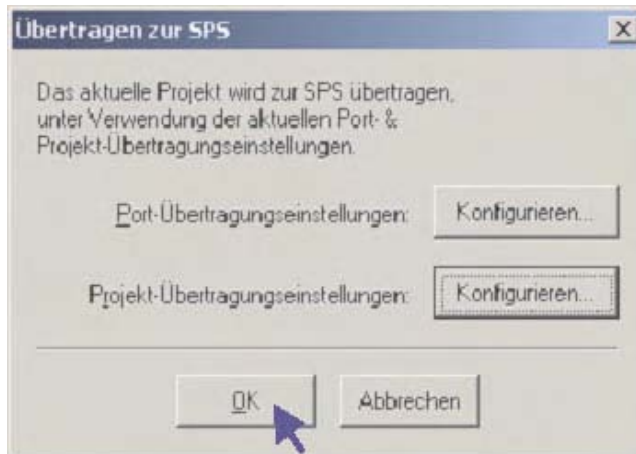




② Wählen Sie **SPS-Parameter und Programm**.

③ Schließen Sie die Einstellung durch einen Klick auf **OK** ab.

④ Starten Sie die Übertragung zur SPS durch einen Klick auf **OK**.




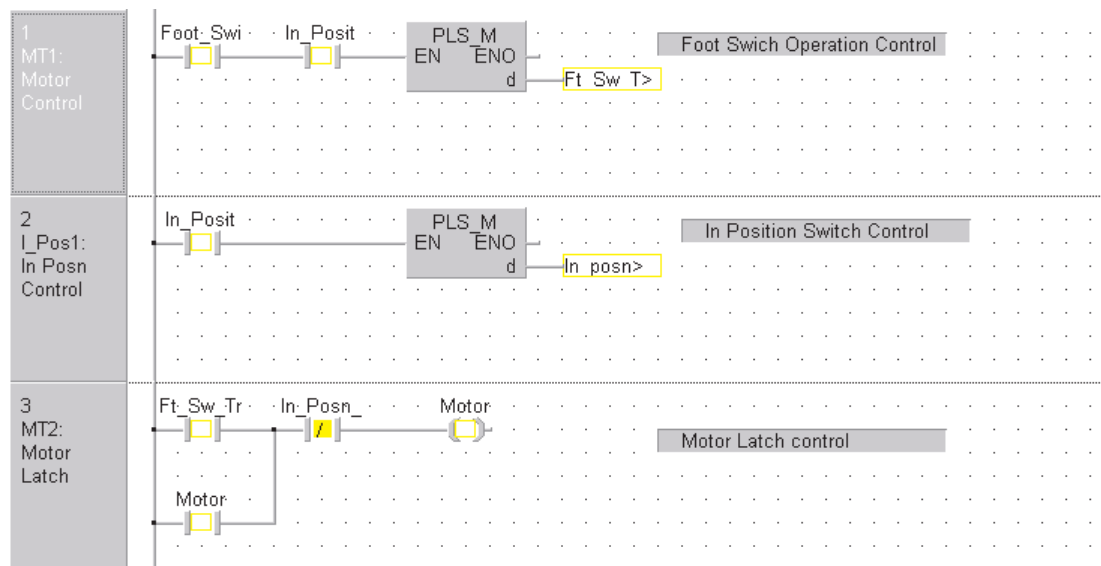
4.4 Ausführung des Programms beobachten

Das Testen und das Optimieren eines Programms, aber auch die Fehlersuche wird erleichtert, wenn die Programmausführung in der SPS beobachtet werden kann. Der GX IEC Developer bietet viele Möglichkeiten zur Anzeige des Status von Programmen und Operanden.

Im **Monitormodus** werden im Programm zusätzlich die Zustände der Operanden angezeigt. Die SPS muss eingeschaltet (RUN) und mit dem Programmiergerät verbunden sein, und es dürfen keine Fehler vorliegen.

Öffnen Sie den Body der POE MAIN.

Klicken Sie auf das Schaltfeld für den Monitormodus  in der Werkzeugleiste und beobachten Sie die Darstellung des Kontaktplans.




HINWEIS

Im Monitormodus werden Kontakte im Programm, deren Bedingungen erfüllt sind, farblich gekennzeichnet (Voreinstellung: Gelb). Werte von z. B. analogen Variablen werden direkt im Netzwerk angezeigt.

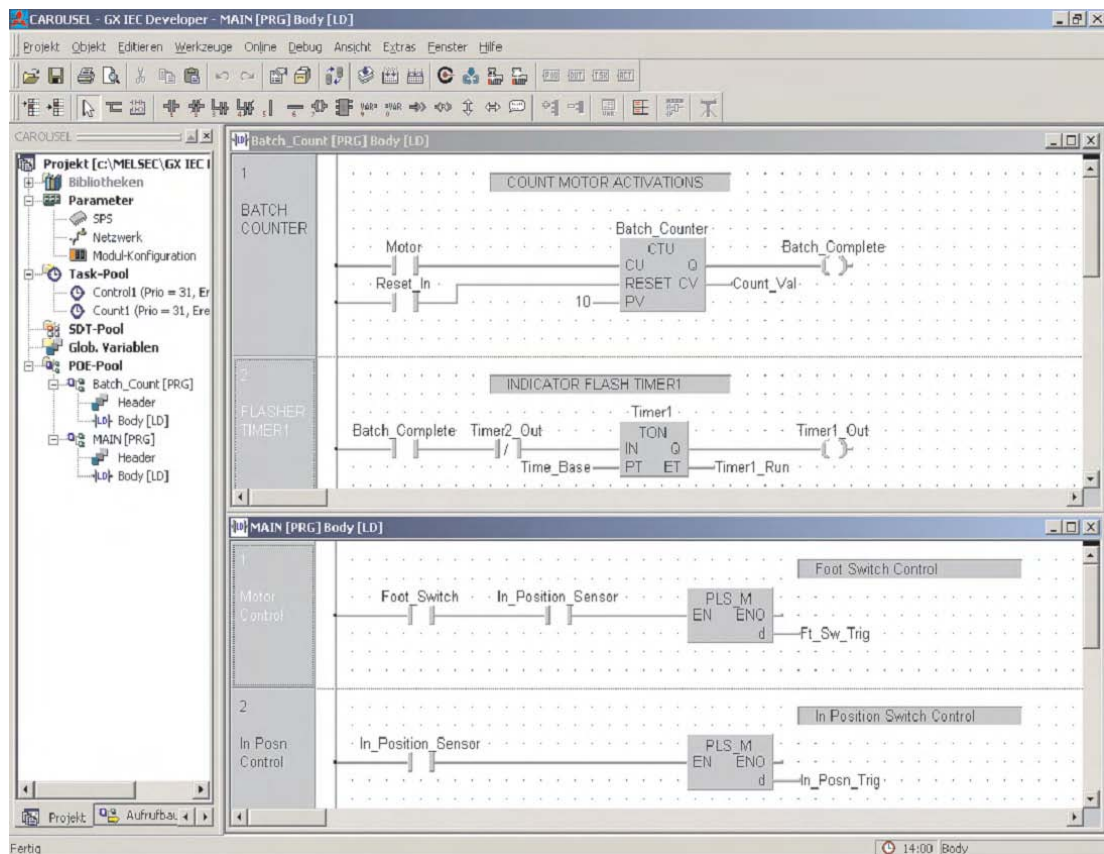
4.4.1 Beobachten in geteilten Fenstern

Um beide POEs des Projekts gleichzeitig zu beobachten, öffnen Sie bitte die BODYS beider Programme. Wählen Sie dann im Menü **Fenster** den Eintrag **Untereinander**.

HINWEIS

Wichtig: Wird der Monitormodus durch das Schaltfeld  aktiviert, wenn nur eine POE angewählt ist, gilt der Monitormodus nur für diese POE. Dadurch wird der Austausch von Daten für Programme verhindert, deren BODYS zwar geöffnet sind, die sich aber im Hintergrund befinden.

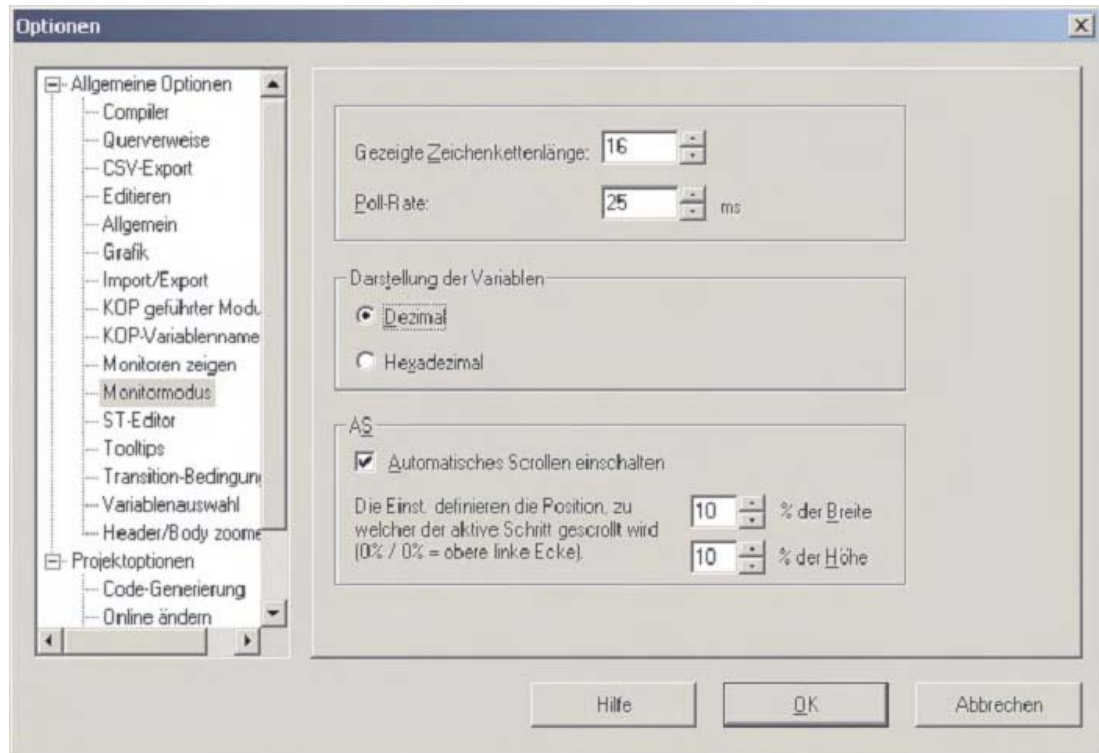
Um den Inhalt weiterer Fenster in den Monitormodus einzubeziehen, klicken Sie in dieses Fenster und wählen dann im Menü **Online** den Eintrag **Monitor starten**.



HINWEIS

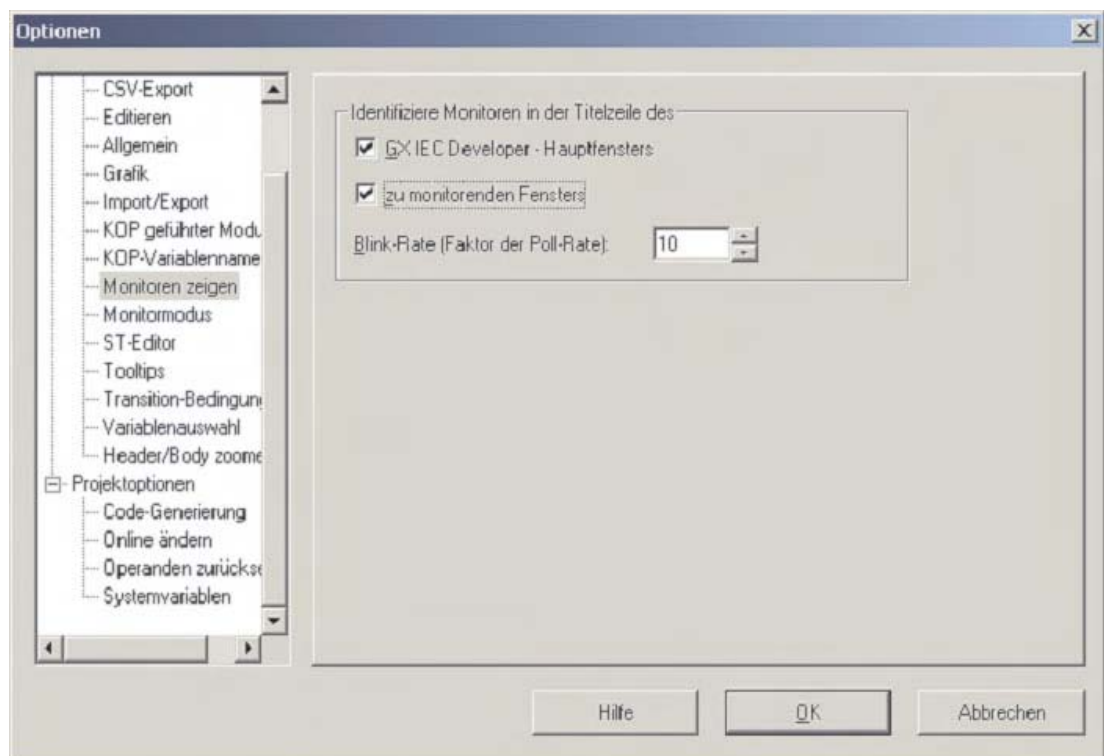
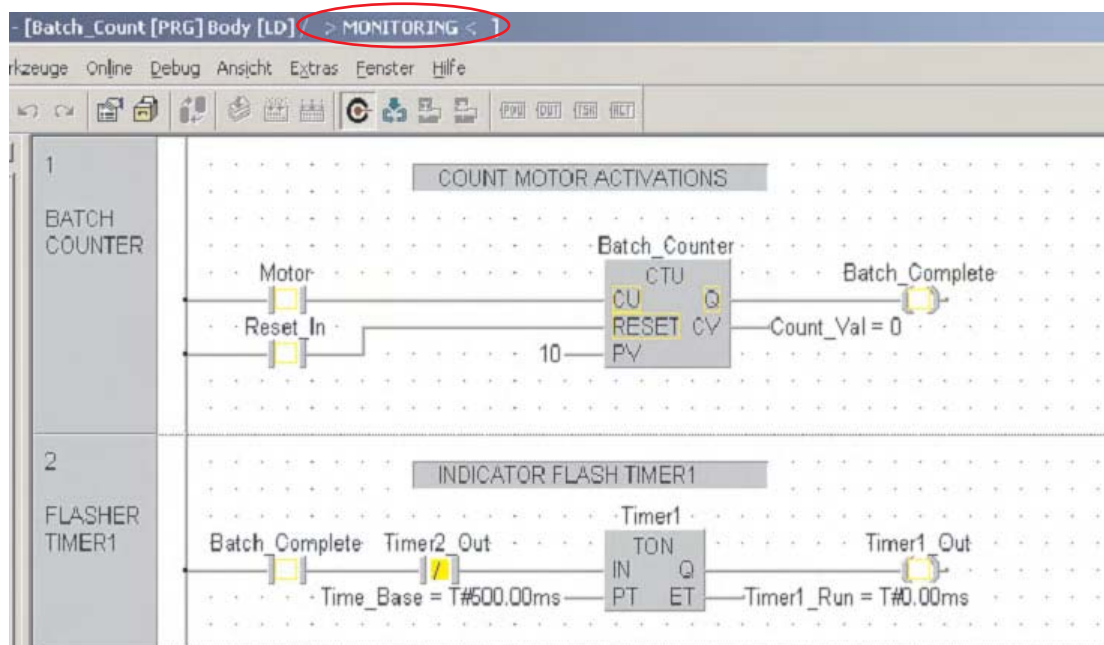
Durch die Handshake-Signale der seriellen Kommunikation kann die Anzeige der Monitorinformationen einige Sekunden verzögert werden.

Das Intervall für die Erfassung der Daten in der SPS kann im Menü **Extras** → **Optionen** → **Monitormodus** eingestellt und bei Bedarf verringert werden. Geben Sie den neuen Wert in das Eingabefeld **Poll-Rate** ein.



4.4.2 Erkennbarkeit des Monitormodus erhöhen

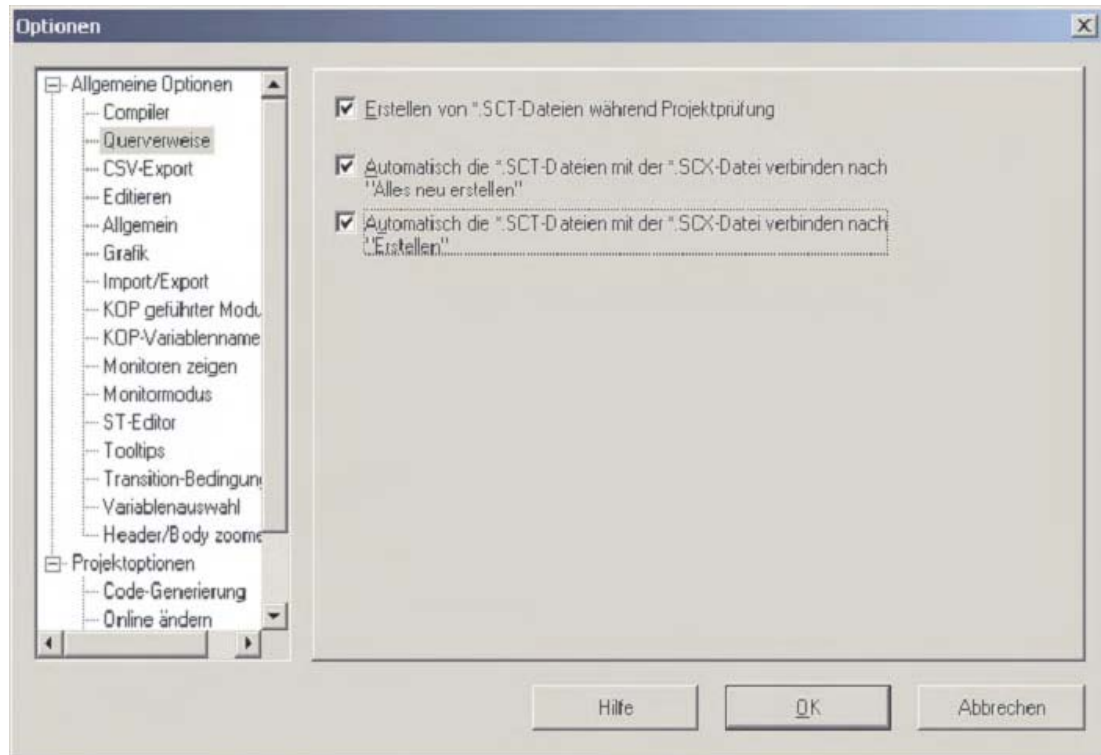
Zur Anzeige, dass der Monitormodus eingeschaltet ist, kann im Menü **Extras** → **Optionen** → **Monitoren zeigen** gewählt werden, ob in der Titelzeile des GX IEC Developers oder im beobachteten Fenster der blinkende Text „MONITORING“ eingeblendet wird. Die Frequenz, mit der der Text blinkt, kann hier ebenfalls eingestellt werden.



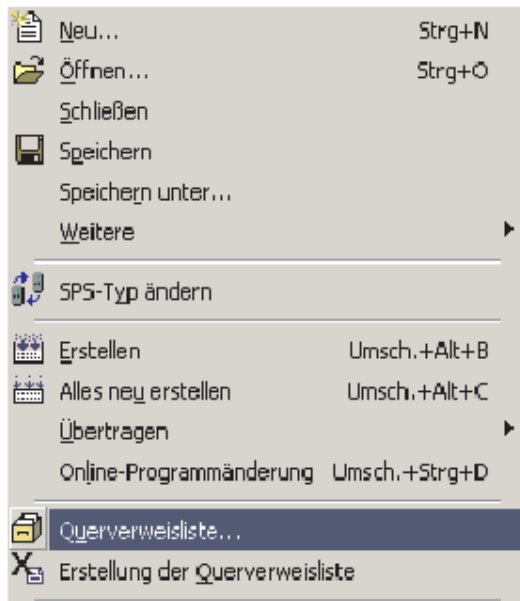
4.5 Querverweisliste


Eine Querverweisliste zeigt an, wo im Programm ein bestimmter Operand verwendet wird. So generieren Sie eine Querverweisliste:

- ① Wählen Sie im Menü **Extras** den Eintrag **Optionen** und dann **Querverweise**.
- ② Prüfen Sie, ob alle Optionen aktiviert sind, und konvertieren Sie anschließend das Projekt noch einmal in den Maschinencode.

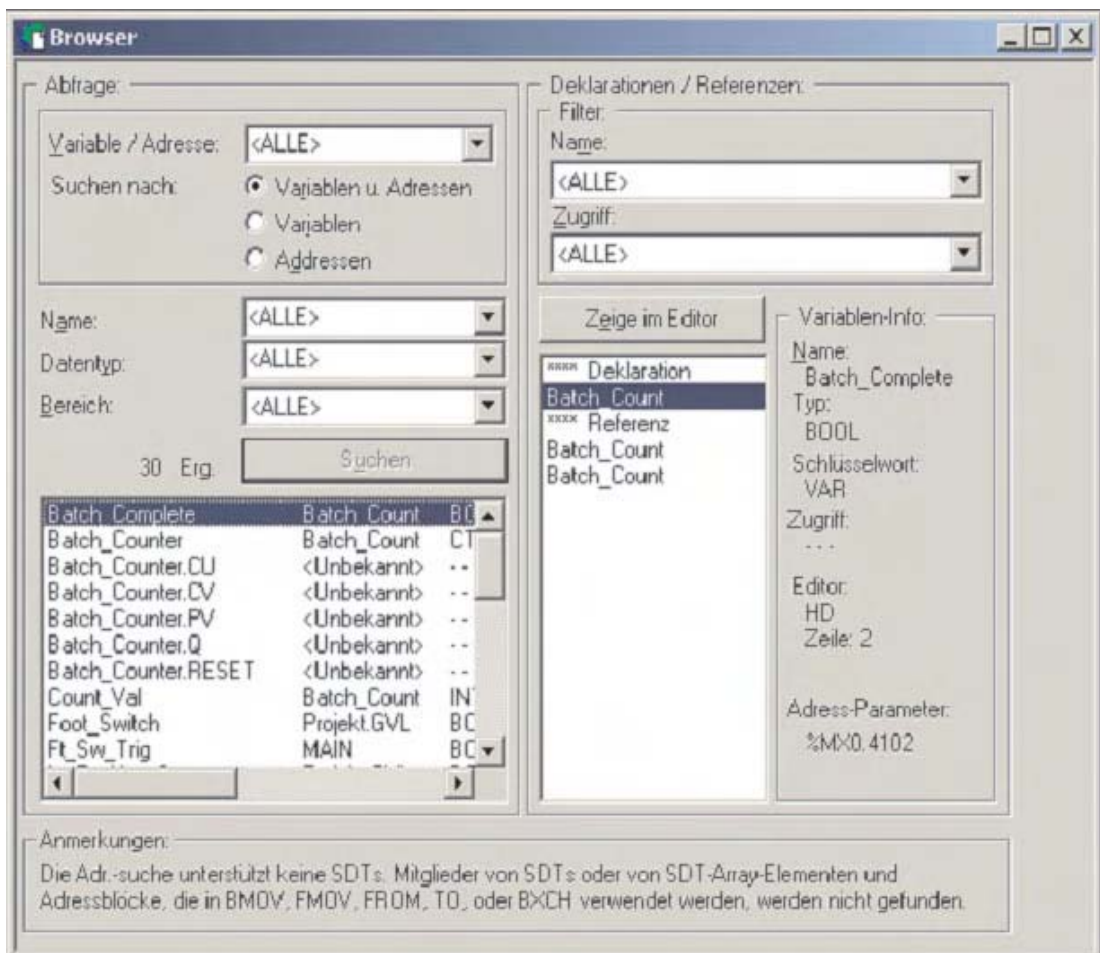


- ③ Wählen Sie dann im Menü **Projekt** den Eintrag **Erstellung der Querverweisliste**.



④ Öffnen Sie die Querverweisliste, entweder im Menü **Projekt** oder mit dem Schaltfeld  in der Werkzeugleiste.

⑤ Zur Anzeige der gesamten Liste klicken Sie bitte auf **Suchen**.

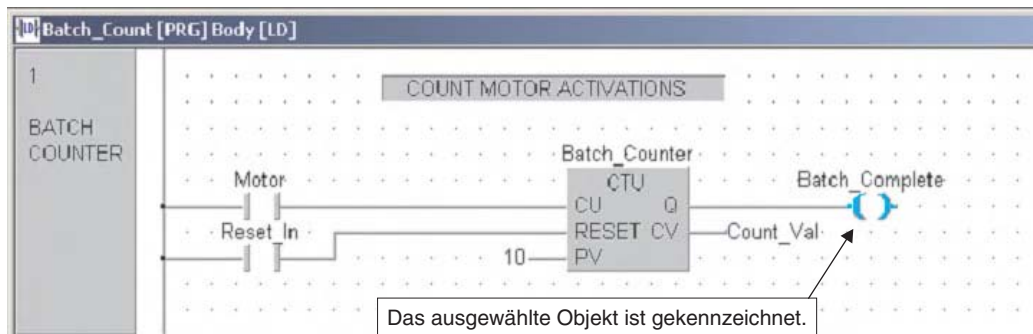


Bestimmte Variablen können durch Einträge in den entsprechenden Schaltfeldern gesucht werden. Weitere Informationen zu einer markierten Variablen werden auf der rechten Seite des Dialogfensters angezeigt.

Durch einen Klick auf das Schaltfeld **Zeige im Editor** wird der Header des in der rechten Spalte markierten Elements geöffnet. Zum Beispiel:

	Klasse	Bezeichner	Typ	Start	Kommentar
1	VAR	Batch_Complete	BOOL	FALSE	Batch Complete
2	VAR	Count_Val	INT	0	
3	VAR	Timer1	TON		Timer Base Timer1
4	VAR	Timer1_Out	BOOL	FALSE	
5	VAR	Timer2_Out	BOOL	FALSE	
6	VAR	Timer2	TON		Timer Base Timer2
7	VAR_CONSTANT	Time_Base	TIME	T#0.5s	
8	VAR	Timer1_Run	TIME	T#0s	
9	VAR	Timer2_Run	TIME	T#0s	

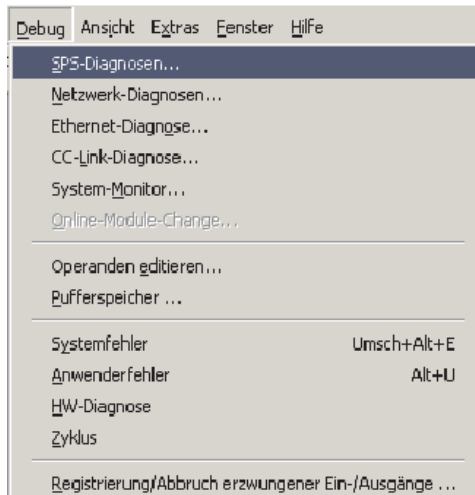
oder



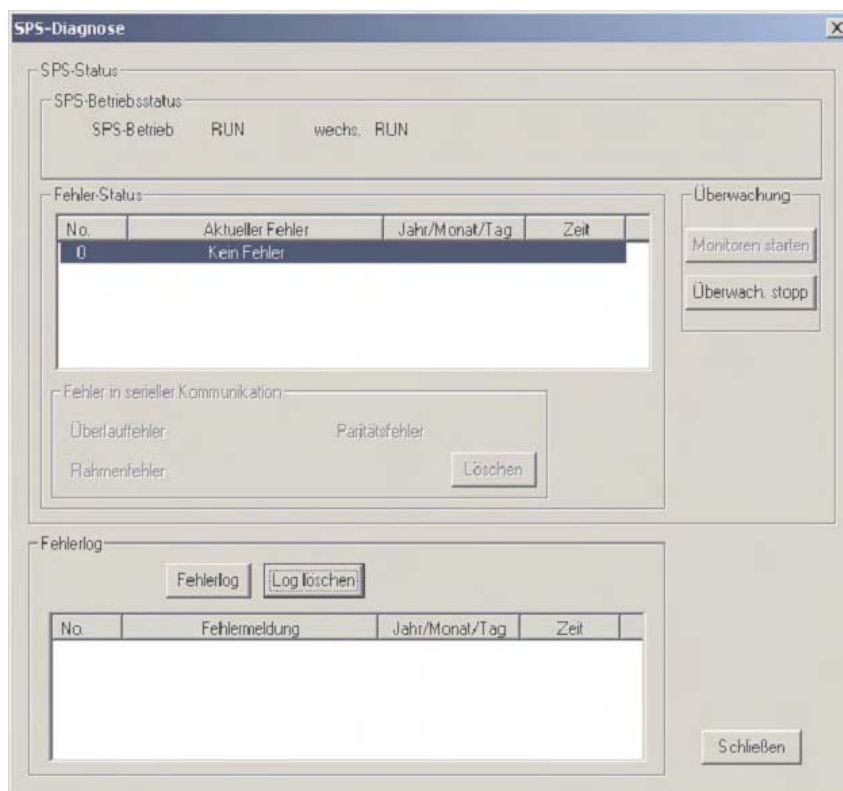
Mit der Druckfunktion des GX IEC Developer kann die Querverweisliste auch ausgedruckt werden.

4.6 SPS-Diagnose

GX IEC Developer bietet umfangreiche Diagnosefunktionen. Im Menü **Debug** stehen Funktionen zur Fehlersuche und -analyse zur Verfügung.



Zur Anzeige des Status der SPS oder von Fehlermeldungen rufen Sie im Menü **Debug** die **SPS-Diagnosen** auf. Dadurch wird das unten abgebildete Dialogfenster geöffnet.



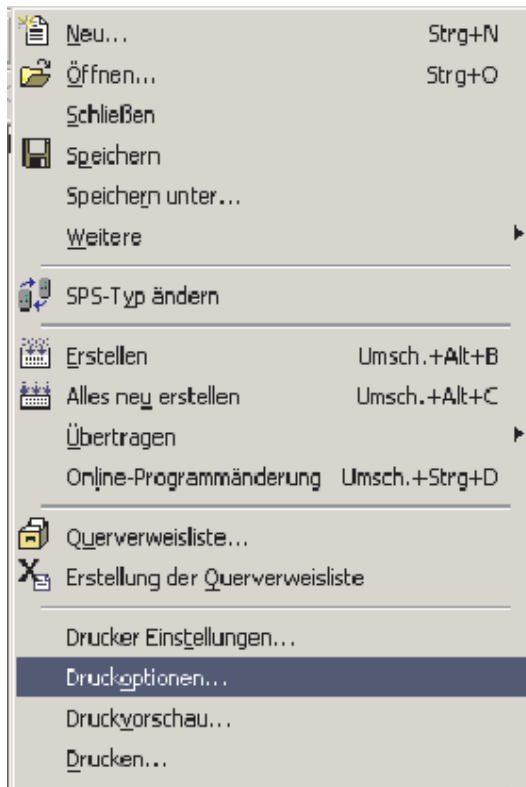
Anzeige der Fehlermeldungen im Klartext

Fehlermeldungen werden in der SPS codiert gespeichert. Im Dialogfenster **SPS-Diagnosen** werden die Fehlercodes ausgewertet und als Textmeldung angezeigt.

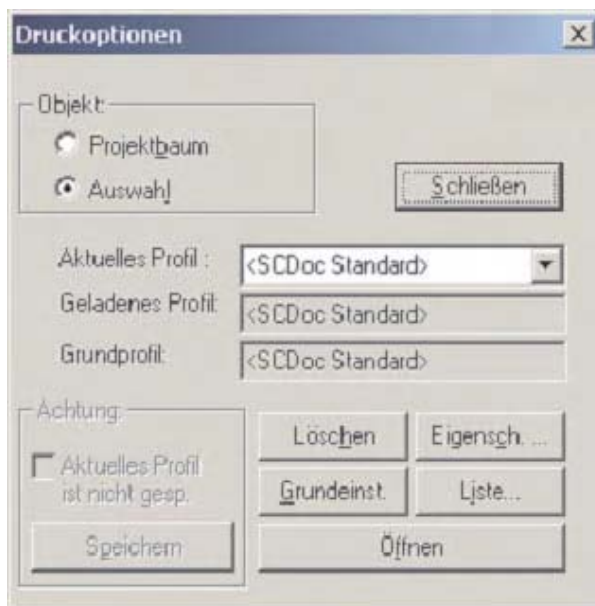
Vom Anwender können selbst Fehlermeldungen festgelegt und in einer Datei abgelegt werden (USER_ERR.TXT). Dadurch wird die Fehlersuche erleichtert. Die letzten acht Anwenderfehler werden in einem FIFO-Speicher eingetragen und nur gelöscht, wenn sie nicht mehr anstehen.

4.7 Dokumentation des Projekts

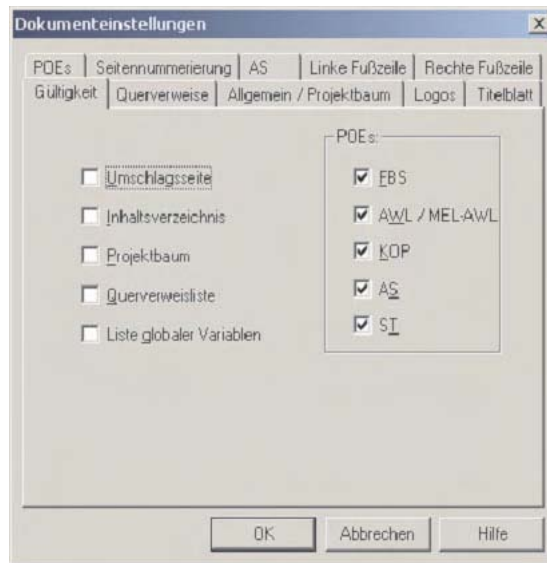
Im Menü **Projekt** können **Druckoptionen** gewählt werden:



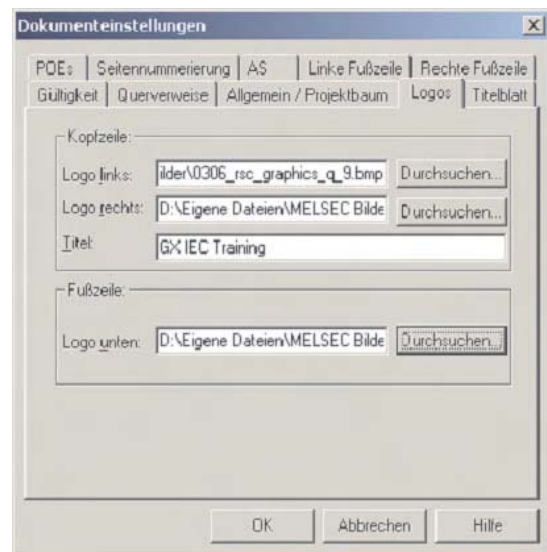
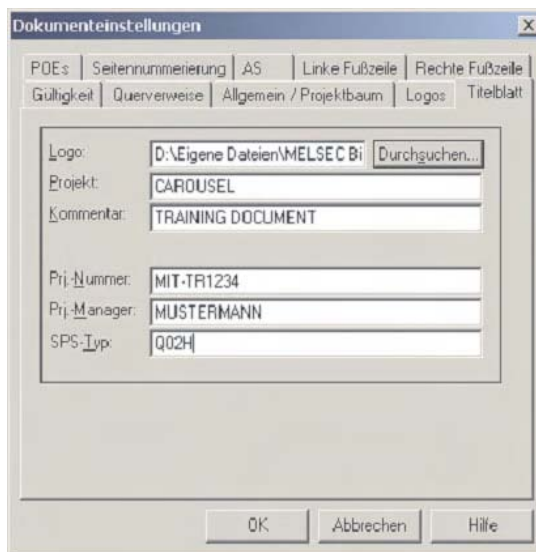
Unten ist das Dialogfenster **Druckoptionen** abgebildet. Hier können Einstellungen (Profile), z. B. von vorhergehenden Projekten, geladen werden. Sie können aber auch mit den Voreinstellungen arbeiten. Wählen Sie entweder **Projektbaum** für alle Elemente oder **Auswahl** für bestimmte Elemente und klicken Sie dann auf **Eigenschaften**.



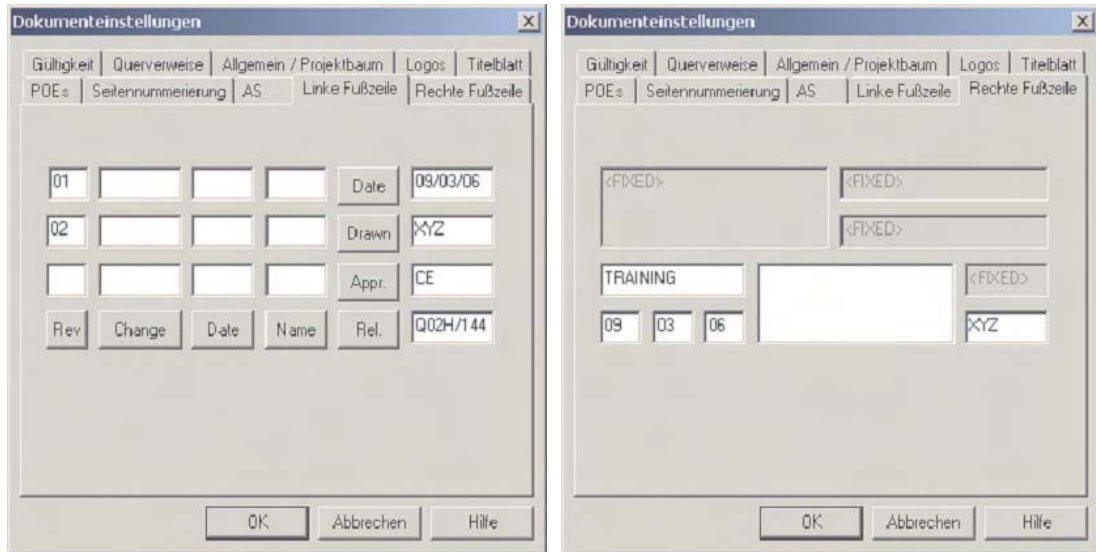
Dadurch wird das Dialogfenster **Dokumenteinstellungen** geöffnet. Nehmen Sie hier die erforderlichen Einstellungen vor. In diesem Beispiel wird nur COUNTER_FB_CE gedruckt, weil **Auswahl** gewählt wurde.



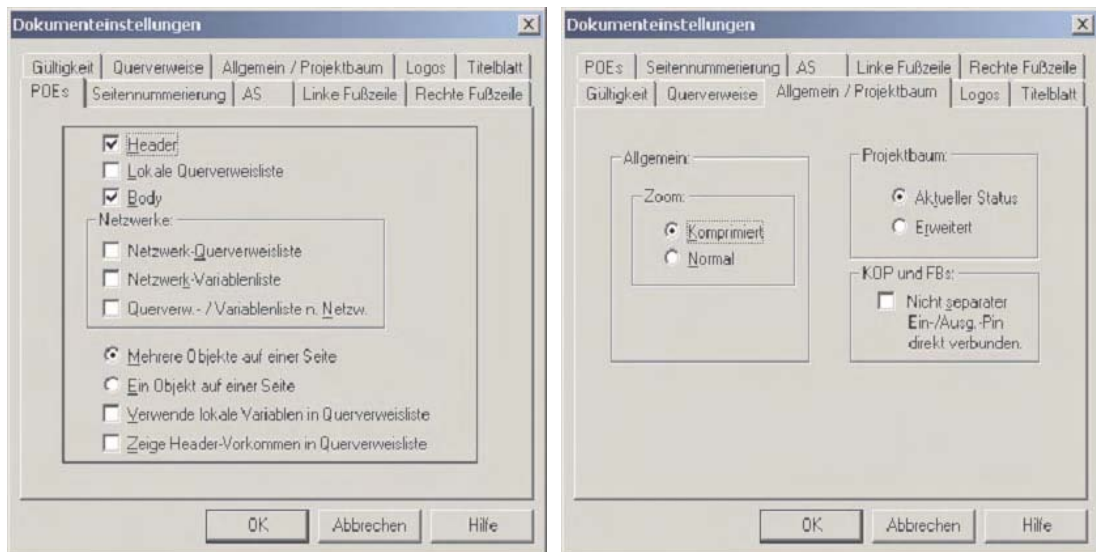
Anwenderdefinierte Logos oder Informationen können der Titelseite zugewiesen werden, wenn Sie auf die Karteikarte **Titelblatt** klicken. Die Kopf- und die Fußzeile können Sie mit eigenem Logos gestalten, nachdem Sie auf die Karteikarte **Logos** geklickt haben.



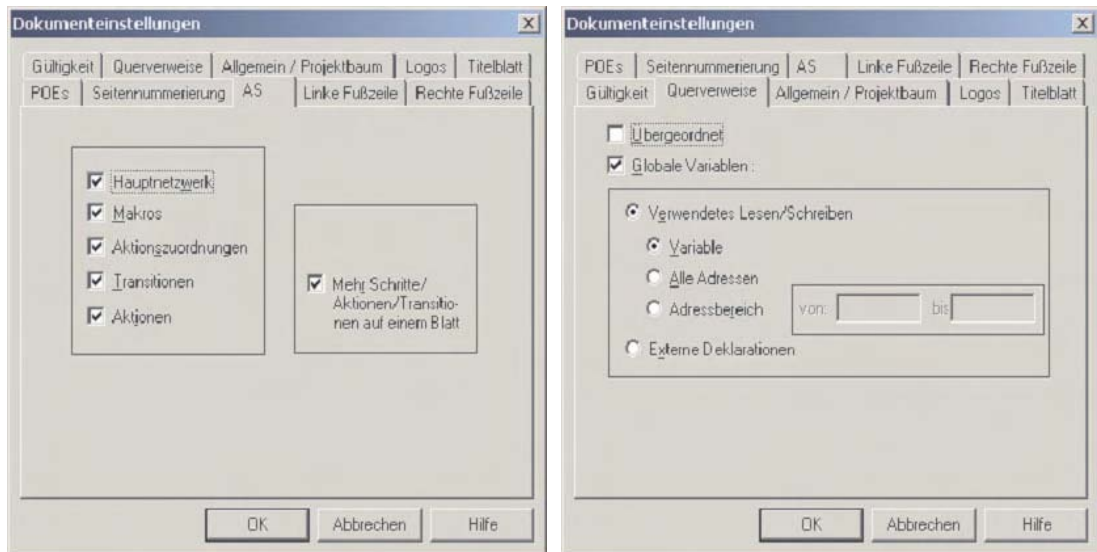
Für die linke und die rechte Fußzeile können unterschiedliche Einstellungen vorgenommen werden. Die Bezeichnungen der Felder der **linken Fußzeile** können Sie bei Bedarf ändern, indem Sie auf das Schaltfeld **Name** klicken.



Das Erscheinungsbild der POEs und des gesamten Projekts kann über die Karteikarten **POEs** bzw. **Allgemein/Projektbaum** eingestellt werden.



In den Ordnern **AS** und **Querverweise** finden Einstellmöglichkeiten für Programme in Ablaufsprache bzw. der Querverweisliste.



Die Einstellungen können gespeichert werden, indem ihnen ein Name gegeben und anschließend das Schaltfeld **Speichern** angeklickt wird. Sie können so jederzeit wieder geladen werden.



5 Programmbeispiel

5.1 QUIZMASTER

In diesem Beispiel werden behandelt:

- Programmierung von Timern
- Programmierung von Countern
- Logische Funktionen: Selbsthaltung – Verriegelungen – Verwendung von Merkern
- Funktionen: Reset (Zurücksetzen) und PLS (Flankenerfassung)

Beschreibung

Dieses Programm erfasst, welcher von vier Kandidaten bei einem Quiz als Erster seinen Taster betätigt hat und dadurch anzeigt, das er eine gestellte Frage beantworten möchte.

Nach der Betätigung des Tasters leuchtet nur die Leuchte, die diesem Kandidaten zugeordnet ist. Wenn andere Kandidaten ihre Taster betätigt später haben, bleiben deren Leuchten ausgeschaltet.

Aufgabe

- Erstellen Sie ein Kontaktplanprogramm für die SPS, das sicherstellt, dass nur eine Leuchte der Kandidaten eingeschaltet wird.
- Nachdem der Quizmaster eine Frage gestellt hat, betätigt er einen Start-Taster. Danach haben die Kandidaten 10 Sekunden Zeit, ihre Antwort-Taster zu betätigen.
- Während der Wartezeit wird die abgelaufene Zeit (0 -10 s) auf der Analoganzeige des Trainings-Racks dargestellt.
- Der Spielleiter kann das System durch einen separaten Taster jederzeit zurücksetzen.

Liste der verwendeten Ein- und Ausgänge

Eingänge

X10	-	Antwort-Taster für Spieler 1
X11	-	Antwort-Taster für Spieler 2
X12	-	Antwort-Taster für Spieler 3
X13	-	Antwort-Taster für Spieler 4
X14	-	Start-Taster (wird vom Quizmaster betätigt)
X15	-	Reset-Taster (wird vom Quizmaster betätigt)

Ausgänge

Y20	-	Antwort-Leuchte für Spieler 1
Y21	-	Antwort-Leuchte für Spieler 2
Y22	-	Antwort-Leuchte für Spieler 3
Y23	-	Antwort-Leuchte für Spieler 2
Y24	-	Anzeige: „Wartezeit ist abgelaufen“

Sondermodul

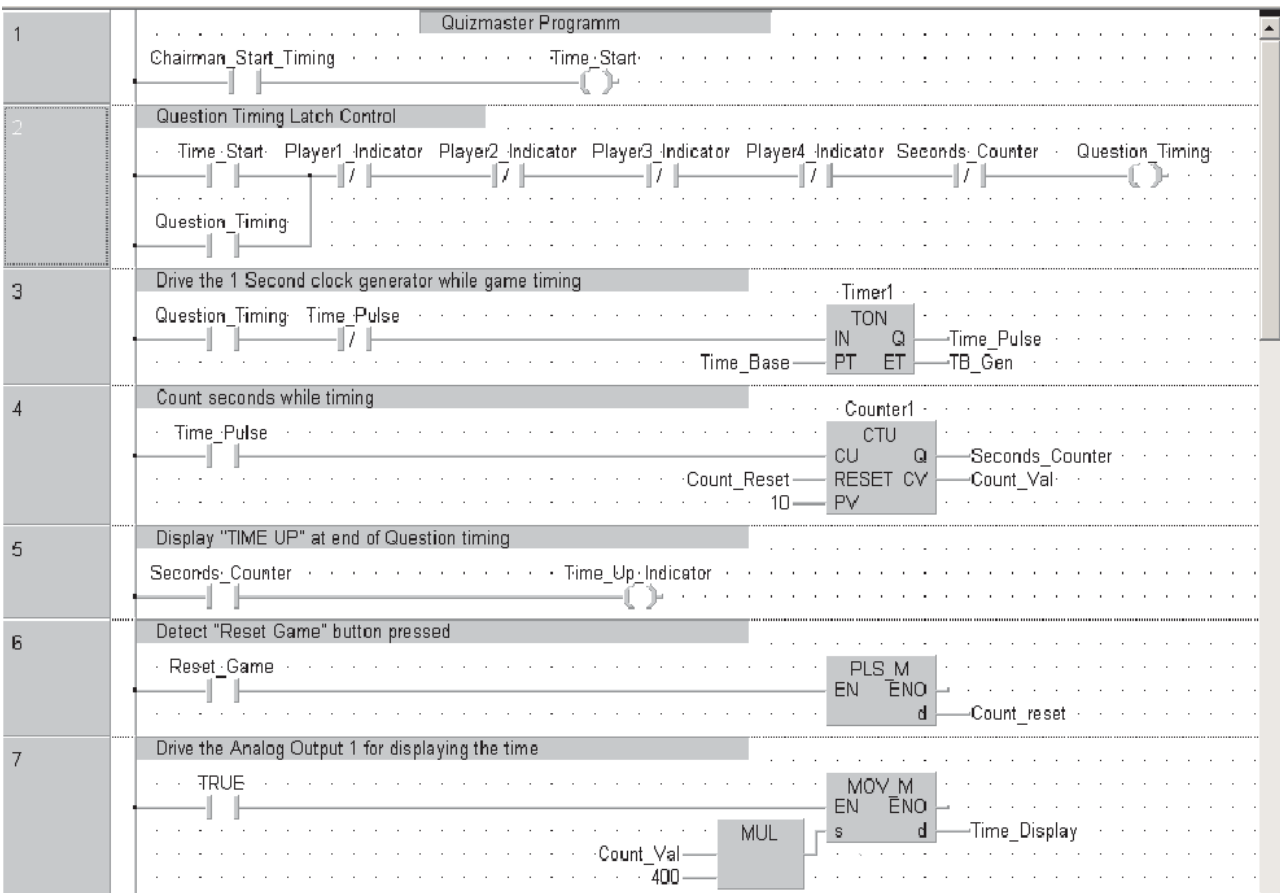
U4\G1	-	Adresse für Analogausgang 1 im Pufferspeicher des Analog-Ausgangsmoduls
-------	---	---

5.1.1 Vorgehensweise

- ① Generieren Sie ein neues Projekt und nennen Sie es „Quizmaster“.
- ② Geben Sie die folgenden Daten in die **Globale Variablenliste** ein:

	Klasse	Bezeichner	MIT-Adr.	IEC-Adresse	Typ	Start
0	VAR_GLOBAL	Player1_Response	X10	%IX16	BOOL	FALSE
1	VAR_GLOBAL	Player2_Response	X11	%IX17	BOOL	FALSE
2	VAR_GLOBAL	Player3_Response	X12	%IX18	BOOL	FALSE
3	VAR_GLOBAL	Player4_Response	X13	%IX19	BOOL	FALSE
4	VAR_GLOBAL	Chairman_Start_Timing	X14	%IX20	BOOL	FALSE
5	VAR_GLOBAL	Reset_Game	X15	%IX21	BOOL	FALSE
6	VAR_GLOBAL	Player1_Indicator	Y20	%QX32	BOOL	FALSE
7	VAR_GLOBAL	Player2_Indicator	Y21	%QX33	BOOL	FALSE
8	VAR_GLOBAL	Player3_Indicator	Y22	%QX34	BOOL	FALSE
9	VAR_GLOBAL	Player4_Indicator	Y23	%QX35	BOOL	FALSE
10	VAR_GLOBAL	Question_Timing	Y24	%QX36	BOOL	FALSE
11	VAR_GLOBAL	Time_Display	U4\G1	%MW14.4	INT	0
12	VAR_GLOBAL	Time_Up_Indicator			BOOL	FALSE

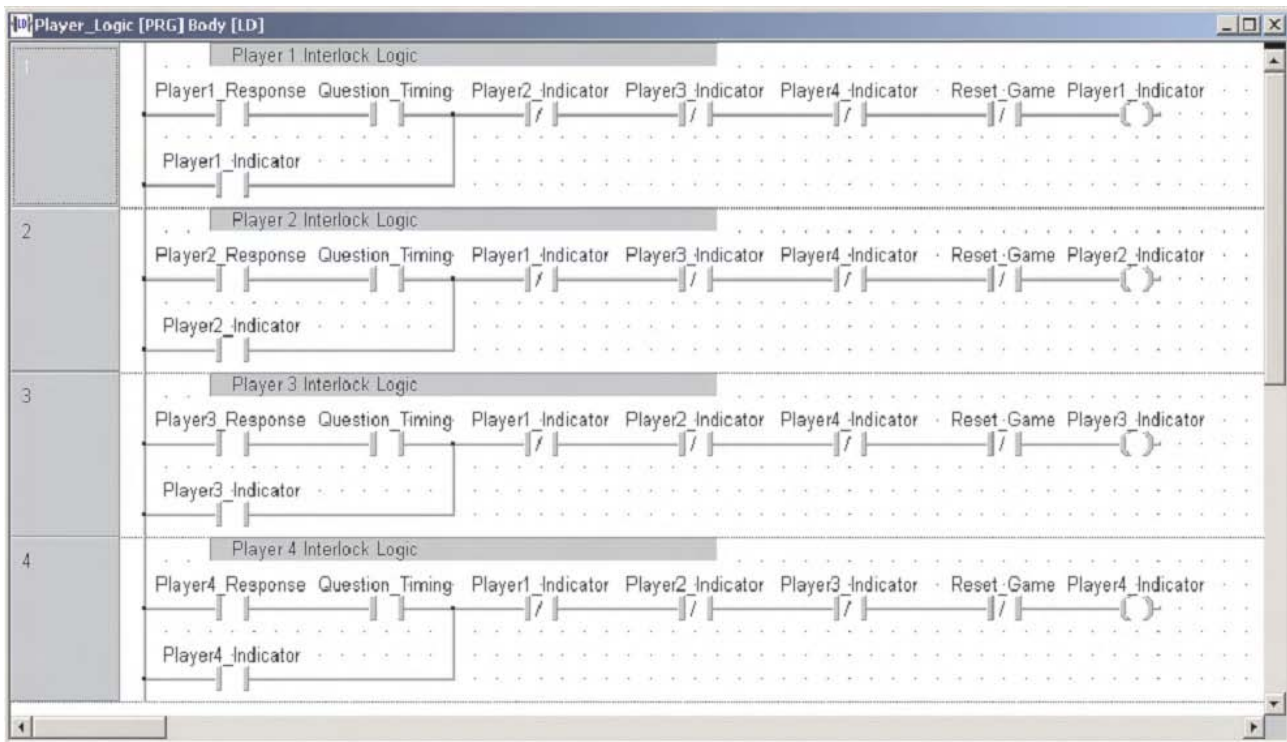
- ③ Legen Sie eine neue POE mit der Programmklasse **PRG** (Programm) an. Als Programmierer wählen Sie bitte **Kontaktplan**. Nennen Sie diese POE „Game_Control“.
- ④ Programmieren Sie diese POE so wie unten abgebildet:



Der Header der POE „Game_Control“ sollte so aussehen:

	Klasse	Bezeichner	Typ	Start	Kommentar
0	VAR	Time_Start	BOOL	... FALSE	
1	VAR	Time_Pulse	BOOL	... FALSE	
2	VAR	TB_Gen	TIME	... T#0s	
3	VAR_CONSTANT	Time_Base	TIME	... T#1s	
4	VAR	Count_Reset	BOOL	... FALSE	
5	VAR	Seconds_Counter	BOOL	... FALSE	
6	VAR	Count_Val	INT	... 0	
7	VAR	Timer1	TON	...	
8	VAR	Counter1	CTU	...	
9	VAR	Config_Analog	BOOL	... FALSE	

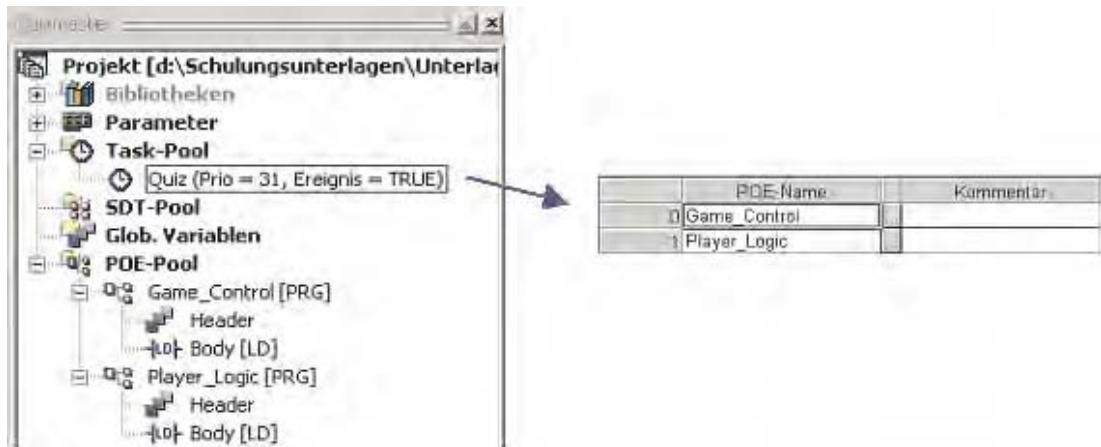
- ⑤ Legen Sie eine neue POE mit der Programmklasse **PRG** (Programm) an. Als Programmmeditor wählen Sie wieder **Kontaktplan**. Nennen Sie diese POE „Player_Logic“.
- ⑥ Programmieren Sie diese POE so wie unten abgebildet:



Der Header der POE „Player_Logic“ sollte so aussehen:

	Klasse	Bezeichner	Typ	Start	Kommentar
0	VAR			...	

- ⑦ Legen Sie im Task-Pool eine neue Task mit dem Namen „QUIZ“ an. Rufen Sie in dieser Task die beiden POEs „Player_Logic“ und „Game_Control“ auf.



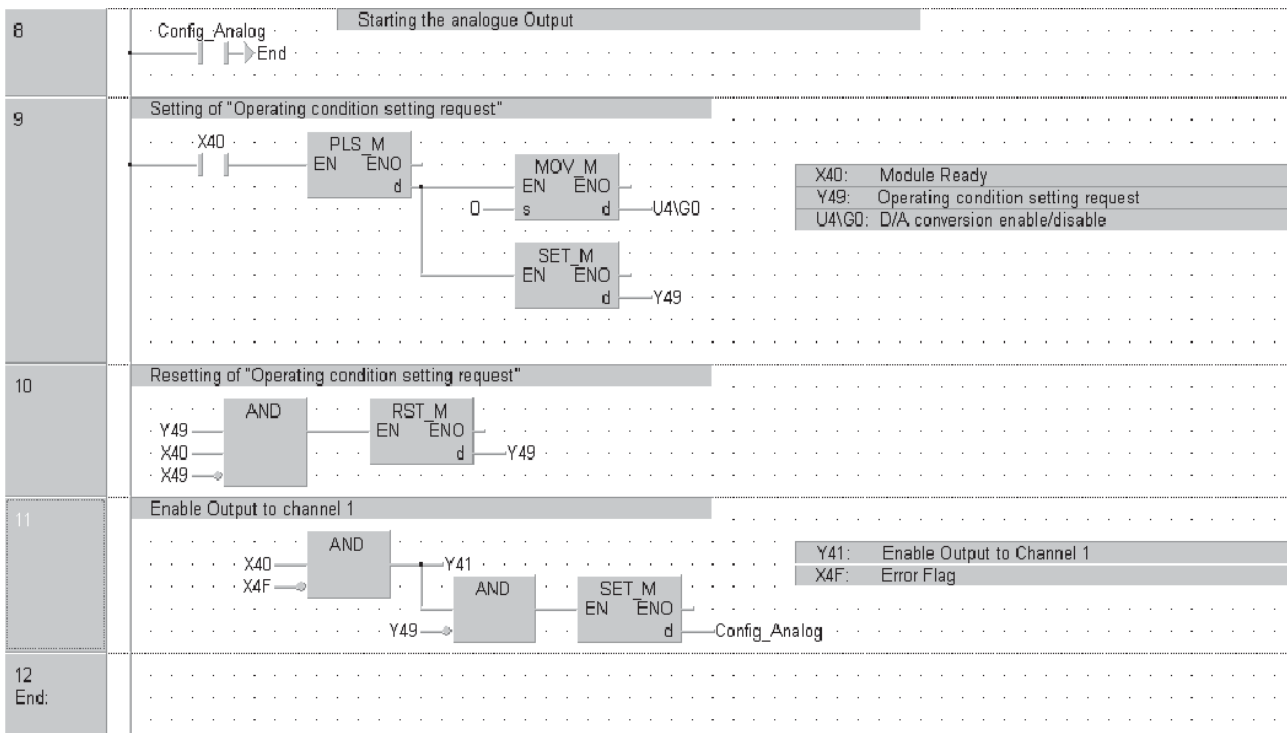
Initialisierung des Analog-Ausgabemoduls

- ⑧ Als nächstes müssen Einstellungen für das Analog-Ausgabemodul Q64DA vorgenommen werden. Klicken Sie dazu im Navigatorfenster doppelt auf **Parameter** und dann auf das Unterverzeichnis **SPS**. Klicken Sie dann auf den Reiter **E/A-Zuweisung**. Nachdem Sie auf **Schalterstellung** geklickt haben, geben Sie bitte die unten gezeigten Werte ein.

Steckpl.	SPS	Typ	Modellname	Adressen	StartW
0	SPS				
1	0(0-0)				
2	1(0-1)	Eingang		16 Adresse	0010
3	2(0-2)	Ausgang		16 Adresse	0020
4	3(0-3)	Sonder		16 Adresse	0030
5	4(0-4)	Sonder		16 Adresse	0040
6					
7					

Steckpl.	Typ	Modellname	Schalt. 1	Schalt. 2	Schalt. 3	Schalt. 4	Schalt. 5
0	SPS						
1	0(0-0)						
2	1(0-1)	Eingang					
3	2(0-2)	Ausgang					
4	3(0-3)	Intelli	5555			0000	
5	4(0-4)	Intelli	4444		0000	0000	
6							

- ⑨ Erweitern Sie die POE „Game_Control“ um die Analogausgabe an Kanal 1. Dieser Ausgang ist mit dem Anzeigeeinstrument verbunden.



5.1.2 Schritte zum Testen des Beispielprogramms „Quizmaster“

- ① Programmieren und kommentieren Sie das Beispielprogramm. Prüfen Sie das Programm mit Hilfe der Funktion in der Werkzeugleiste. Lassen Sie es dann vom GX IEC Developer in den Maschinencode konvertieren und speichern Sie es.
- ② Übertragen Sie das Projekt in die SPS des MELSEC System Q.
- ③ Aktivieren Sie den Monitormodus, um die Funktion des Programms zu beobachten.
- ④ Betätigen Sie kurz den am Eingang X14 angeschlossenen Schalter, um die Wartezeit zu starten.
- ⑤ Prüfen Sie, ob durch einem der Schalter an den Eingängen X10, X11, X12 und X13 die entsprechende Lampe eingeschaltet wird.
- ⑥ Prüfen Sie, ob die die Wartezeit von 10 Sekunden abläuft und ob die Zeit am Analoginstrument angezeigt wird.
- ⑦ Nach Ablauf der Wartezeit dürfen die Antwort-Taster der Kandidaten nicht mehr erfasst werden. Die LED, die das Ende der Wartezeit anzeigt, muss leuchten.
- ⑧ Durch den Reset-Taster (Eingang X15) müssen alle Ausgänge und Merker gelöscht werden können.

5.1.3 Erläuterung des Beispielprogramms „Quizmaster“

POE „Game_Control“

- Netzwerk 1
Bei der Betätigung des Start-Tasters wird durch die Anweisung „PLS_M“ die lokale Variable „Time_Start“ nur für einen SPS-Zyklus eingeschaltet.
- Netzwerk 2
Nach dem Start des Spiels wird „Question_Timing“ eingeschaltet. Diese Variable bleibt durch eine Selbsthaltung solange gesetzt, bis die Antwort-Lampe eines Spielers eingeschaltet wird oder die Wartezeit abgelaufen ist.
- Netzwerk 3
Wenn „Question_Timing“ gesetzt ist, beginnt der Timer für den 1-Sekunden-Takt zu laufen. Nach Ablauf dieser Zeit wird die Ausgangsvariable „Time_Pulse“ gesetzt. Diese unterbricht im nächsten Programmzyklus die Eingangsbedingung – der Timer wird angehalten. Dadurch wird auch die Ausgangsvariable „Time_Pulse“ zurückgesetzt. Dies wiederum startet den Timer im nächsten Zyklus erneut. Dies wiederholt sich, solange die Variable „Question_Timing“ gesetzt ist.
- Netzwerk 4
Das periodische Einschalten von „Time_Pulse“ wird von der CTU-Funktion „Count UP“ gezählt. Wenn „Time_Pulse“ zehn mal eingeschaltet wurde, sind 10 Sekunden vergangen und die Ausgangsvariable „Seconds_Counter“ wird eingeschaltet.
- Netzwerk 5
Durch die Ausgangsvariable „Seconds_Counter“ wird eine Leuchte eingeschaltet, die den Ablauf der Wartezeit anzeigt.
- Netzwerk 6
Die Betätigung des Reset-Tasters wird in einen Impuls gewandelt, der den Zähler im Netzwerk 4 zurücksetzt.

- Netzwerk 7

Die Variable TRUE ist immer eingeschaltet (logisch „1“). Dadurch wird der Zahlwert „Count_Val“ ständig mit dem Offset von 400 Einheiten/Volt multipliziert und in der Variablen „Time_Display“ an das Analog-Ausgangsmodul übertragen.

POE „Player_Logic“

- Netzwerke 1- 4

Diese Programmsequenzen dienen zur Verriegelung der Signale der einzelnen Spieler. Wenn zum Beispiel der 1. Spieler seinen Antwort-Taster zuerst betätigt, wird die Variable „Player1_Indicator“ eingeschaltet. In den weiteren Netzwerken ist diese Variable als Öffnerkontakt verarbeitet und sperrt dadurch Tasterbetätigungen der anderen Spieler. Ebenso sind die Ausgangsvariablen der anderen Spieler im Programm verknüpft. Die Kandidaten können eine Antwort nur anbieten, solange die Wartezeit läuft.

6 Funktion und Funktionsbaustein

Die folgende Tabelle zeigt einen Vergleich zwischen einer „Funktion“ und einem „Funktionsbaustein“.

Merkmal	Funktion	Funktionsbaustein
Speicherung interner Variablen	keine Speicherung	Speicherung
Instanzierung	Nicht erforderlich	Erforderlich
Ausgänge	Ein Ausgang	keinen, einen oder mehrere Ausgänge
Wiederholte Ausführung mit gleichen Eingangswerten	Gleicher Eingangswert ergibt immer den gleichen Ausgangswert	Gleicher Eingangswert ergibt nicht immer den gleichen Ausgangswert

6.1 Funktionen

- Funktionen gehören zum Befehlsvorrat der SPS
- Funktionen sind Unterprogramme, die mehrfach im Programm wie eine Programmieranweisung aufgerufen werden können.
- Funktionen sind in der Hersteller- und Standardbibliothek enthalten. Zum Beispiel ist TIMER_M ebenso eine Funktion aus dem Mitsubishi-Befehlsvorrat wie MOV_M und PLUS_M und in der Herstellerbibliothek enthalten.
- Funktionen können auch vom Anwender programmiert werden. So können geprüfte Programmteile leicht in Funktionen gewandelt werden.

So können zum Beispiel Berechnungen für Prozessgrößen als Funktionen in Bibliotheken gespeichert und dadurch mehrfach, mit verschiedenen Variablen, verwendet werden. Dies ist ähnlich der Verwendung z. B. einer MOV-Anweisung, nur mit dem Vorteil, dass die selbst programmierte Funktion anwenderspezifisch ist.

In vielen Programmen werden Berechnungen benötigt. Beispielsweise müssen häufig analoge Signale aufbereitet oder Messwerte skaliert werden. Und oft werden diese Kalkulationen mehrmals im Projekt angewendet. Durch eine anwenderprogrammierte Funktion lässt sich viel Zeit bei der Programmierung einsparen.

6.1.1 Beispiel: Programmierung einer Funktion

Ziel:

Programmieren Sie eine Funktion, die einen Wert in Grad Fahrenheit in Grad Celsius umrechnet.

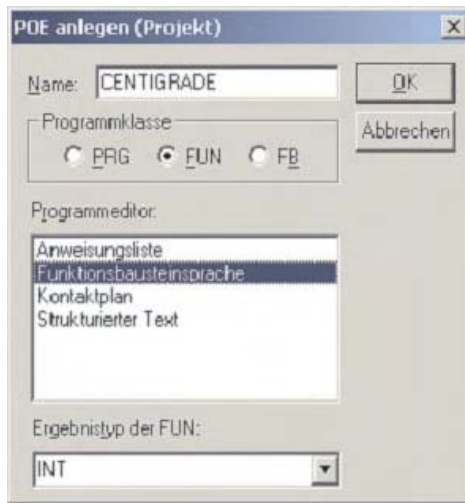
Die Formel dazu ist:

$$Grad_{Celsius} = \frac{(Grad_{Fahrenheit} - 32) \times 5}{9}$$

Die Funktion bekommt den Namen „CENTIGRADE“ (englisch für „Grad Celsius“) und die Eingangsvariable wird „Fahrenheit“ genannt.

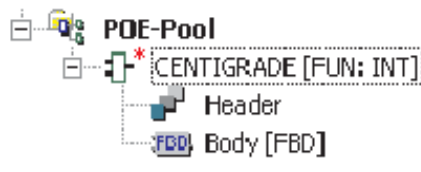
Vorgehensweise

- ① Legen Sie eine neue POE an und nennen Sie sie „CENTIGRADE“.

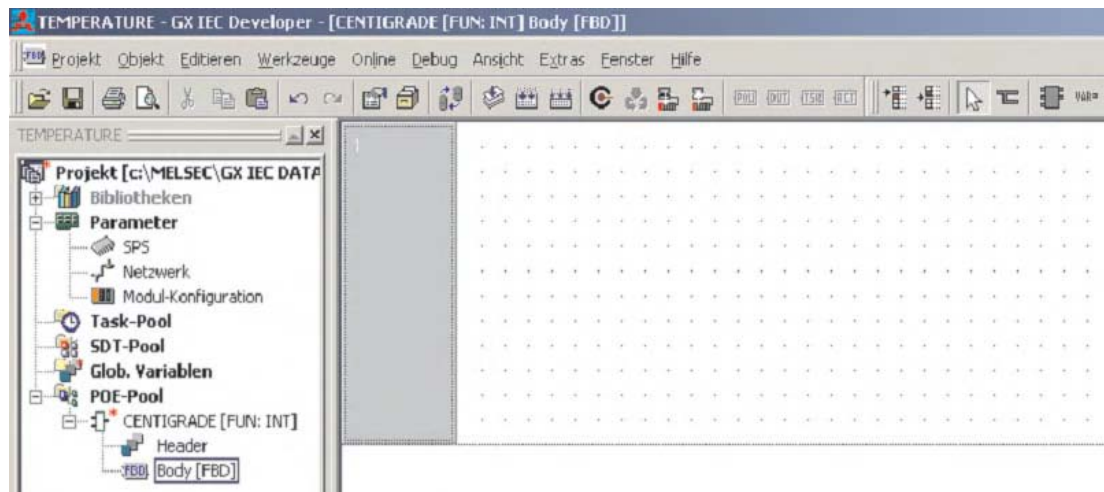


Dieses Mal wählen Sie als Programmklasse **FUN** anstatt wie bisher **PRG**. Diese Funktion wird in der **Funktionsbausteinsprache** programmiert. Als **Ergebnistyp der FUN** kann **INT** (Integer) übernommen werden.


Nachdem Sie auf OK geklickt haben, erscheint die Funktion „CENTIGRADE“ im POE-Pool.



- ② Klicken Sie doppelt auf den Body der Funktion, um ihn zu öffnen.

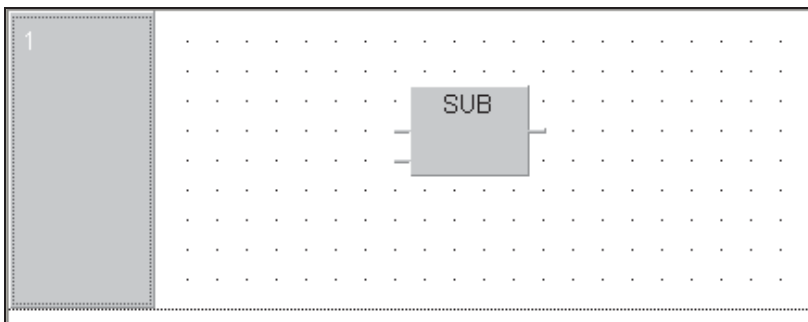


Programmierung

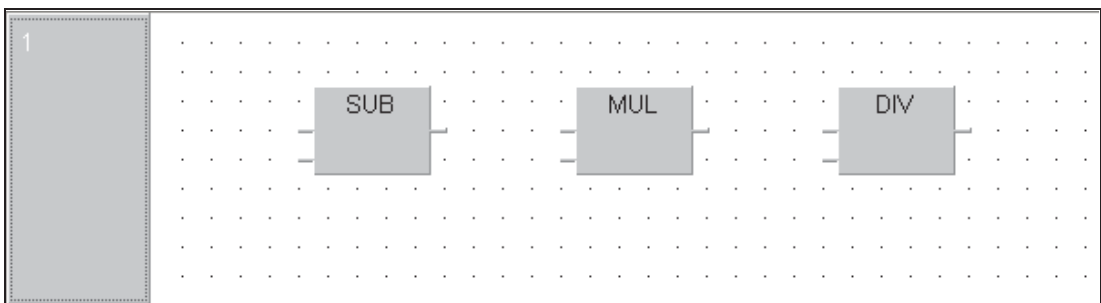
- ① Klicken Sie in der Werkzeugleiste auf das Schaltfeld  (Funktionsblock) und wählen Sie aus der Liste der Operatoren die Anweisung **SUB**.



- ② Klicken Sie auf **Anwenden** oder doppelt auf das ausgewählte Objekt und dann in den Body, um es abzulegen.



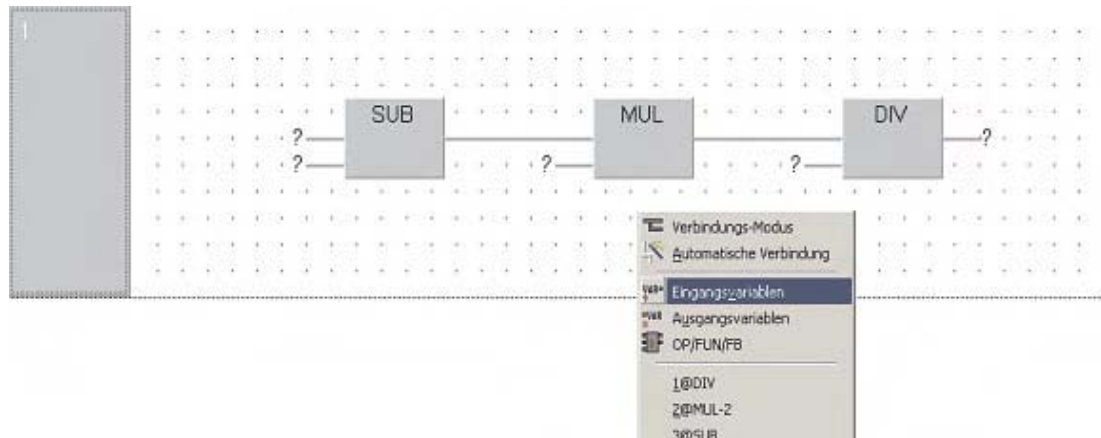
- ③ Wiederholen Sie diese Schritte, um die folgenden Objekte zu platzieren:




Deklaration der Variablen

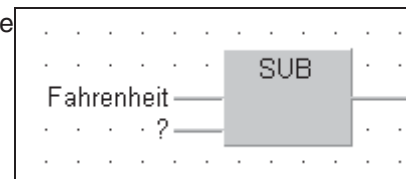
Es gibt verschiedene Möglichkeiten zur Deklaration von Variablen. In diesem Beispiel wird demonstriert, wie Variablen während der Programmierung deklariert werden können.

- ① Betätigen Sie im Netzwerk die rechte Maustaste. Aus dem Menü, das dann angezeigt wird, wählen Sie Ein- und Ausgangsvariablen und positionieren Sie wie unten gezeigt an die Anweisungen.

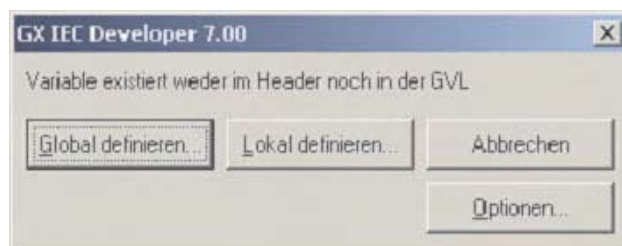


Mit den Schaltfeldern  können Sie Ein- und Ausgangsvariable auch in der Werkzeugleiste auswählen.

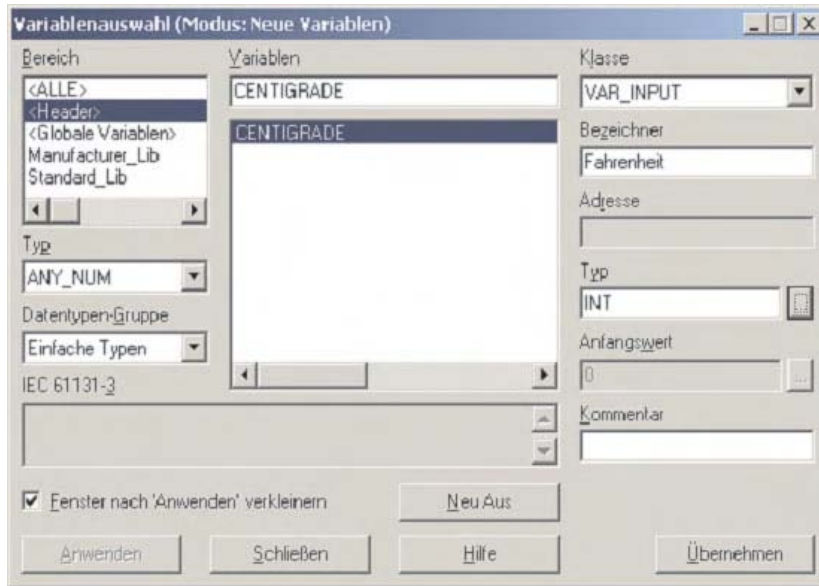
- ② Deklarieren Sie die Variable „Fahrenheit“, indem Sie sie einfach als Eingangsvariable eintippen.



Weil der Name dieser Variablen noch nicht im Header (Lokale Variablenliste) eingetragen wurde, wird eine Meldung angezeigt. Sie können diese Variable nun global oder lokal definieren.



- ③ Klicken Sie auf **Lokal definieren**.
- ④ Danach wird ein Dialogfenster zur Deklaration der Variable geöffnet. Tragen Sie in das Eingabefeld **Klasse** VAR_INPUT und in das Eingabefeld **Typ** INT ein.



HINWEISE

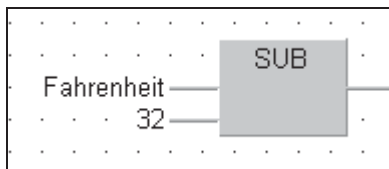
Als **Klasse** muss VAR_INPUT angegeben werden, weil dadurch beim Aufruf der Funktion in einem Programm die Eingabe von Werten in die Funktion ermöglicht wird. Das grafische Symbol der Funktion erhält durch einen VAR_INPUT einen Eingang an der linken Seite

Beachten Sie bitte, dass die Variable CENTIGRADE automatisch in den Header eingetragen wurde. Das kommt daher, weil der Name der Ausgangsvariablen einer Funktion identisch mit dem Namen der Funktion sein muss.

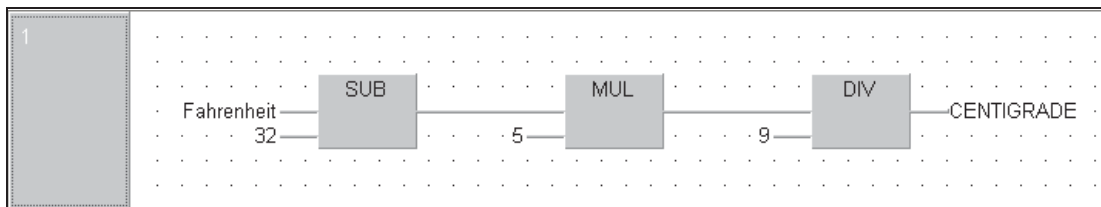
- ④ Klicken Sie auf Übernehmen. Dadurch wird die neue Variable in den Header der Funktion CENTIGRADE übernommen. Sie können dies prüfen, indem Sie den Header öffnen.

Festlegung von Konstanten

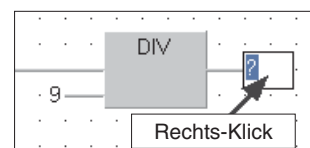
- ① Eine Konstante wird angegeben, indem sie in das Eingabefeld für eine Variable eingetragen wird:

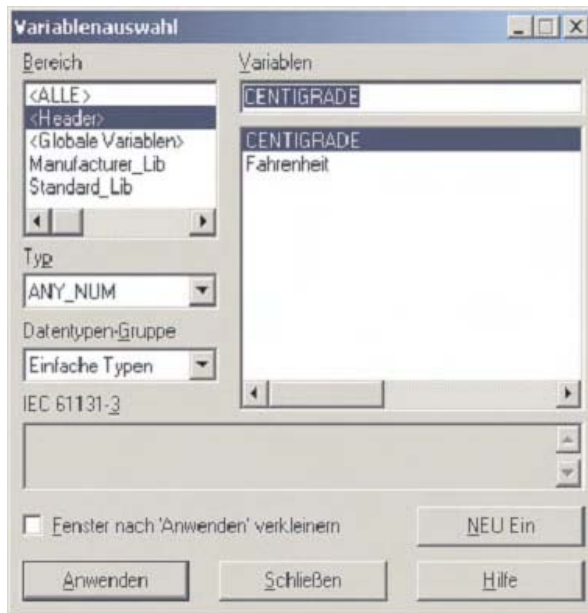


- ② Vervollständigen Sie das Programm der Funktion CENTIGRADE so wie nachfolgend gezeigt:



Tipp: Die Variable CENTIGRADE muss nicht eingetippt werden. Klicken Sie einfach mit der rechten Maustaste in das Eingabefeld (oder betätigen Sie die Taste F2). Im Dialogfenster **Variablenauswahl** klicken Sie dann entweder doppelt auf CELSIUS oder nur einmal und anschließend auf **Anwenden** (siehe folgende Abbildung).





CENTIGRADE wurde automatisch in den Header eingetragen, weil es der Name der Funktion ist. Deshalb muss CENTIGRADE auch als Ausgang der Funktion angegeben werden.

Zur Sicherheit können Sie den Header der Funktion CENTIGRADE prüfen, er sollte so aussehen:

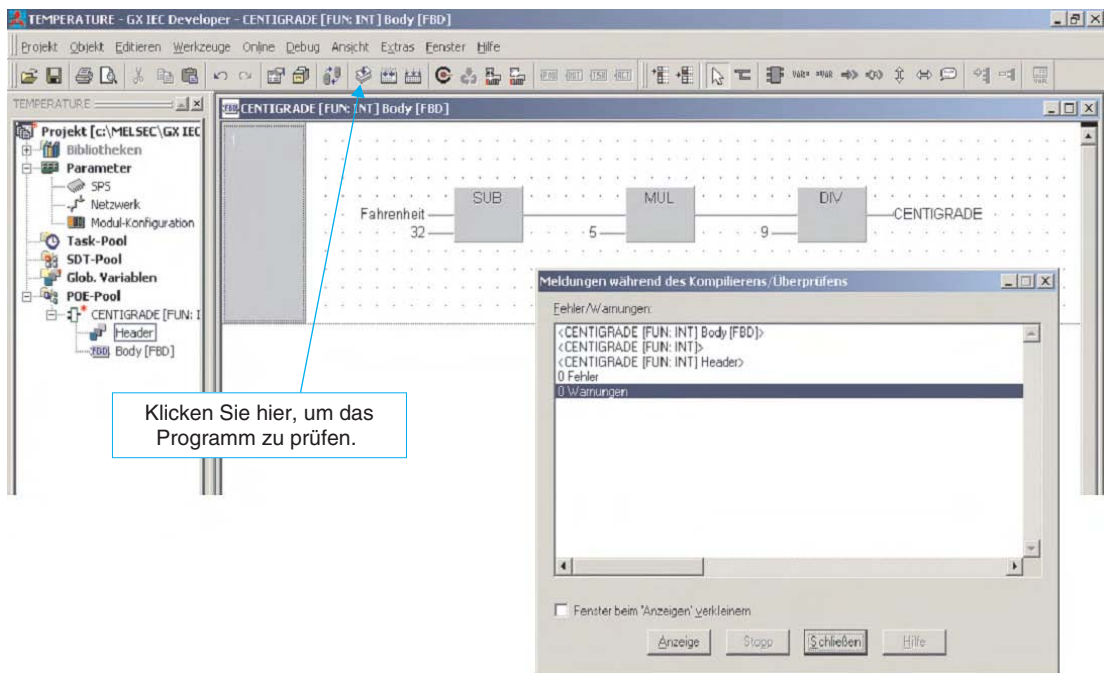
	Klasse	Bezeichner	Typ	Start	Kommentar
<input type="checkbox"/>	VAR_INPUT	Fahrenheit	INT	0	

HINWEIS

Die Variable „Fahrenheit“ kann auch vor der Programmierung direkt in den Header eingetragen und während der Programmierung ausgewählt werden.

Programm prüfen

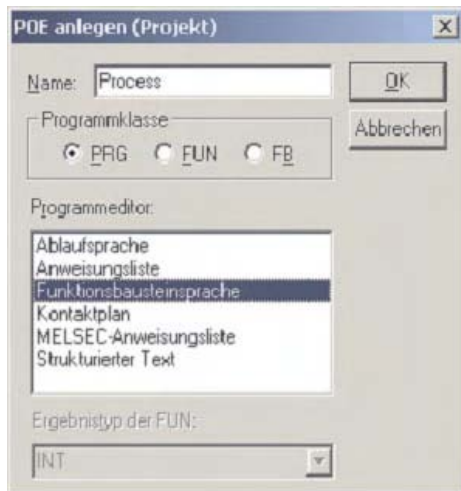
- ① Prüfen Sie das Netzwerk, es sollten keine Fehler oder Warnungen gemeldet werden.



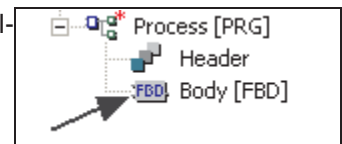
- ② Schließen Sie alle Dialog- und Arbeitsfenster.

Neue Programm-POE anlegen


- ① Legen Sie ein neue POE mit dem Namen „**Prozess**“ an. Als Programmklasse wählen Sie bitte **PRG** und als Programmiersprache wieder **Funktionsbausteinsprache**.

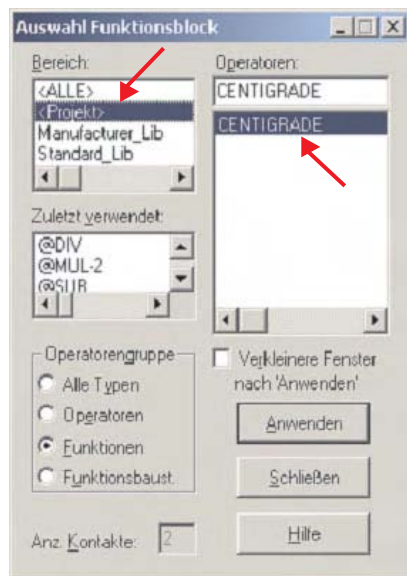


- ② Öffnen Sie den Body der POE „**Prozess**“ durch einen Doppelklick.



Aufruf einer anwenderdefinierten Funktion

- ① Klicken Sie wieder auf das Schaltfeld  (Funktionsblock) in der Werkzeugleiste. Dieses Mal wählen Sie aber bitte **Funktionen** und als Bibliothek **<Projekt>**. Bitte beachten Sie, dass die neue Funktion CENTIGRADE nun in der Liste der Operatoren angezeigt wird.

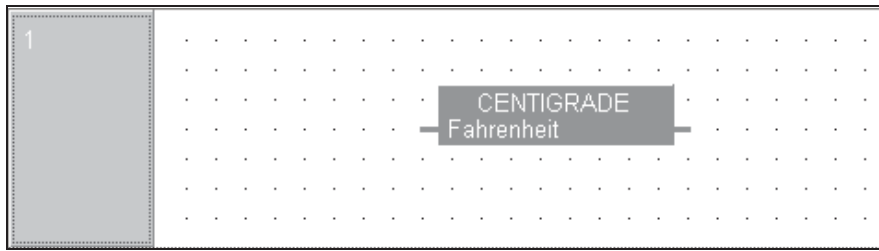


- ② Klicken Sie auf CENTIGRADE und anschließend auf **Anwenden**.

HINWEIS

Das Dialogfenster **Auswahl Funktionsblock** wird nach einem Klick auf **Anwenden** automatisch minimiert, wenn die Option **Verkleinere Fenster nach Anwenden** aktiviert ist. Ohne diese Option bleibt das Dialogfenster geöffnet. Wählen Sie eine dieser Möglichkeiten entsprechend Ihren Arbeitsgewohnheiten.

Die Funktion wird in den Body der POE übernommen:



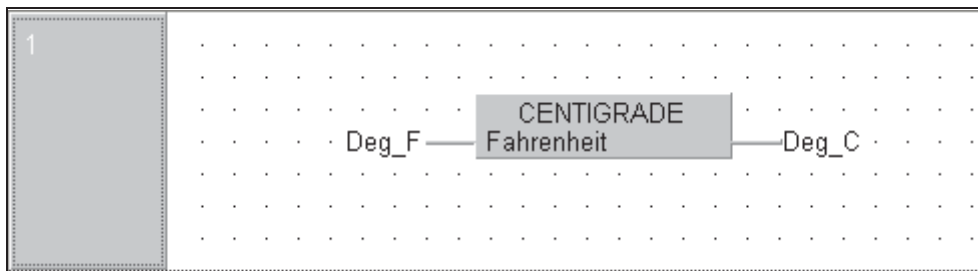
Zuweisung von Globalen Variablen

Nun können der Funktion Variablen zugewiesen werden.

① Tragen Sie in die Globale Variablenliste die folgenden Variablen ein:

Globale Variablenliste							
	Klasse	Bezeichner	MIT-Adr.	IEC-Adress	Typ	Start	
0	VAR_GLOBAL	Deg_F	D0	%MWO.0	INT	0	
1	VAR_GLOBAL	Deg_C	D1	%MWO.1	INT	0	

Weisen Sie diese Variablen der Funktion zu:



② Öffnen Sie die Task „Main“.

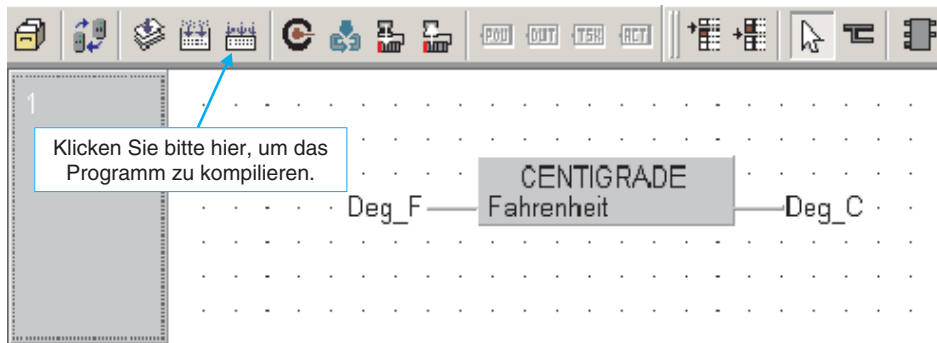


③ Rufen Sie in dieser Task die POE „Process“ auf.

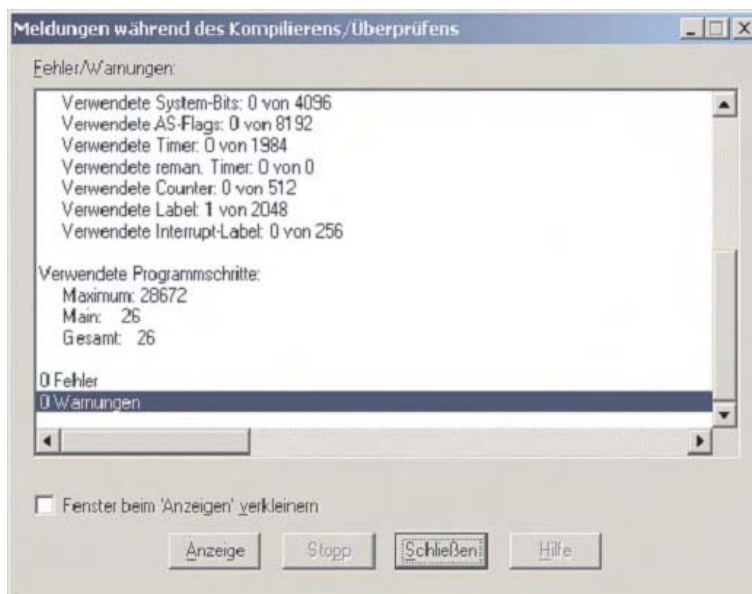
	POE-Name	Kommentar
0	Process	...

Wandlung des Programms in den Maschinencode

Konvertieren Sie das Programm, indem Sie in der Werkzeugleiste auf das Schaltfeld **Alles neu erstellen** klicken:




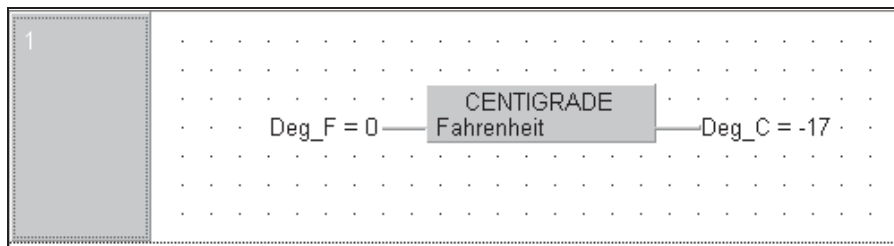
Es dürfen keine Fehler oder Warnungen angezeigt werden.



Falls doch ein oder mehrere Fehler aufgetreten sind, klicken Sie auf die Fehlermeldung und anschließend auf **Anzeige**. Dadurch wird der Fehler im Programm markiert. Korrigieren Sie dann das Programm.

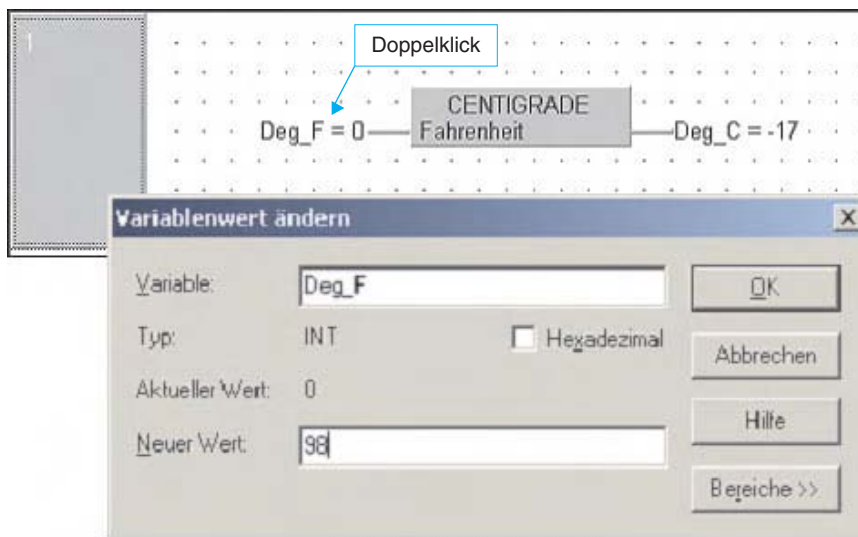
Programm im Monitormodus überprüfen

- ① Übertragen Sie das Programm in die SPS und aktivieren Sie den Monitormodus mit dem Schaltfeld  in der Werkzeugleiste:



- ② Die Werte von Variablen können im Monitormodus verändert werden. In diesem Fall wird der Wert der Eingangsvariablen „Deg_F“ angepasst.

Klicken Sie doppelt auf die Eingangsvariable und geben Sie danach den gewünschten Wert in das Dialogfenster **Variablenwerte ändern** ein.



Hinweis: 100 °F entsprechen 37 °C (genau genommen 37,7 °C)

6.1.2 Verarbeitung von Gleitkommazahlen

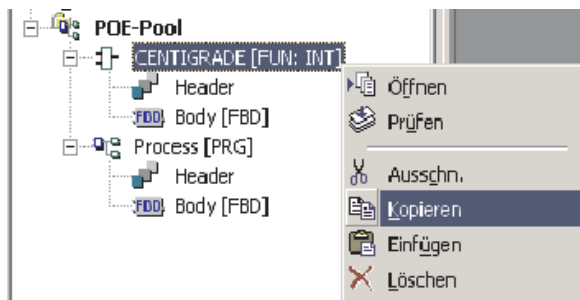
Die anwenderdefinierte Funktion CENTIGRADE kann zur Zeit nur ganzzahlige 16-Bit Werte im Bereich von -32768 bis +32767 verarbeiten. Dies entspricht der Voreinstellung beim Anlegen von Funktionen. Im nächsten Beispiel wird die Funktion CENTIGRADE so verändert, das auch Gleitkommazahlen vom Datentyp REAL umgerechnet werden können*.

* Gleitkommazahlen können nicht von allen CPUs verarbeitet werden.

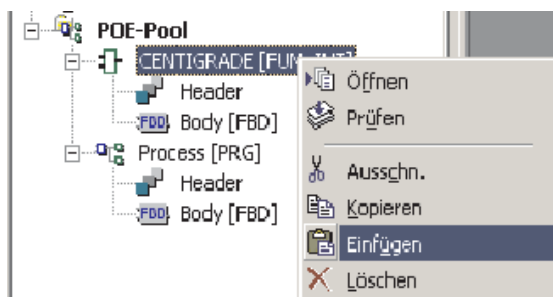
Kopieren einer Funktion

Die bestehende Funktion „CENTIGRADE“ wird kopiert und als „CENTIGRADE1“ gespeichert.

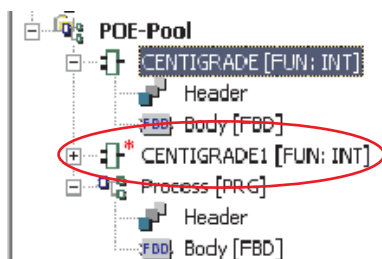
- ① Klicken Sie mit der rechten Maustaste im POE-Pool auf CENTIGRADE. Wählen Sie aus dem Menü, das dann angezeigt wird, **Kopieren**.



- ② Klicken Sie dann mit der rechten Maustaste auf **POE-Pool**. Wählen Sie nun **Einfügen** aus dem Menü.

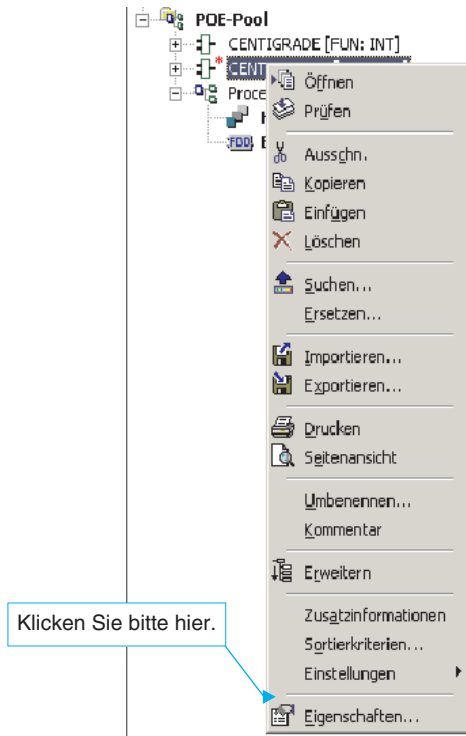


Das System fügt eine Kopie der Funktion CENTIGRADE in den POE-Pool eine und ändert den Namen automatisch in CENTIGRADE1.

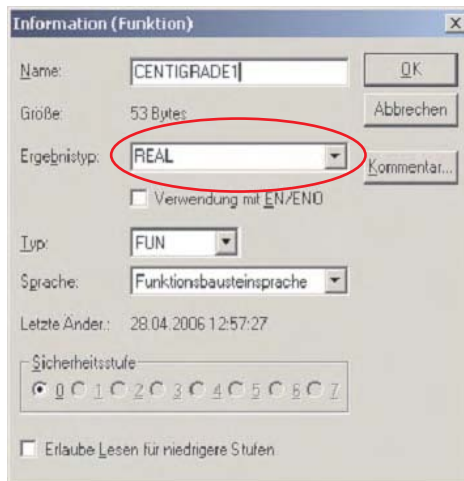


Ergebnistyp der Funktion ändern

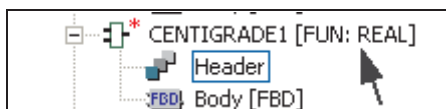
- 1 Klicken Sie mit der rechten Maustaste auf die soeben erzeugte Funktion CENTIGRADE1 und wählen Sie **Eigenschaften**.



- 2 Dadurch wird das Dialogfenster **Information (Funktion)** geöffnet. Ändern Sie hier den **Ergebnistyp** in REAL.



Im Navigatorfenster wird hinter dem Namen der Funktion der Ergebnistyp angezeigt. Für CENTIGRADE1 sollte nun REAL angegeben werden:



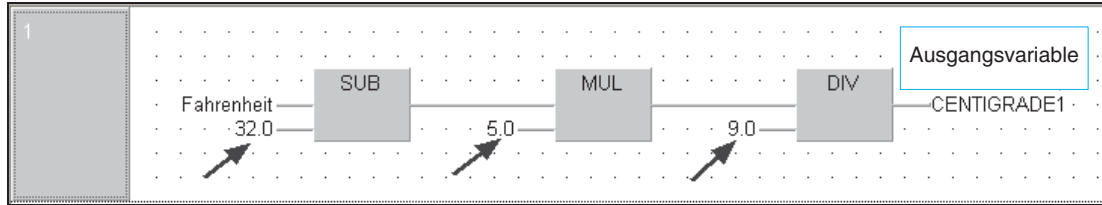
- 3 Ändern Sie im Header der Funktion CENTIGRADE1 die Variable „Fahrenheit“ in den Typ „REAL“:

	Klasse	Bezeichner	Typ	Start	Komr
0	VAR_INPUT	Fahrenheit	REAL	0.0	

Konstante in Gleitkommazahlen ändern

- ① Öffnen Sie den Body der Funktion CENTIGRADE1 und geben Sie an Stelle der ganzzahligen Konstanten Werte mit Vor- und Nachkommastellen an (Zum Beispiel 32→ 32.0).

Ändern Sie auch die Ausgangsvariable von CENTIGRADE1 in CENTIGRADE1.



- ② Schließen Sie alle Editoren und sichern Sie alle Änderungen.

Aufruf der Gleitkomma-Funktion CENTIGRADE1 in der POE „Process“

- ① Erweitern Sie die Liste der Globalen Variablen um zwei Einträge:

	Klasse	Bezeichner	MIT-Adr.	IEC-Adress	Typ	Start
0	VAR_GLOBAL	Deg_F	D0	%MWD.0	INT	0
1	VAR_GLOBAL	Deg_C	D1	%MWD.1	INT	0
2	VAR_GLOBAL	Deg_F_Real	D2	%MDD.2	REAL	0.0
3	VAR_GLOBAL	Deg_C_Real	D4	%MDD.4	REAL	0.0

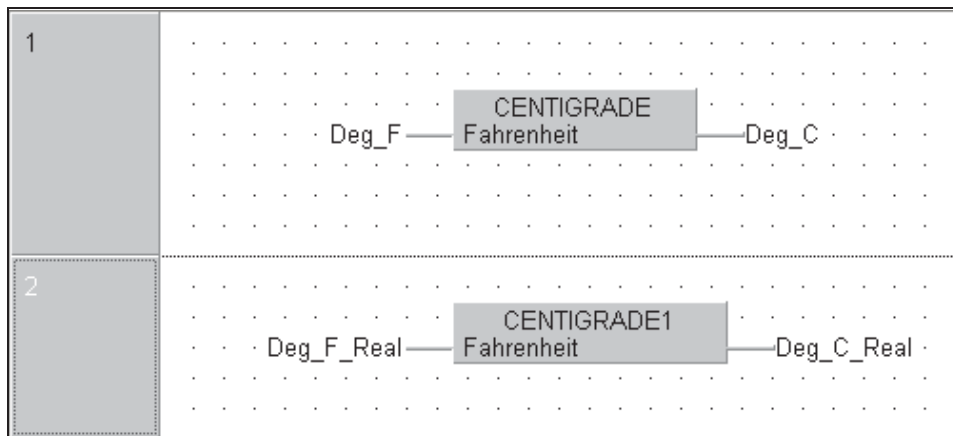
- ② Öffnen Sie den Body der POE „Process“ und fügen Sie die Funktion CENTIGRADE1 ein.




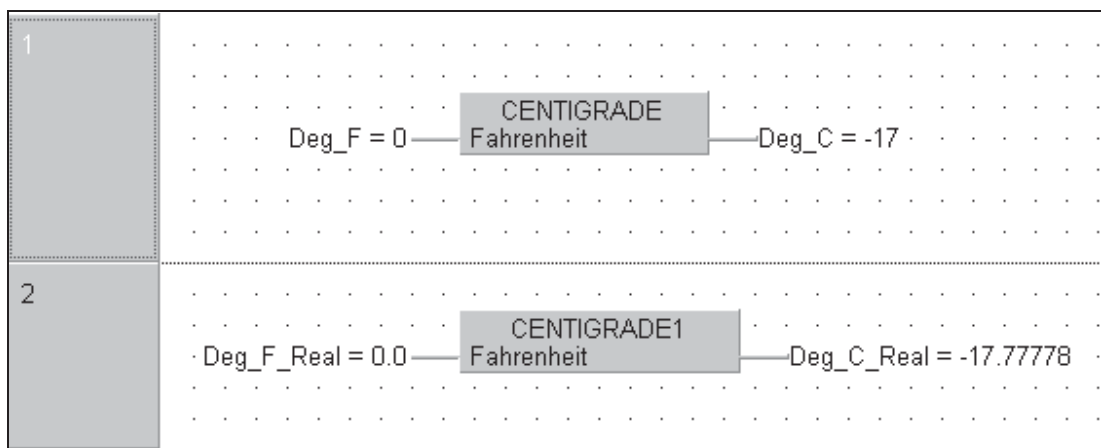
HINWEIS

Gleitkommazahlen vom Typ REAL werden in einem besonderen Format gespeichert und belegen in der SPS zwei aufeinanderfolgende Register (32 Bit). Aus diesem Grund werden in der Globalen Variablenliste die Register D2 (D2, D3) und D4 (D4, D5) als Operanden für diese Variablen angegeben.

③ Vervollständigen Sie das Programm für die POE „Process“ so, wie hier gezeigt:



- ④ Speichern Sie das Projekt, schließen Sie alle Dialogfenster und konvertieren Sie es in den Maschinencode (**Alles neu erstellen**).
- ⑤ Übertragen Sie das Projekt in die SPS und beobachten Sie die Ausführung des Programms im Monitormodus. Diesen aktivieren Sie mit dem Schaltfeld  in der Werkzeugleiste.



Verändern Sie den Wert der Eingangsvariablen „Deg_F_Real“ und beobachten Sie das Ergebnis der Berechnung. Bitte beachten Sie, dass der Ausgangswert nun als Gleitkommazahl mit 7 Stellen und einer höheren Genauigkeit angezeigt wird.

6.2 Programmierung eines Funktionsbausteins

Ziel

Es soll ein Funktionsbaustein programmiert werden, der als Stern-Dreieck-Schalter arbeitet. Deklarieren Sie dazu die folgenden Variablen.


- Start-Taster: **START**
- Stopp-Taster: **STOP**
- Kontakt der Sicherung: **OVERLOAD**
- Umschaltzeit: **TIMEBASE**
- Register für Timer: **TIME_COIL**
- Ausgang für Stern-Schütz: **STAR_COIL**
- Ausgang für Dreieck-Schütz: **DELTA_COIL**

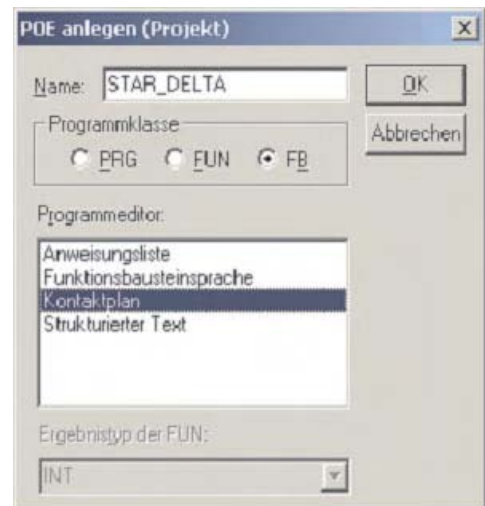
Nennen Sie den Funktionsbaustein **STAR_DELTA**.

Vorgehensweise

① Legen Sie im GX IEC Developer ein neues „leeres“ Projekt ohne POEs an. Geben Sie dem Projekt den Namen „**Motor Control**“.

② Klicken Sie in der Werkzeugleiste auf das

Schaltfeld , um eine neue POE mit dem Namen „**STAR_DELTA**“ anzulegen. Als **Programmklasse** wählen Sie FB (Funktionsbaustein) und als **Programmeditor** Kontaktplan.



Die neue POE wird danach im POE-Pool angezeigt.

③ Klicken Sie auf das Pluszeichen vor dem Namen der POE, um die Unterverzeichnisse für den Header und den Body anzuzeigen.

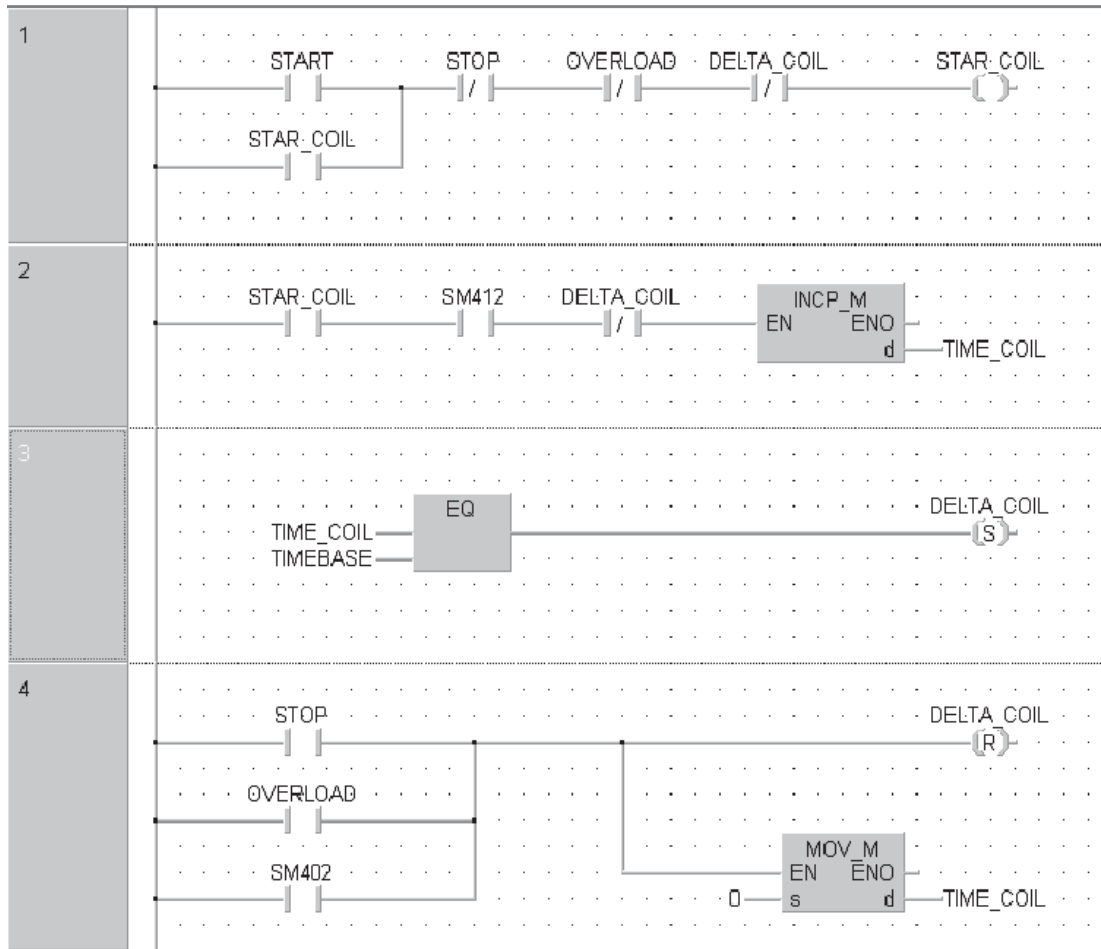
④ Klicken Sie doppelt auf den Header, um ihn zu öffnen.

Deklaration der Lokalen Variablen

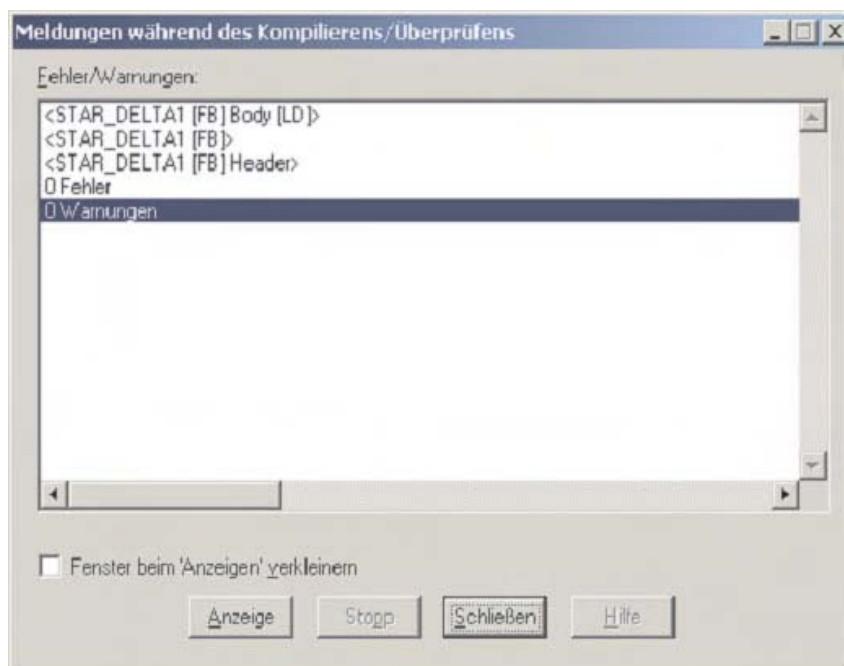
① Tragen Sie in den Header die folgenden Lokalen Variablen ein:

	Klasse	Bezeichner	Typ	Start	Kommentar
0	VAR_INPUT	START	BOOL	... FALSE	
1	VAR_INPUT	STOP	BOOL	... FALSE	
2	VAR_INPUT	OVERLOAD	BOOL	... FALSE	
3	VAR_INPUT	TIMEBASE	INT	... 0	
4	VAR_OUTPUT	DELTA_COIL	BOOL	... FALSE	
5	VAR_OUTPUT	STAR_COIL	BOOL	... FALSE	
6	VAR_OUTPUT	TIME_COIL	INT	... 0	

- ② Prüfen und speichern Sie den Header. Schließen Sie danach das Eingabefenster.
- ③ Öffnen Sie den Body der POE und geben Sie das unten abgebildete Programm ein.

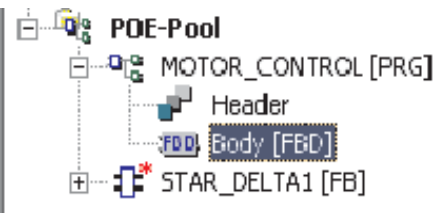


- ④ Prüfen Sie das eingegebene Programm, es sollten keine Fehler oder Warnungen angezeigt werden.



Anlegen einer neuen Programm-POE „MOTOR_CONTROL“

- ① Schließen Sie alle Dialog- und Arbeitsfenster.
- ② Legen Sie eine neue POE mit dem Namen „MOTOR_CONTROL“ an. Als **Programm-klasse** wählen Sie **PRG** und als **Programmediator** die **Funktionsbausteinsprache**.



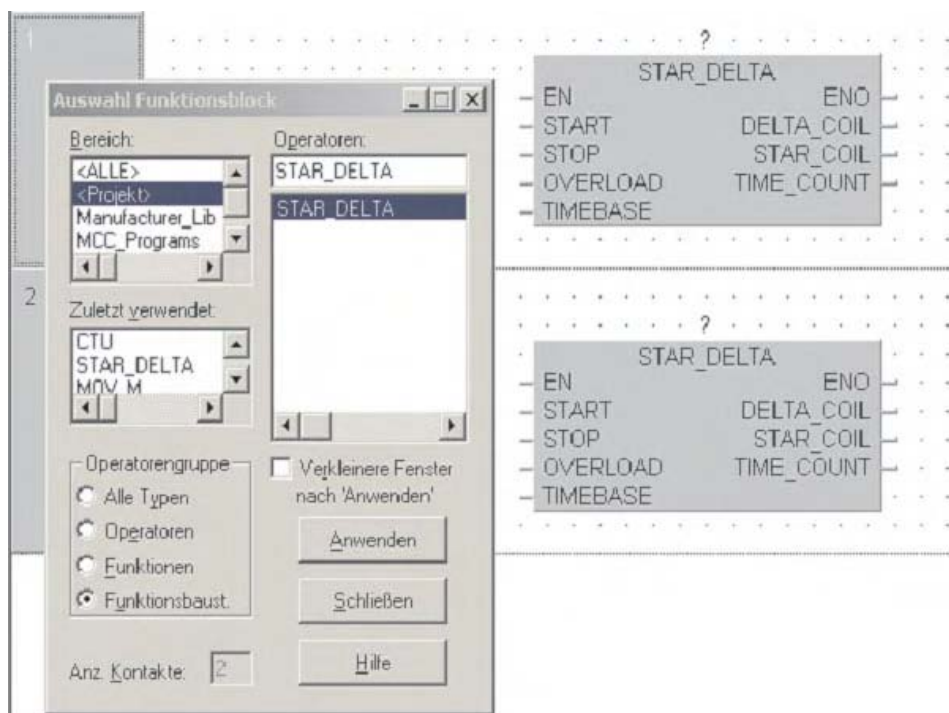
Anlegen einer neuen Globalen Variablenliste

Öffnen Sie die GVL und geben Sie die folgenden Variablen ein:

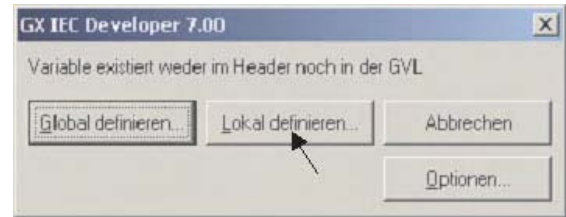
	Klasse	Bezeichner	MIT-Adr.	IEC-Adress	Typ	Start
0	VAR_GLOBAL	START1	X10	%IX16	BOOL	FALSE
1	VAR_GLOBAL	STOP1	X11	%IX17	BOOL	FALSE
2	VAR_GLOBAL	OVERLOAD1	X12	%IX18	BOOL	FALSE
3	VAR_GLOBAL	STAR_COIL1	Y20	%QX32	BOOL	FALSE
4	VAR_GLOBAL	DELTA_COIL1	Y21	%QX33	BOOL	FALSE
5	VAR_GLOBAL	TIME_COIL1	D0	%MWD.0	INT	0
6	VAR_GLOBAL	START2	X13	%IX19	BOOL	FALSE
7	VAR_GLOBAL	STOP2	X14	%IX20	BOOL	FALSE
8	VAR_GLOBAL	OVERLOAD2	X15	%IX21	BOOL	FALSE
9	VAR_GLOBAL	STAR_COIL2	Y22	%QX34	BOOL	FALSE
10	VAR_GLOBAL	DELTA_COIL2	Y23	%QX35	BOOL	FALSE
11	VAR_GLOBAL	TIME_COIL2	D1	%MWD.1	INT	0

Zuweisung von Instanzen-Namen

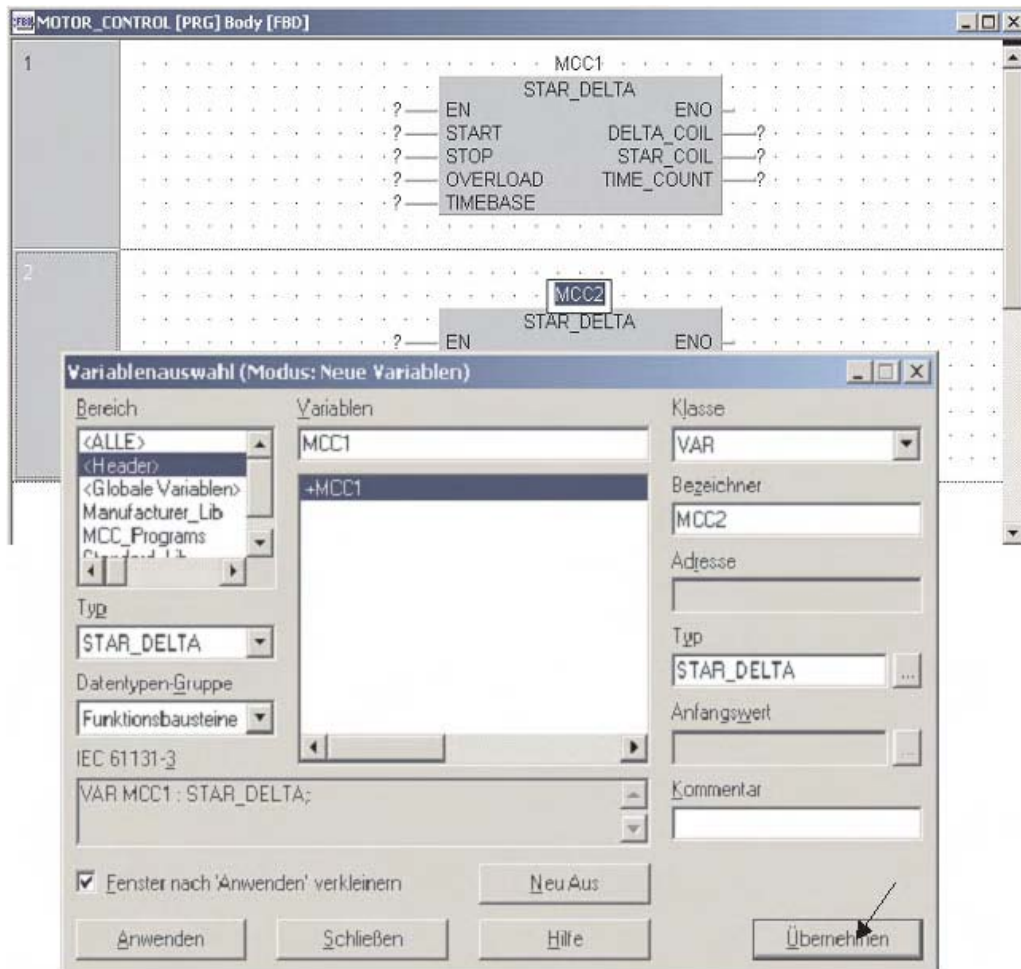
- ① Öffnen Sie den Body der POE MOTOR_CONTROL und richten Sie zwie Netzwerke ein. Plazieren Sie in jedes Netzwerk eine Instanz des Funktionsbausteins STAR_DELTA:



- ② Weisen Sie beiden Instanzen des Funktionsbausteins Namen zu, indem Sie „MCC1“ und „MCC2“ in das Eingabefeld über dem FB eintippen. Die Abfrage des Systems bestätigen Sie bitte mit **Lokal definieren**.



- ③ Deklarieren Sie die Instanzen-Namen „MCC1“ und „MCC2“ so wie hier gezeigt:

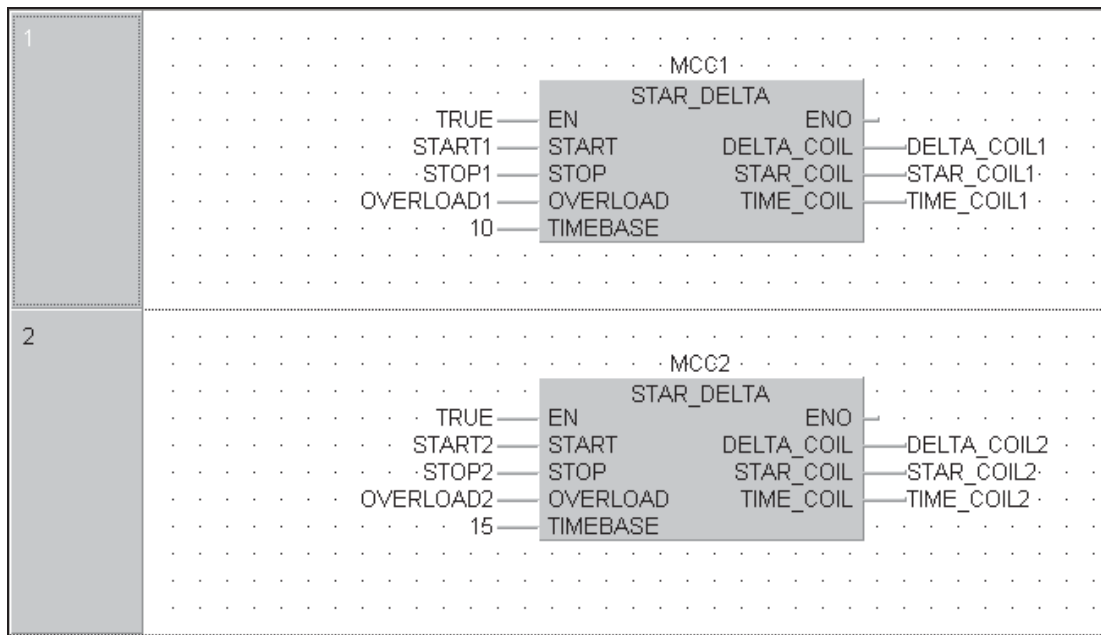


Eine Instanz ist eine Kopie des Funktionsbausteins für diese POE. Geben Sie einfach **MCC1** und **MCC2** ein. Nach der Eingabe werden die Instanzen im Dialogfenster Variablenauswahl als +MCC1 und +MCC2 mit dem Typ STAR_DELTA angezeigt (Das ist der Name des Funktionsbaustein.)

Die Instanzen müssen im Header der POE deklariert werden. Wie die vorhergehenden Abbildungen zeigen, werden die Namen von Instanzen genau so eingefügt, wie neue Variablen, die direkt im Body der POE angegeben wurden.

Funktionsbausteinen Variablen zuweisen

Vervollständigen Sie nun die POE, indem Sie den Funktionsblöcken Variablen zuweisen:



HINWEISE

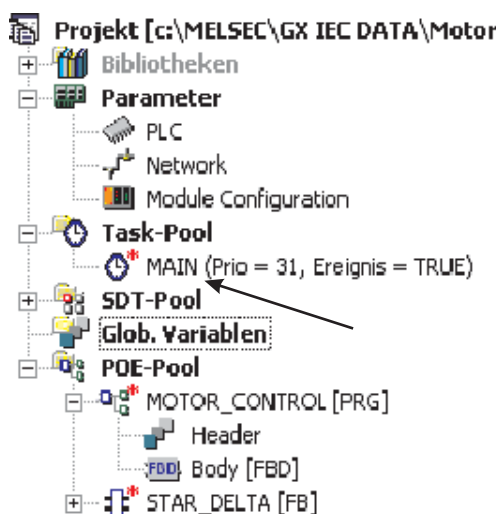
Es können absolute Mitsubishi-Adressen oder symbolische Namen verwendet werden. Bei der Verwendung von Mitsubishi-Adressen entspricht das Programm jedoch nicht mehr den IEC-Bestimmungen.

Die Variable TRUE hat dieselbe Funktion wie ein Merker, der ständig gesetzt ist (In den CPU-Modulen des MELSEC-System Q entspricht dies dem Sondermerker SM400.) Die Angabe von TRUE ist allerdings einfacher nachvollziehbar und entspricht zudem den IEC-Bestimmungen.

Der Funktionsbaustein STAR_DELTA kann beliebig oft im Programm verwendet werden. Es müssen nur unterschiedliche Instanzen-Namen vergeben werden.

Anlegen einer neuen Task:

- ① Legen Sie im Task-Pool eine neue Task „MAIN“ an.

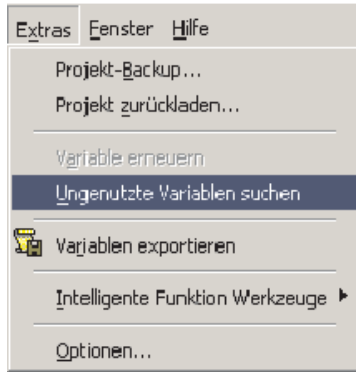


- ② Klicken Sie doppelt auf den Namen der Task und rufen Sie in dieser Task die POE „MOTOR_CONTROL“ auf.

POE-Name	Kommentar
MOTOR_CONTROL	...

- ③ Speichern Sie das Programm und schließen Sie alle Fenster und Dialoge.

Ungenutzte Variablen suchen



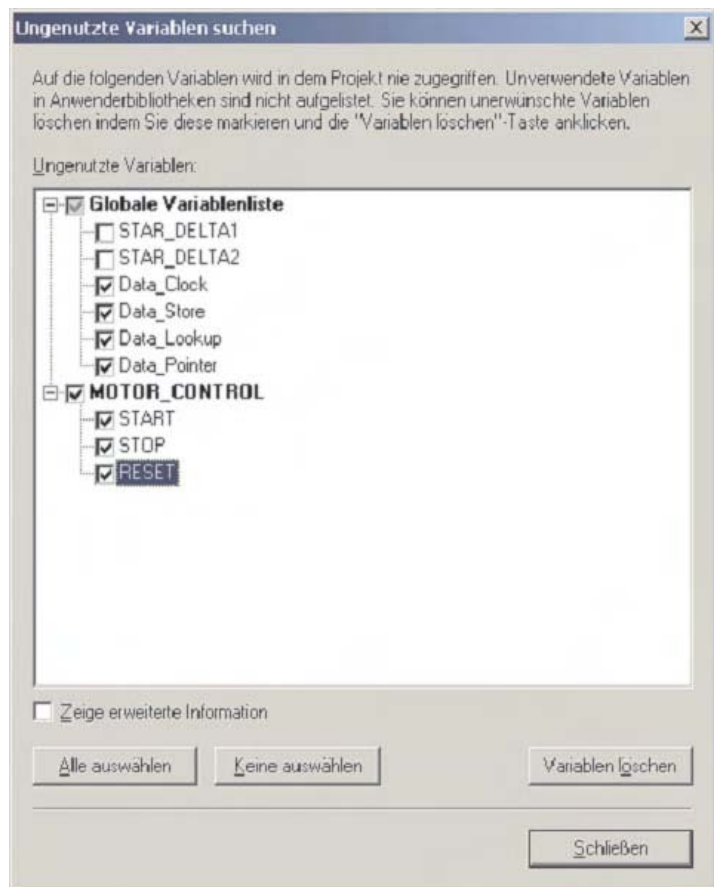
Mit der Funktion **Ungenutzte Variablen suchen** im Menü **Extras** können alle Variablen gesucht werden, die zwar als Lokale oder Globale Variablen deklariert wurden, aber im Projekt nicht verwendet werden.

Bis auf Variablen in anwenderdefinierten Bibliotheken können ungenutzte Variablen anschließend gelöscht werden.

HINWEIS

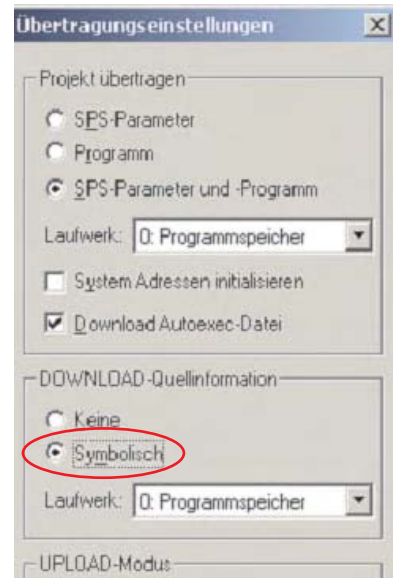
Ungenutzte Variablen können nur gesucht werden, wenn das Projekt in den Maschinencode gewandelt und danach nicht verändert wurde. Falls ungenutzte Variablen nicht gesucht werden können, erscheint eine Warnmeldung.

Jede ungenutzte Variable wird unter dem Verzeichnis angezeigt, in dem Sie deklariert wurde: Globale Variablen erscheinen in der Globalen Variablenliste und Lokale Variablen in der jeweiligen POE. Es werden nur die Verzeichnisse angezeigt, die ungenutzte Variablen enthalten. Die Verzeichnisse sind durch fette Schrift gekennzeichnet. Außerdem erscheinen Sie auf einer höheren Ebene als ihr Inhalt.

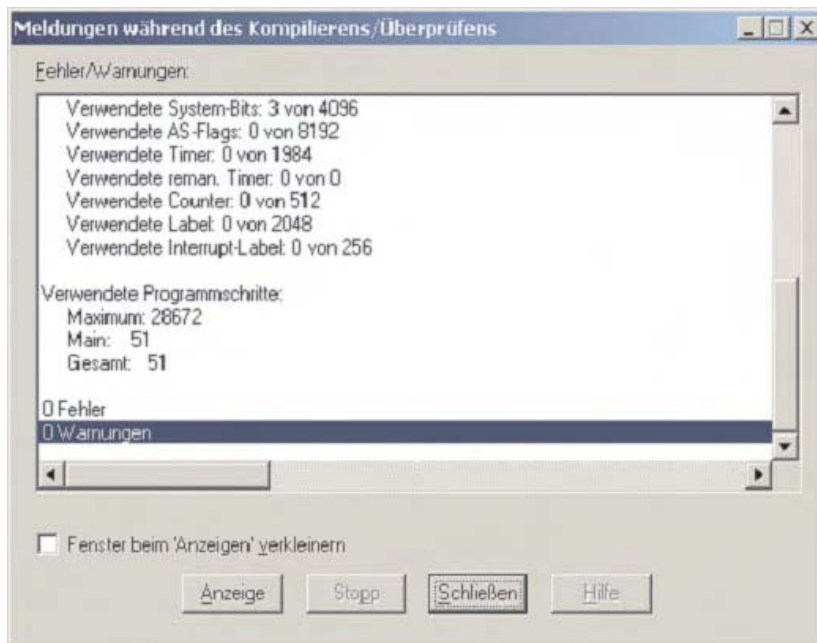


HINWEIS

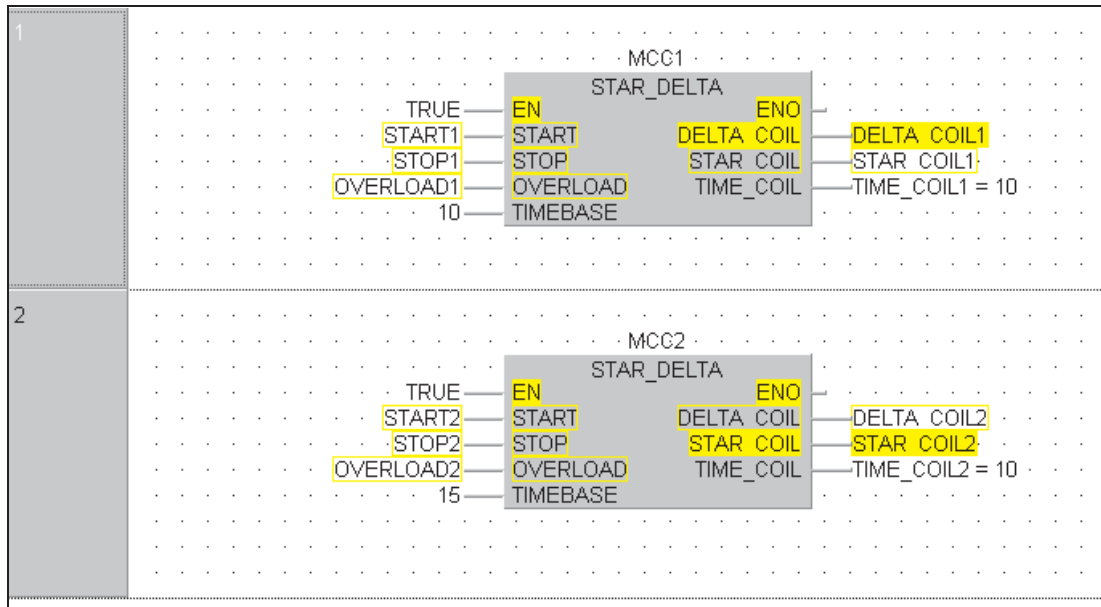
Durch das Löschen ungenutzter Variablen kann der Umfang des Maschinencodes teilweise erheblich reduziert werden. Dies ist besonders wichtig, wenn unter **Download-Quellinformationen** die Option **Symbolik** aktiviert wurde.



Kompilieren Sie mit Hilfe des Schaltfelds  das Programm wie gewohnt .



Öffnen Sie den Body der POE MOTOR_CONTROL, aktivieren Sie mit dem Schaltfeld  den Monitormodus und prüfen Sie die korrekte Funktion des Programms.

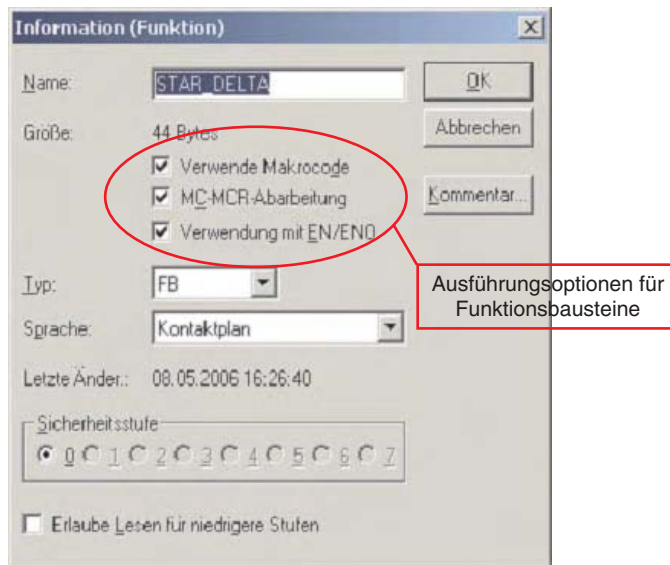


6.3 Ausführungsarten für Funktionsbausteine

Funktionsbausteine können auf verschiedene Arten ausgeführt werden:

- Makro-Code-Abarbeitung
- MC-MCR-Abarbeitung
- Verwendung mit EN/ENO

Die Einstellung der Ausführungsart wird im Eigenschaften-Fenster des Funktionsbausteins vorgenommen.



Einstellung der Ausführungsart

- ① Markieren Sie den Funktionsbaustein im Navigator-Fenster.
- ② Öffnen Sie das Eigenschaften-Fenster des Funktionsbausteins, indem Sie mit der rechten Maustaste auf den Namen des FB klicken und anschließend **Eigenschaften** wählen.
- ③ Kreuzen Sie die gewünschten Optionen an. Die Option **MC-MCR-Abarbeitung** lässt sich erst dann aktivieren, wenn die beiden anderen Optionen bereits aktiviert sind.

Die Bildung einer Instanz und die Programmierung der Instanzen in den verschiedenen Programmiersprachen ist unverändert.

6.3.1 Makro-Code-Abarbeitung

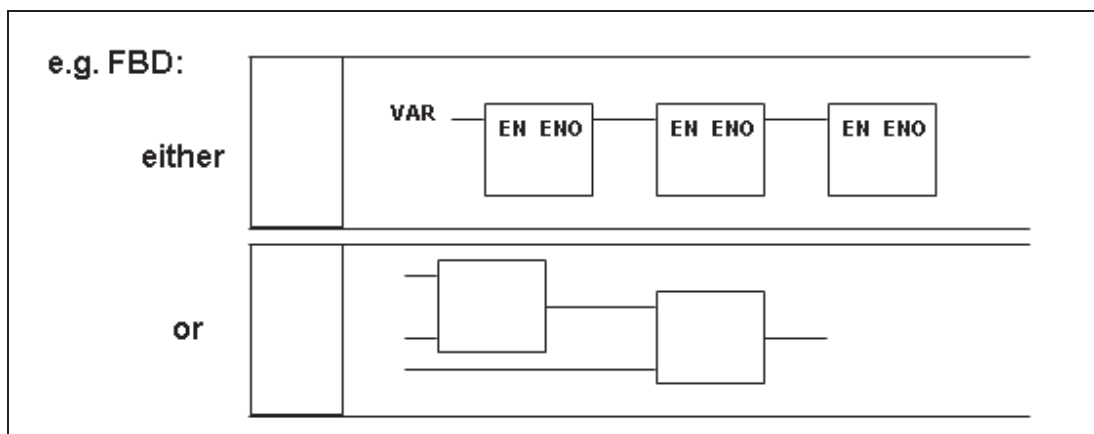
- ohne Makro-Code: Funktionsbaustein wird über System-Label aufgerufen.
- mit Makro-Code: Funktionsbaustein wird intern expandiert

Vorteile eines Funktionsbausteins mit Makro-Code

Mit Makro-Code	Ohne Makro-Code (herkömmliche Abarbeitung)
Für die Abarbeitung einer Funktionsbaustein-Instanz werden keine internen System-Label benötigt. <i>Folge:</i> Die Anzahl der einzusetzenden Funktionsbausteine ist lediglich durch den SPS-Speicher begrenzt, da die Funktionsbausteine unabhängig von den System-Labeln sind.	Für jede Instanz werden interne System-Label (Pointer) verwendet. <i>Folge:</i> Da die verfügbare Anzahl der System-Label begrenzt ist (FX: 128, A: 256, Q: 1024), kann theoretisch auch nur eine begrenzte Anzahl an Funktionsbausteinen verwendet werden. Da System-Label auch für andere interne Abarbeitungsarten benötigt werden, verringert sich Anzahl der einzusetzenden Funktionsbausteine weiter.
Anwenderorientierte Abarbeitung der Funktionsbausteine	Umsetzung der Funktionsbaustein-Thematik nach IEC61131-3
Keine Einschränkungen bei der Abarbeitung von Timern und Ausgängen innerhalb des Funktionsbausteins	Einschränkungen bei der Abarbeitung von Timern und Ausgängen innerhalb eines Funktionsbausteins (Unterprogramms)

6.3.2 EN-Eingang und ENO-Ausgang

- Mit dem EN-Eingang (Enable = Freigabe) kann die Bearbeitung einer Funktion oder Funktionsbausteins gesteuert werden. Nur wenn der EN-Eingang den Zustand „1“ hat, wird die Funktion oder der Funktionsbaustein ausgeführt.
- Am ENO-Ausgang wird der Zustand der EN-Eingangs weitergegeben.
- In einem Netzwerk können nur Anweisungen mit EN-Eingang und ENO-Ausgang oder ohne EN-Eingang und ENO-Ausgang verwendet werden. Die beiden Anweisungsarten können nicht gemischt werden.
- Für eine Reihenschaltung von EN-Ein- und ENO-Ausgängen sollten die Bedingungen am Beginn der Kette definiert werden.



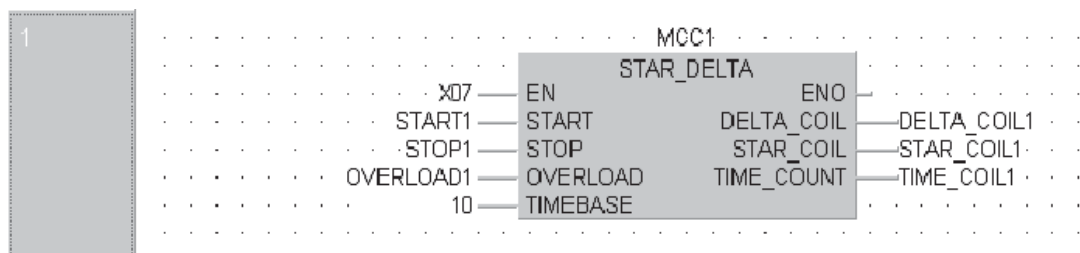
Hinweise zu Funktionen und Funktionsbausteinen

- In der IEC-Standardbibliothek sind fast alle Anweisungen doppelt vorhanden. Die Anweisungen mit EN-Eingang und einen ENO-Ausgang sind durch die Endung „_E“ gekennzeichnet.

- Alle Anweisungen mit der Endung „_M“ sind herstellerspezifisch und in der Herstellerbibliothek gespeichert.
- Bitte beachten Sie bei der Programmierung, besonders in der Funktionsbausteinsprache, die Mitsubishi-Programmierregeln. Besonders für Berechnungen werden, wie in der Abbildung oben gezeigt, oft viele Anweisungen hintereinander angeordnet. Wenn jedoch eine der verwendeten Anweisungen normalerweise als letzte Anweisung eines Netzwerks programmiert werden muss, darf sie nun, nur wegen der Funktionsbausteinsprache, nicht einfach in der zwischen zwei anderen Anweisungen plaziert werden.
- Wählen Sie für eine Aufgabe die passende Anweisung. In den meisten Fällen wird es eine IEC-Anweisung sein.
- Bitte beachten Sie, dass bei den Mitsubishi-Anweisungen zum Speichern des Ergebnisses einer Multiplikation von zwei 16-Bit-Werten 32 Bit benötigt werden. Falls Variable verwendet werden, muss als Datentyp für Multiplikand und Multiplikator INT und für das Ergebnis DINT angegeben werden.

Übung: Ergänzung eines Funktionsbausteins um einen EN-Eingang und einen ENO-Ausgang

Ändern Sie den Funktionsbaustein STAR_DELTA so, dass er einen EN-Eingang und einen ENO-Ausgang erhält. Steuern Sie den EN-Eingang mit dem SPS-Eingang X07 an.



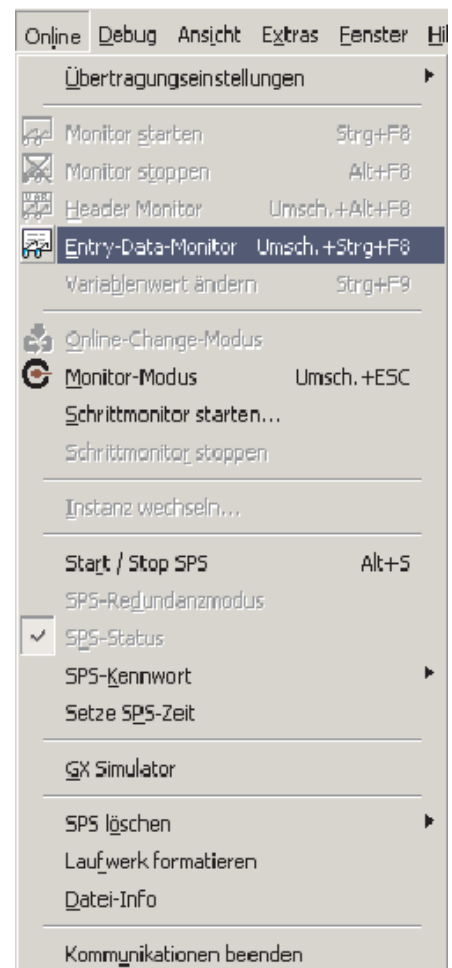
7 Erweiterte Monitorfunktionen

Die folgenden Abbildungen sollen nur die Handhabung der Monitorfunktionen veranschaulichen. Es wird die Ausführung des Funktionsbausteins STAR_DELTA geprüft.

7.1 Entry-Data-Monitor

Wenn gleichzeitig verschiedene Daten aus unterschiedlichen Programmteilen überwacht werden sollen, können Sie den Entry-Data-Monitor (EDM) verwenden.

- ① Zum Start dieser Funktion klicken Sie im Monitormodus im Menü **Online** auf den Menüpunkt **Entry-Data-Monitor**.



Dadurch wird die folgende Tabelle angezeigt:

Pos	Adresse (MIT)	Name	Wert (DEZ)
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			

- ② Bevor Sie die Operandenzustände beobachten können, müssen Sie die Operanden in die Liste eintragen. Klicken Sie dazu in die linke Spalte (**Adresse MIT**) und geben Sie den gewünschten Operanden ein. Falls einem Operanden ein Variablenname zugewiesen wurde, wird dieser zusammen mit dem aktuellen Wert des Operanden in der Tabelle angezeigt.

Die Breite der Spalten kann angepasst werden, indem der Cursor im Kopf der Tabelle auf eine Spaltengrenze positioniert, dann die linke Maustaste betätigt und anschließend die Begrenzung verschoben wird.

Pos	Adresse (MIT)	Name	Wert (DEZ)
1	D0	TIME_COIL1	0
2	D1	TIME_COIL2	0
3	X10	START1	0
4	X11	STOP1	0
5	X12	OVERLOAD1	0
6	X13	START2	0
7	X14	STOP2	0
8	X15	OVERLOAD2	0

7.1.1 Anpassung des Entry-Data-Monitor

- ① Klicken Sie mit der rechten Maustaste in die Tabelle. Dadurch wird das unten abgebildete Menü angezeigt. Wählen Sie **Konfiguration**.

Pos	Adresse (MIT)	Name	Wert (DEZ)
1	D0	TIME_COIL1	0
2	D1	TIME_COIL2	
3	X10	START1	
4	X11	STOP1	
5	X12	OVERLOAD1	
6	X13	START2	
7	X14	STOP2	
8	X15	OVERLOAD2	
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			

- Objekte einfügen... F2
- Nächstes Objekt F3
- Festgelegte Eingänge einfügen
- Gesetzte Eingänge einfügen
- Gesetzte Ausgänge einfügen
- Lösche Operandendatei
- Zeile einfügen Einfg
- Löschen Entf
- Alles löschen
- Lesen < SPS
- Schreiben > SPS...
- Lesen aus Datei
- Schreiben in Datei
- Konfiguration...**
- Immer oben

Im Dialogfenster **Konfiguration** können Sie die EDM anpassen. Wenn Sie in diesem Fenster die rechte Maustaste betätigen, wird ein Menü angezeigt, mit dem Sie z. B. Spalten in die EDM-Tabelle eingefügt oder gelöscht werden können. In diesem Beispiel werden zusätzliche Spalten für die IEC-Adresse und den hexadezimalen Wert der Operanden eingefügt.



- ② Klicken Sie in die Zeile **Name**, um sie zu markieren und betätigen Sie anschließend die rechte Maustaste. Wählen Sie dann **Zeile einfügen**.



Dadurch wird über der Zeile **Name** eine neue, leere Zeile eingefügt.



- ③ Klicken Sie doppelt in das leere Feld oder betätigen Sie die Taste F2 der PC-Tastatur. Es wird die Feldliste geöffnet, aus der gewählt werden kann, was angezeigt werden soll. Wählen Sie **Adresse (IEC)** und klicken Sie dann auf **OK**.



- ④ Konfigurieren Sie die EDM so, dass in der 5. Position der Tabelle der hexadezimale Wert angezeigt wird. Wählen Sie dazu aus der Feldliste **Wert (Hex)**.



- ⑤ Schließen Sie das Dialogfenster Konfiguration und beachten Sie den geänderten Entry-Data-Monitor.

Pos	Adresse (MIT)	Adresse (IEC)	Name	Wert (DEZ)	Wert (HEX)
1	D0	%MWO.0	TIME_COIL1	0	0
2	D1	%MWO.1	TIME_COIL2	0	0
3	X10	%IX16	START1	0	0
4	X11	%IX17	STOP1	0	0
5	X12	%IX18	OVERLOAD1	0	0
6	X13	%IX19	START2	0	0
7	X14	%IX20	STOP2	0	0
8	X15	%IX21	OVERLOAD2	0	0

Auf diese Weise kann der EDM konfiguriert werden, um verschiedene Daten in ein und derselben Tabelle anzuzeigen.

Versuchen Sie, die Spaltenbreiten zu verstellen. Um die gesamte Tabellenbreite anzuzeigen, kann im Menü **Ansicht** unter dem Menüpunkt **Zoom** der Vergrößerungsfaktor eingestellt werden. Die Größe der Anzeige ist abhängig vom der Bildschirmauflösung des PC.

In der EDM können auch Werte für die angezeigten Objekte eingegeben werden. Der Wert eines Datenregisters lässt sich zum Beispiel ändern, indem das entsprechende Feld markiert und der Wert dort eingegeben wird.

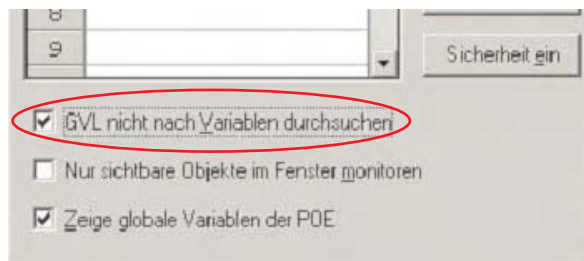
HINWEIS

Die Zustände und die Werte von SPS-Operanden werden durch das SPS-Programm bestimmt. Operanden werden durch das Überschreiben im Entry-Data-Monitor nur kurzzeitig beeinflusst und nehmen dann wieder ihren vom Programm zugewiesenen Status bzw. Wert an.

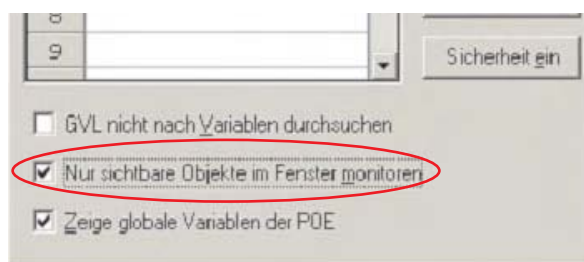
In diesem Beispiel werden die Inhalte der Datenregister D0 und D1 zyklisch vom Programm verändert.

Konfigurationseinstellungen**● GVL nicht nach Variablen durchsuchen**

Bei Eingabe einer Mitsubishi-Adresse im Entry-Data-Monitor, wie zum Beispiel M0, sucht das System automatisch in der Globalen Variablenliste nach einem Bezeichner für diesen Operanden. Bei großen Projekten kann dies lange dauern. Diese automatische Suche wird nicht ausgeführt, wenn die Option **GVL nicht nach Variablen durchsuchen** aktiviert wird.

**● Nur sichtbare Objekte im Fenster monitorieren**

Grundsätzlich werden alle Elemente des Entry-Data-Monitor überwacht, auch wenn Sie nicht angezeigt werden. Mit der Option **Nur sichtbare Objekte im Fenster monitorieren** kann die Reaktionszeit verkürzt werden.



7.1.2 Zustände von Bit-Operanden in der EDM verändern

Beim Testen eines Programms können Zustände von Bit-Operanden (Datentyp BOOL) auch direkt im Entry-Data-Monitor beeinflusst werden. Wenn beispielsweise zum Einleiten eines bestimmten Prozesses das Eingangssignal eines Schalters in der Anlage benötigt wird, kann dieser Eingang im EDM gesetzt und danach der weitere Programmverlauf beobachtet werden.

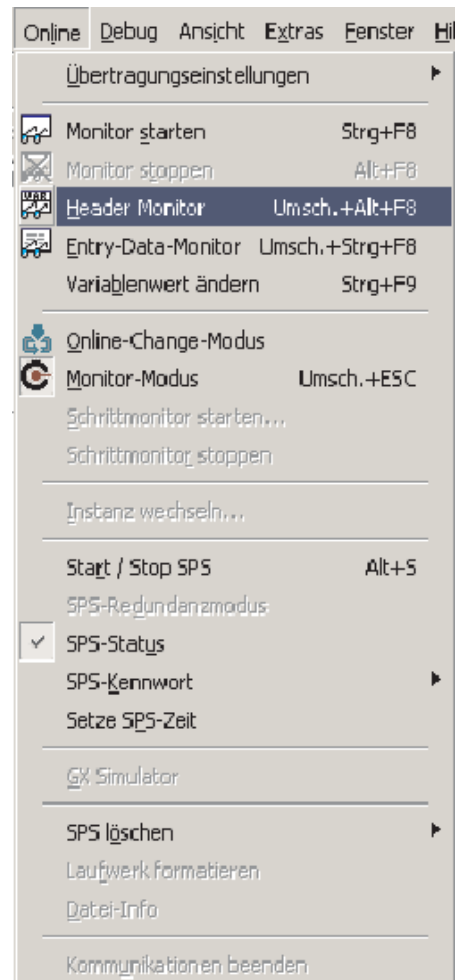
Wenn in das Feld mit dem Wert einer Booleschen Variable doppelt geklickt wird, ändert sich abwechselnd deren Zustand (z. B. 0 → 1 → 0 → 1 usw.).

Pos	Adresse (MIT)	Adresse (IEC)	Name	Wert (DEZ)	Wert (HEX)
1	D0	%MWD.0	TIME_COIL1	10	A
2	D1	%MWD.1	TIME_COIL2	0	0
3	X10	%IX16	START1	1	1
4	X11	%IX17	STOP1	0	0
5	X12	%IX18	OVERLOAD1	0	0
6	X13	%IX19	START2	0	0
7	X14	%IX20	STOP2	0	0
8	X15	%IX21	OVERLOAD2	1	1
9					
10					
11					
12					
13					
14					

Klicken Sie hier doppelt, um den Zustand des Operanden umzuschalten.

7.2 Header-Monitor

Ein weiteres nützliches Werkzeug des GX IEC Developers ist der **Header-Monitor**. Er kann angewählt werden, wenn im Monitormodus der Body einer POE geöffnet ist. Klicken Sie dazu im Menü **Online** auf den Eintrag **Header Monitor**. Diese Funktion kann aber auch in der Werkzeugleiste **Online** angewählt werden.




Im Header-Monitor werden alle Elemente der geöffneten POE mit ihren Zuständen angezeigt:

Pos	Adresse (MIT)	Adresse (IEC)	Name	Wert (DEC)	Wert (HEX)
1			-MOTOR_CONTROL		
2	X10	%IX16	START1	1	1
3	X11	%IX17	STOP1	0	0
4	X12	%IX18	OVERLOAD1	0	0
5	Y21	%QX33	DELTA_COIL1	1	1
6	Y20	%QX32	STAR_COIL1	0	0
7	D0	%MWO.0	TIME_COIL1	10	A
8	X13	%IX19	START2	0	0
9	X14	%IX20	STOP2	0	0
10	X15	%IX21	OVERLOAD2	1	1
11	Y23	%QX35	DELTA_COIL2	0	0
12	Y22	%QX34	STAR_COIL2	0	0
13	D1	%MWO.1	TIME_COIL2	0	0
14			+MCC1		
15			+MCC2		

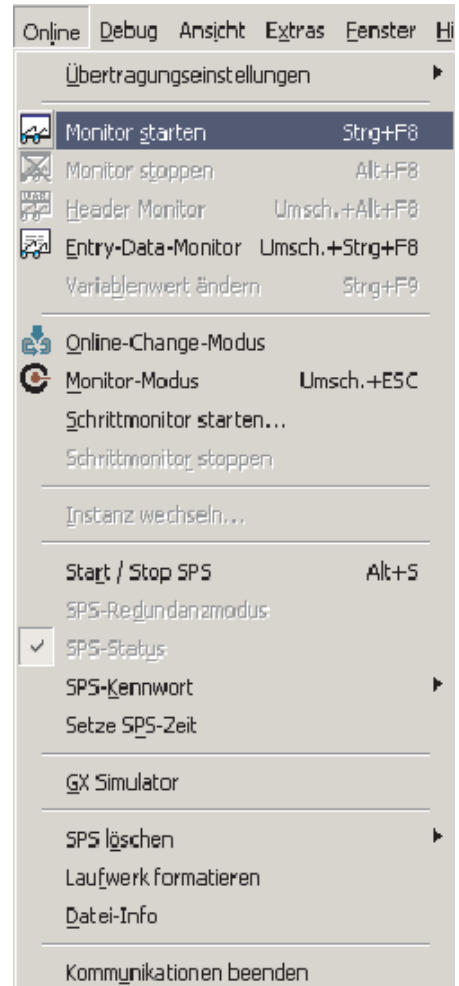
Beachten Sie, dass die Booleschen Variablen in dieser Liste farbig gekennzeichnet werden.

7.3 Gleichzeitige Anzeige mehrerer Fenster

Um mehrere Fenster gleichzeitig im Monitormodus anzuzeigen, müssen diese separat geöffnet und dann im Menü **Fenster** zum Beispiel **Untereinander** angeordnet werden.

Wird der Monitormodus durch das Schaltfeld  aktiviert, wenn nur ein Fenster angewählt ist, gilt der Monitormodus nur für dieses Fenster.

Um den Inhalt weiterer Fenster in den Monitormodus einzubeziehen, klicken Sie in dieses Fenster und wählen dann im Menü **Online** den Eintrag **Monitor starten**.



HINWEIS

Diese Methode zum Start des Monitormodus wurde gewählt, um den Austausch von Daten für Fenster zu verhindern, die zwar geöffnet sind, sich aber im Hintergrund befinden. Durch den Austausch unnötiger Daten zwischen PC und SPS wird sonst die Übertragungszeit verlängert und damit die Reaktionszeit im Monitormodus vergrößert. Besonders stark wirkt sich dieser Effekt bei den Steuerungen der MELSEC FX-Familie oder der A-Serie aus, die mit GX IEC Developer Daten über eine langsamere serielle Verbindung austauschen.

Gleichzeitige Anzeige von Header und Body im Monitormodus

Die folgende Abbildung zeigt ein Beispiel für die gleichzeitige Anzeige von Header und Body im Monitormodus.

Pos	Adresse (MIT)	Adresse (IEC)	Name	Wert (DEZ)	Wert (HEX)
1			-MOTOR_CONTROL		
2	X10	%IX16	START1	1	1
3	X11	%IX17	STOP1	0	0
4	X12	%IX18	OVERLOAD1	0	0
5	Y21	%QX33	DELTA_COIL1	1	1
6	Y20	%QX32	STAR_COIL1	0	0
7	D0	%MWO.0	TIME_COIL1	10	A
8	X13	%IX19	START2	0	0
9	X14	%IX20	STOP2	0	0
10	X15	%IX21	OVERLOAD2	1	1
11	Y23	%QX35	DELTA_COIL2	0	0
12	Y22	%QX34	STAR_COIL2	0	0
13	D1	%MWO.1	TIME_COIL2	0	0
14			+MCC1		
15			+MCC2		

The screenshot shows the FBD editor for the program 'MOTOR_CONTROL [PRG] Body [FBD]'. It displays two MCC blocks:

- MCC1:**
 - EN: TRUE
 - START: START1
 - STOP: STOP1
 - OVERLOAD: OVERLOAD1
 - TIMEBASE: 10
 - DELTA_COIL: DELTA_COIL1
 - STAR_COIL: STAR_COIL1
 - TIME_COIL: TIME_COIL1 = 10
- MCC2:**
 - EN: TRUE
 - START: START2
 - STOP: STOP2
 - OVERLOAD: OVERLOAD2
 - TIMEBASE: 15
 - DELTA_COIL: DELTA_COIL2
 - STAR_COIL: STAR_COIL2
 - TIME_COIL: TIME_COIL2 = 0

7.4 Anzeige von Bit-Gruppen

Um mehrere aufeinanderfolgende Bit-Operanden mit einer Anweisung anzusprechen, wird die Adresse des ersten Bit-Operanden zusammen mit einem Faktor „K“ angegeben, der die Anzahl der Operanden angibt. Dieser Faktor „K“ gibt die Anzahl der Einheiten zu je 4 Operanden an: K1 = 4 Operanden, K2 = 8 Operanden, K3 = 12 Operanden usw.

Mit der Angabe „K2M0“ werden zum Beispiel die acht Merker M0 bis M7 definiert. Möglich sind Faktoren von K1 (4 Operanden) bis K8 (32 Operanden).

Diese Bitgruppen können auch im Entry-Data-Monitor angezeigt werden. In folgenden Beispiel werden mit K1X10 die Zustände der Eingänge X10 - X13 angezeigt:

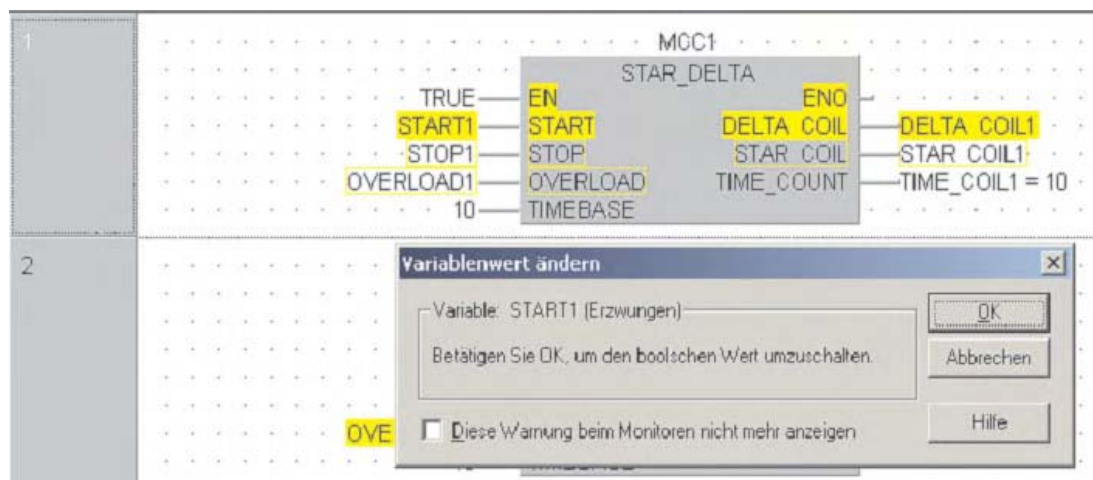
Pos	Adresse (MIT)	Adresse (IEC)	Name	Wert (DEZ)	Wert (HEX)
1	D0	%MWO.0	TIME_COIL1	10	A
2	D1	%MWO.1	TIME_COIL2	0	0
3	X10	%IX16	START1	1	1
4	X11	%IX17	STOP1	0	0
5	X12	%IX18	OVERLOAD1	0	0
6	X13	%IX19	START2	0	0
7	X14	%IX20	STOP2	0	0
8	X15	%IX21	OVERLOAD2	1	1
9					
10	K1X10	%IW19.1.16	K1X10	1	1
11					
12					

7.5 Werte von Variablen im Body der POE ändern

Im Monitormodus ist es auch möglich, die Werte von Variablen direkt im Body der POE zu ändern. Bei Bit-Operanden (Datentyp POOL) kann der Zustand abwechselnd umgeschaltet werden (z. B. 0 → 1 → 0 → 1 usw.). Bei Variablen vom Typ INT/DINT oder REAL kann direkt ein Wert als dezimaler oder hexadezimale Zahl eingegeben werden. Ein Beispiel hierfür finden Sie im Abschnitt 6.1.

Sie leiten diese Funktion ein, indem Sie doppelt auf die Variable (z.B. START1) klicken und dadurch das Dialogfenster zur Eingabe öffnen. Dieses Dialogfenster kann auch ausgeblendet werden. Dadurch können die Variablen einfach mit der Maus gesteuert werden.

Bei der Steuerung von Operanden in Ausgabeanweisungen, wie zum Beispiel Ausgänge, hat die Bearbeitung durch das Programm eine höhere Priorität. Diese Operanden werden durch die manuelle Zuweisung im Monitormodus nur kurzzeitig beeinflusst und nehmen danach wieder ihren vom Programm zugewiesenen Status ein.




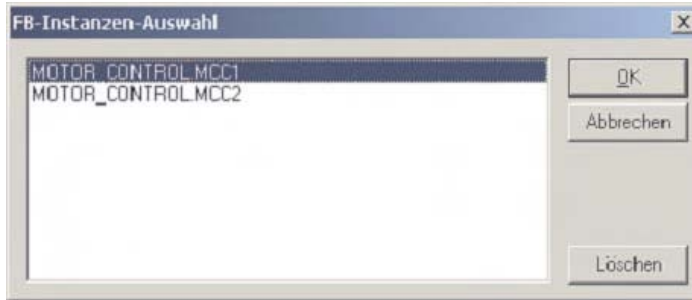
HINWEIS

Die Werte oder Zustände von Variablen können auch verändert werden, wenn sie als absolute MELSEC-Adresse im Body der POE angegeben sind.

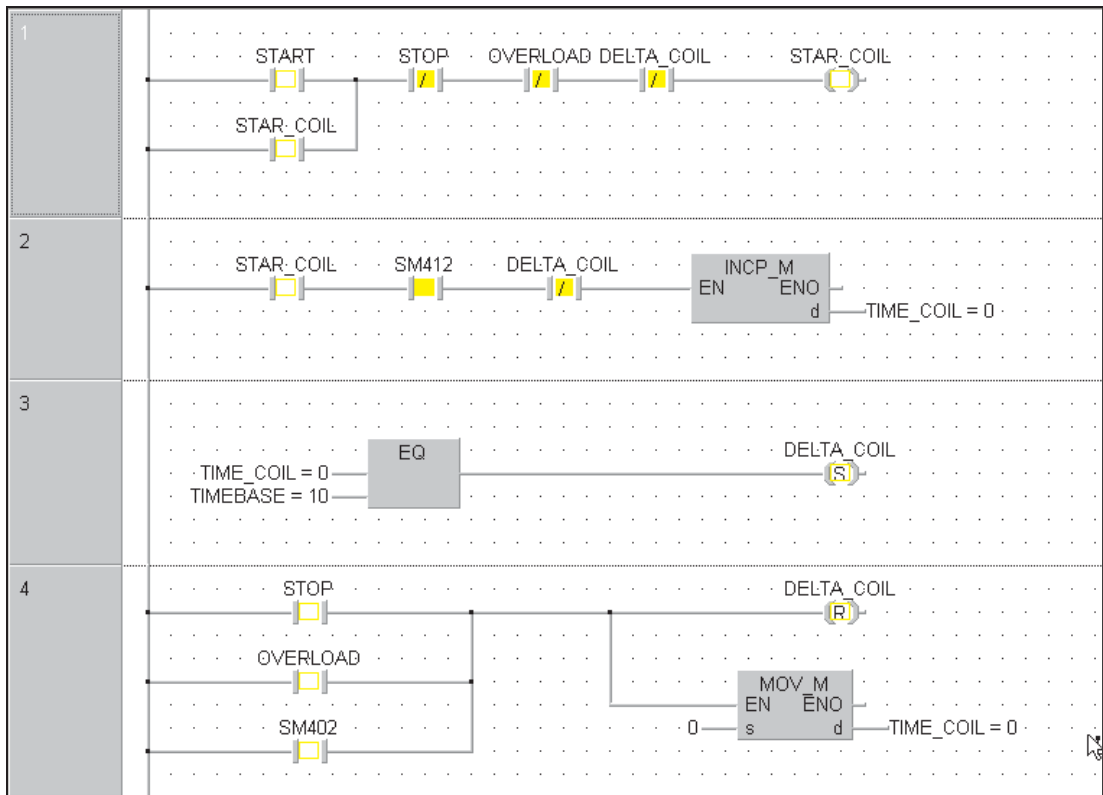
7.6 Instanzen von Funktionsbausteinen monitoren

Die einzelnen Instanzen von Funktionsbausteinen können unabhängig voneinander im Monitormodus beobachtet werden.

- Um eine Instanz der POE des Funktionsbausteins STAR_DELTA im aktuellen Projekt zu beobachten, öffnen Sie den Body der POE und klicken danach in der Werkzeugleiste auf das Schaltfeld  (Monitormodus). Das folgende Dialogfenster wird angezeigt:



- Wählen Sie die Instanz MOTOR_CONTROL.MCC1 und beobachten Sie die Ausführung des Programms.

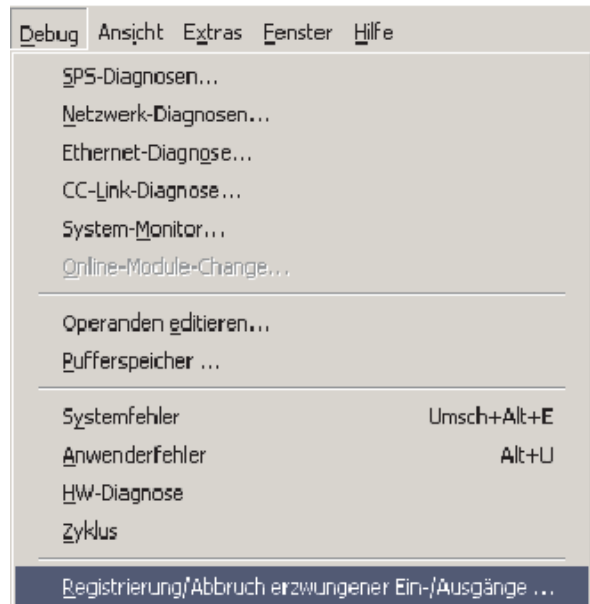


Auf diese Art und Weise kann jede Instanz eines Funktionsbausteins geprüft werden.

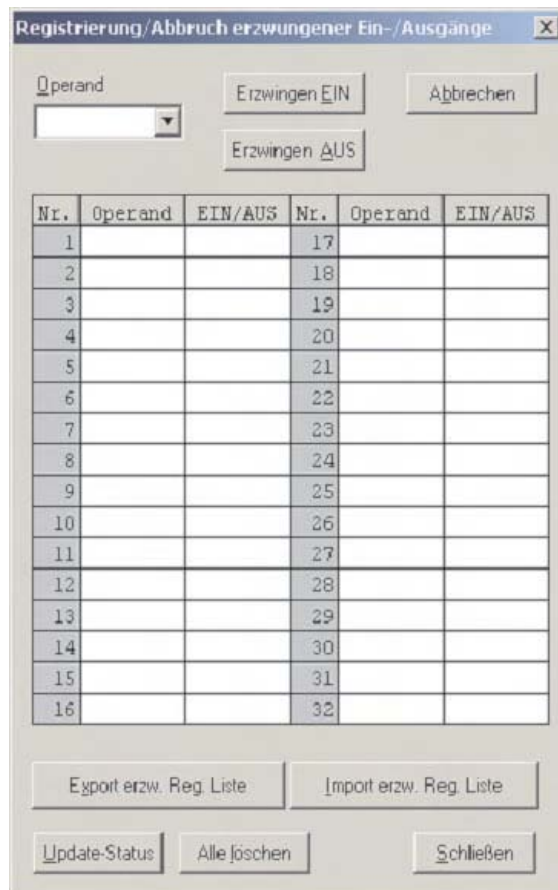
8 Ein- und Ausgänge steuern

Die Programmier-Software GX-IEC Developer erlaubt auch die Steuerung von Ein- und Ausgängen der SPS. Dabei werden die Ein- und Ausgänge unabhängig vom SPS-Programm zwangsweise ein- oder ausgeschaltet.

- ① Zur Steuerung der Ein- und Ausgänge klicken Sie im Menü **Debug** auf den Eintrag **Registrierung/Abbruch erzwungener Ein-/Ausgänge**



- Das folgende Fenster wird angezeigt:



- ② Geben Sie in das Feld **Operand** X10 und X15 ein und klicken Sie nach jeder Eingabe auf das Feld **Erzwingen EIN**.

The dialog box 'Registrierung/Abbruch erzwungener Ein-/Ausgänge' contains the following table:

Nr.	Operand	EIN/AUS	Nr.	Operand	EIN/AUS
1	X10	ON	17		
2	X15	ON	18		
3			19		
4			20		
5			21		
6			22		
7			23		
8			24		
9			25		
10			26		
11			27		
12			28		
13			29		
14			30		
15			31		
16			32		

The ladder logic diagram shows two networks, MCC1 and MCC2. MCC1 has inputs TRUE, START1, STOP1, OVERLOAD1, and TIMEBASE. MCC2 has inputs TRUE, START2, STOP2, OVERLOAD2, and TIMEBASE. Both networks have outputs ENO, DELTA COIL, STAR COIL, and TIME_COIL.

- ③ Um den Zustand von X10 oder X15 umzuschalten, klicken Sie doppelt in das jeweilige Feld **EIN/AUS**.

The dialog box 'Registrierung/Abbruch erzwungener Ein-/Ausgänge' contains the following table:

Nr.	Operand	EIN/AUS	Nr.	Operand	EIN/AUS
1	X10	AUS	17		
2	X15	AUS	18		
3			19		
4			20		
5			21		
6			22		
7			23		
8			24		
9			25		
10			26		
11			27		
12			28		
13			29		
14			30		
15			31		
16			32		

- ④ Geben Sie auch die Ausgänge Y20, Y21 und Y22 in die Liste ein und beobachten Sie Auswirkungen auf diese Operanden.

- ⑤ Um einen Operanden aus der Liste der zwangsweise gesteuerten Ein- und Ausgänge zu entfernen, klicken Sie entweder doppelt auf den Eintrag in der Liste oder geben den Operanden in das Feld **Operand** ein. (Sie brauchen den Operanden nicht einzutippen, es genügt, wenn Sie auf das Symbol „▼“ neben dem Eingabefeld klicken und den Operanden aus der Liste wählen, die dann angezeigt wird.) Klicken Sie dann auf **Abbrechen**.

Registrierung/Abbruch erzwungener Ein-/Ausgänge

Operand: Y22

Erzwingen EIN

Erzwingen AUS

Abbrechen

Nr.	Operand	EIN/AUS	Nr.	Operand	EIN/AUS
1	X10	ON	17		
2	X15	ON	18		
3	Y20	ON	19		
4	Y21	ON	20		
5	Y22	ON	21		
6			22		
7			23		
8			24		
9			25		
10			26		
11			27		
12			28		
13			29		
14			30		
15			31		
16			32		

Export erzw. Reg. Liste

Import erzw. Reg. Liste

Update-Status

Alle löschen

Schließen

- ⑥ Nach dem Löschen von Y22 sieht die Liste so aus:

Registrierung/Abbruch erzwungener Ein-/Ausgänge

Operand:

Erzwingen EIN

Erzwingen AUS

Abbrechen

Nr.	Operand	EIN/AUS	Nr.	Operand	EIN/AUS
1	X10	ON	17		
2	X15	ON	18		
3	Y20	ON	19		
4	Y21	ON	20		
5			21		
6			22		
7			23		
8			24		
9			25		
10			26		
11			27		
12			28		
13			29		
14			30		
15			31		
16			32		

Export erzw. Reg. Liste

Import erzw. Reg. Liste

Update-Status

Alle löschen

Schließen

HINWEIS

Wenn in der CPU der Zustand eines Ein- oder Ausgangs erzwungen wurde, blinkt die Leuchtdiode MODE der CPU mit einer Frequenz von 2 Hz.

- ⑧ Um alle erzwungenen Ein- und Ausgänge in der CPU gleichzeitig zu löschen, klicken Sie auf das Schaltfeld **Alle löschen**.



- ⑨ Vor dem Löschen aller Einträge wird die folgende Warnmeldung angezeigt. Bestätigen Sie mit **Ja**.

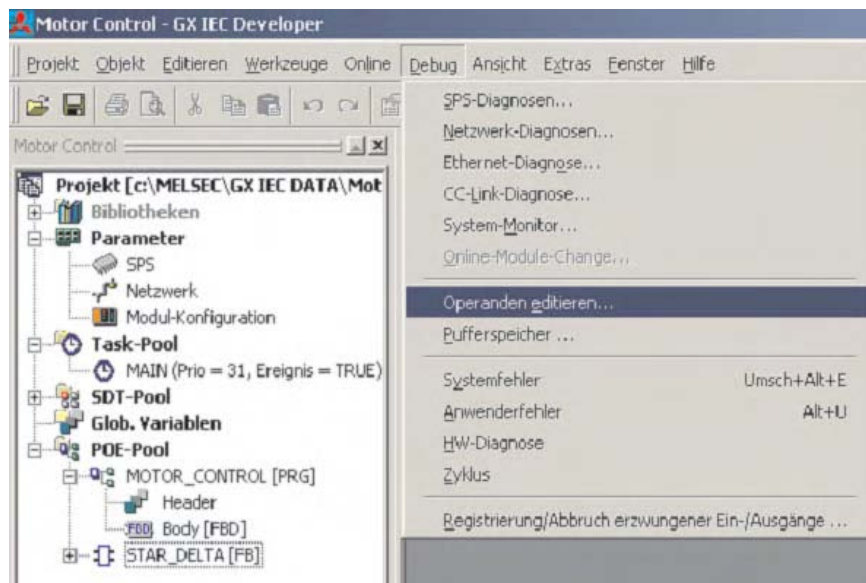
**HINWEIS**

Einzelne Einträge in der Liste der zwangsweise gesteuerten Ein- und Ausgänge können mit Hilfe des Schaltfelds **Abbrechen** entfernt werden..

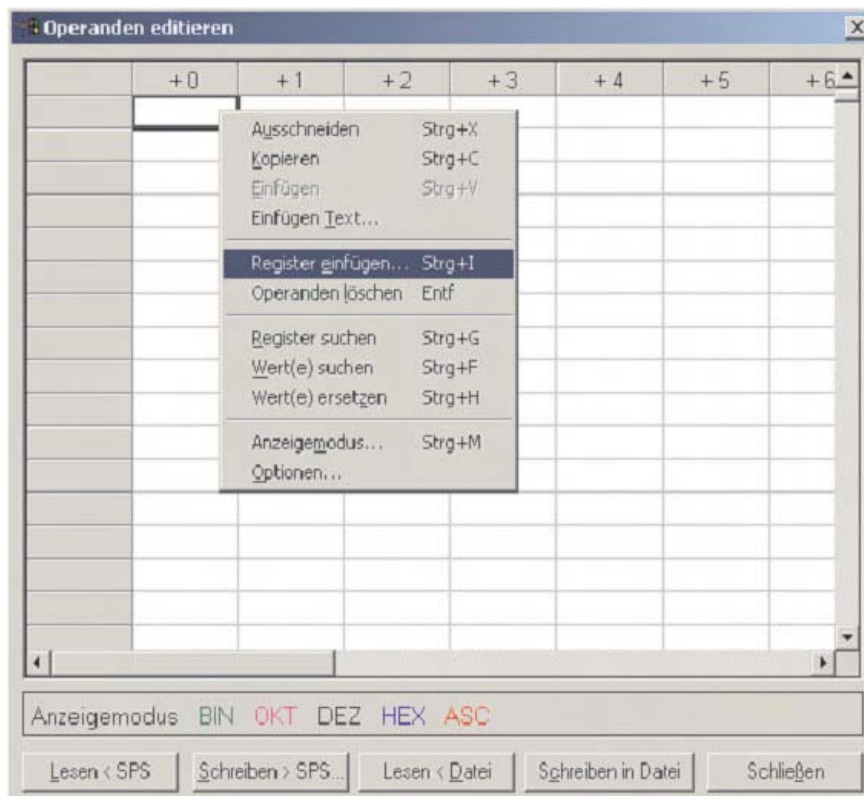
9 Operanden editieren

Mit der Funktion **Operanden editieren** können die Zustände von Operanden beobachtet werden, die sich in einen zusammenhängenden Bereich befinden. Diese Funktion entspricht der Funktion **D,W,R-Setzen** in MELSEC MEDOC und **Operanden-Batch** im GX Developer.

- ① Um diese Testfunktion zu starten, klicken Sie im Menü **Debug** auf den Menüpunkt **Operanden editieren**.



- ② Klicken Sie in die Zeller in der linken oberen Ecke. Betätigen Sie dann die rechte Maustaste. Aus dem Menü, das dann angezeigt wird, wählen Sie **Register einfügen**.



- ③ Tragen Sie in das Feld **Operand** den Typ des Operanden ein. Wenn alle Operanden dieses Typs angezeigt werden sollen, klicken Sie bitte **OK** an. Falls Sie sich nur einen bestimmten Operandenbereich ansehen möchten, klicken Sie auf **Adresse** und geben in den Eingabefeldern die Anfangs- und die Endadresse des Bereichs ein. Klicken Sie dann auf **OK**.

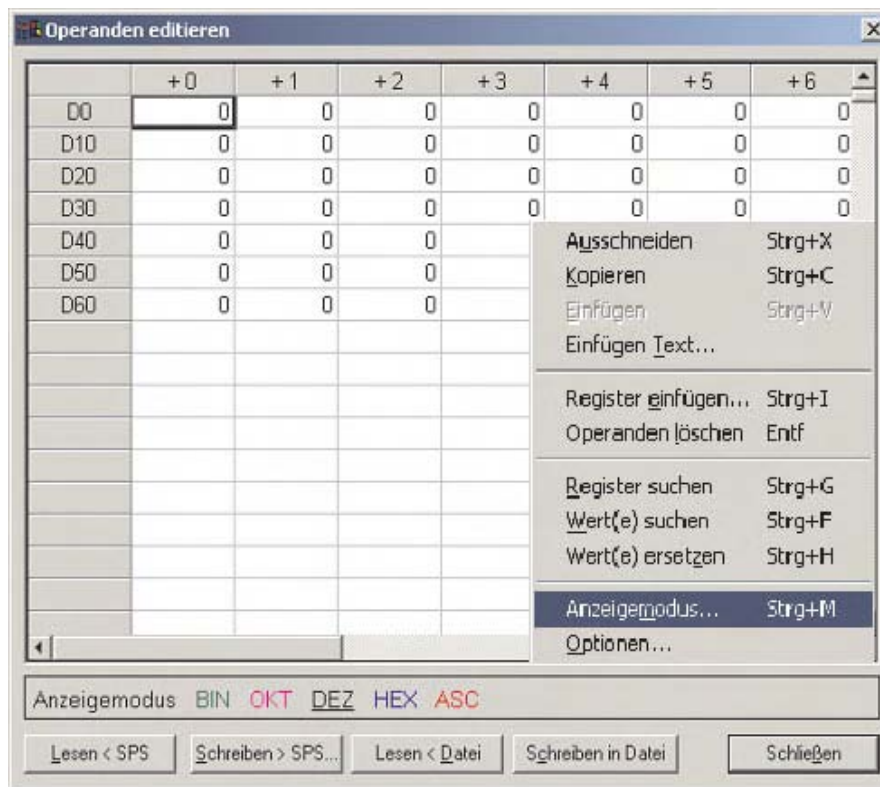


Die Tabelle kann frei konfiguriert, als Datei gespeichert oder in die SPS übertragen werden. Eine gespeicherte Konfiguration kann auch aus der SPS gelesen werden.

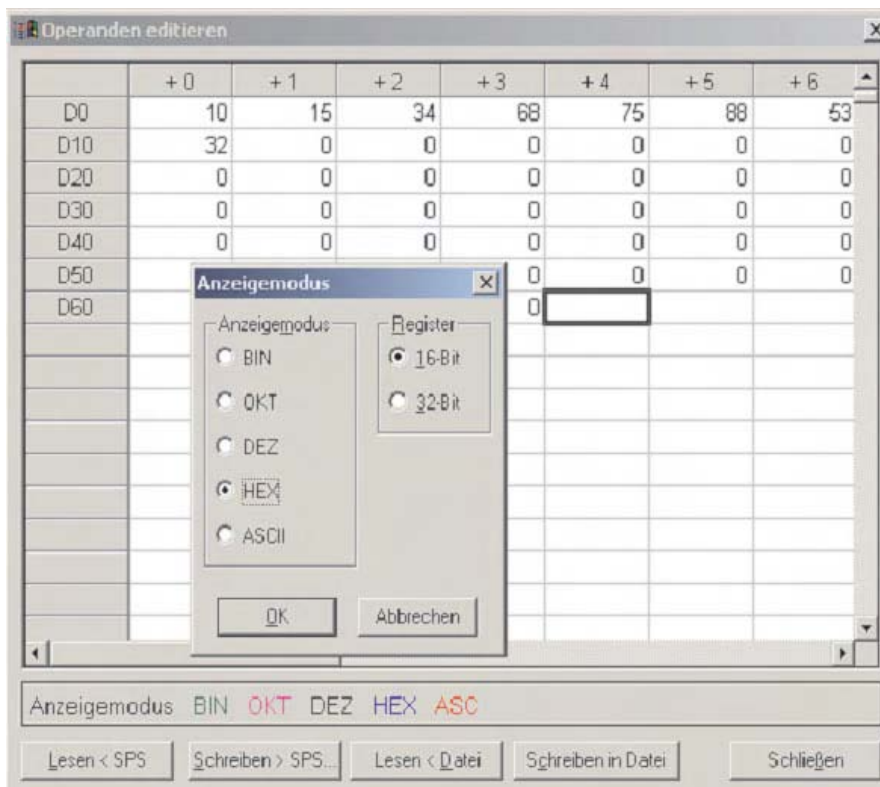
	+0	+1	+2	+3	+4	+5	+6
D0	0	0	0	0	0	0	0
D10	0	0	0	0	0	0	0
D20	0	0	0	0	0	0	0
D30	0	0	0	0	0	0	0
D40	0	0	0	0	0	0	0
D50	0	0	0	0	0	0	0
D60	0	0	0	0	0	0	0

Wenn Sie in der Tabelle mit der rechten Maustaste klicken, öffnet sich ein Menü, mit dem Sie z. B. Operanden löschen oder Werte suchen können.

- ④ Klicken Sie in eine Zelle, z. B. für D0, um sie zu markieren, und betätigen Sie dann die rechte Maustaste. Wählen Sie dann den **Anzeigemodus**.



Danach können Sie das Anzeigeformat für diesen Operanden auswählen. Klicken Sie auf **HEX**.



Die Werte der ausgewählten Zelle werden nun hexadezimal dargestellt, während das Anzeigeformat der anderen Zellen nicht verändert wird. Dadurch wird die Funktion **Operanden editieren** sehr flexibel.

10 Online-Modus

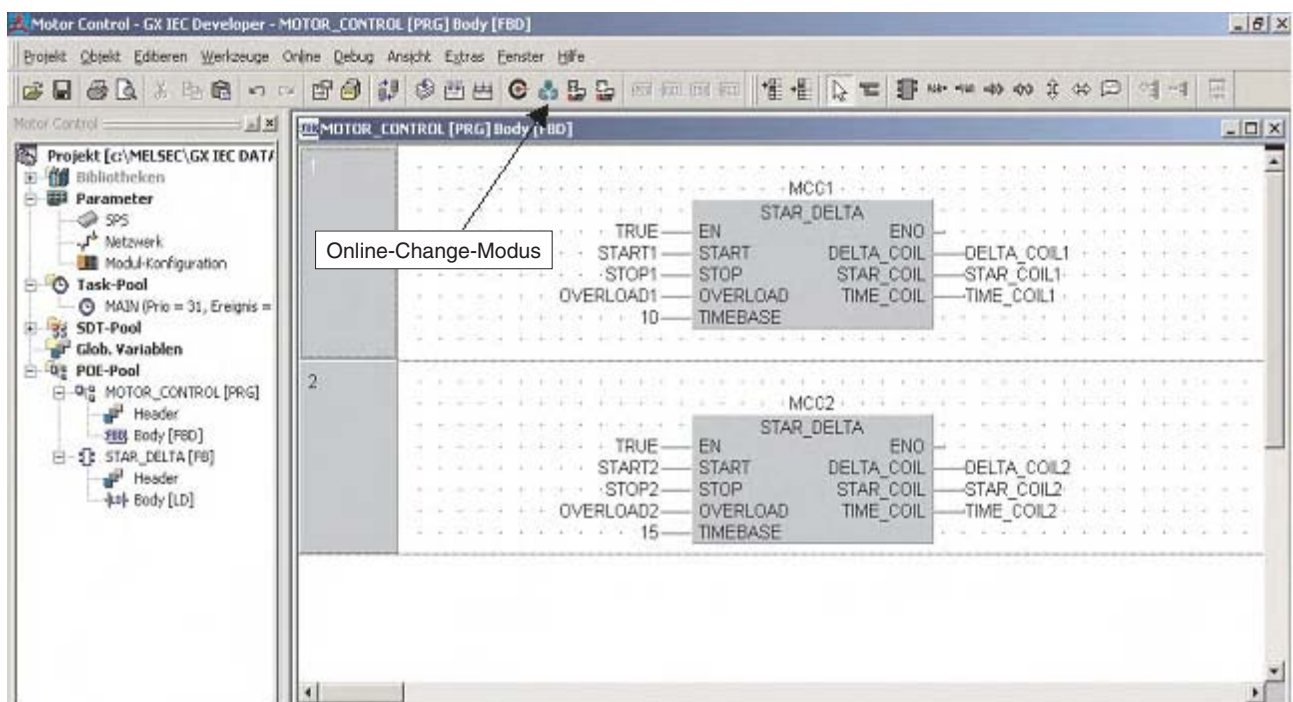
Der im Abschnitt 4.4 und im Kapitel 7 beschriebene Monitormodus eignet sich zum Beobachten von Operandenzuständen und der Programmausführung. Für den Fall, dass während der Überwachung das Programm in der SPS geändert werden soll, bietet Ihnen der GX IEC Developer zwei Möglichkeiten, die in folgenden Kapitel beschrieben werden.

Erzeugen Sie zuerst eine Kopie des aktuellen Projekts. Klicken Sie dazu im Menü **Projekt** auf dem Eintrag **Speichern unter...** Nennen Sie die Kopie „MOTOR_CONTROL_Mod“. Die folgende Beschreibungen beziehen sich auf dieses geänderte Programm.

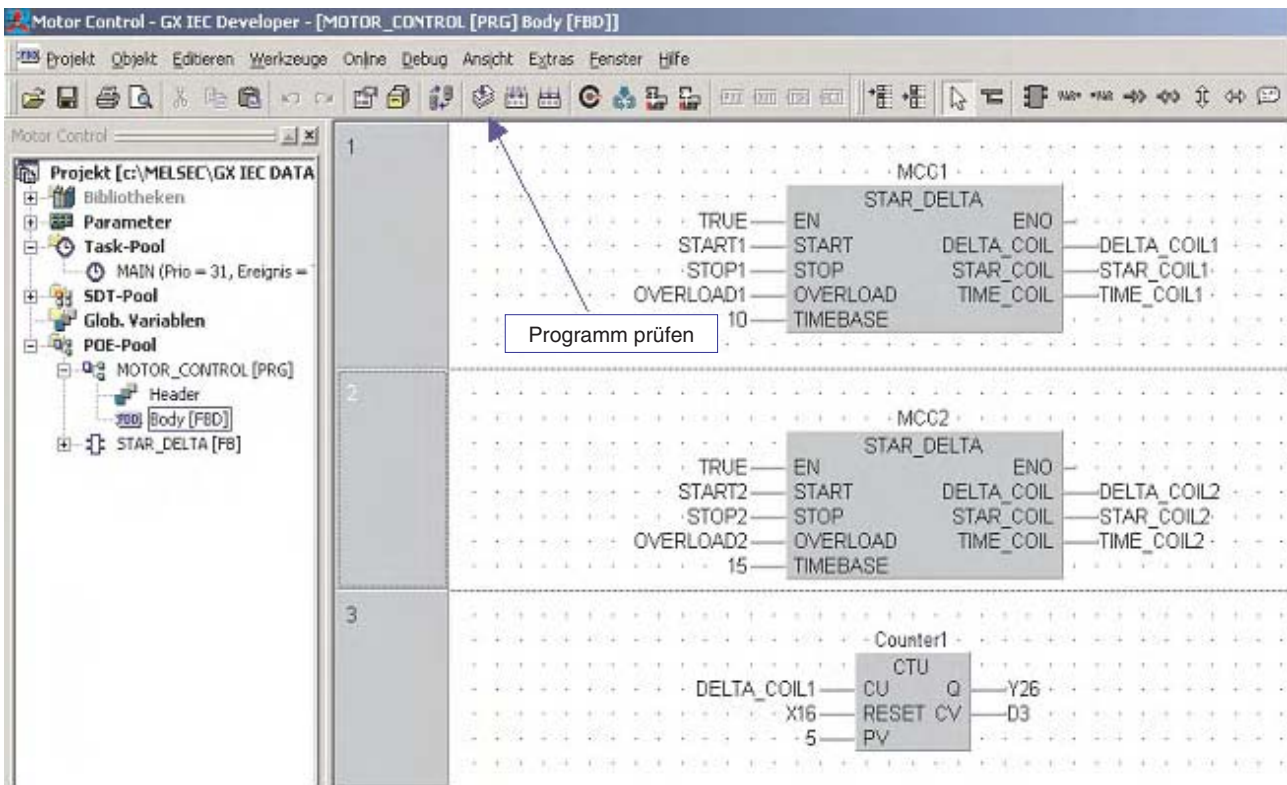
Kompilieren Sie das Projekt und übertragen Sie es in die SPS.

10.1 Online-Change-Modus

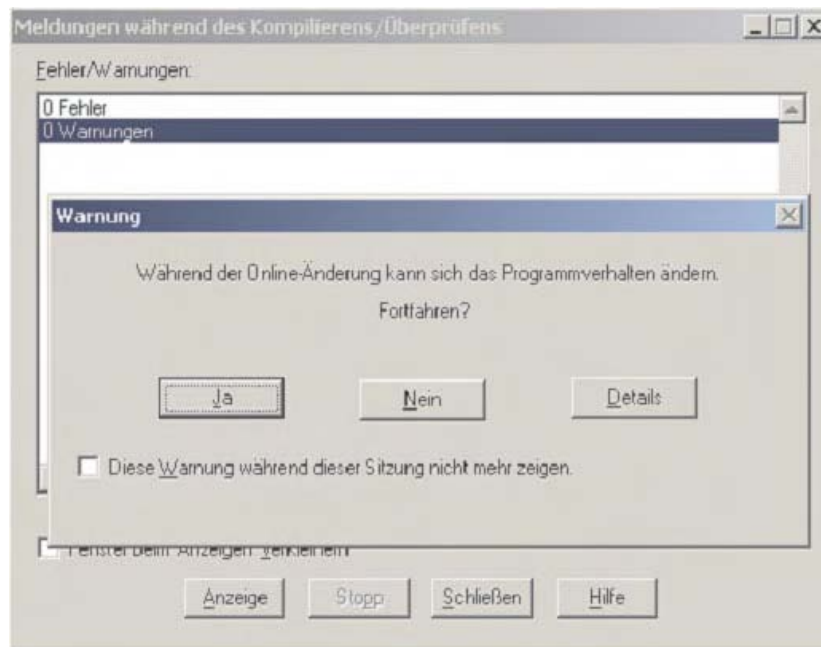
- ① Öffnen Sie den Body der POE „MOTOR_CONTROL“ und wählen Sie in der Werkzeugleiste **Online-Change-Modus**:



- ② Fügen Sie, wie unten gezeigt, ein zusätzliches Netzwerk ein.



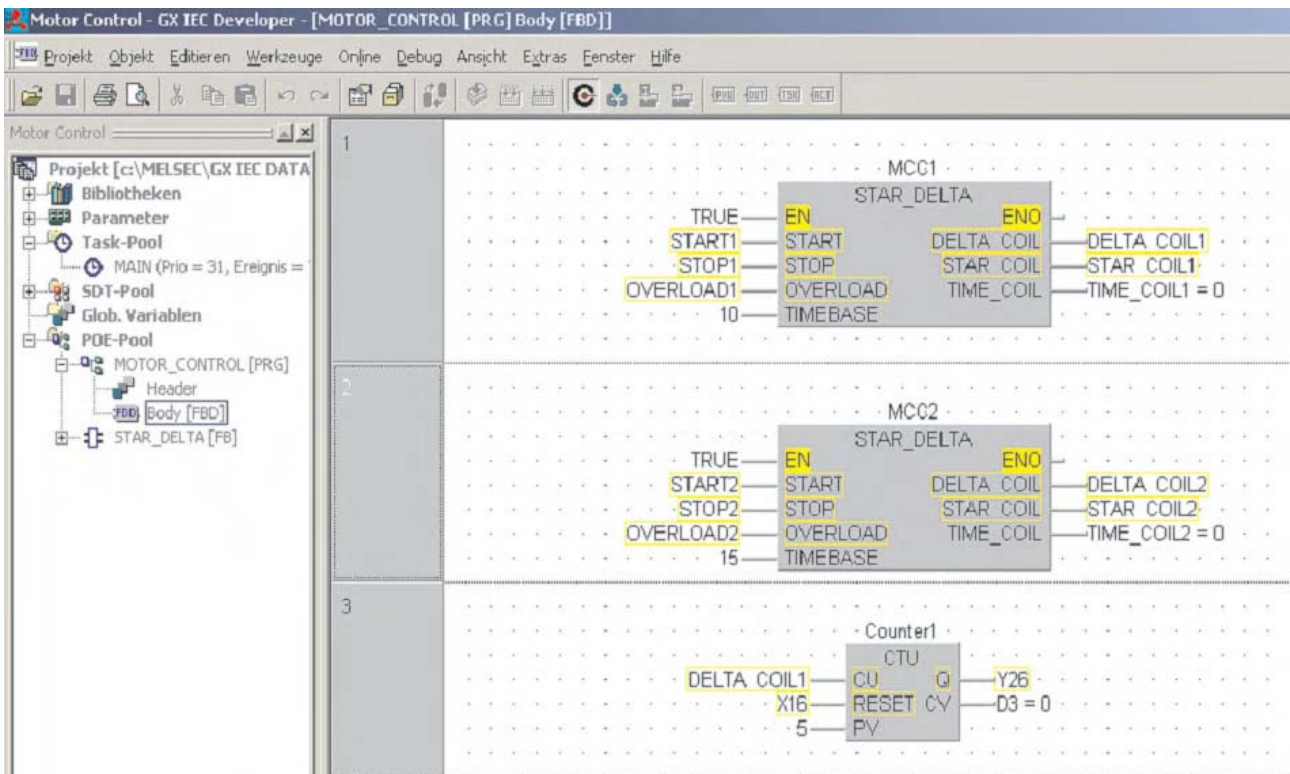
- ③ Klicken Sie dann auf ein anderes Netzwerk oder in der Werkzeugleiste auf das Schaltfeld „Prüfen“. Dadurch werden die Änderungen automatisch kompiliert und in die SPS übertragen. Vorher erscheint aber noch ein Warnhinweis:



HINWEIS

Eine Programmänderung im Online-Modus ist nur möglich, wenn der Code des bearbeiteten Projekts mit dem in der SPS identisch ist.

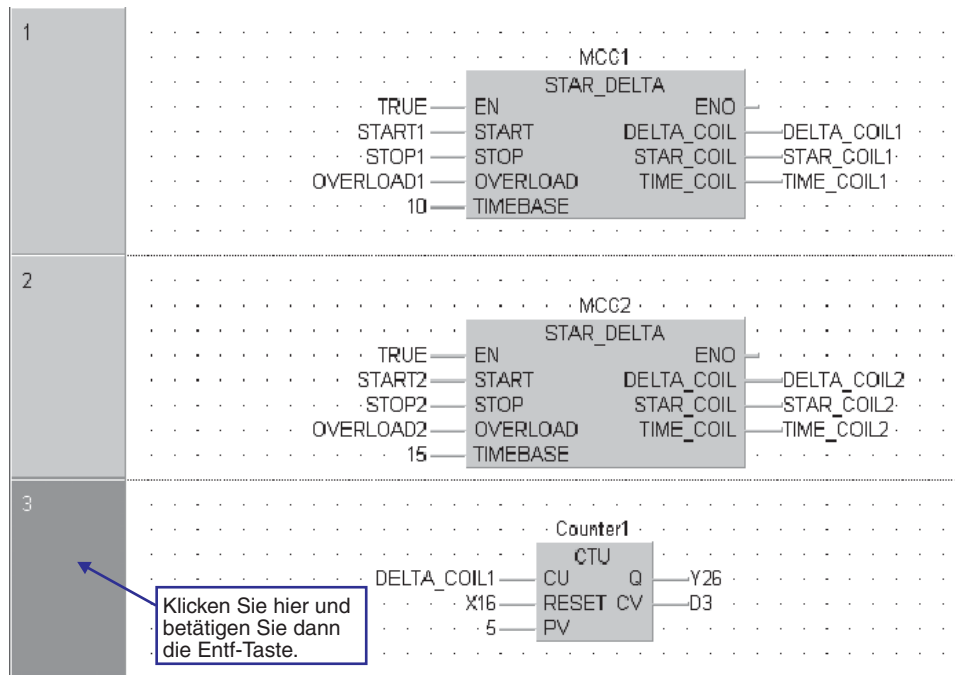
- ④ Aktivieren Sie den Monitormodus und beobachten Sie die Ausführung des geänderten Programms.



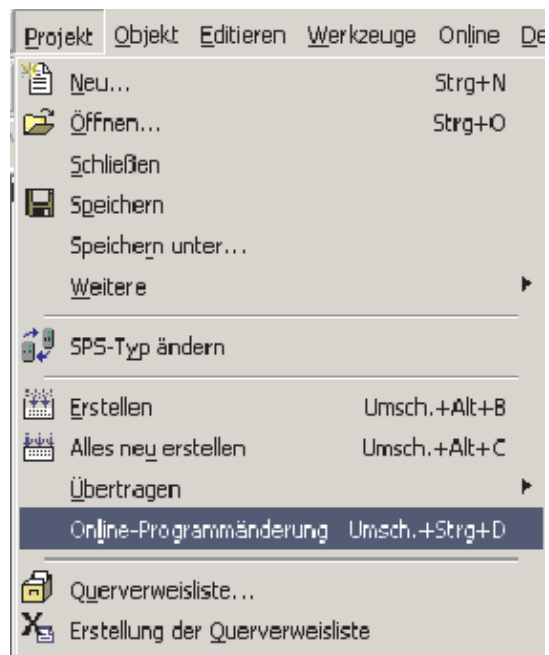
10.2 Online-Programmänderung

Wenn in einem laufenden Programm größere Änderungen vorgenommen werden sollen, wird die Funktion **Online-Programmänderung** empfohlen. Um zum Beispiel das im vorherigen Abschnitt eingefügte Netzwerk wieder zu löschen, führen Sie die folgenden Bedienschritte aus. Beachten Sie aber bitte, dass vor einer Änderung die Programme im GX IEC Developer und SPS identisch sein müssen.

- ① Klicken Sie in das 3. Netzwerk der POE „MOTOR_CONTROL“, um es zu markieren und betätigen Sie dann die Entfernen-Taste (Entf) der PC-Tastatur.

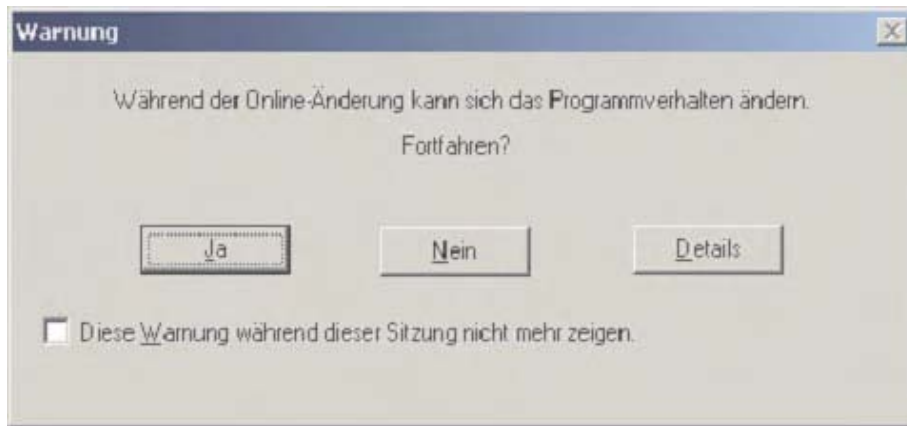


- ② Klicken Sie im Menü **Projekt** auf den Eintrag **Online-Programmänderung**. Dadurch werden die Änderungen automatisch kompiliert und in die SPS übertragen.

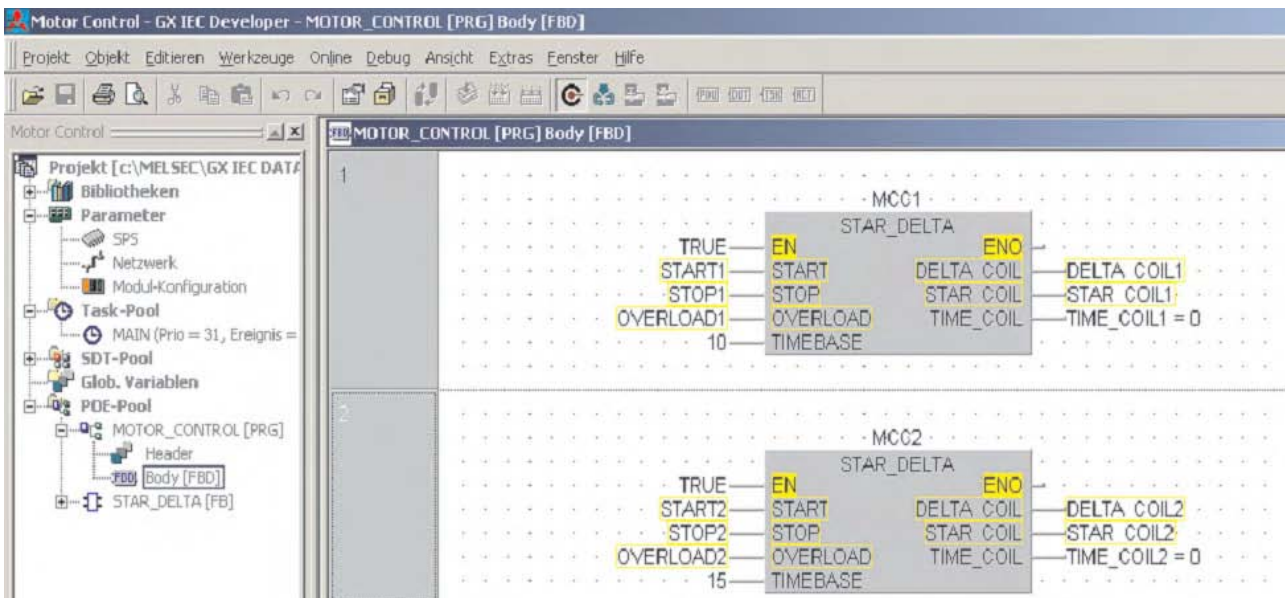


Vorher erscheint aber noch eine Warnmeldung. Die Programmänderung kann zu diesem Zeitpunkt noch abgebrochen oder fortgesetzt werden.

- ③ Klicken Sie auf **Ja** und warten Sie, bis der Transfer des Programms beendet ist.



- ④ Prüfen Sie die korrekte Ausführung des Programms im Monitor-Modus.



11 Strukturierter Datentyp (SDT)

In diesem Kapitel wird die Anwendung eines Strukturierten Datentyps (SDT) beschrieben. Das Erzeugen und die Verwendung eines SDT wird anhand des schon bekannten Beispiels „Motor Control“ erläutert.

Strukturierte Datentypen können vom Anwender selbst zusammengestellt werden. Dies ist hilfreich bei Programmen, die gemeinsame Elemente enthalten, wie zum Beispiel mehrere identische Stern-Dreieck-Schalter. Für so ein Programm kann eine SDT mit dem Namen „SD“ definiert werden, die verschiedene Elemente, wie z. B. INT, BOOL usw., enthält.

Bei der Deklaration der Globalen Variablenliste können Bezeichner vom Datentyp „SD“ angegeben werden. Das bedeutet, dass die vorher definierte Variablensammlung mit dem Namen „SD“ und den zur Steuerung jedes Motors benötigten Elementen verwendet wird. Die SDT kann, ebenso wie Funktionsbausteine, immer wieder verwendet werden. Dadurch wird die für die Programmierung benötigte Zeit reduziert.

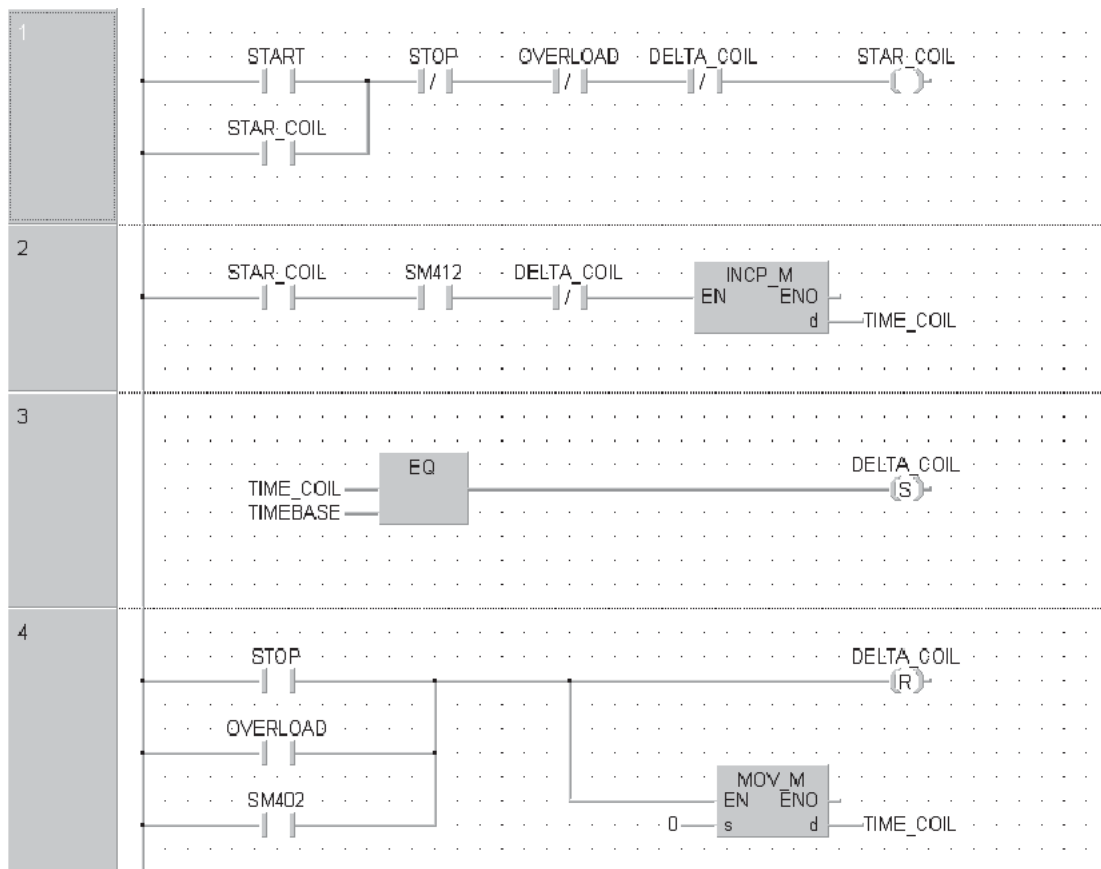
Falls in „SD“ ein Element mit der Bezeichnung „START“ enthalten ist, kann es für jede Instanz des Funktionsbausteins „Star Delta“ verwendet und in der Globalen Variablenliste deklariert werden (z. B. STAR_DELTA1.START, STAR_DELTA2.START usw.).

Das bedeutet, dass von einer Deklaration viele Anleitungen verwendet werden können. Das ist besonders bei der Gruppierung und Übermittlung von Daten an eine Prozessvisualisierung von Vorteil. Durch die reduzierte Datenmenge und die zusammenhängenden Daten wird die Kommunikation gegenüber dem Austausch vieler einzelner, nicht zusammenhängender Daten, schneller und effizienter.

11.1 Beispiel für die Anwendung einer SDT

- ① Legen Sie ein neues Projekt mit dem Namen „Motor Control DUT“ an.
- ② Legen Sie eine neue Programm-POE mit dem Namen MOTOR_CONTROL an.
- ③ Richten Sie im Task-Pool eine neue Task ein und nennen Sie diese Task MAIN. Rufen Sie in dieser Task die POE MOTOR_CONTROL auf.
- ④ Programmieren Sie einen neuen Funktionsbaustein „STAR_DELTA“. Das Programm ist unten abgebildet. Sie können aber auch den schon bestehenden Funktionsbaustein aus dem Projekt „Motor Control“ kopieren.

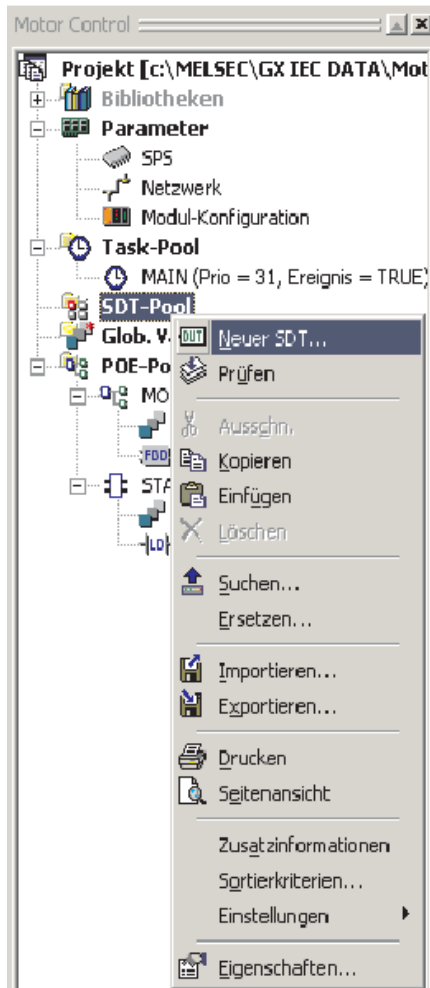
Body: STAR_DELTA




Header: STAR_DELTA

	Klasse	Bezeichner	Typ	Start	Kommentar
0	VAR_INPUT	START	BOOL	... FALSE	
1	VAR_INPUT	STOP	BOOL	... FALSE	
2	VAR_INPUT	OVERLOAD	BOOL	... FALSE	
3	VAR_INPUT	TIMEBASE	INT	... 0	
4	VAR_OUTPUT	DELTA_COIL	BOOL	... FALSE	
5	VAR_OUTPUT	STAR_COIL	BOOL	... FALSE	
6	VAR_OUTPUT	TIME_COIL	INT	... 0	

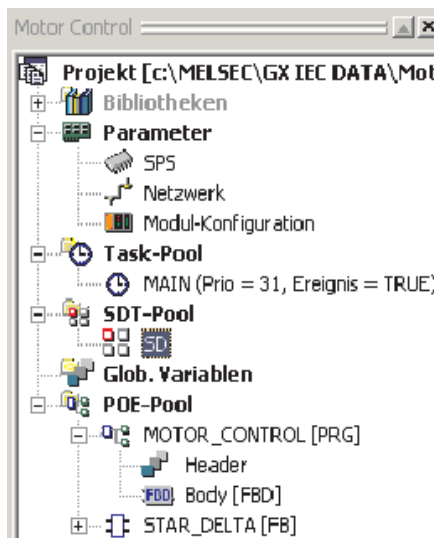
Der Header enthält die Definitionen (Maske) der Datentypen, die bei der Erstellung der SDT „SD“ verwendet werden.



- ⑤ Erzeugen Sie eine neue SDT, indem Sie im Navigatorfenster mit der rechten Maustaste **SDT-Pool** klicken. Alternativ können Sie auch in der Werkzeugleiste auf das Schaltfeld  klicken.



- ⑥ Geben Sie in das Dialogfenster den Namen der neuen SDT ein. Für dieses Beispiel tippen Sie „SD“ ein und klicken dann auf **OK**.



Die neue SDT wird nun im **SDT-Pool** im Navigatorfenster angezeigt.

- ⑦ Öffnen Sie die SDT, indem Sie doppelt auf deren Namen klicken. Dadurch bekommen Sie die folgende Eingabemöglichkeit.

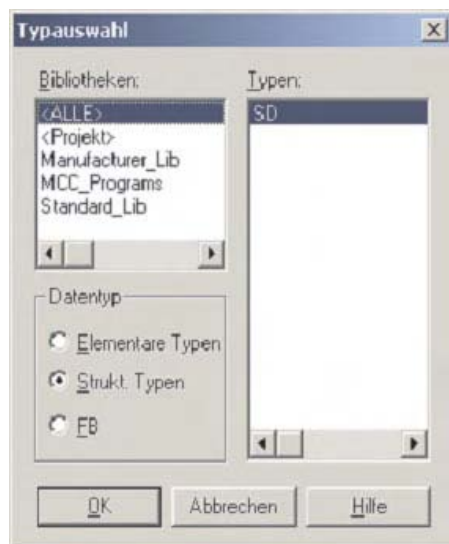
	Bezeichner	Typ	Start	Kommentar
0			...	

- ⑧ Geben Sie in die SDT „SD“ die folgenden Daten ein:

	Bezeichner	Typ	Start	Kommentar
0	STAR	BOOL	... FALSE	
1	DELTA	BOOL	... FALSE	
2	START	BOOL	... FALSE	
3	STOP	BOOL	... FALSE	
4	O_L	BOOL	... FALSE	
5	TB	INT	... 0	
6	TV	INT	... 0	

- ⑨ Schließen Sie die SDT und speichern Sie das Programm.
 ⑩ Öffnen Sie die Globale Variablenliste und deklarieren Sie die beiden Variablen STAR_DELTA1 und STAR_DELTA2 neu.
 ⑪ Klicken Sie auf das Feld ... rechts neben **Typ** und wählen Sie für beide Einträge **Strukt. Typen**.

	Klasse	Bezeichner	MIT-Adr.	IEC-Adress	Typ	Start
- 0	VAR_GLOBAL	STAR_DELTA1			SD	...
- 1	VAR_GLOBAL	STAR_DELTA2			SD	...



- ⑫ Klicken Sie dann in das Feld **MIT-Adr.** der Variablen STAR_DELTA1, um SDT-Einträge anzuzeigen.

	Klasse	Bezeichner	MIT-Adr.	IEC-Adress	Typ	Start
- 0	VAR_GLOBAL	STAR_DELTA1			SD	...
- 1	VAR_GLOBAL	STAR_DELTA2			SD	...

Klicken Sie zur Auswahl hier.

Danach wird dieses Fenster angezeigt:

Adressen der SDT-Variable

STAR_DELTA1 (SD)

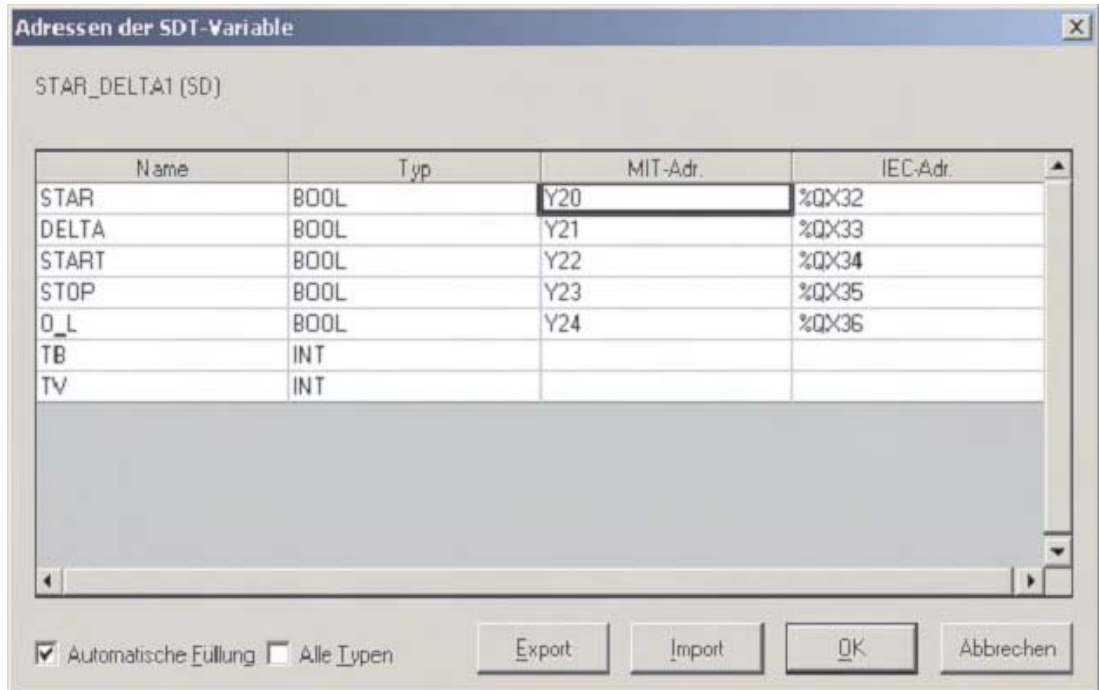
Name	Typ	MIT-Adr.	IEC-Adr.
STAR	BOOL		
DELTA	BOOL		
START	BOOL		
STOP	BOOL		
O_L	BOOL		
TB	INT		
TV	INT		

Automatische Füllung Alle Typen

Export Import OK Abbrechen

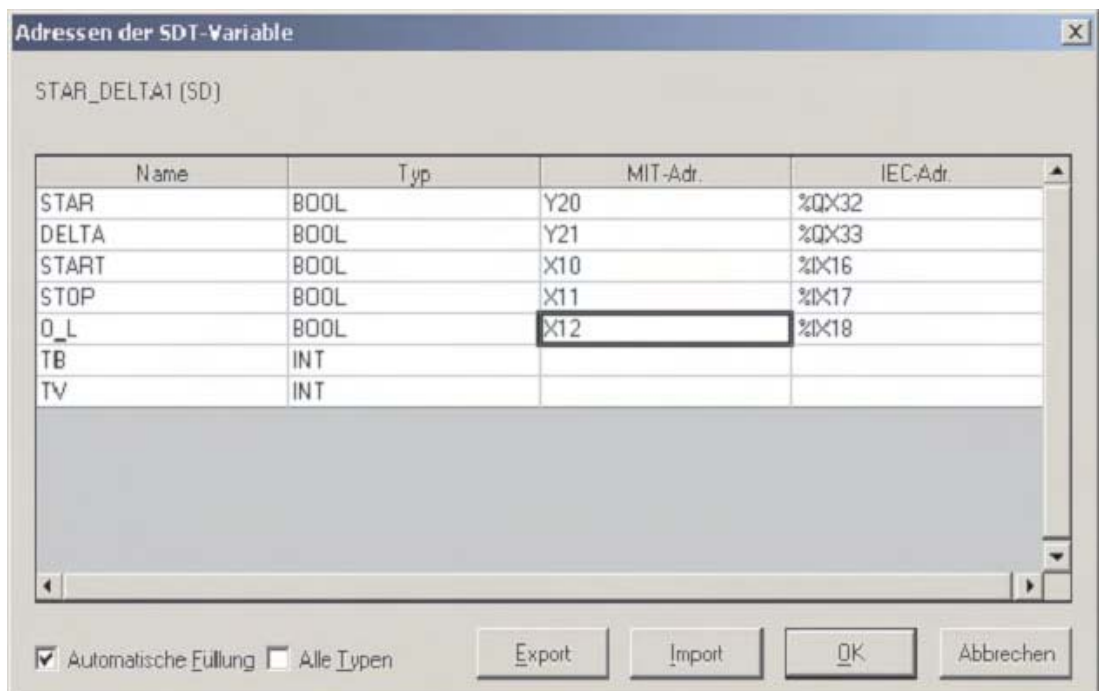
11.2 Automatisches Füllen von Variablen

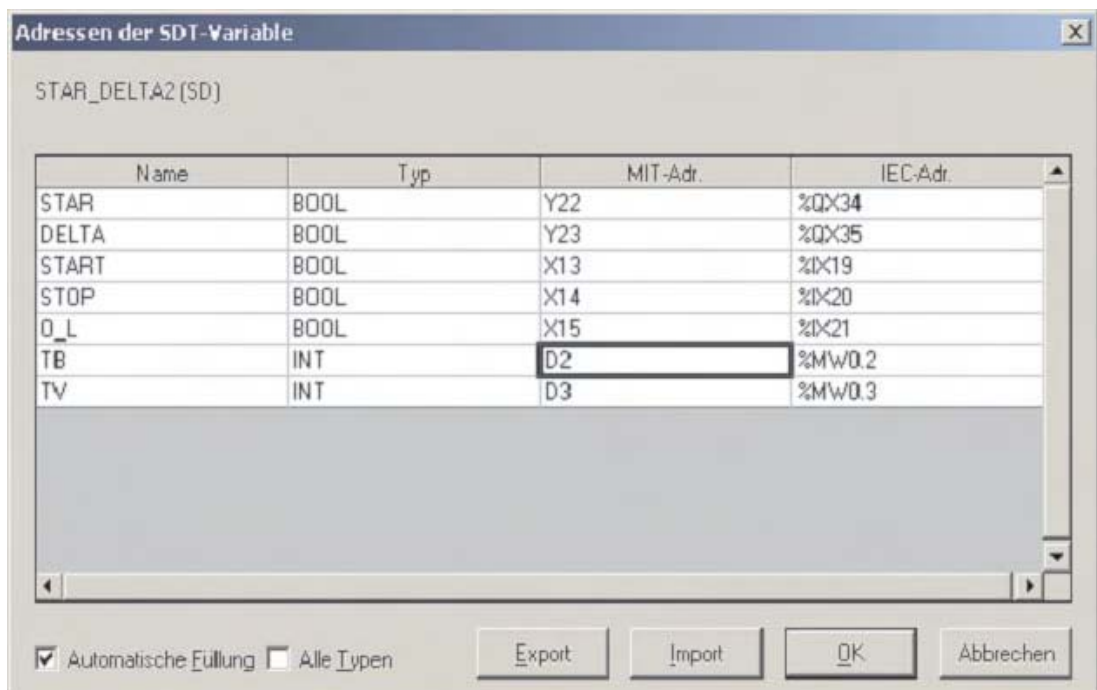
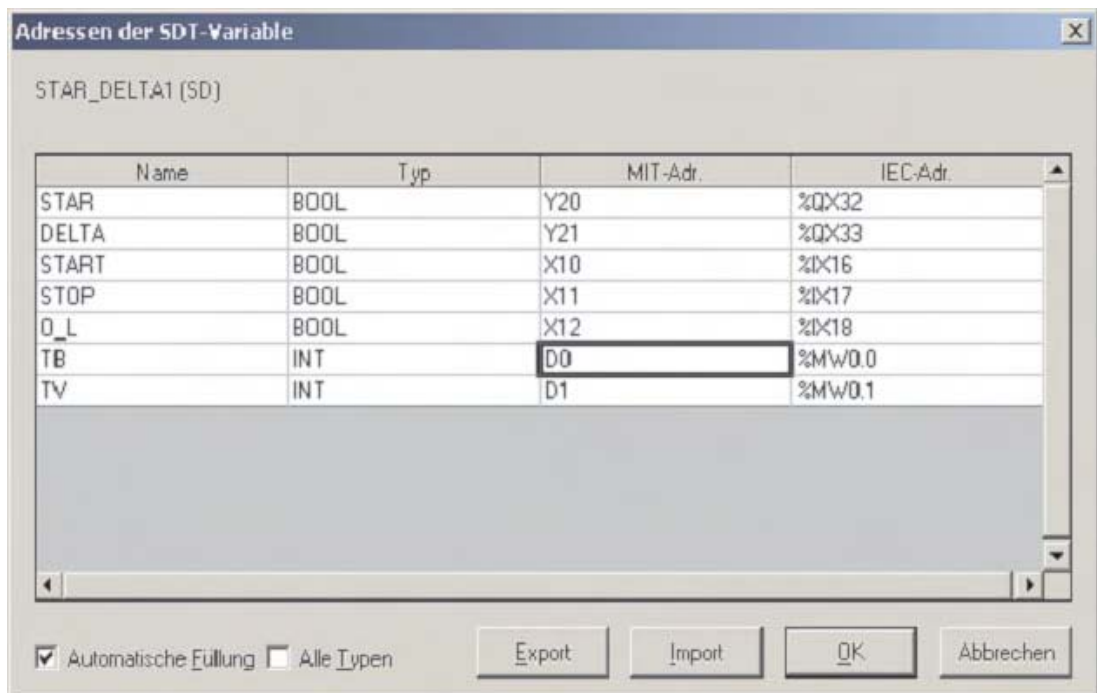
- ① Deaktivieren Sie die Option **Alle Typen**. Diese Option kann bei gemischten Variablentypen nicht verwendet werden.
- ② Geben Sie im Feld **MIT-Adr.** der Variablen „DELTA“ Y20 ein.



Durch die Option **Automatische Füllung** werden alle Variablen vom Typ BOOL mit fortlaufenden Adressen belegt. Obwohl diese Funktion in vielen Anwendungen sehr nützlich ist, führt sie in diesem Beispiel nur teilweise zum Erfolg.

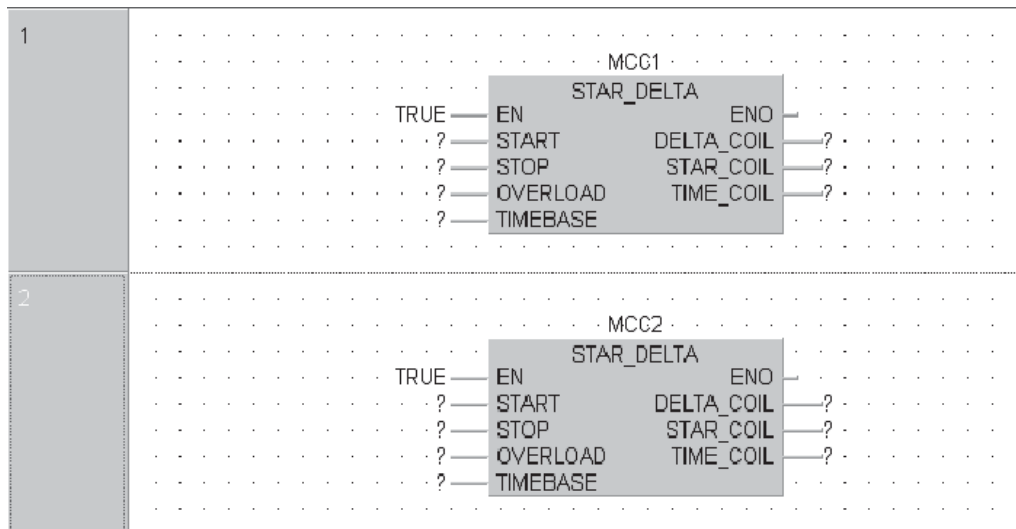
- ③ Überschreiben Sie daher die Adressen der Variablen „START“ und „STOP“ mit X10 bzw. X11.





	Klasse	Bezeichner	MIT-Adr.	IEC-Adresse	Typ	Start
+0	VAR_GLOBAL	STAR_DELTA1	STAR:	STAR:	SD	...
+1	VAR_GLOBAL	STAR_DELTA2	STAR:	STAR:	SD	...

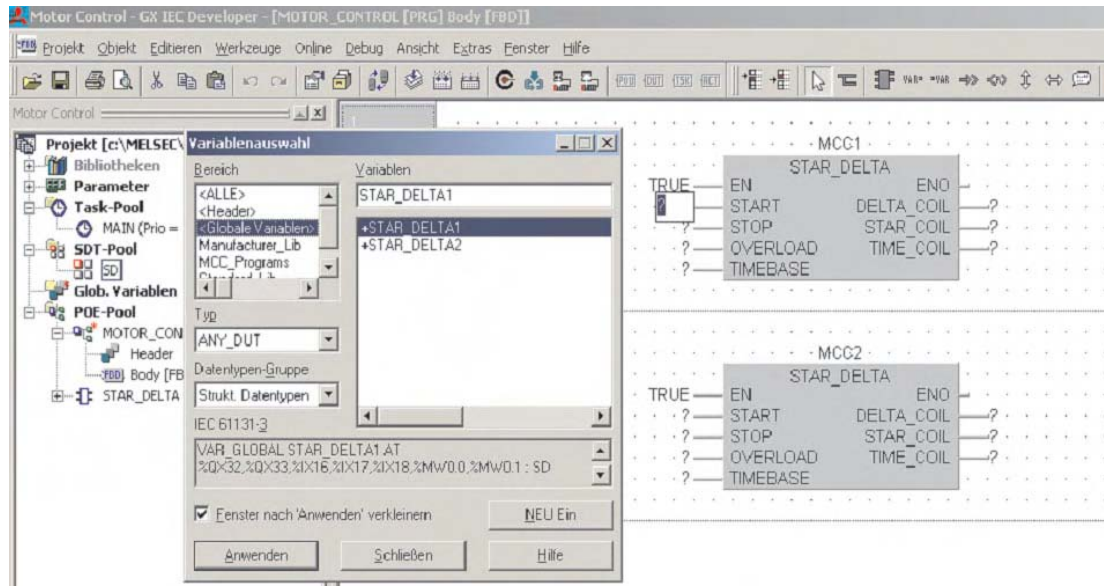
Öffnen Sie die POE MOTOR_CONTROL und fügen Sie zwei Instanzen des anwenderdefinierten Funktionsbausteins STAR_DELTA ein:



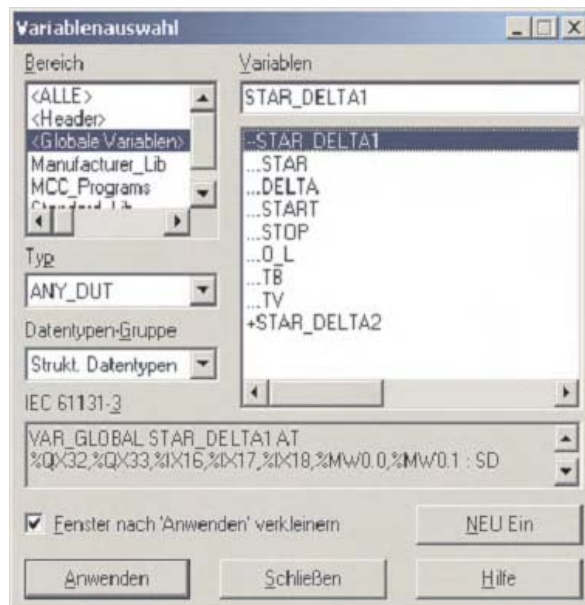
11.3 SDT als Variable eines Funktionsbausteins

Um einen Strukturierten Datentyp (SDT) als Variable eines Funktionsbausteins zu verwenden...

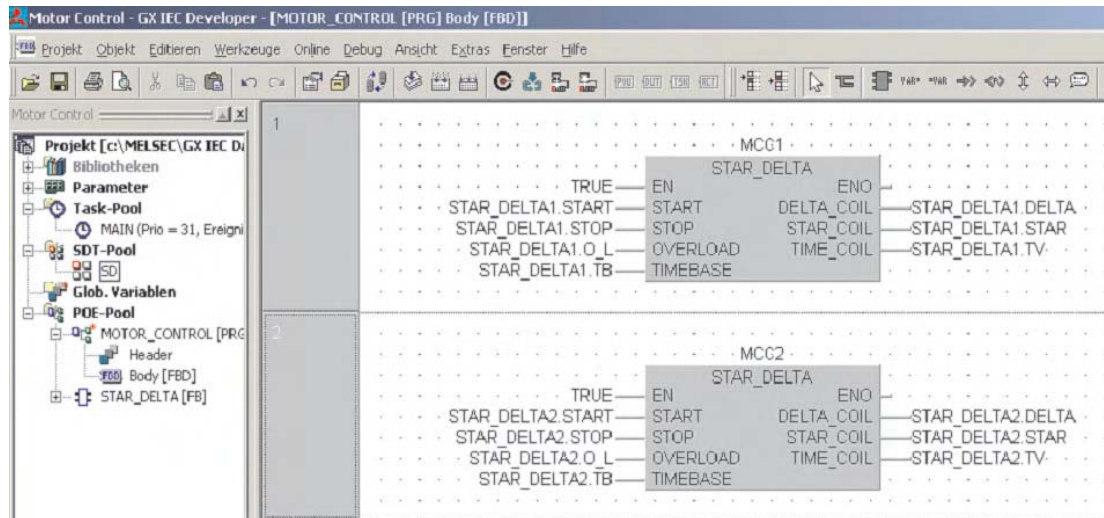
- ① ...klicken Sie mit der rechten Maustaste auf die Variable oder betätigen die Taste F2 der PC-Tastatur. Dadurch wird das Dialogfenster Variablenauswahl geöffnet.



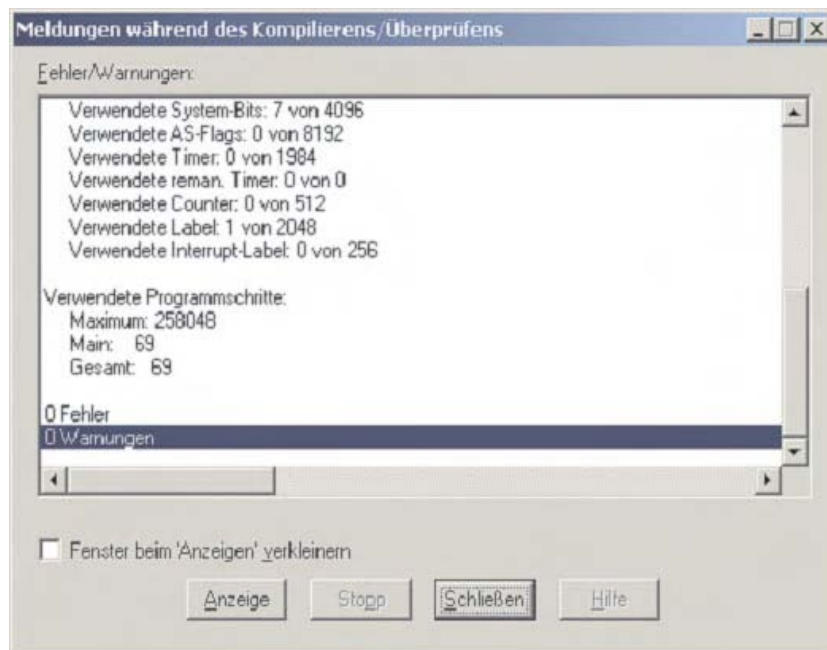
- ② Wählen Sie als Bereich **Header**, als Datentypen-Gruppe **Strukt. Datentypen** und als Typ **ANY_DUT**.
- ③ Klicken Sie doppelt auf +STAR_DELTA1. Dadurch wird die Liste um die SDT-Elemente erweitert.



- ④ Weisen Sie den beiden Funktionsbausteinen STAR_DELTA in der POE MOTOR_CONTROL die SDT-Elemente als Ein- und Ausgangsvariablen zu:

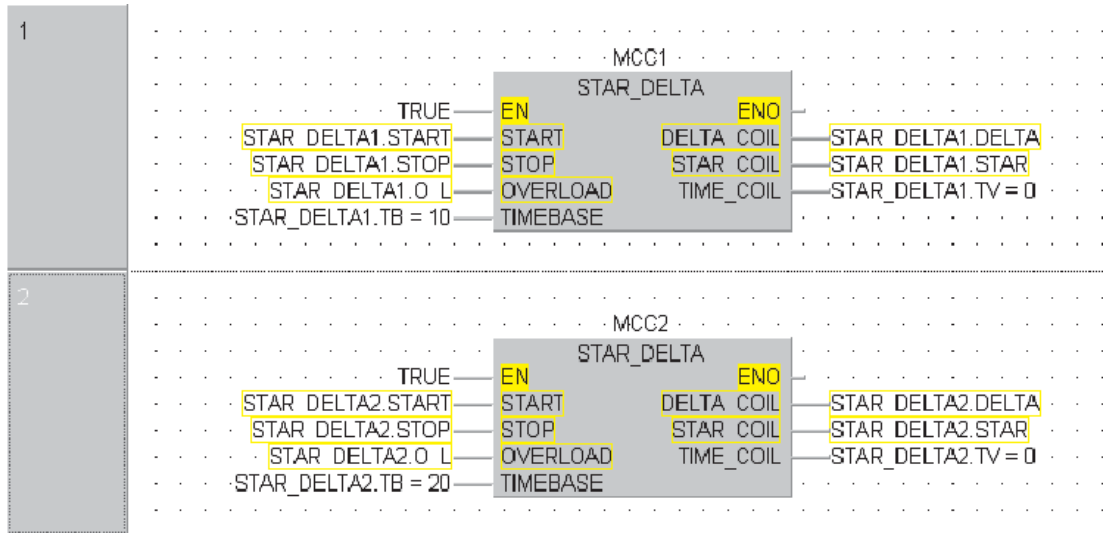


Sichern und kompilieren Sie anschließend das Projekt.



Transferieren Sie das Projekt in die SPS und prüfen Sie die Funktion im Monitormodus. Zum Betrieb der Funktionsbausteine ist es erforderlich, Werte in die Eingangsvariablen TIMEBASE einzutragen. Ändern Sie dazu die Inhalte der Variablen STAR_DELTA1.TB und STAR_DELTA2.TB so, wie in Abschnitt 7.5 beschrieben.

Simulieren Sie, wie in der folgenden Abbildung gezeigt, den Betrieb beider Funktionsbausteine und prüfen Sie, ob sie so arbeiten, wie erwartet.



12 Arrays

12.1 Übersicht

Ein Array (Feld) besteht aus Variablen des gleichen Typs. Zum Beispiel ist ein **ARRAY [0..2] OF INT** ein eindimensionales Array mit drei Integer-Elements (0,1,2). Falls als Startadresse des Arrays das Register D0 angegeben wird, besteht das Array aus D0, D1 and D2.

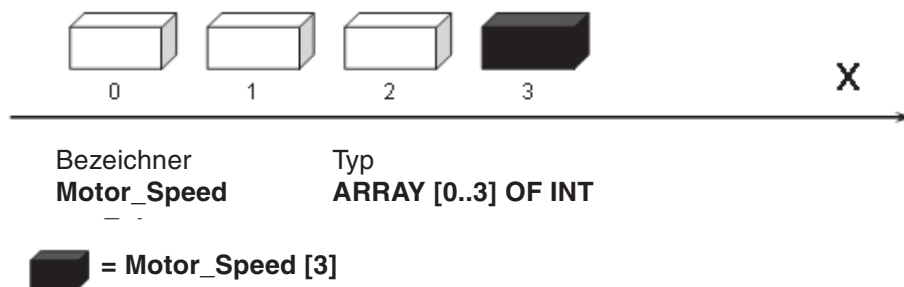
Im Programm können einzelne Array-Elemente angegeben werden. Zum Beispiel wird bei der Angabe von „Motor_Volts[1]“ und „ Motor_Volts[2]“ auf die Register D1 und D2 zugegriffen.

Arrays können bis zu 3 Dimensionen haben. Zum Beispiel hat ARRAY [0..2, 0..4] drei Elemente in der ersten und fünf Elemente in der zweiten Dimension.

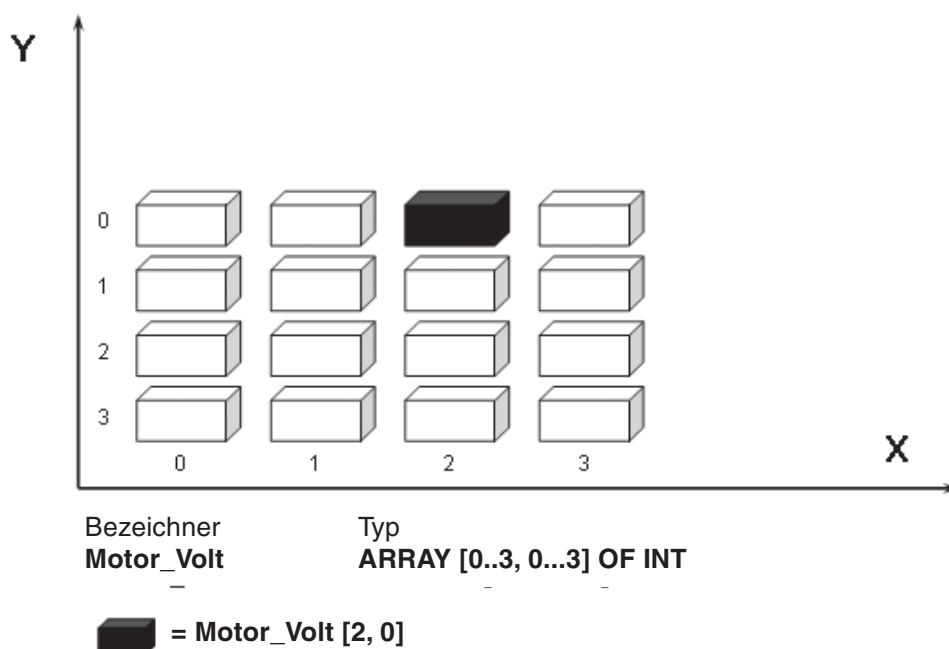
Arrays bieten eine bequeme Möglichkeit, Bezeichner zu vergeben. Durch nur eine Definition in der Liste der Lokalen oder der Globalen Variablen kann auf viele Elemente zugegriffen werden.

Zum besseren Verständnis werden die drei Array-Typen im folgenden grafisch dargestellt.

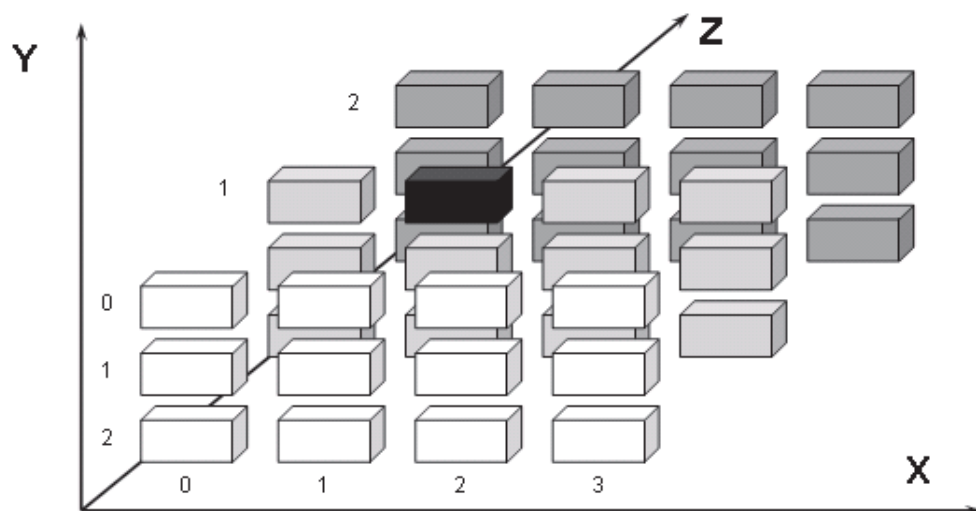
Eindimensionales Array



Zweidimensionales Array



Dreidimensionales Array



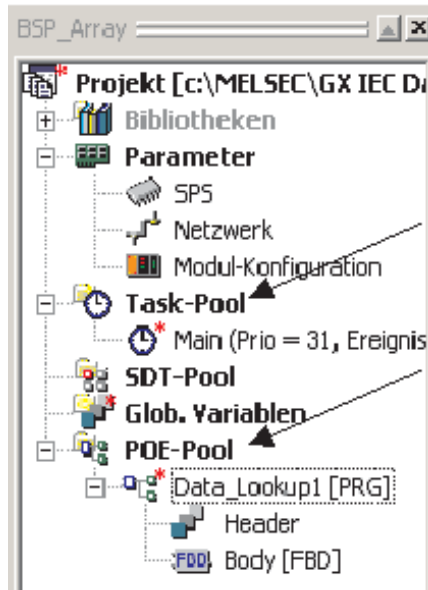
Bezeichner	Typ
Motor_Strom	ARRAY [0..3, 0..2, 0..2] OF INT

 = Motor_Volt [1, 0, 1]

12.2 Beispiel für ein eindimensionales Array

Das folgende Beispiel soll die Bildung und Verwendung eines eindimensionalen Arrays demonstrieren. Das Array ist 10 Worte lang und belegt die Datenregister D100 bis D109. In diesem Beispiel werden nur Standard IEC-Operatoren, Funktionen und Funktionsbausteine verwendet.

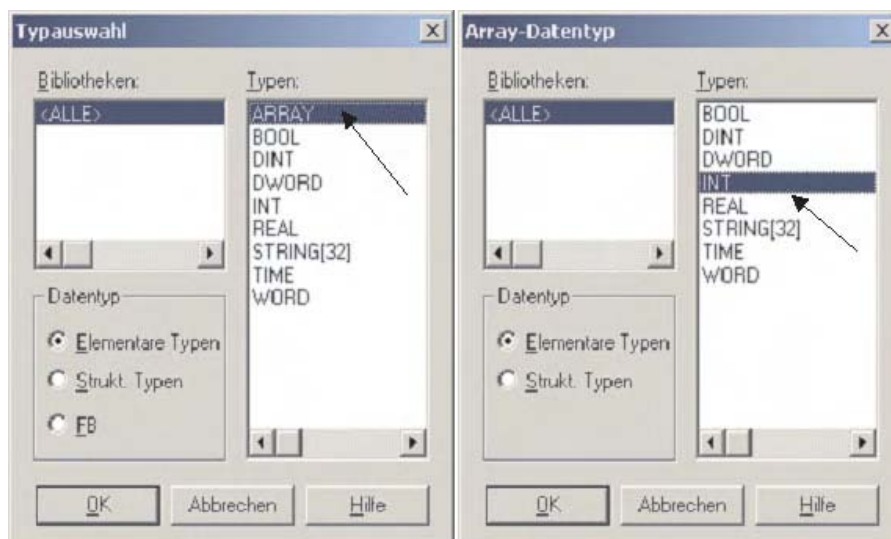
- ① Legen Sie ein neues Projekt an. Definieren Sie darin ein neue POE der Programmklasse PRG (Programm). Programmiert werden soll diese POE in der **Funktionsbausteinsprache**. Nennen Sie diese POE „Data_Lookup1“.
- ② Legen Sie im Task-Pool eine neue Task mit dem Namen „Main“ an. Rufen Sie in dieser Task die POE „Data_Lookup1“ auf.



- ③ Öffnen Sie die Globale Variablenliste und nehmen Sie die folgenden Einträge vor:

	Klasse	Bezeichner	MIT-Adr.	IEC-Adress	Typ	Start
0	VAR_GLOBAL	Data_Clock	X10	%IX16	BOOL	FALSE
1	VAR_GLOBAL	Data_Store	D100	%MWD.100	ARRAY [0..9] OF INT	{10(0)}
2	VAR_GLOBAL	Data_Lookup	D10	%MWD.10	INT	0
3	VAR_GLOBAL	Data_Pointer	D11	%MWD.11	INT	0

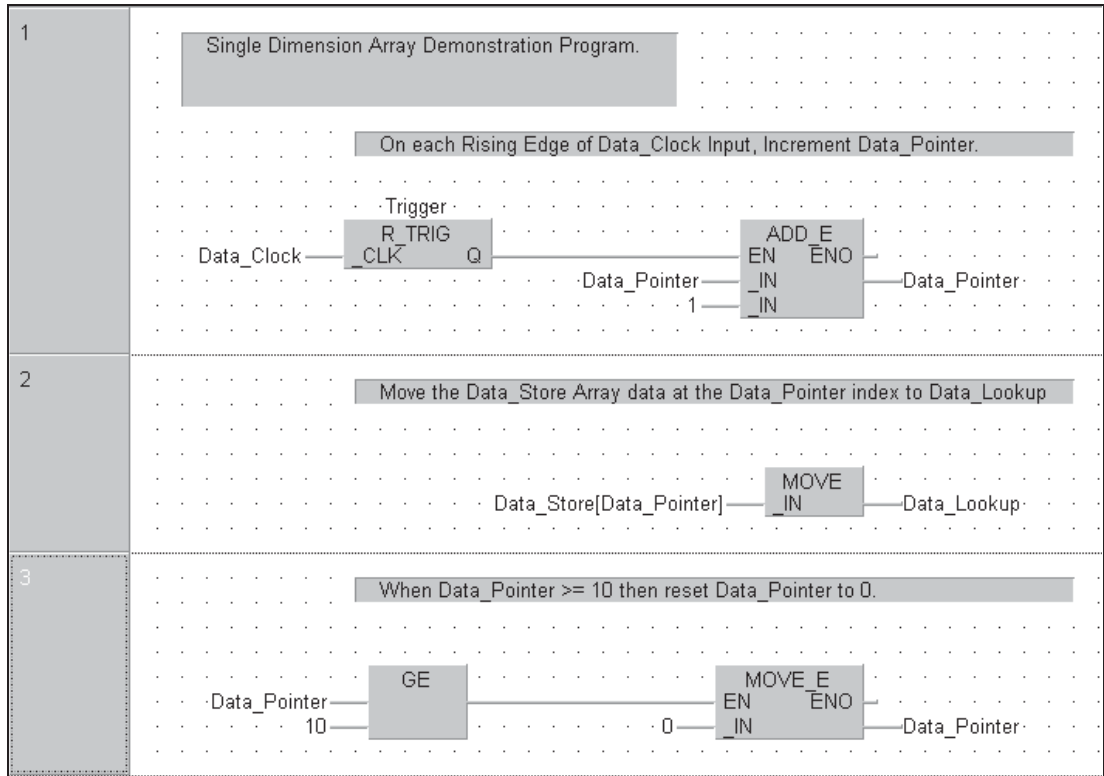
Variablen vom Typ „Array“ werden so eingegeben:



Bitte beachten Sie, dass in der Globalen Variablenliste für die Größe des Array die Standardwerte ARRAY [0..3] OF INT eingesetzt werden. Passen Sie die Größe so wie unten gezeigt dem in diesen Beispiel verwendeten Array an ([0..9] of INT).

```
1 VAR_GLOBAL Data_Store D100 [%MWD.100] ARRAY [0..9] OF INT [10x0]
```

④ Öffnen Sie die POE „Data_Lookup1“ und geben Sie das folgende Programm ein:

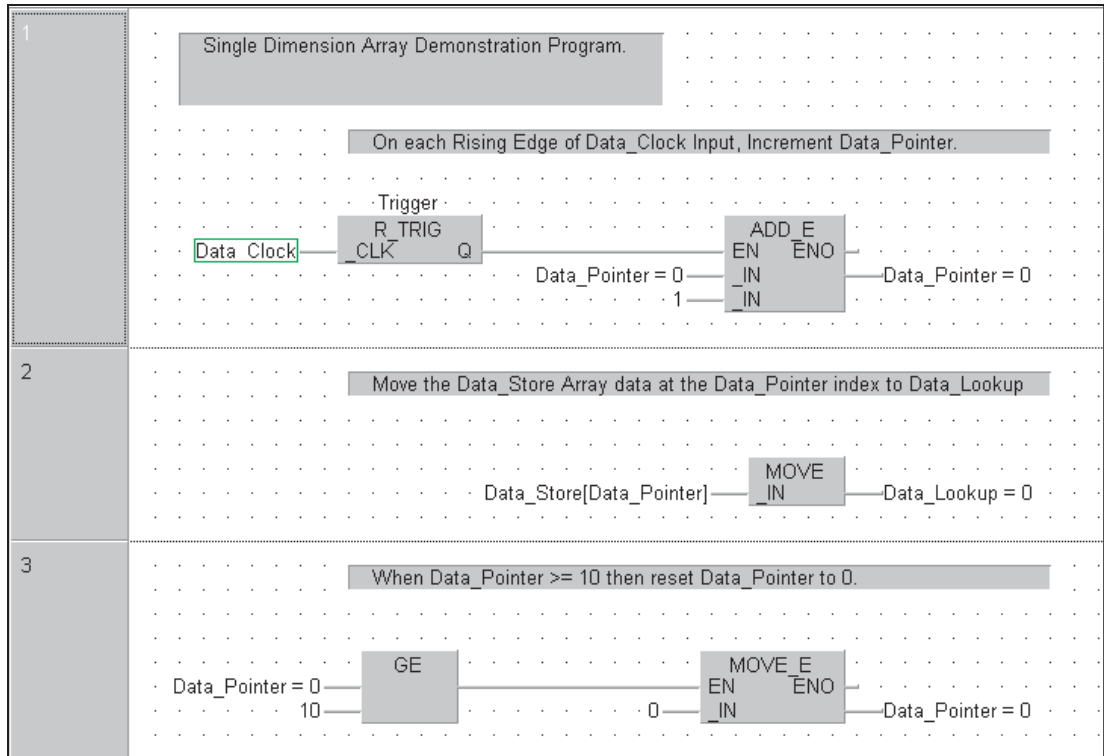


HINWEIS | Definieren Sie den Funktionsbaustein „R_TRIG“ mit dem Instanznamen „Trigger“.

⑤ Prüfen Sie, ob der Header so aussieht:

	Klasse	Bezeichner	Typ	Start	Kor
0	VAR_GLOBAL	Trigger	R_TRIG	...	

- ⑥ Speichern und kompilieren Sie das Programm.
- ⑦ Übertragen Sie das Programm in die SPS.
- ⑧ Beobachten Sie die Ausführung des Programms im Monitormodus.

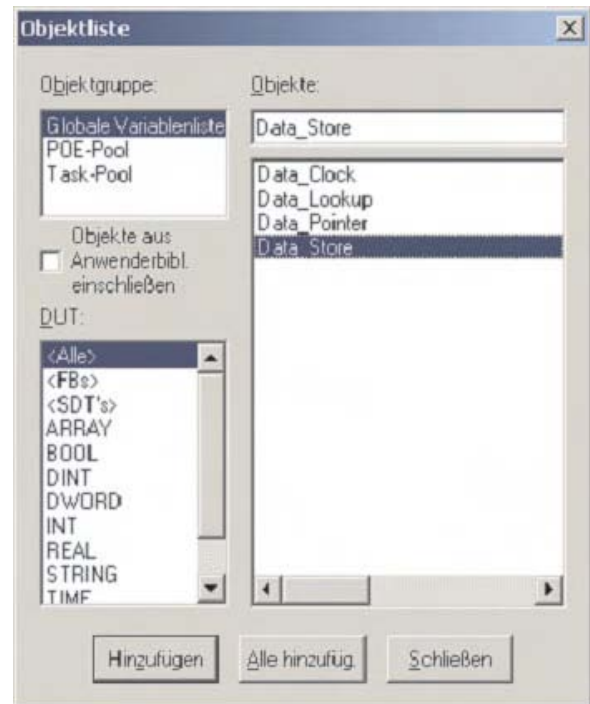


Damit das Programm einwandfrei arbeitet, müssen Werte in die SPS-Operanden eingetragen werden, die durch das Array belegt werden. Dies kann auf zwei verschiedene Arten erreicht werden:

- Wählen Sie im Menü **Debug** die Funktion **Operanden editieren** (siehe Kapitel 9). Mit **Register einfügen** geben Sie D100 bis D109 ein. In diese 10 Register geben Sie dann beliebige Werte zwischen -32768 und +32767 ein und übertragen diese Werte in die SPS (Schaltfeld **Schreiben > SPS**).
- oder Sie öffnen im Menü Online den **Entry-Data-Monitor**.
 - Klicken Sie mit der rechten Maustaste in die Tabelle. Dadurch wird das unten abgebildete Menü angezeigt. Wählen Sie **Objekte einfügen**.

Pos	Adresse (MIT)	Name	Wert (DEZ)
1			
2		Objekte einfügen...	F2
3		Nächstes Objekt	F3
4		Festgelegte Eingänge einfügen	
5		Gesetzte Eingänge einfügen	
6		Gesetzte Ausgänge einfügen	
7		Lösche Operandendatei	
8		Zeile einfügen	Einfg
9		Löschen	Entf
10		Alles löschen	
11			
12		Lesen < SPS	
13		Schreiben > SPS...	
14		Lesen aus Datei	
15		Schreiben in Datei	
16		Konfiguration...	
17		Immer oben	
18			

- Aus der Objektliste wählen Sie **Data_Store** und klicken anschließend auf **Hinzufügen**.



- Weil der Variablenname „Data_Store“ auf ein Array verweist, wird ihm vom System ein Pluszeichen (+) vorangestellt. Durch einen Klick auf den Variablennamen wird die Liste der Operanden erweitert:

Pos	Adresse (MIT)	Name	Wert (DEZ)
1		-Data Store	
2	D100	[0]	0
3	D101	[1]	0
4	D102	[2]	0
5	D103	[3]	0
6	D104	[4]	0
7	D105	[5]	0
8	D106	[6]	0
9	D107	[7]	0
10	D108	[8]	0
11	D109	[9]	0

- Wenn Sie auf das Minuszeichen (-) vor dem Variablennamen klicken, wird die Liste der Array-Elemente wieder verkleinert.
- Geben Sie 10 beliebige Werte zwischen -32768 und +32767 ein:

Pos	Adresse (MIT)	Name	Wert (DEZ)
1		-Data_Store	
2	D100	[0]	1234
3	D101	[1]	4321
4	D102	[2]	7654
5	D103	[3]	4236
6	D104	[4]	17
7	D105	[5]	32766
8	D106	[6]	8912
9	D107	[7]	43
10	D108	[8]	186
11	D109	[9]	9999

13 Der Umgang mit Bibliotheken

13.1 Anwenderbibliotheken

Alle Funktionen und Funktionsbausteine existieren bis jetzt nur im aktuellen Projekt und können auch nur in diesem Projekt verwendet werden.

Eine Anwenderbibliothek ist eine Sammlung von POEs, Funktionen, Funktionsbausteinen usw., die vom Anwender selbst erstellt worden sind. Diese Bibliotheken sind projektübergreifend verfügbar, das heißt, dass sie auch in anderen Projekte verwendet werden können.

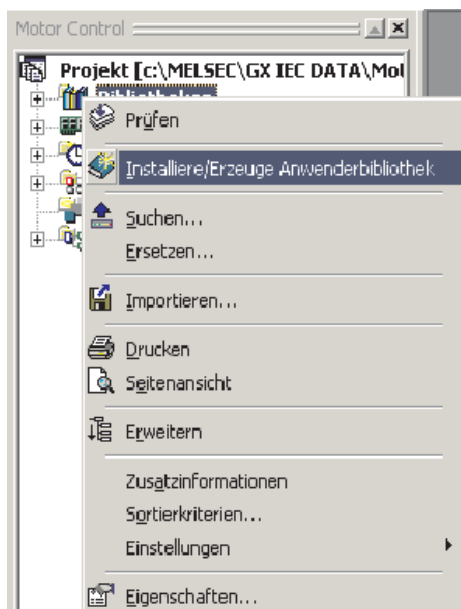
Dadurch können verschiedene Programmierer, die an unterschiedlichen Projekten arbeiten, gemeinsame Bibliotheken mit Standard-Funktionen nutzen.

Wie Sie bereits bei der Programmierung beim Aufruf von Funktionen gesehen haben, enthält die **Standardbibliothek** IEC-Funktionen. Die **Herstellerbibliothek** enthält herstellerspezifische Funktionen. Diese sind durch den Anhang _M gekennzeichnet. Dieses „M“ bedeutet übrigens Manufacturer (= Hersteller) und nicht „Mitsubishi“!

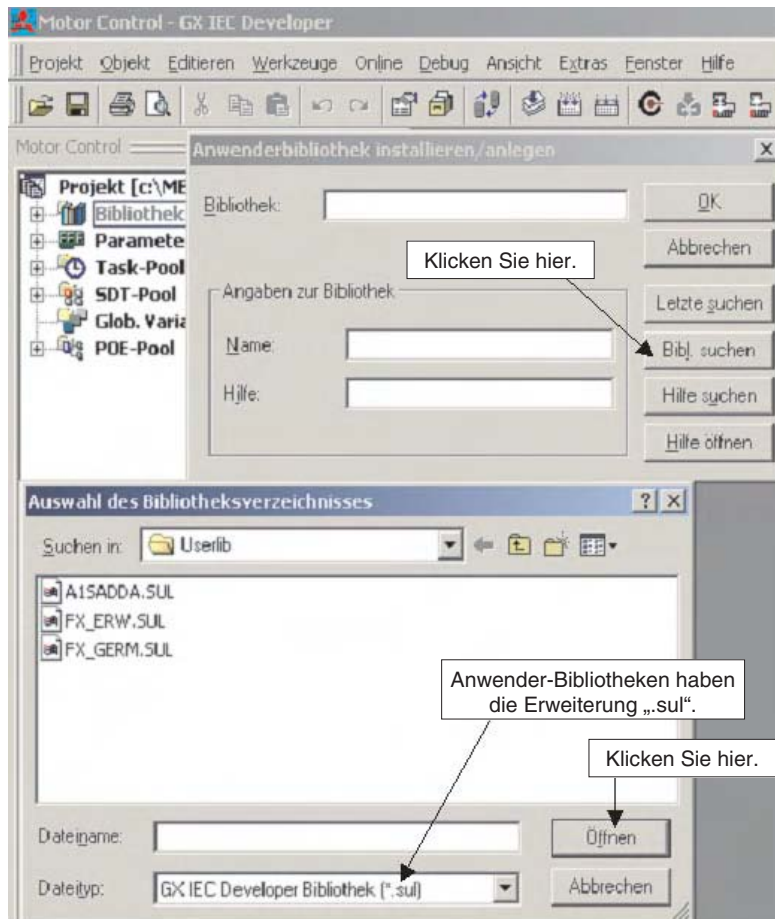
13.1.1 Beispiel: Anlegen einer neuen Bibliothek

Der Funktionsbaustein STAR_DELTA soll in einer neuen Bibliothek abgelegt werden.

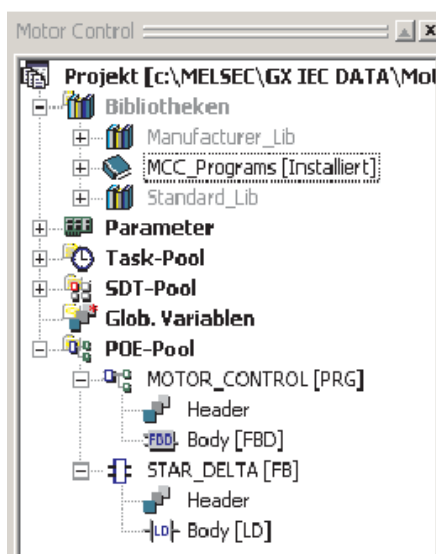
- ① Klicken Sie im Navigatorfenster mit der rechten Maustaste auf **Bibliotheken**. Im Menü, das dann angezeigt wird, klicken Sie auf **Installiere/Erzeuge Anwenderbibliothek**.



- ③ Klicken Sie im Dialogfenster **Anwenderbibliothek installieren/anlegen** auf das Schaltfeld **Bibl. suchen**. Daraufhin öffnet sich ein neues Dialogfenster, in dessen Eingabefeld **Dateiname** Sie bitte "MCC_Programs" eintragen. Der Pfad, in dem die Bibliothek eingetragen wird, kann bei Bedarf geändert werden. In diesem Beispiel wird vorausgesetzt, dass der voreingestellte Standard-Pfad "C:\MELSEC\GX IEC DEVELOPER 7.00\Userlib" verwendet wird.



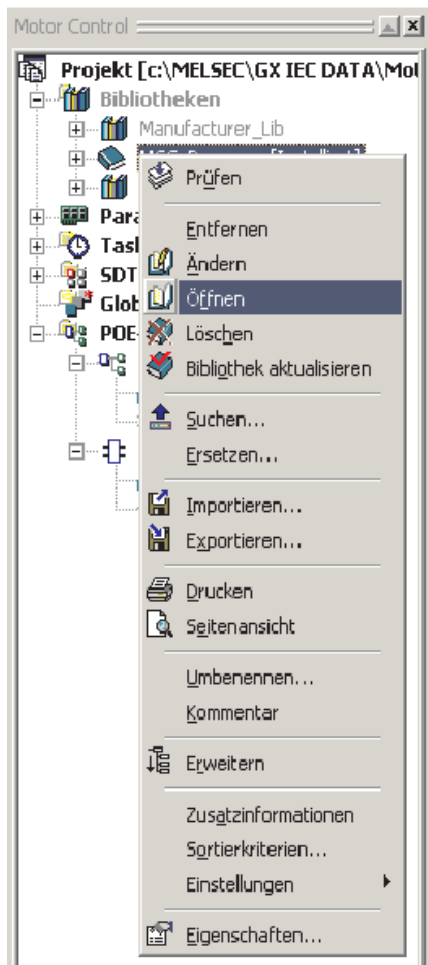
- ④ Klicken Sie anschließend auf **Öffnen**.



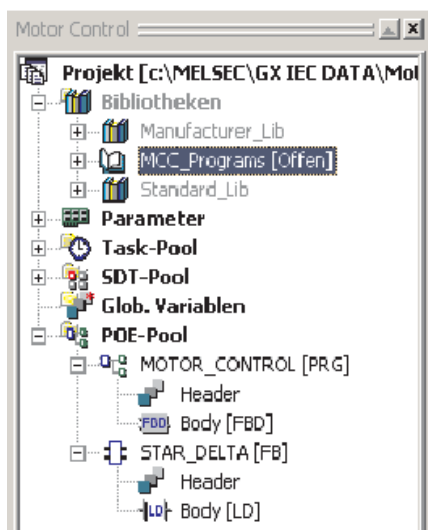
Beachten Sie, dass die neue Bibliothek „MCC_Programs“ nun im Navigatorfenster unter Bibliotheken angezeigt wird.

13.1.2 Öffnen einer Bibliothek

- ① Öffnen Sie die Bibliothek, indem Sie mit der rechten Maustaste auf „MCC_Programs“ klicken und aus dem Menü, das dann angezeigt wird, **Öffnen** wählen.



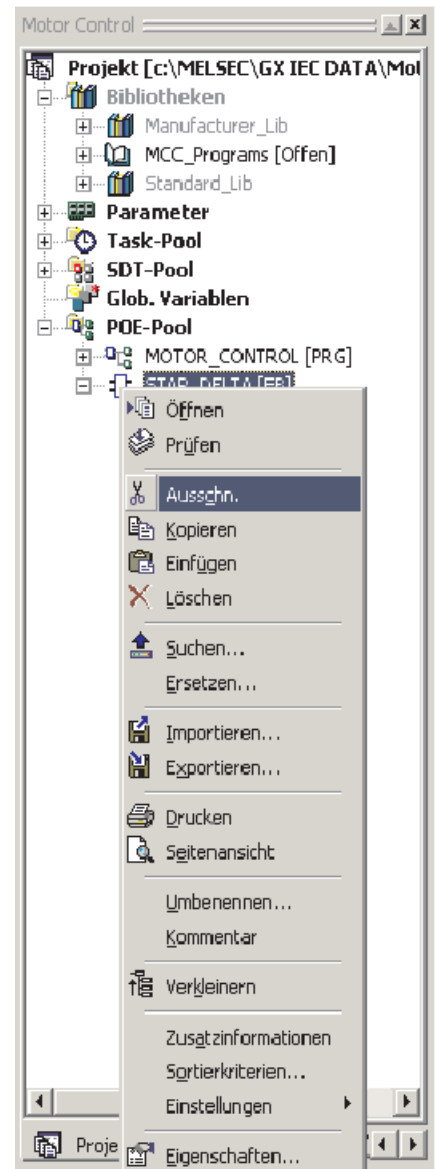
Die Bibliothek ist nun geöffnet und ihr Inhalt kann verändert werden.



13.1.3 Speichern eines Funktionsbausteins in eine geöffnete Bibliothek

Der Funktionsbaustein STAR_DELTA wird nun in die Bibliothek „MCC_Programs“ übertragen.

- ① Klicken Sie mit der rechten Maustaste im POE-Pool des Navigatorfensters auf STAR_DELTA und anschließend auf **Ausschn.** (Ausschneiden).

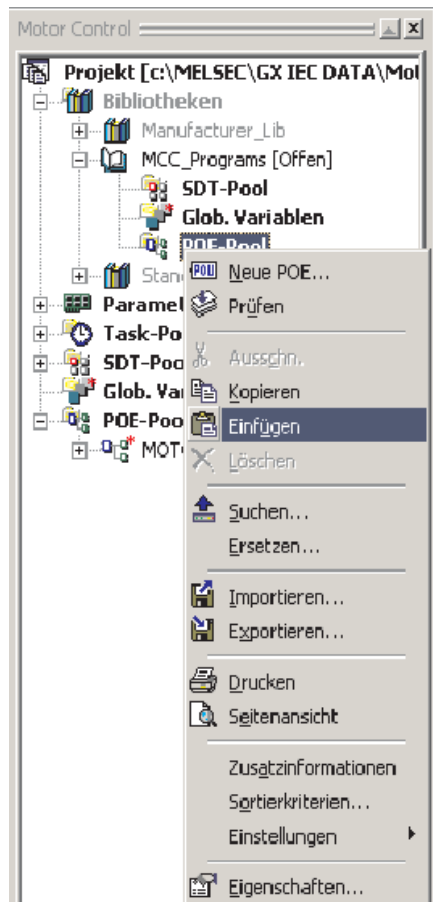


Danach wird die folgende Warnmeldung angezeigt:

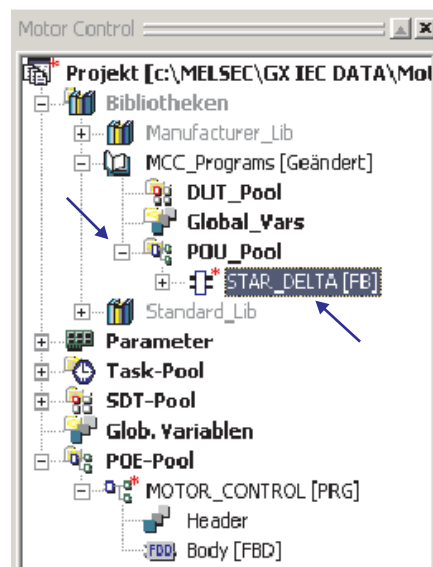


- ② Klicken Sie auf **Ja**.

- ③ Klicken Sie mit der rechten Maustaste im Navigatorfenster auf die Anwenderbibliothek und wählen Sie aus dem Menü, das dann angezeigt wird, **Einfügen**.



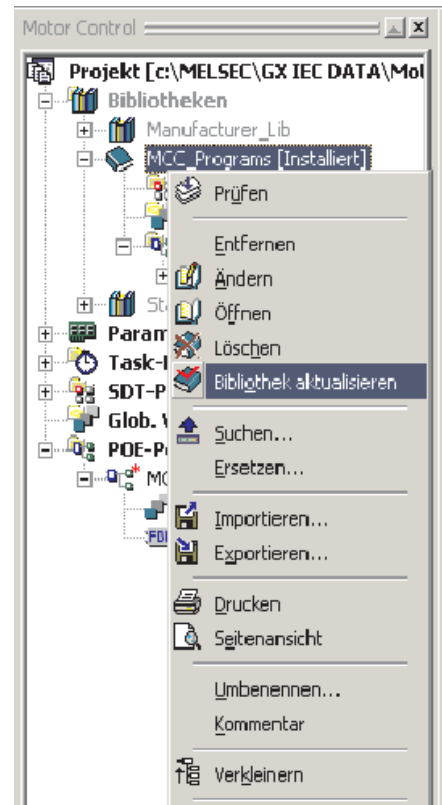
- ④ Klicken Sie auf das Pluszeichen vor dem Unterverzeichnis POE-Pool der Anwenderbibliothek, um den Inhalt anzuzeigen.



Der Funktionsbaustein-POE „STAR_DELTA“ gehört jetzt zur Bibliothek „MCC_Programs“ und nicht mehr zum POE-Pool des Projekts.

Auf diese Weise kann jede POE, ganz gleich ob Programm, Funktion oder Funktionsbaustein, in eine Anwenderbibliothek übertragen werden.

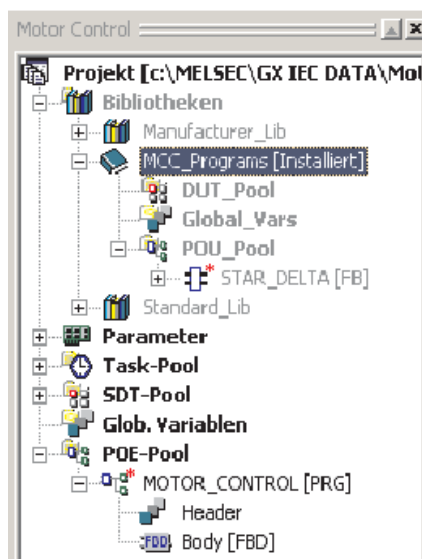
- ⑤ Wenn die Bearbeitung der Bibliothek abgeschlossen ist, klicken Sie mit der rechten Maustaste auf den Namen der Bibliothek und wählen anschließend **Bibliothek aktualisieren** aus dem Menü.



Danach wird die folgende Meldung angezeigt:



- ⑥ Klicken Sie auf **Ja**. Dadurch wird die Bibliothek aktualisiert, gespeichert und geschlossen.



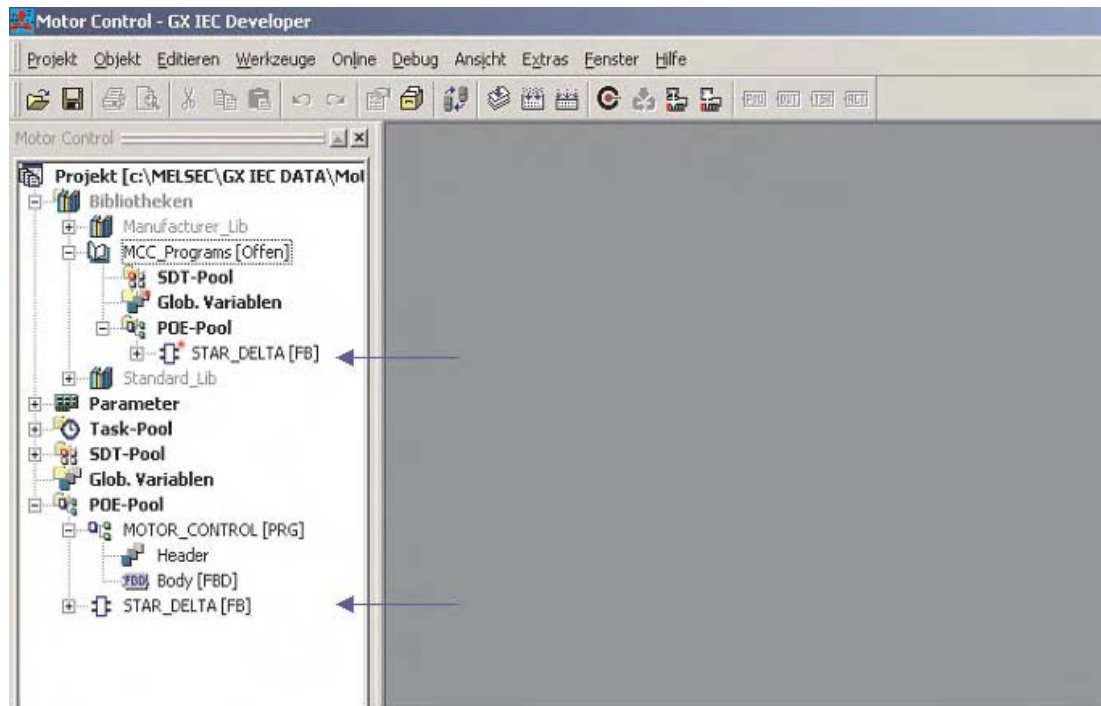
Die Bibliothek wird in dem Pfad gespeichert, der bei deren Einrichtung angegeben wurde. In diesem Beispiel ist das „C:\MELSEC\GX IEC DEVELOPER 7.00\Userlib“ (Abschnitt 13.1.1).

13.2 Hinweise zu Bibliotheken

Wenn eine Bibliothek in einem Projektpfad als Unterverzeichnis angelegt wird, dürfen die Elemente der Bibliothek nicht gleichzeitig im POE-Pool des Projekts vorhanden sein.

In diesem Fall erscheint beim Kompilieren des Projekts eine Fehlermeldung und die Bibliothekselemente müssen aus dem POE-Pool des Projekts gelöscht werden.

Es wird **keine** Fehlermeldung angezeigt, wenn die Bibliothek, wie oben beschrieben, außerhalb des Projekts eingerichtet wurde.



13.3 Bibliothek in ein Projekt importieren

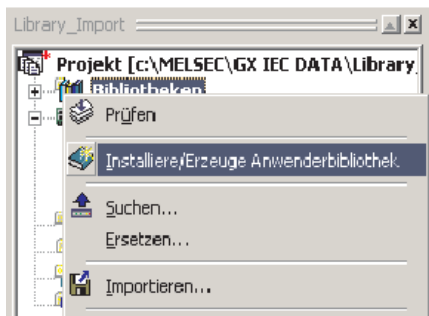
Anwenderbibliotheken können in Projekte importiert werden. Dadurch ist es möglich, einmal programmierte Routinen auch in anderen Projekten zu nutzen. Für oft benötigte Funktionen hat Mitsubishi Electric bereits viele Bibliotheken angelegt. Zum Beispiel enthält eine Bibliothek erprobte Funktionsbausteine zur Handhabung der Analogein- und -ausgabemodule. Diese und weitere Funktionsbausteine können kostenlos von der Mitsubishi-Homepage (www.mitsubishi-automation.de) geladen werden. Einige sind auch auf der CD mit der Programmier-Software GX IEC Developer enthalten.

Die folgenden beiden Beispiele beschreiben, wie Bibliotheken in ein Projekt importiert werden können.

13.3.1 Import einer Anwenderbibliothek

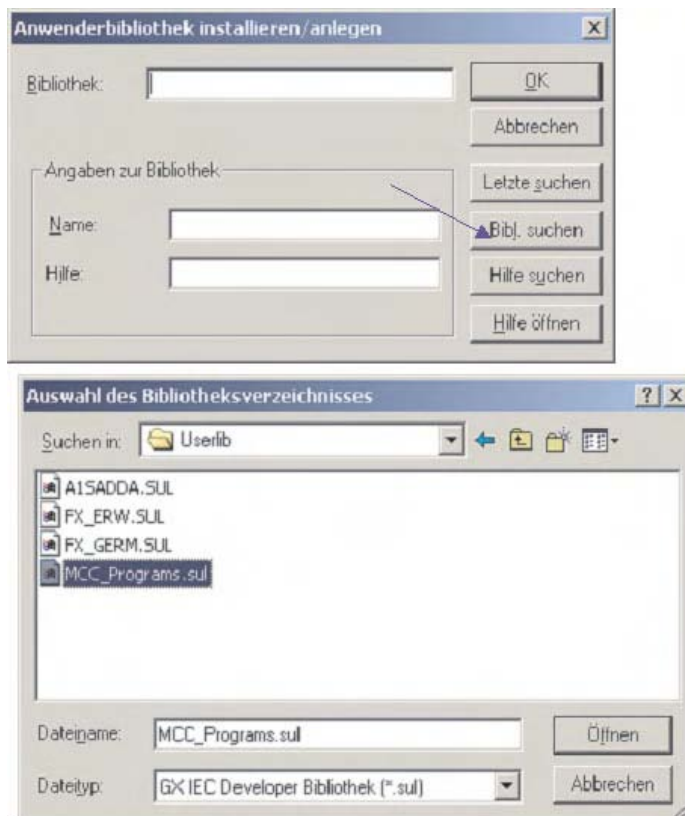
Die zuvor gespeicherte Anwenderbibliothek „MCC_Programs“ wird in das aktuelle Projekt importiert und der in der Bibliothek enthaltene Funktionsbaustein wiederverwendet.

- ① Legen Sie ein neues leeres Projekt ohne POEs an und nennen Sie es „Library Import“.



- ② Klicken Sie mit der rechten Maustaste auf **Bibliotheken** und anschließend im Menü auf **Installiere/Erzeuge Anwenderbibliothek**.

- ② Klicken Sie auf **Bibl. suchen** und wählen Sie die Bibliothek „MCC_Programs“.



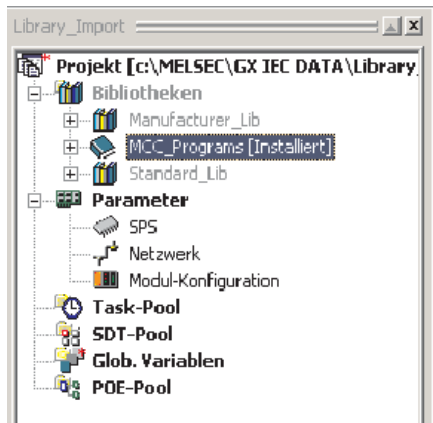
- ③ Klicken Sie dann auf **OK**, um die Eingaben zu übernehmen.



HINWEIS

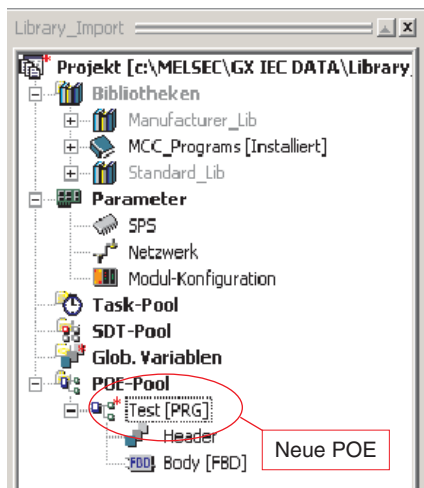
Die im oben abgebildeten Dialogfenster angebotene Hilfe bezieht sich auf anwenderdefinierte Hilfe-Dateien, mit denen die Funktion der Elemente in der Bibliothek beschrieben werden kann. Diese Dateien können mit Microsoft® Word erzeugt und z. B. im HTML-Format mit der Erweiterung CHM gespeichert werden (z. B. Hilfe.CHM). Mit der Funktion **Hilfe suchen** können diese Dateien dann genauso in das Projekt übernommen werden wie die Bibliothek selbst (siehe auch Abschnitt 13.3.3).

Die soeben importierte Bibliothek wird in das aktuelle Projekt übernommen und erscheint im Verzeichnis **Bibliotheken** des Navigatorfensters:

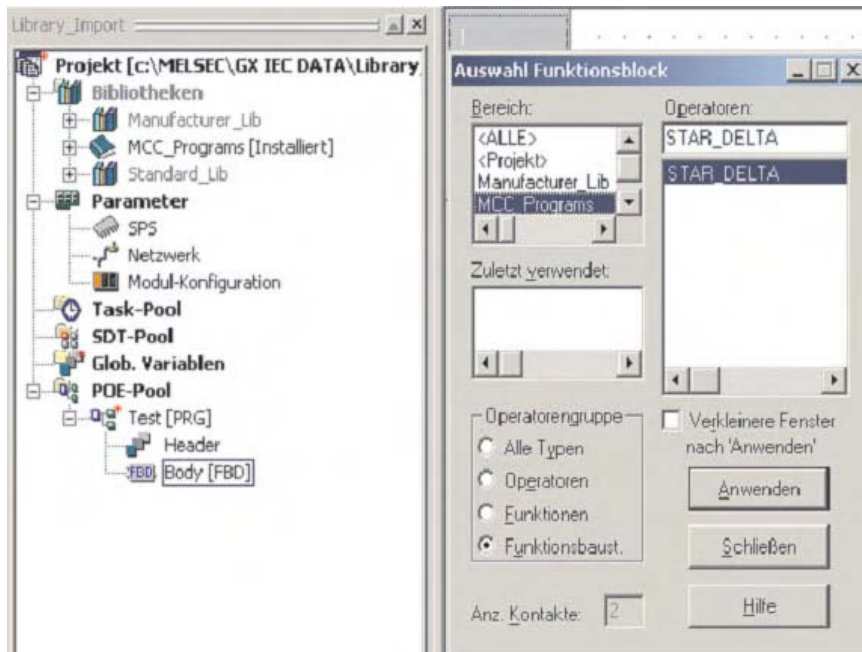


Wie die folgenden Abbildungen zeigen, lässt sich der Inhalt von Bibliotheken einfach auswählen und im Projekt verwenden.

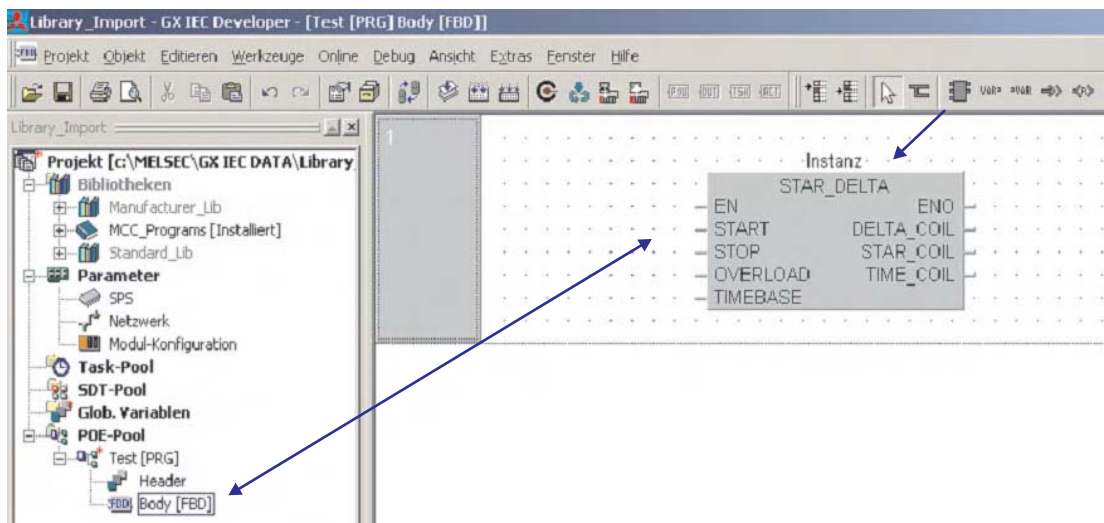
- ① Legen Sie eine neue POE mit dem Namen „Test“ an. Als Programmierer wählen Sie bitte die **Funktionsbausteinsprache**.



- ② Öffnen Sie die neue POE und öffnen Sie, wie hier gezeigt, das Dialogfenster zur Funktionsbausteinauswahl.



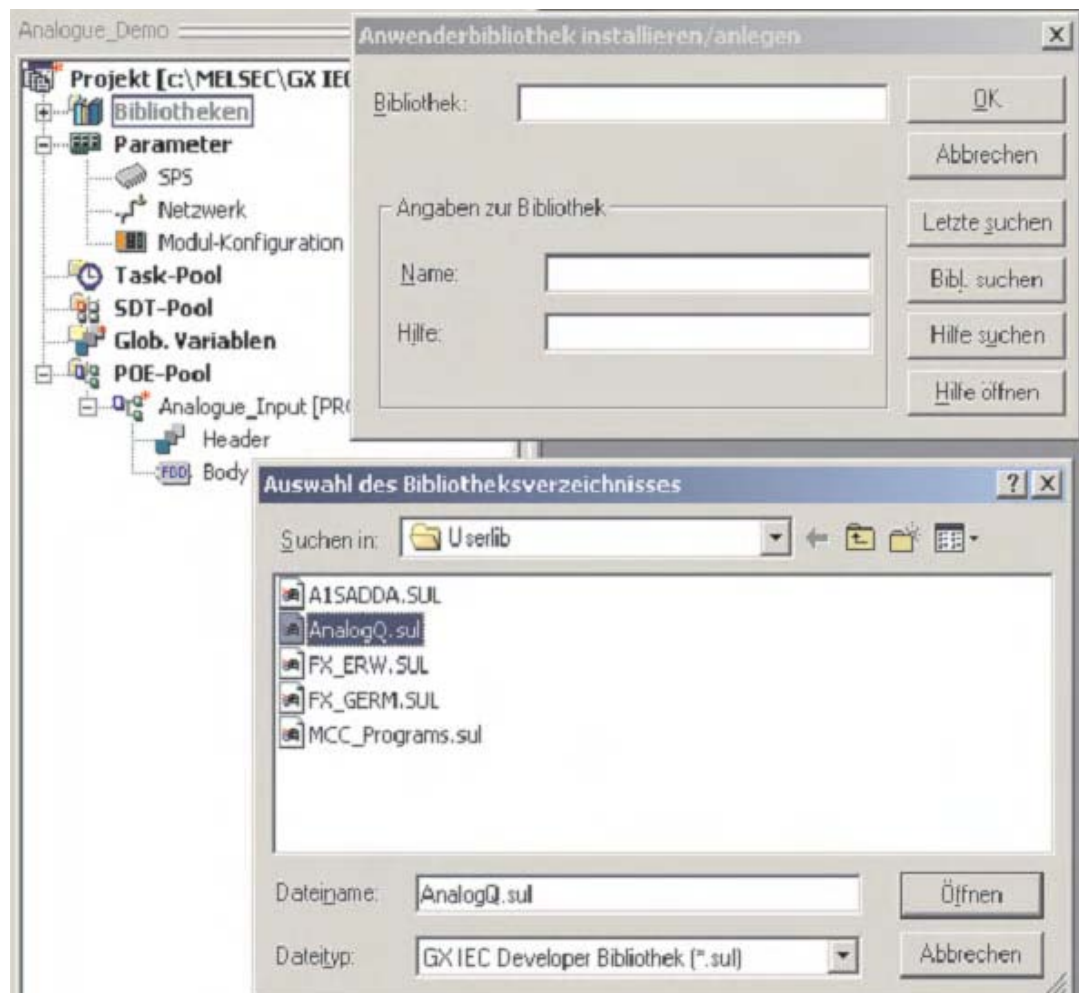
Die neue Bibliothek kann geöffnet und deren Inhalt ausgewählt werden.



13.3.2 Import eines Funktionsbausteins aus einer Mitsubishi-Bibliothek

In diesem Beispiel wird ein Funktionsbaustein zum Lesen von Analogwerten des Analogeingangsmoduls Q64AD in das aktuelle Projekt importiert. Dieser Funktionsbaustein wird von Mitsubishi zur Verfügung gestellt. Aus diesem Grund muss die Bibliothek „AnalogQ“ in das Projekt übernommen werden. Diese Bibliothek kann von der Mitsubishi-Homepage (www.mitsubishi-automation.de) oder – mit Hilfe des Installationsassistenten – von der CD mit der Programmier-Software GX IEC Developer geladen werden. Danach kann diese Bibliothek wie eine Anwenderbibliothek behandelt werden.

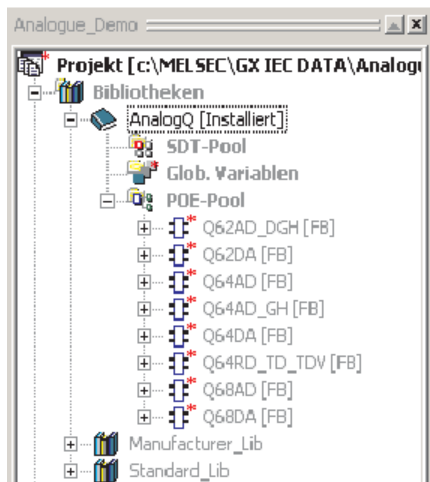
- ① Legen Sie ein neues leeres Projekt ohne POEs an und nennen Sie es „Analogue_Demo“.
- ② Legen Sie eine neue POE mit dem Namen „Analogue_Input“ an. Als Programmklass wählen Sie bitte **PRG** (Programm) und als Programmeditor die **Funktionsbausteinsprache**.
- ③ Klicken Sie mit der rechten Maustaste auf **Bibliotheken** und anschließend im Menü auf **Installiere/Erzeuge Anwenderbibliothek**. Im neuen Dialogfenster klicken Sie dann bitte auf **Bibl. suchen**. Wählen Sie die Bibliothek *AnalogQ.sul* und klicken Sie anschließend auf **Öffnen**.



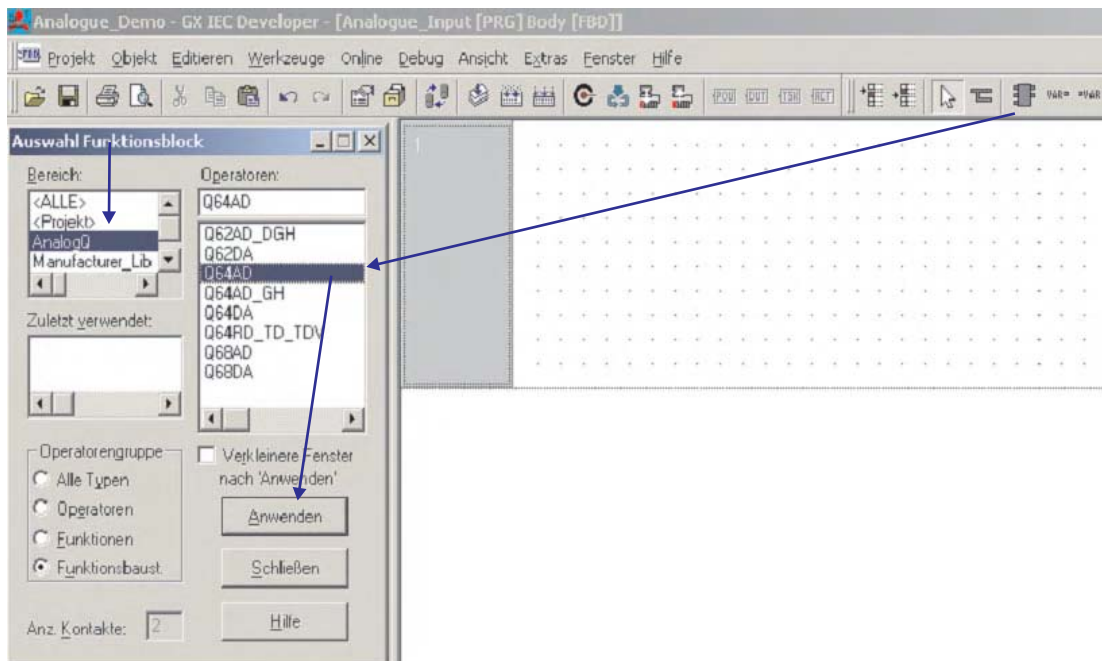
- ④ Klicken Sie dann im Dialogfenster **Installiere/Erzeuge Anwenderbibliothek** auf **OK**.



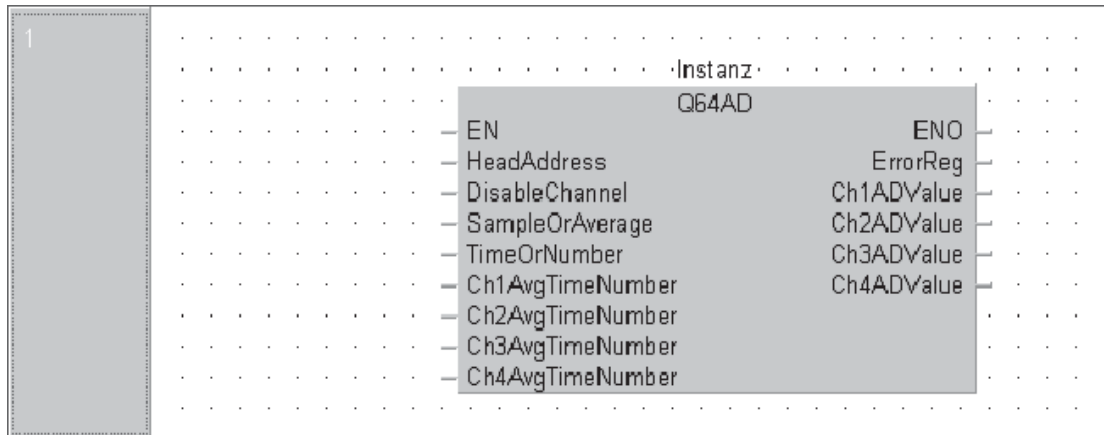
Danach wird die neue Bibliothek AnalogQ im Navigatorfenster angezeigt.



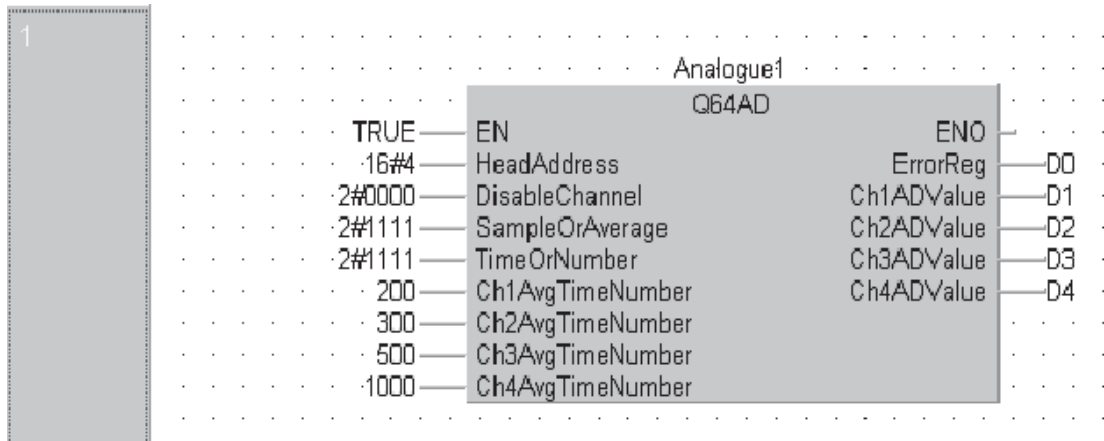
- ⑤ Legen Sie im Task-Pool eine neue Task mit dem Namen „MAIN“ an. Rufen Sie in dieser Task die POE „Analogue_Input“ auf.
- ⑥ Wählen Sie in dieser POE den Funktionsbaustein Q64AD aus:



Dieser Funktionsbaustein wird so dargestellt:



⑦ Definieren Sie alle Variablen so wie hier gezeigt:



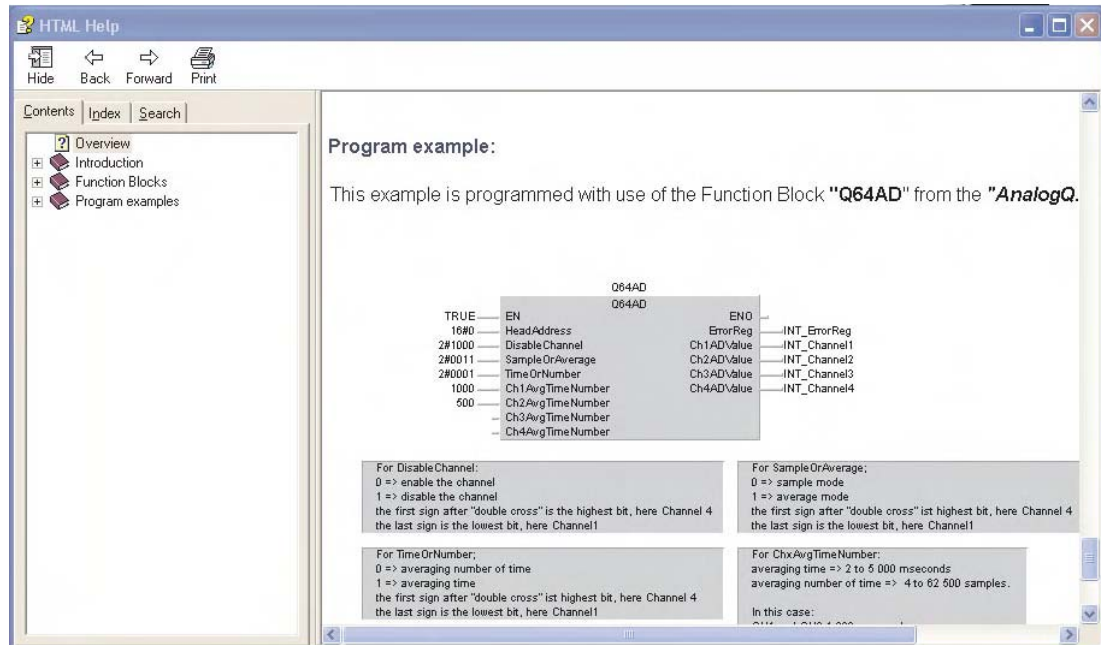
⑧ Kompilieren Sie das Programm und übertragen Sie es in die SPS.

⑨ Prüfen Sie im Monitormodus die korrekte Funktion des Programms. Verfolgen Sie die Reaktion der Analogausgänge bei unterschiedlichen Einstellungen für die Analogwerfassung.

13.3.3 Hilfefunktion der Bibliotheken

Wenn auch die Hilfe-Datei einer Bibliothek importiert wurde, steht eine ausführliche Beschreibung der Funktionsbausteine mit Beispielen zur Verfügung.

Klicken Sie dazu auf den Funktionsblock und betätigen Sie anschließend die F1-Taste. Dadurch wird zum Beispiel für den Funktionsbaustein Q64AD die folgende Hilfe angeboten:



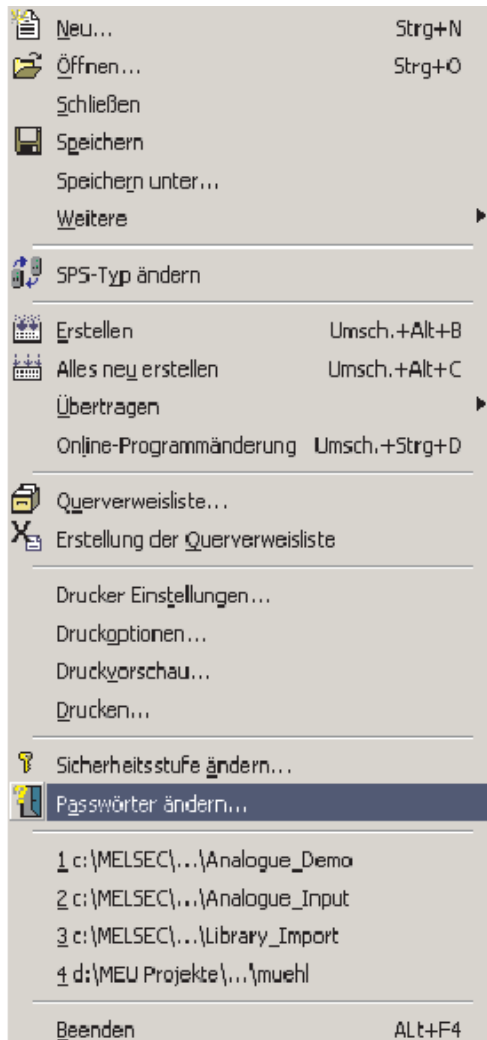
Die Hilfethemen umfassen alle Bereiche, von der Parametrierung der Analogmodule des MEL-SEC System Q bis hin zum Einsatz der Funktionsbausteine aus der Bibliothek.

14 Schutz der Programme

14.1 Vergabe eines Passworts

Alle Teile eines Programms können durch ein Passwort geschützt werden. Passwörter können zum Schutz von Programmteilen vor Überschreiben oder auch sogar als Schutz vor einem unberechtigten Lesen eingerichtet werden. Dies ist besonders bei Funktionsbausteinen wichtig, die selbst programmiert wurden. Zusätzlich kann durch ein Passwort auch der Zugriff auf eine SPS verhindert werden.

14.1.1 Einrichten eines Passworts

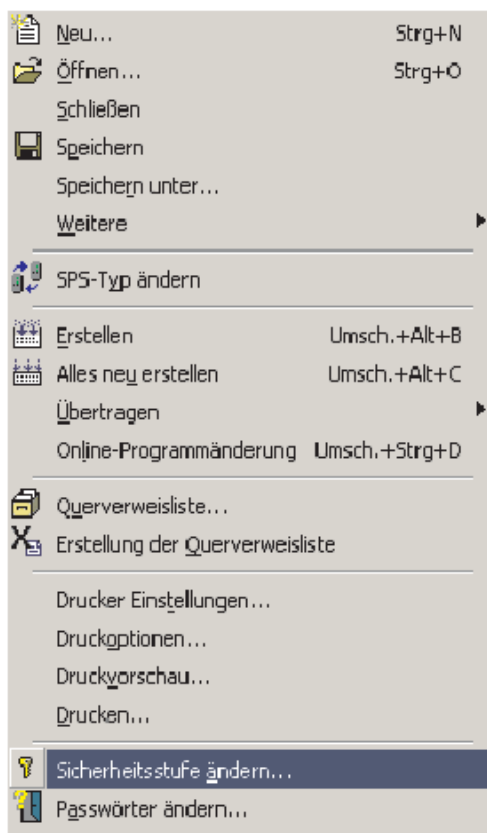


Passwörter können im Menü **Projekt** eingerichtet werden. Hier lässt sich auch die Sicherheitsstufe ändern.

Zur Verdeutlichung der Passwortfunktion, wählen Sie eine **Sicherheitsstufe** (wählen Sie hier der Einfachheit wegen Sicherheitsstufe 7) und geben in das Eingabefeld **Neues Passwort** ein Passwort für diese Sicherheitsstufe ein. Geben Sie dasselbe Passwort noch einmal in das Feld **Passwortwiederh.** (Passwortwiederholung) ein und klicken Sie anschließend auf **Ändern**.



14.1.2 Sicherheitsstufe ändern



① Wählen Sie im Menü **Projekt** den Eintrag **Sicherheitsstufe ändern**.

② Geben Sie das Passwort für die Sicherheitsstufe 7 ein. Nach Eingabe des korrekten Passworts sind Sie auf dieser Sicherheitsstufe eingeloggt.



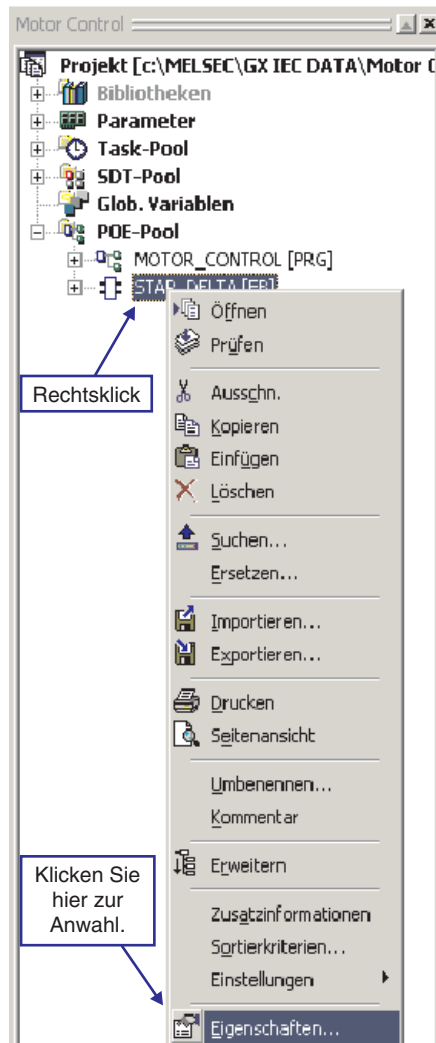
Nach dem Einloggen können die Sicherheitseinstellungen für viele Elemente geändert werden. Eine weit verbreitete Anwendung zum Beispiel ist der Schutz von POEs, und hier besonders von Funktionen und Funktionsbausteinen, vor einem unbefugten Zugriff.

14.1.3 Sicherheitsstufe einer POE ändern

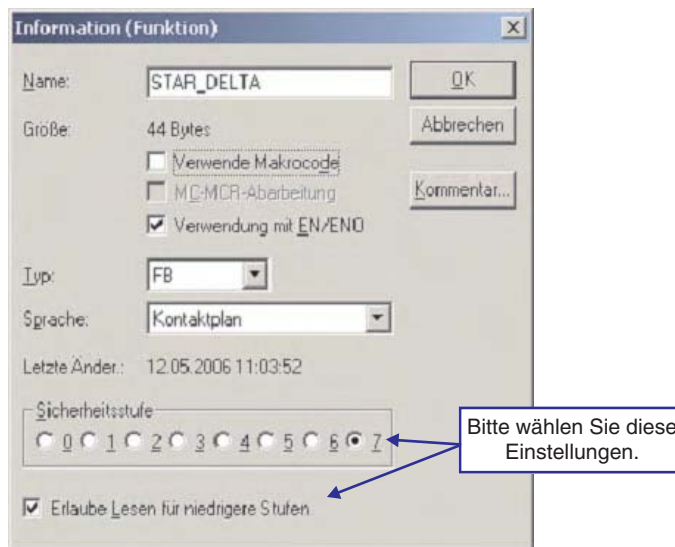
Wenn Sie auf der Sicherheitsstufe eingeloggt sind, die für eine Anwender-POE eingestellt wurde, können Sie die Sicherheitseinstellungen dieser POE ändern und sie so vor einen Schreib- oder Lesezugriff schützen.

Einstellung der Sicherheitsstufe

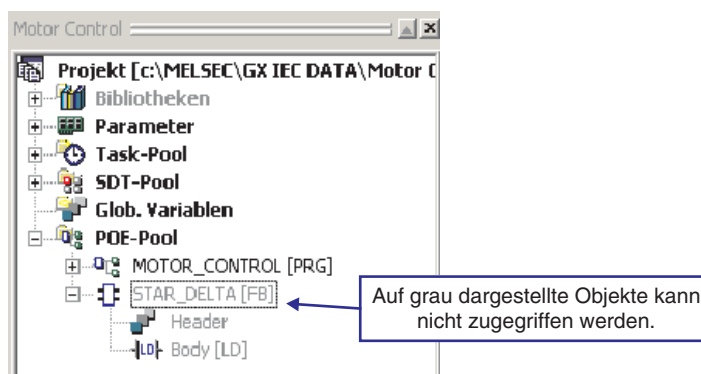
- ① Öffnen Sie das Projekt „Motor Control“ und klicken Sie im POE-Pool anschließend mit der rechten Maustaste auf den Funktionsbaustein „STAR_DELTA“.



- ② Wählen Sie **Sicherheitsstufe 7** und aktivieren Sie die Option **Erlaube Lesen für niedrigere Stufen**. Dadurch können sich Anwender, die auf einer niedrigeren Sicherheitsstufe eingeloggt sind, den Inhalt des Headers und Body dieses Funktionsbausteins ansehen („Lesen“). Ändern („Schreiben“) können Sie diesen Funktionsbaustein aber nicht.



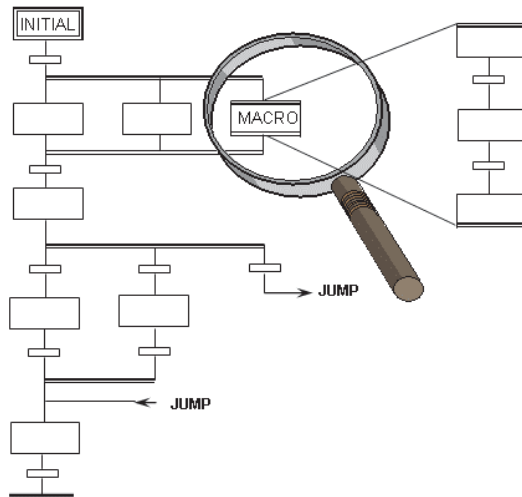
- ③ Loggen Sie sich mit der Sicherheitsstufe 0 ein und öffnen Sie Header und Body des Funktionsbausteins „STAR_DELTA“. Prüfen Sie die Sicherheitseinstellungen. Lesen, z. B. zur Fehlerdiagnose, ist erlaubt, ändern jedoch nicht.
- ④ Loggen Sie sich wieder die Sicherheitsstufe 7 ein und deaktivieren Sie in den Einstellungen für den Funktionsbausteins „STAR_DELTA“ die Option **Erlaube Lesen für niedrigere Stufen**. Dadurch haben Anwender, die auf einer niedrigeren Sicherheitsstufe eingeloggt sind, keinen Zugriff mehr auf Header und Body dieses Funktionsbausteins.
- ⑤ Loggen Sie sich mit der Sicherheitsstufe 0 ein, und versuchen Sie, den Body des Funktionsbausteins „STAR_DELTA“ zu öffnen. Dies darf jetzt nicht mehr möglich sein. Im Navigatorfenster wird der Name sowie der Header und der Body einer POE, auf die nicht mehr zugegriffen werden kann, grau dargestellt.



Sicherheitsstufen können für jedes einzelne Objekt oder komplette Ordner im Navigatorfenster eingestellt werden. Durch die Zuweisung unterschiedlicher Sicherheitsstufen an einzelne Objekte ist eine hohe Flexibilität beim Schutz der Programme gewährleistet.

15 Programmieren in Ablaufsprache (AS)

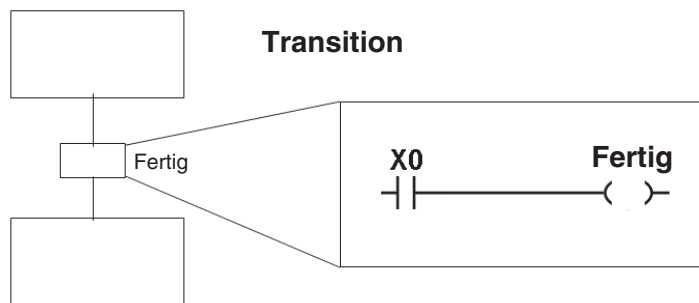
15.1 Was ist Ablaufsprache?



- Die Ablaufsprache ist eine Strukturierungssprache, mit der Prozessabläufe übersichtlich dargestellt werden können.
- Die Ablaufsprache unterteilt Prozesse in Schritte und Transitionen und bildet so eine Schrittkette.
- Der Programmierer für Ablaufsprache ist ein geführter Editor.
- Ablaufsprache basiert auf dem französischen Grafcet (IEC 848).
- Jedem Schritt können entweder keine, eine oder mehrere Aktionen (keine POEs) und/oder das direkte Schalten von Bit-Operanden zugeordnet werden.
- Jeder Transition wird eine Weiterschaltbedingung (Bezeichnung der Transition) zugeordnet. (Anstelle eines Transitionenamens kann auch eine absolute Adresse verwendet werden.)
- Aktionen können in jeder beliebigen Programmiersprache - außer in der Ablaufsprache - erstellt werden.
- Transitionen können in jeder beliebigen Programmiersprache - außer in der Ablaufsprache - erstellt werden.
- Nur bei Steuerungen der MELSEC A-Serie wird der Code für Programme in Ablaufsprache im Mikrocomputer-Bereich der SPS abgelegt. Reservieren Sie Speicherplatz in den SPS-Parametern.

15.2 Elemente von AS-Programmen

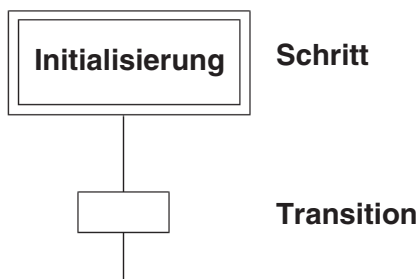
15.2.1 Transitionen



- Jeder Transition wird eine Transitionsbedingung (Weiterschaltbedingung) zugeordnet. Wenn die Transitionsbedingung erfüllt ist, wird der nächste Schritt aktiviert.
- Die Transitionsbedingung kann in jeder beliebigen Programmiersprache - außer in der Ablaufsprache - programmiert werden.
- An Stelle eines symbolischen Namens kann auch direkt ein Bit verwendet werden.

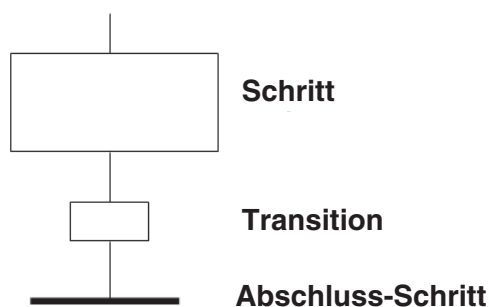
15.2.2 Initialisierungs-Schritt

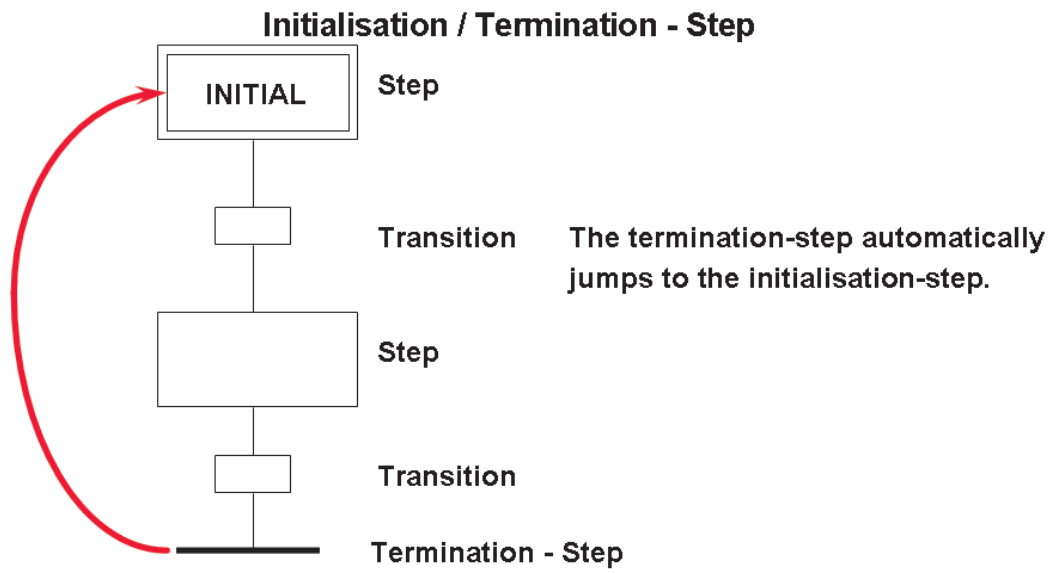
Der Initialisierungs-Schritt bildet den Anfang einer Schrittkette.



15.2.3 Abschluss-Schritt

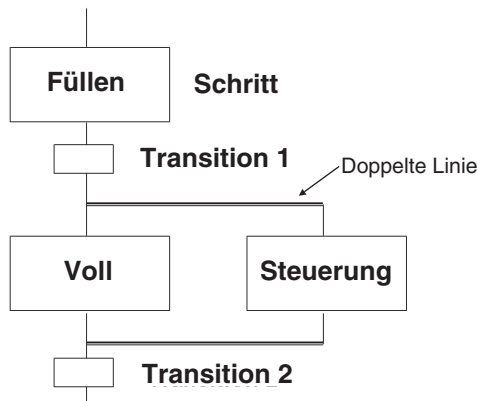
Jede Schrittkette endet mit einem Abschluss-Schritt.



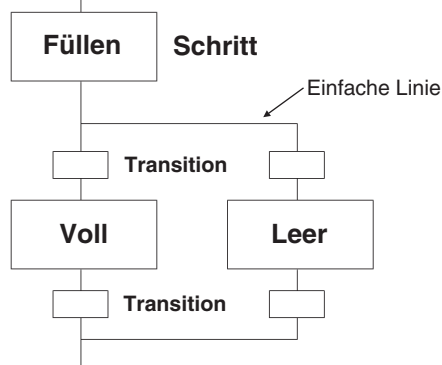


15.3 Verzweigungen und Sprünge

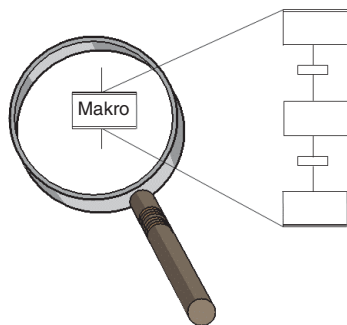
Parallele Verzweigung



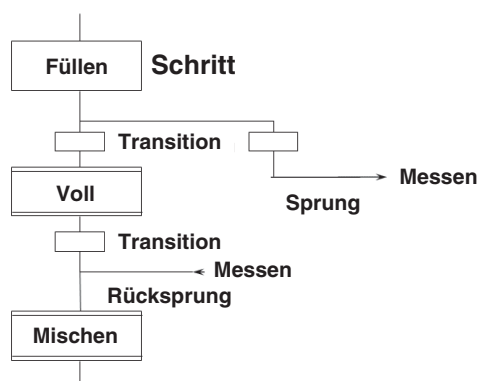
Selektive Verzweigung



Makro-Schritt

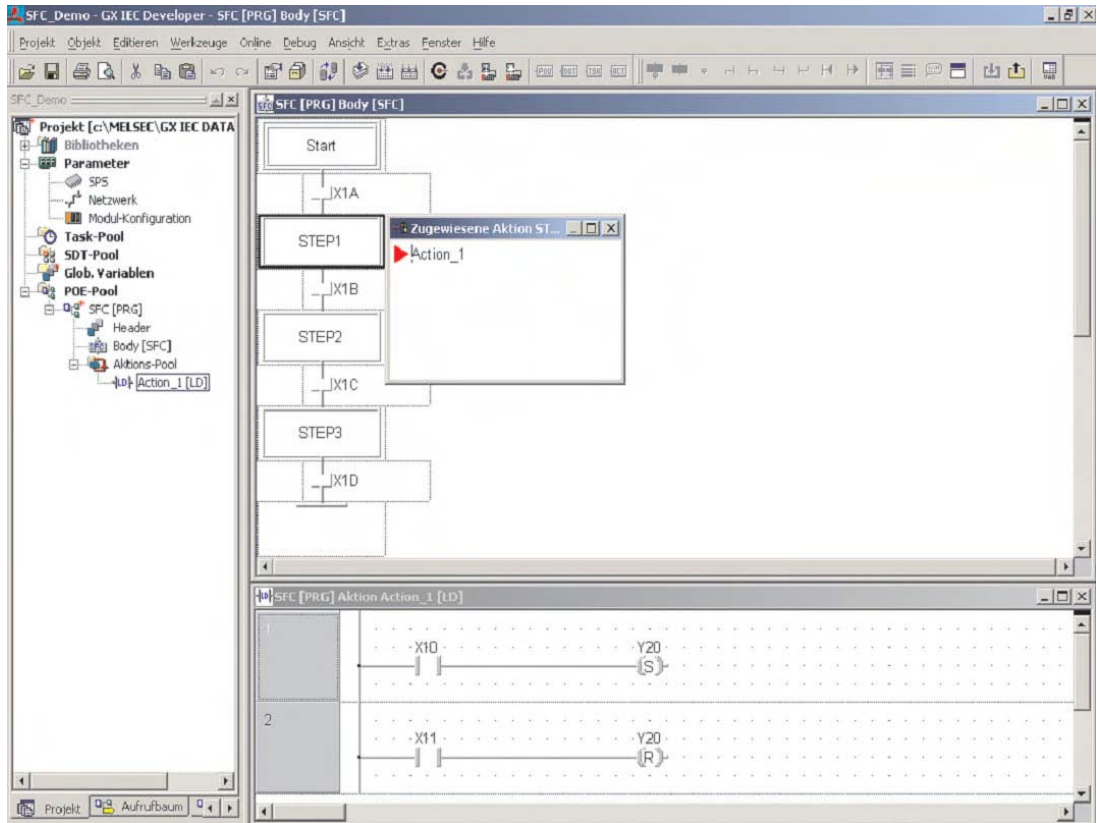


Sprung

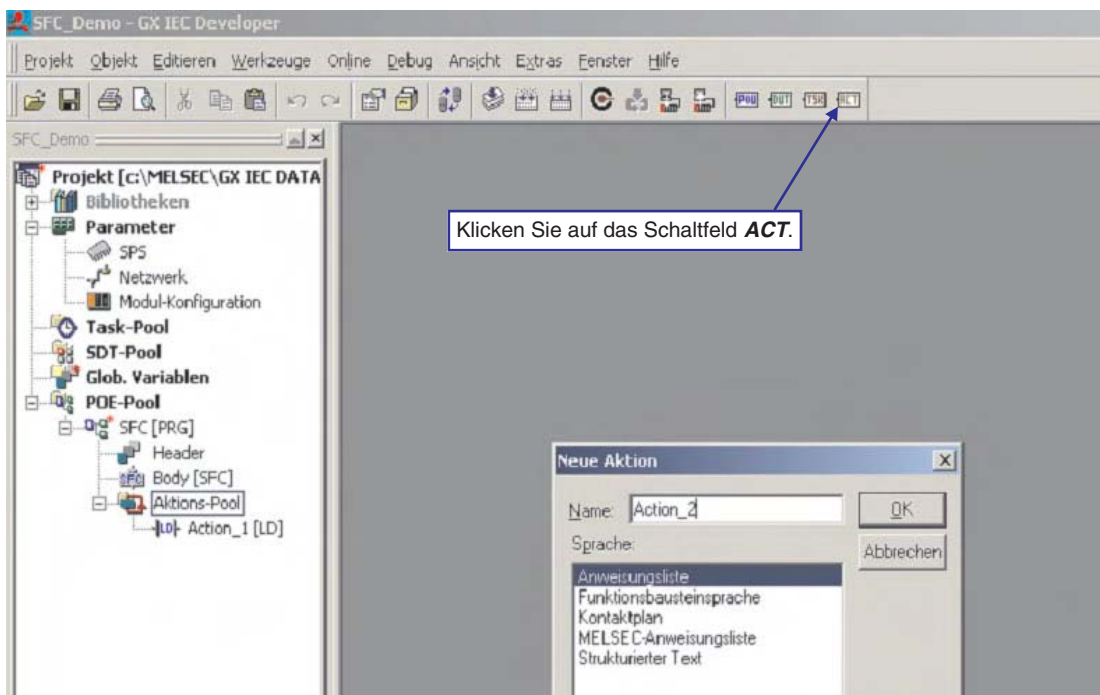


15.4 Aktionen

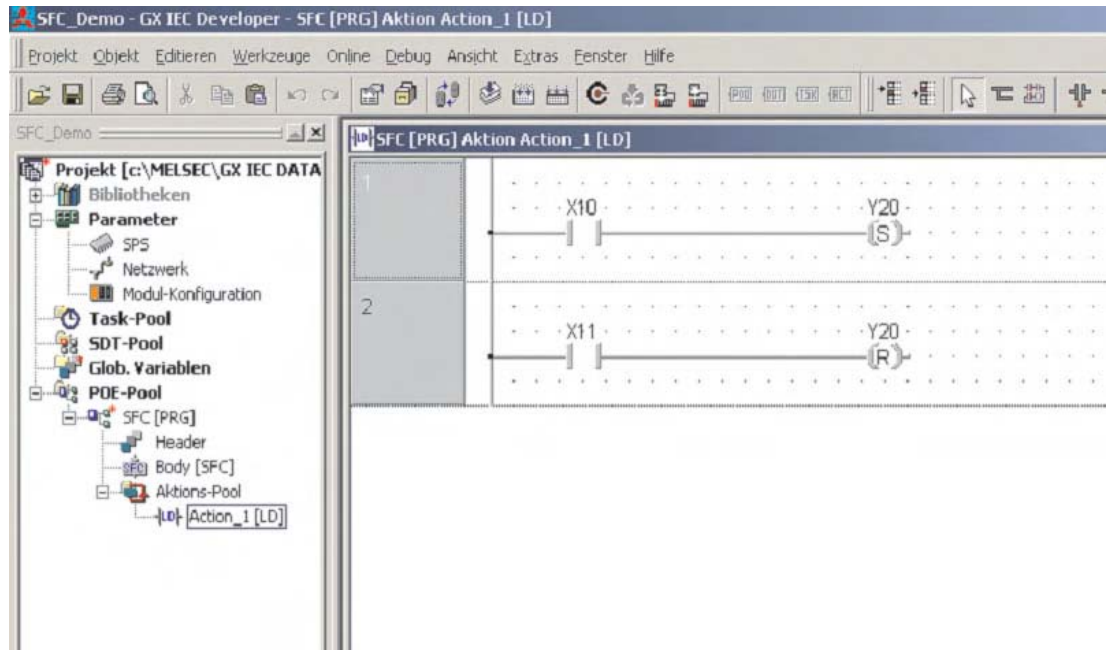
Jedem Schritt können entweder keine, eine oder mehrere Aktionen zugeordnet werden. Eine Aktion ist ein Programm, ähnlich einer POE. Dieses Programm kann in der IEC- oder MELSEC-Anweisungsliste, Kontaktplan, der Funktionsbausteinsprache oder Strukturiertem Text geschrieben worden sein.



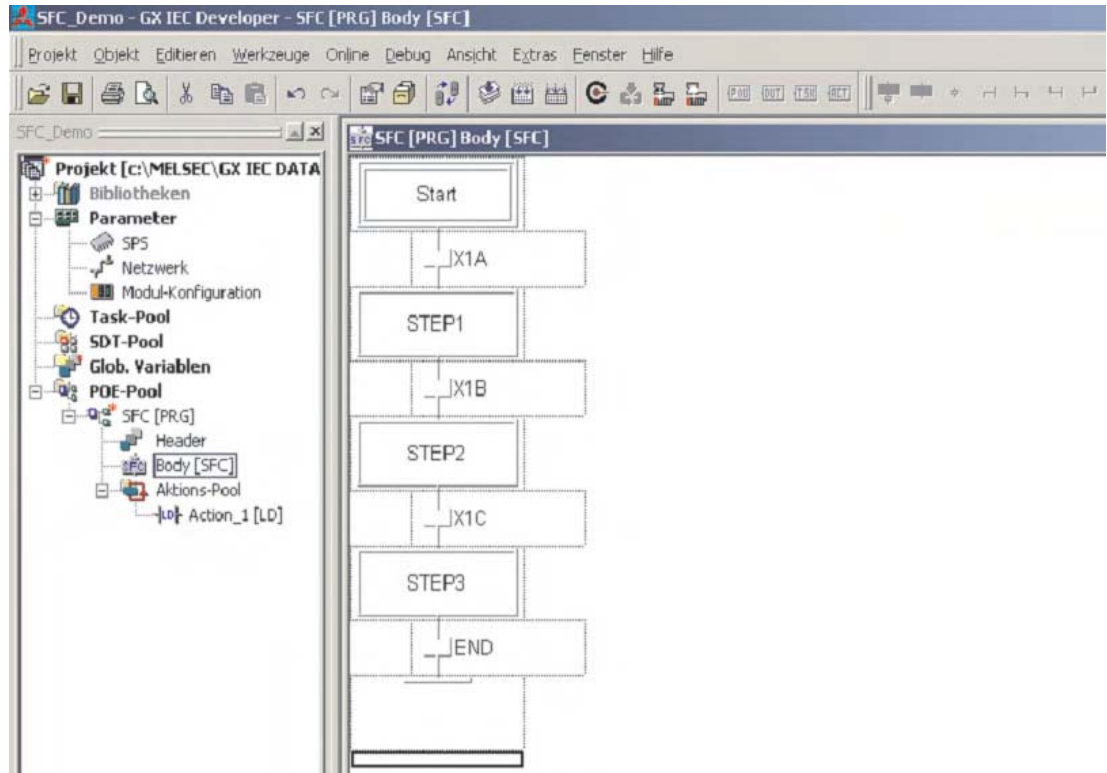
Um neue Aktionen zu programmieren, klicken Sie in der Werkzeugleiste auf das Schaltfeld **AKT**. Wählen Sie dann, wie auch bei einer POE, den Programmeditor.



Aktionen können auch umfangreiche Programme sein. Aktion_1 kann z. B. eine Verriegelung sein, die im Kontaktplan programmiert ist und mehrere Netzwerke belegt.

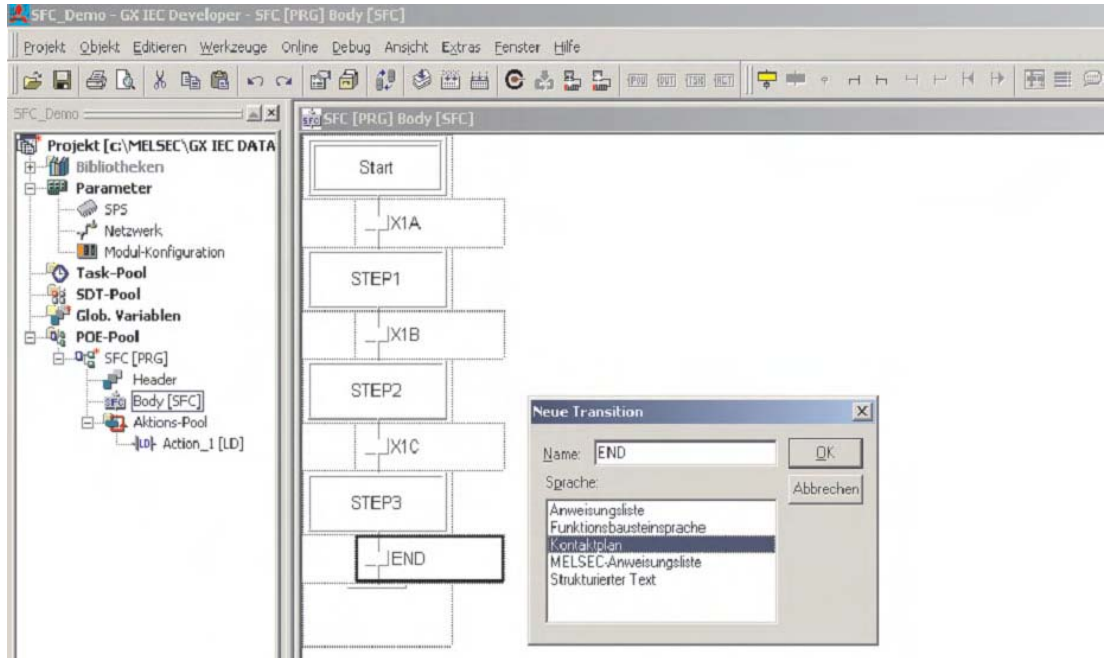


Eine Transition kann ein einfacher Operand sein, wie z. B. der Eingang XA, oder der Bezeichner einer Variablen, aber auch ein Programm, das ein einzelnes Netzwerk belegt. Dieses Programm kann in der IEC- oder MELSEC-Anweisungsliste, Kontaktplan, der Funktionsbausteinsprache oder Strukturiertem Text geschrieben worden sein.

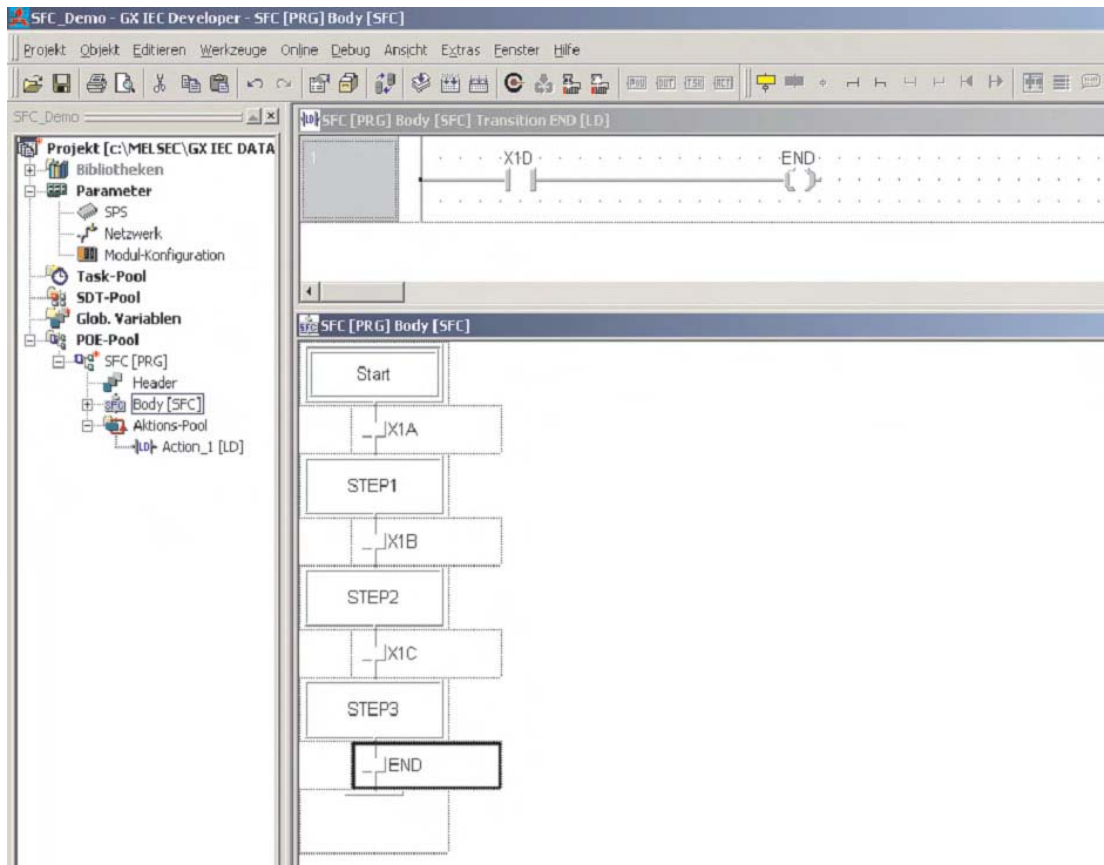


15.5 Komplexe Transitionen

Zur Programmierung einer komplexen Transition geben Sie den Namen der Transition ein und betätigen dann die Übernahme-Taste. Wählen Sie dann, genauso wie bei Aktionen, den Programmierer.

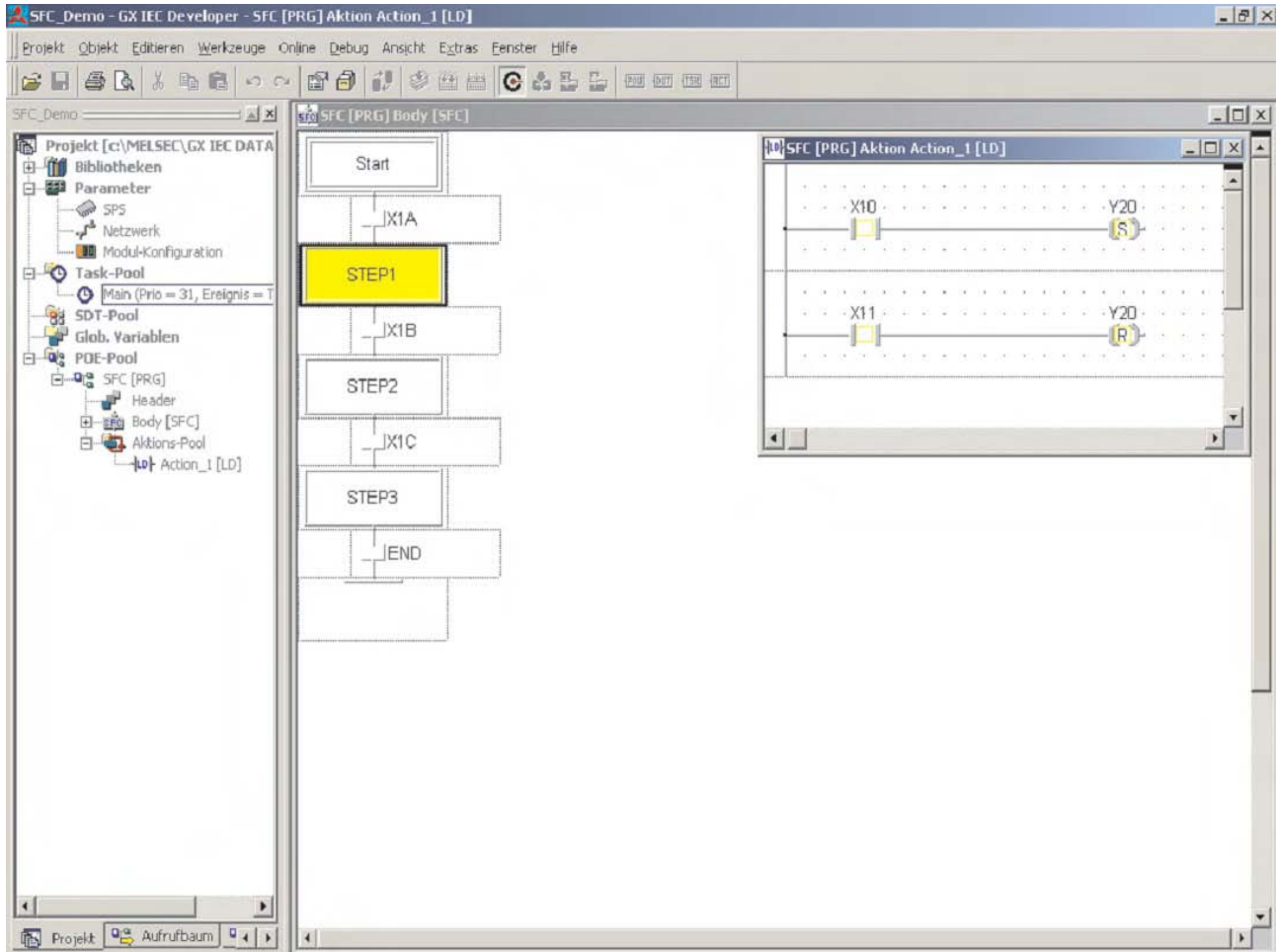


Die Transition kann eine komplexe Verknüpfung sein, darf aber nur ein Netzwerk belegen.



15.6 Anzeige von AS-Programmen im Monitormodus

Im Monitormodus wird der aktuelle Schritt eines AS-Programms besonders hervorgehoben. Bei einer Fehlerdiagnose zum Beispiel kann dadurch schnell festgestellt werden, an welcher Stelle der Schrittkette das Programm steheengeblieben ist.



16 IEC-Anweisungsliste

- Der Anweisungslisteneditor ist ein Fließtexteditor.
- Die einzelnen Zeilen werden nicht nummeriert.
- In der Anweisungsliste können Funktionen und Funktionsbausteine aufgerufen werden.
- Eine IEC-Anweisungsliste kann zusätzlich zu IEC-Netzwerken auch MELSEC-Netzwerke enthalten.
- Kommentare werden besonders gekennzeichnet (z. B. *Kommentar*)
- Ein Programm kann auch mit einem anderen Texteditor wie z. B. Microsoft® Word erstellt und mit Hilfe der Standard Windows-Funktionen in den GX IEC Developer kopiert werden.

16.1 Beispiel für IEC-Anweisungsliste (IL)

```
LD      X4      (* Abfrage von X4 *)
ANDN    M5      (* ANDN M5 *)
ST      Y20     (* Verknüpfungsergebins Y20 zuweisen *)
```

```
LD      TEST    (* TEST in Akku laden *)
BCD_TO_INT  (* Accu-Inhalt wandeln *)
ST      RESULT  (* Akku-Inhalt in RESULT speichern *)
```

16.1.1 Einige nützliche Tipps

„+ D0 D1 D2“ kann in der IEC-Anweisungsliste so ausgedrückt werden:

```
LD      D0
ADD     D1
ST      D2
```

“ + D0 D1 D2 ” und anschließend “ + D2 K50 D3 ” wird zu:

```
LD      D0
ADD     D1,D2,50
ST      D3
```

Durch die „_E“-Funktion kann noch weiter vereinfacht werden. Die Operation „ + D0 D1 D2 “ und anschließend “ + D2 K50 D3 ” wird durch den Eingang X0 gesteuert:

```
LD      X0
ADD_E   D0,D1,D2,50,D3
```

Dies ist möglich, weil die ADD_E-Funktion einen ENO-Ausgang besitzt.

16.2 Mischen von IEC- und MELSEC-Anweisungsliste

In einer POE können Netzwerke mit IEC-Anweisungsliste (IL) und Netzwerke mit MELSEC-Anweisungsliste (MELSEC-IL) enthalten sein.

Um in einer POE, die in IEC-Anweisungsliste programmiert wird, ein Netzwerk für MELSEC-Anweisungsliste einzufügen, klicken Sie links in den grauen Bereich eines Netzwerks, um es zu markieren. Wählen Sie dann im Menü **Editieren** den Menüpunkt **Neues Netzwerk**. Im Untermenü klicken Sie dann auf **MELSEC vorher**. Dadurch wird ein Netzwerk für MELSEC-Anweisungsliste vor dem markiertem Netzwerk eingefügt.

The screenshot displays the GX IEC Developer interface. The left pane shows a project tree for 'Projekt [c:\MELSEC\GX IEC DATA]' with a POE-Pool containing an IL [PRG] sub-project. The right pane shows the 'IL [PRG] Body [IL]' editor with three rungs of ladder logic:

Rung	Instruction	Address
1	LD	SM400
	OUT	Y20
	TO	H4 K1 RESULT_2 K1
MELSEC		
2	LD	X10
	ANDN	X11
	ANDN	X12
	ST	MO
3	LD	D10
	ADD	D11,D12,D13,D14
	ST	D20

17 Strukturierter Text (ST)

Der Programmierer für Strukturierten Text (ST) ist ein Texteditor, der zwar vom Aussehen her an PASCAL erinnert, der aber eine besonders für industrielle Anwendungen geeignet ist.

Mit Strukturiertem Text können POEs für Programme, Funktionen und Funktionsbausteine programmiert werden.

Einige Beispiele für IEC-konformen Strukturierten Text:

IFTHEN ELSE Bedingungen
CASE ...ELSE END_CASE Strukturen
REPEAT
RETURN
Mathematische Ausdrücke
Deklaration von Variablen etc.

Komplexe mathematische Ausdrücke können mit diesen Programmelementen in wenigen Textzeilen realisiert werden.

17.1 Elemente für Strukturierten Text

Element	Beschreibung	Vorrang bei der Programmausführung
(...)	Klammer	Höchster
Funktion (...)	Parameterliste einer Funktion, Auswertung einer Funktion	
**	Exponentialfunktion, Potenzierung	
–	Negation	
NOT	Boolsches Komplement	
*	Multiplikation	
/	Division	
MOD	Modulo-Operation	
+	Addition	
–	Subtraktion	
<, >, <=, >=	Vergleiche	
=	Gleichheit	
<>	Ungleichheit	
AND, &	UND-Verknüpfung	
XOR	Exklusiv-ODER-Verknüpfung	
OR	Oder-Verknüpfung	Niedrigster

17.2 Beispiel für Strukturierten Text

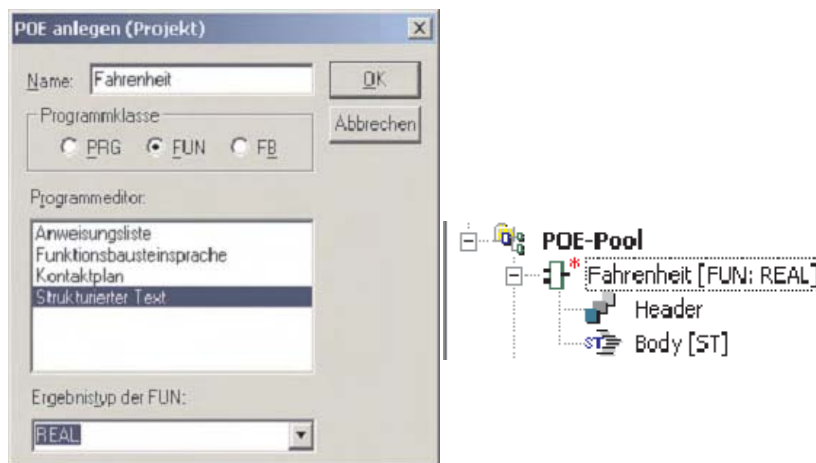
In diesem Beispiel wird eine Funktion im Strukturierten Text programmiert, die einen Wert in Grad Celsius in Grad Fahrenheit umrechnet. Diese Funktion ist das Gegenstück zu der Funktion, die im Abschnitt 6.1 in der Funktionsbausteinsprache programmiert wurde, und die einen Wert in Grad Fahrenheit in Grad Celsius wandelt.

Zur Umrechnung der Temperaturwerte wird die folgende Formel verwendet:

$$Grad_{Fahrenheit} = \frac{Grad_{Celsius} \times 9}{5} + 32$$

Als Ein- und Ausgangsvariablen werden Gleitkommazahlen verwendet (Datentyp REAL).

- ① Legen Sie ein neues Projekt mit dem Namen „Structured_Text“ an.
- ② Legen Sie eine neue POE mit dem Namen „Fahrenheit“ und der Programmklasse **FUN** (Funktion) an. Als Programmierer wählen Sie bitte **Strukturierter Text** und als Ergebnistyp der FUN **REAL**.



- ③ Tragen Sie im Header der Funktion „Fahrenheit“ die folgenden Daten ein:

	Klasse	Bezeichner	Typ	Start	Kommentar
0	VAR_INPUT	Centigrade	REAL	0.0	

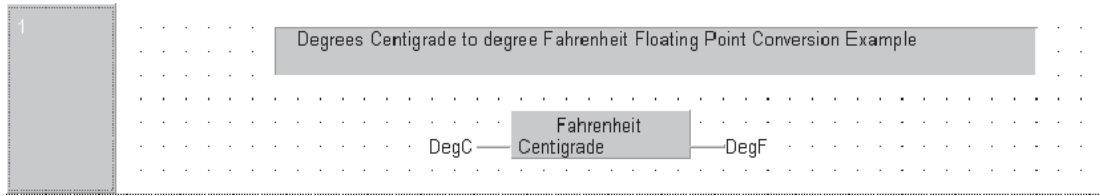
- ④ Öffnen Sie den Body der Funktion „Fahrenheit“ und geben Sie das folgende einfache ST-Programm ein:

Fahrenheit := (Centigrade*9.0/5.0+32.0);

- ⑤ Legen Sie eine neue POE mit dem Namen „Temp_Conv“ an. Als Programmklasse wählen Sie **PRG** und als Programmierer die **Funktionsbausteinsprache**.



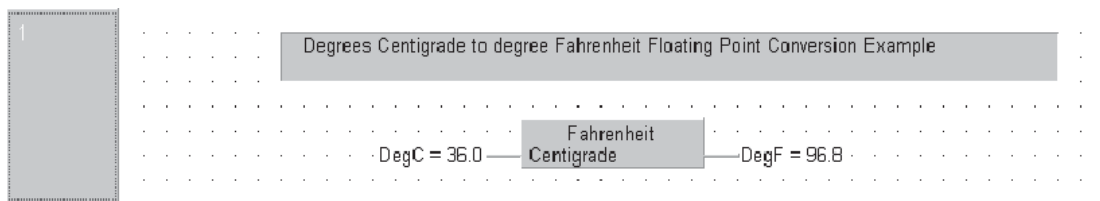
- ⑥ Öffnen Sie den Body der POE „Temp_Conv“ und programmieren Sie das unten abgebildete Beispiel:



- ⑦ Erweitern Sie die Lokale Variablenliste (Header) der POE „Temp_Conv“ um zwei lokale Variablen:

	Klasse	Bezeichner	Typ	Start	Kommentar
0	VAR	DegF	REAL	0.0	
1	VAR	DegC	REAL	0.0	

- ⑧ Schließen Sie alle Editoren und kompilieren Sie das Projekt mit der Funktion **Alles neu erstellen**. Speichern Sie dann das Projekt und übertragen Sie es in die SPS.
- ⑨ Beobachten Sie die Ausführung des Programms „Temp_Conv“ im Monitormodus.
- ⑩ Klicken Sie doppelt auf die Variable „DegC“, ändern Sie dann deren Wert und beachten Sie die Änderung der Ausgangsvariablen.



HINWEIS

In diesem Beispiel werden Lokale Variablen verwendet, um im Monitormodus des GX IEC Developer Werte direkt einzugeben. Normalerweise werden zur Eingabe von Werten Globale Variablen verwendet.

18 Kommunikation über ETHERNET

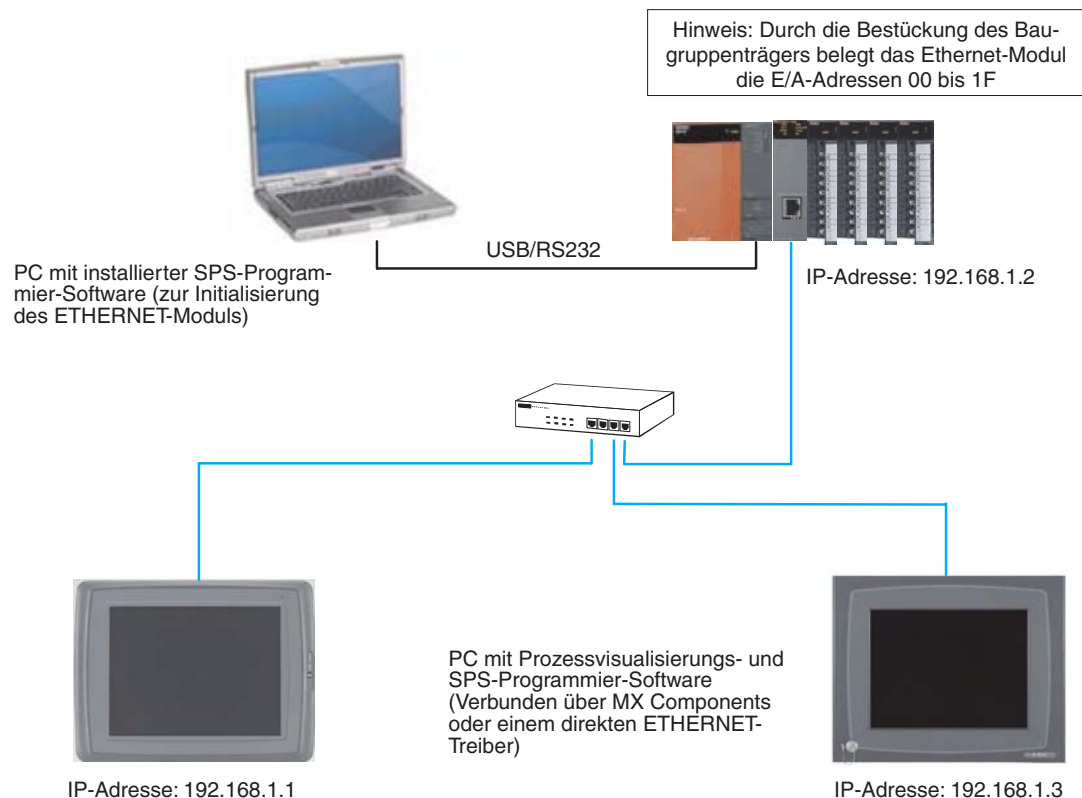
18.1 Parametrierung eines ETHERNET-Moduls

In diesem Abschnitt wird schrittweise beschrieben, wie ein ETHERNET-Modul QJ71E71 mit Hilfe der Parametereinstellung im GX IEC Developer (ab Version 7.00) konfiguriert werden kann.

Als Beispiel wird das ETHERNET-Modul für die TCP/IP-Kommunikation zwischen einer Q02HCPU und einen Prozessvisualisierungs-PC sowie einem grafischen Bediengerät vom Typ E1071 eingesetzt. Der PC für die Prozessvisualisierung kann – bei installierter Programmier-Software – auch zur Programmierung der SPS verwendet werden. Deshalb wird in diesem Kapitel auch gezeigt, wie mit dem GX IEC Developer über das ETHERNET auf die SPS-CPU zugegriffen werden kann.

Die folgende Abbildung zeigt die Konfiguration des Netzwerks für dieses Beispiel. Die verwendeten IP-Adressen sind ebenfalls angegeben.

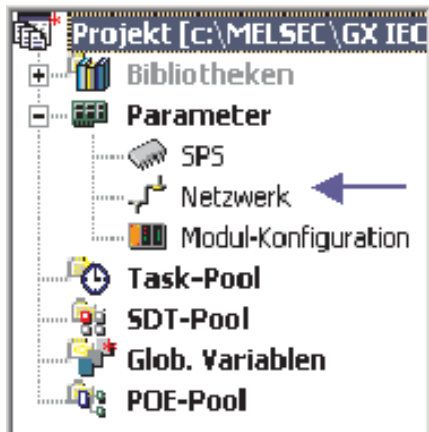
Bitte haben Sie Verständnis dafür, dass die Einstellungen für die SPS ausführlicher beschrieben sind, als die für den PC oder das Bediengerät. Bei diesen Geräten sind aber oft sehr spezifische Einstellungen notwendig, die im Rahmen dieses Handbuchs nicht behandelt werden können.



18.1.1 Einstellung der Netzwerk-Parameter

Bei dieser Konfiguration wird vorausgesetzt, dass die Programmier-Software mit den Voreinstellungen betrieben wird.

- ① Starten Sie den GX IEC Developer, öffnen Sie ein vorhandenes Projekt oder legen Sie ein neues Projekt an. Zur Einstellung der Netzwerk-Parameter klicken Sie dann im Navigatorfenster doppelt auf **Netzwerk**.



- ② Dadurch wird das unten abgebildete Auswahlfenster angezeigt. Klicken Sie auf **MELSECNET/Ethernet**.



Danach wird ein Dialogfenster geöffnet, in dem Parameter für bis zu vier Netzwerkmodule eingestellt werden können.

- ③ Klicken Sie in der Spalte für Modul 1 auf das Pfeilsymbol (▼) neben dem Eingabefeld **Netzwerktyp**. Dadurch wird eine Liste mit allen wählbaren Netzwerken angezeigt.

	Modul 1
Netzwerktyp	Keine ▼
Start-E/A-Adr.	
Netzwerk Nr.	
Anz. Stationen	
Gruppe Nr.	
Station Nr.	
Modus	▼

- ④ „ETHERNET“ finden Sie am Ende der Liste. Klicken Sie auf diesen Listeneintrag. Dadurch wird ETHERNET als Netzwerktyp übernommen.

Modul 1	
Netzwerktyp	Keine
Start-E./A-Adr.	MNET/10H-Modus (Normale Station)
Netzwerk Nr.	MNET/10-Modus (Kontrollstation)
Anz. Stationen	MNET/10-Modus (Normale Station)
Gruppe Nr.	MNET/10(H) Bereitschaftsstation
Station Nr.	MNET/H(Remote-Master)
Modus	Ethernet

- ⑤ In der Spalte für Modul 1 werden nun alle Einstellmöglichkeiten für ein ETHERNET-Modul angezeigt. Die unbedingt erforderlichen Einstellungen werden mit roter Schrift dargestellt. Die Einstellungen, die nicht unbedingt vorgenommen werden müssen, werden lila dargestellt. Stellen Sie diese Parameter bei Bedarf ein.

Modul 1	
Netzwerktyp	Ethernet
Start-E./A-Adr.	
Netzwerk Nr.	
Anz. Stationen	
Gruppe Nr.	0
Station Nr.	
Modus	Online
	Betriebseinstellungen
	Timer-DNS Einstellungen
	Verbindungs-Einstellungen
	Router-Relais-Parameter
	Stationsnr.<->IP-Information
	FTP-Parameter
	E-Mail-Einstellungen
	Interrupt-Einstellungen

- ⑥ Klicken Sie in die Eingabefelder in der oberen Hälfte der Spalte und geben Sie die benötigten Daten ein. Die folgende Abbildung zeigt die Einstellungen für die Beispielkonfiguration von Seite 18-1.

Modul 1	
Netzwerktyp	Ethernet
Start-E/A-Adr.	0000
Netzwerk Nr.	1
Anz. Stationen	
Gruppe Nr.	0
Station Nr.	2
Modus	Online
	Betriebseinstellungen
	Timer-DNS Einstellungen
	Verbindungs-Einstellungen
	Router-Relais-Parameter
	Stationsnr.<->IP-Information
	FTP-Parameter
	E-Mail-Einstellungen
	Interrupt-Einstellungen

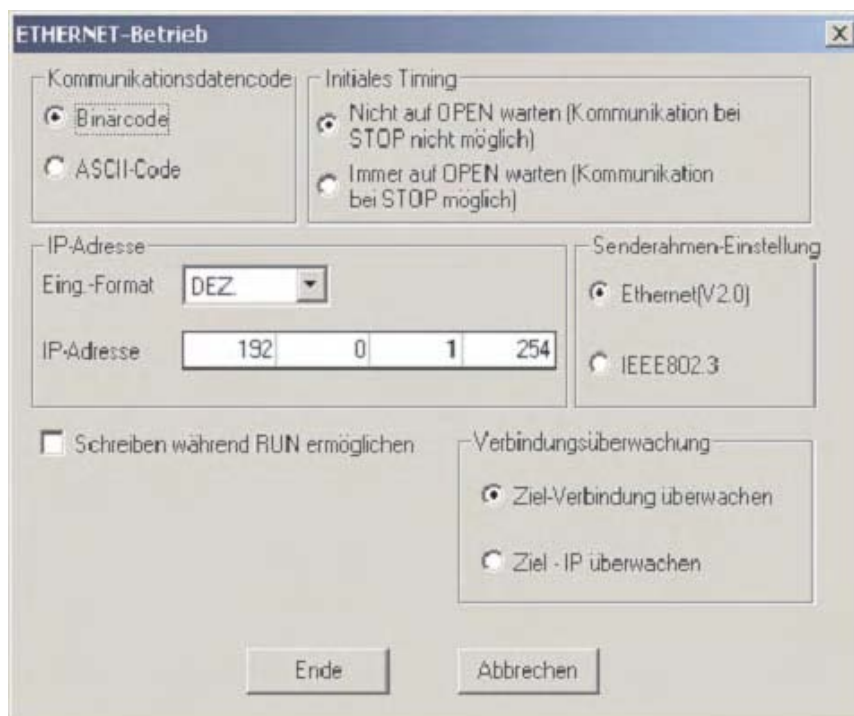
← Bitte beachten Sie den folgenden Hinweis.

← Bitte beachten Sie den folgenden Hinweis.

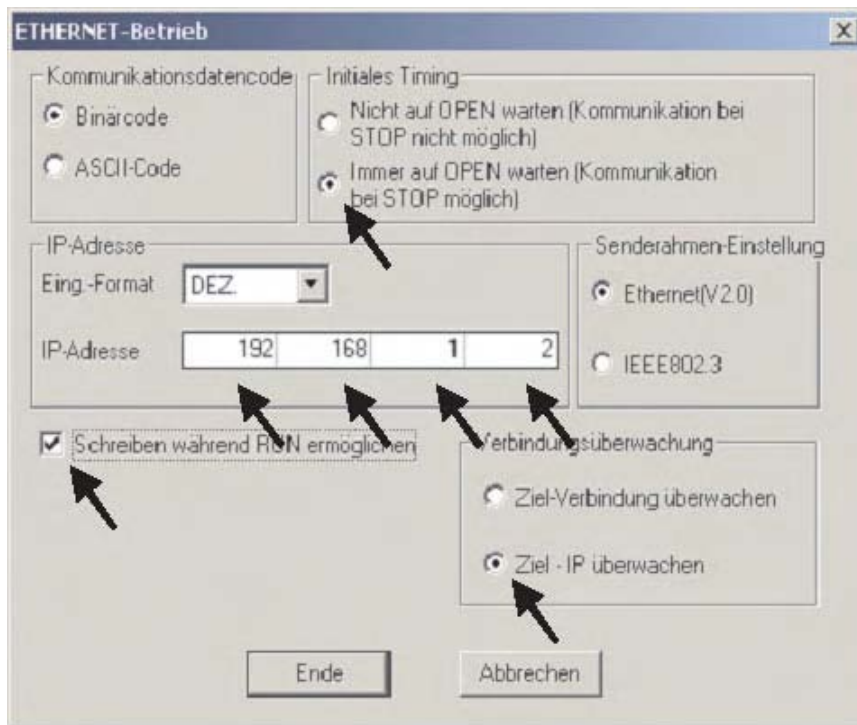
HINWEIS

Die „Netzwerk Nr.“ und die „Station Nr.“ werden zur Identifikation des Moduls bei der Kommunikation zwischen zwei ETHERNET-Modulen benötigt. In diesem Handbuch wird diese Art des Datenaustausches nicht behandelt. Diese Einstellungen sind auch erforderlich, wenn mit der Programmier-Software über das ETHERNET auf die SPS zugegriffen wird. Diese Möglichkeiten wird im Abschnitt 18.3 beschrieben.

- ⑦ Klicken Sie dann auf **Betriebseinstellungen**, um das unten abgebildete Dialogfenster zu öffnen. Die schon vorhandenen Einträge sind die Voreinstellungen, die die Programmier-Software vorgenommen hat.



- ⑧ Die folgende Abbildung zeigt die Betriebseinstellungen, die für die oben beschriebene Beispielkonfiguration erforderlich sind. Die Pfeile kennzeichnen die Unterschiede zu den Voreinstellungen.



- ⑨ Nachdem Sie die Einstellungen vorgenommen haben, klicken Sie auf **Ende**, um das Dialogfenster **Betriebseinstellungen** zu schließen. Bitte beachten Sie, dass die Schriftfarbe des Schaltfelds **Betriebseinstellungen** sich von rot nach blau geändert hat. Dadurch wird angezeigt, dass die Einstellungen verändert worden sind.

	Modul 1
Netzwerktyp	Ethernet
Start-E/A-Adr.	0000
Netzwerk Nr.	1
Anz. Stationen	
Gruppe Nr.	0
Station Nr.	2
Modus	Online
	Betriebseinstellungen
	Timer-DNS Einstellungen
	Verbindungs-Einstellungen
	Router-Relais-Parameter
	Stationsnr. <-> IP-Information
	FTP-Parameter
	E-Mail-Einstellungen
	Interrupt-Einstellungen

- ⑩ Klicken Sie dann auf **Verbindungs-Einstellungen**. Dadurch wird das unten abgebildete Dialogfenster angezeigt. Hier werden die Einstellungen für die Prozessvisualisierung und das grafische Bediengerät vorgenommen.

HINWEIS

Falls das ETHERNET-Modul **nur** dazu verwendet wird, um mit der Programmier-Software über das ETHERNET auf die SPS zuzugreifen, sind keine Verbindungseinstellungen erforderlich (siehe Abschnitt 18.3).

	Protokoll	System öffnen	Feste Puffer	Feste Puffer mit Prozedur	Paarige Verbindung	Verbindungsüberwachung	Lokale Port Nr.	Ziel IP-Adresse	Ziel Port Nr.
1									
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									

Ende Abbrechen

Die folgende Abbildung zeigt die Einstellungen, die in diesem Beispiel für die Kommunikation mit der Prozessvisualisierung und dem grafischen Bediengerät erforderlich sind. Die Einträge können Sie vornehmen, indem Sie auf das Pfeilsymbol (▼) neben einem Eingabefeld klicken und aus der dann angezeigten Liste eine mögliche Option wählen. Bei Eingabefeldern ohne diese Auswahlmöglichkeit geben Sie bitte die Daten direkt ein.

	Protokoll	System öffnen	Feste Puffer	Feste Puffer mit Prozedur	Paarige Verbindung	Verbindungsüberwachung	Lokale Port Nr.	Ziel IP-Adresse	Ziel Port Nr.
1	TCP ▼	Unpassiv ▼	Empfang ▼	möglich ▼	Keine Pa ▼	Bestätigen ▼	0401		
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									

Ende Abbrechen

Für HMI

- ⑪ Nachdem Sie alle Einstellungen vorgenommen haben, klicken Sie bitte auf **Ende**, um das Dialogfenster zu schließen und zur Einstellung der Netzwerk-Parameter zurückzukehren.

	Modul 1	Modul 2	Modul 3
Netzwerktyp	Ethernet	Keine	Keine
Start-E/A-Adr.	0000		
Netzwerk Nr.	1		
Anz. Stationen			
Gruppe Nr.	0		
Station Nr.	2		
Modus	Online		
	Betriebseinstellungen		
	Timer-DNS Einstellungen		
	Verbindungs-Einstellungen		
	Router-Relais-Parameter		
	Stationsnr.<->IP-Information		
	FTP-Parameter		
	E-Mail-Einstellungen		
	Interrupt-Einstellungen		

Notwendige Ohne Einst / Ist gesetzt) Gesetzt falls nötig(Ohne Eins / Ist gesetzt)

Start-E/A-Adr.: Zulässiges Modul bei Zugriff von anderen: 1

Übertragungsparameter Interlink: Geben Sie bitte die Start-E/A-Adr. des Moduls in HEX (16 Bit) ein.

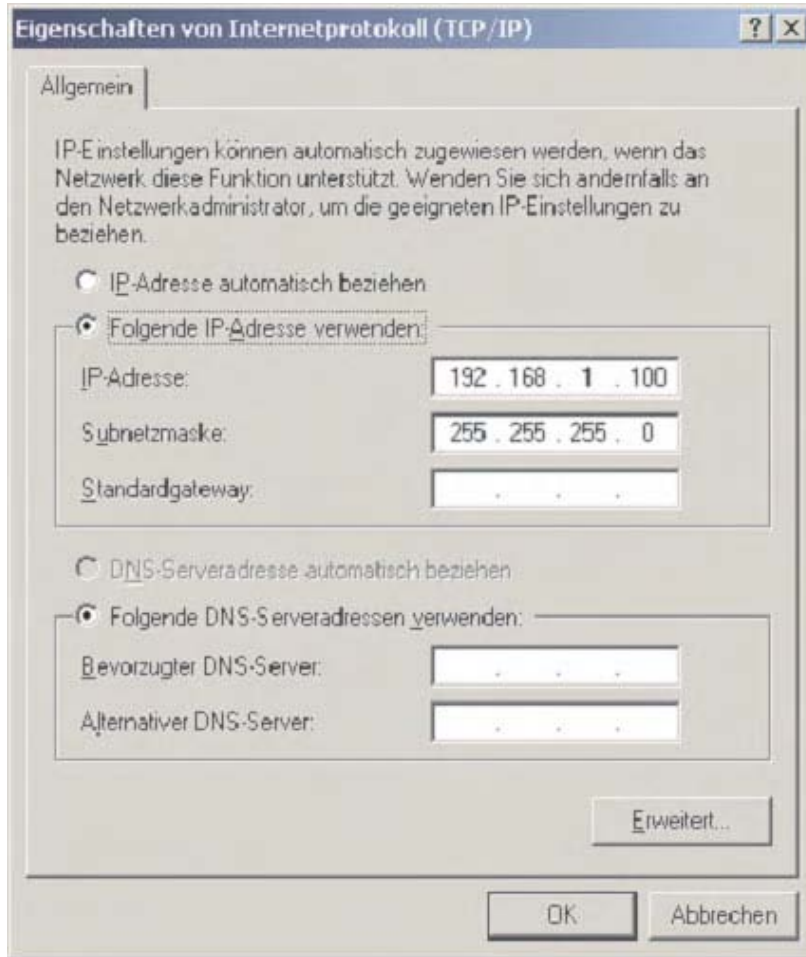
E/A-Zuweisung übernehmen Routing-Parameter Zuweisungskennz. Prüfen Ende Abbrechen

Für den Datenaustausch mit der Prozessvisualisierung und dem grafischen Bediengerät sind keine weiteren Einstellungen erforderlich.

- ⑫ Klicken Sie im Dialogfenster **Netzwerk-Parameter** auf das Schaltfeld **Ende**. Dadurch werden die Einstellungen geprüft und das Dialogfenster geschlossen. Wenn beim nächsten Transfer des Projekts in die SPS die Parameter mit übertragen werden, werden auch diese Einstellungen übertragen.

18.2 Einstellungen am PC für ein ETHERNET-Netzwerk

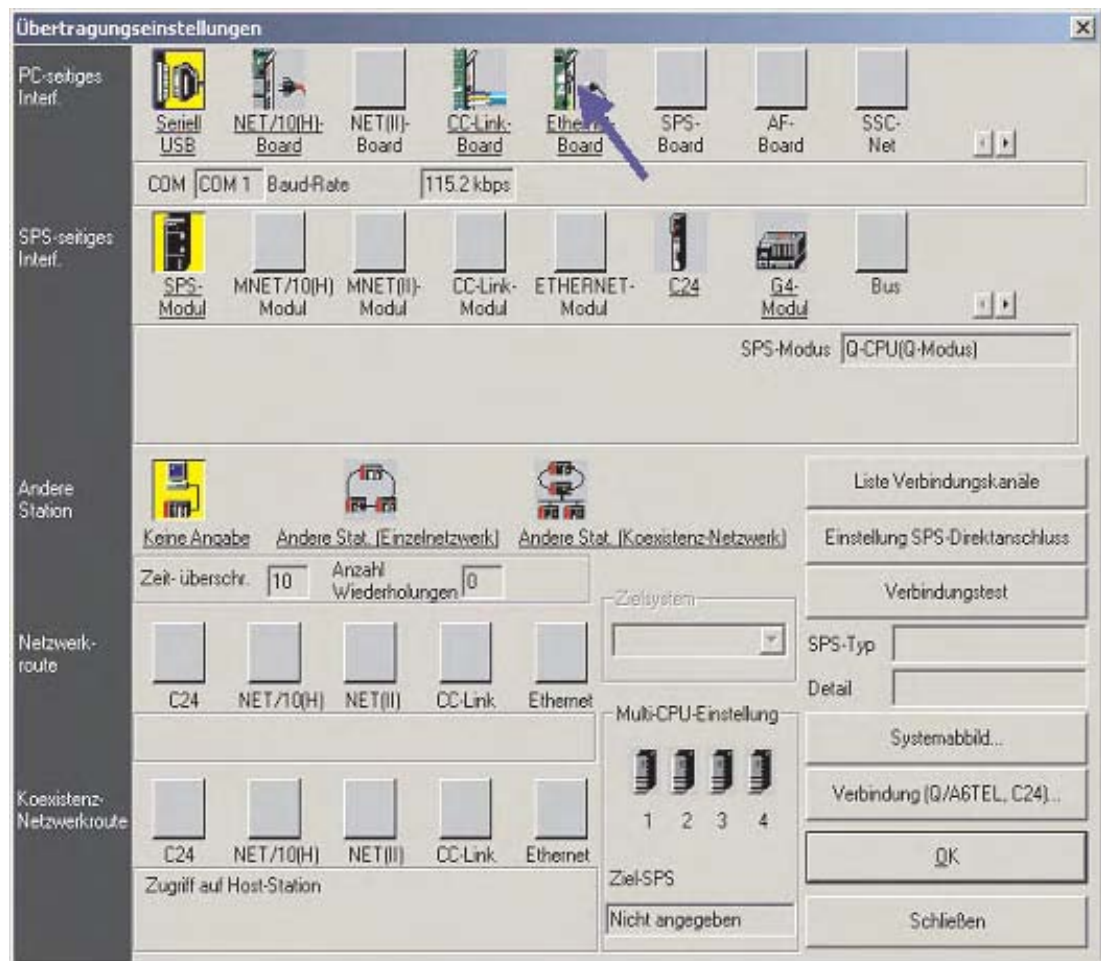
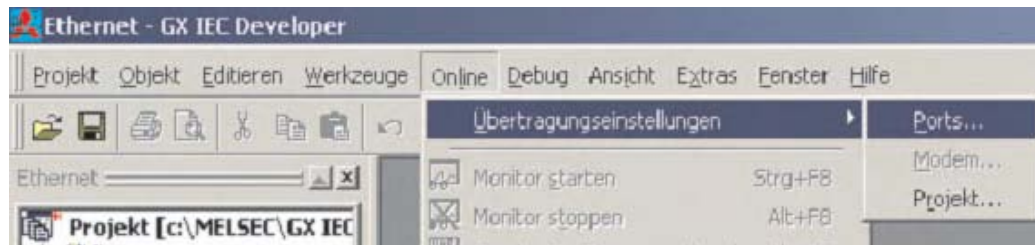
- ① Öffnen Sie Netzwerkeigenschaften unter Windows® und geben Sie im Dialogfenster TCP/IP-Eigenschaften eine IP-Adresse und eine Subnet-Mask an. Bitte beachten Sie, dass nach der Änderung der IP-Adresse ein Neustart des PC erforderlich ist.



18.3 Einstellungen zum Zugriff auf die SPS über das ETHERNET

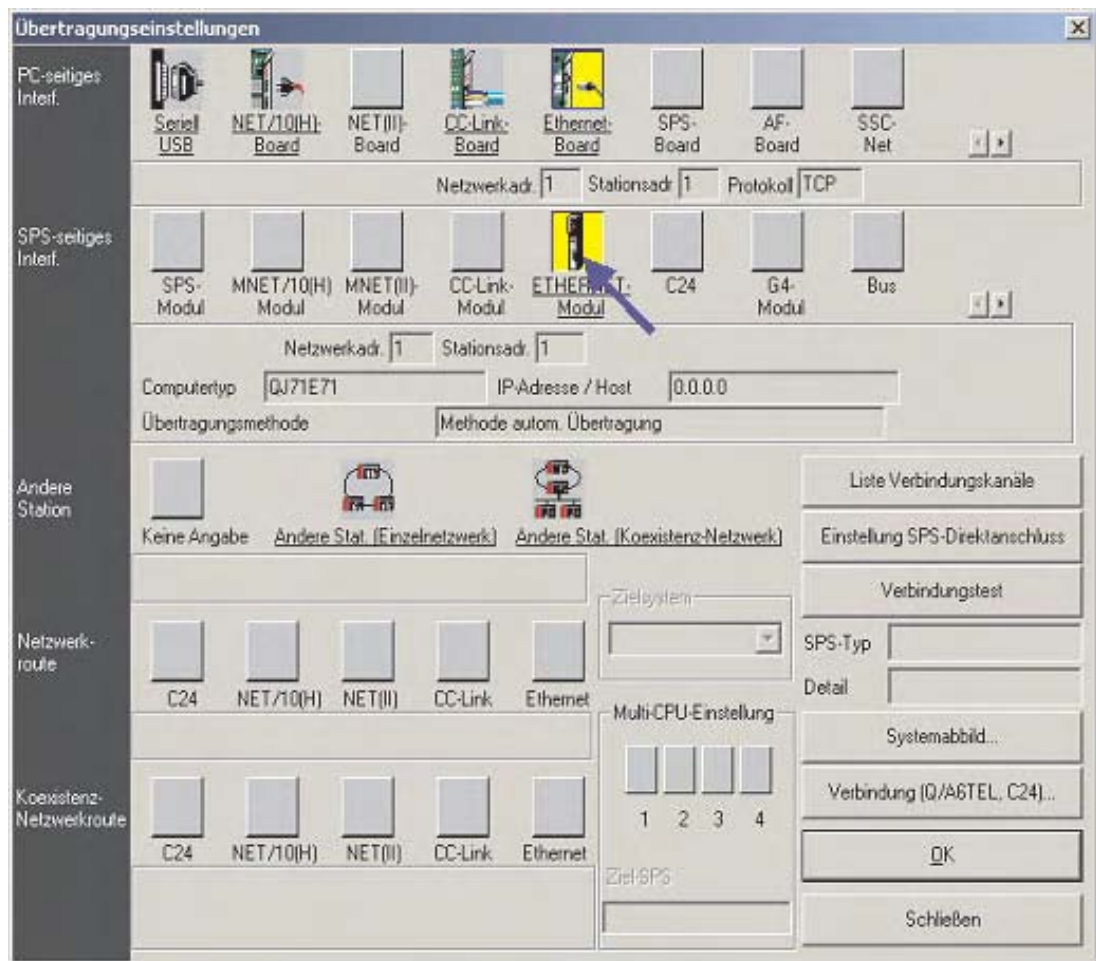
Um mit der Programmier-Software GX IEC Developer über ein ETHERNET-Netzwerk und einem ETHERNET-Modul auf eine SPS zuzugreifen, nehmen Sie bitte die folgenden Einstellungen vor.

- ① Klicken Sie im Menü **Online** auf **Übertragungseinstellungen** und dann auf **Ports**.

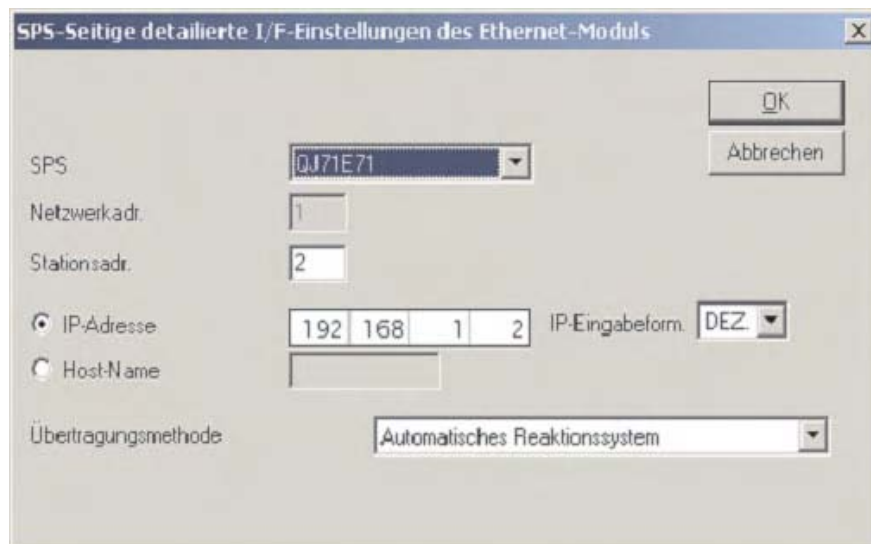


- ② Als Voreinstellung zur Kommunikation mit einer SPS ist als **PC-seitiges Interf.** (Schnittstelle am PC) eine serielle Schnittstelle gewählt (**Seriell, USB**). Ändern Sie diese Einstellung auf **Ethernet-Board**, indem Sie, wie oben gezeigt, auf diese Option klicken. Danach wird eine Meldung angezeigt, die Sie darauf hinweist, dass durch diese Änderung die aktuellen Einstellungen verloren gehen. Bestätigen Sie diese Meldung durch einen Klick auf **Ja**.

- ③ Für die Schnittstelle am PC (**PC-seitiges Interf.**) werden nun, wie unten gezeigt, als Voreinstellungen automatisch die **Netzwerkadr. 1** und die **Stationsadr. 1** eingetragen. Als **Protokoll** wird TCP verwendet. Falls diese Vorgaben nicht angezeigt werden, klicken Sie bitte doppelt auf **Ethernet-Board** und nehmen diese Einstellungen selbst vor.



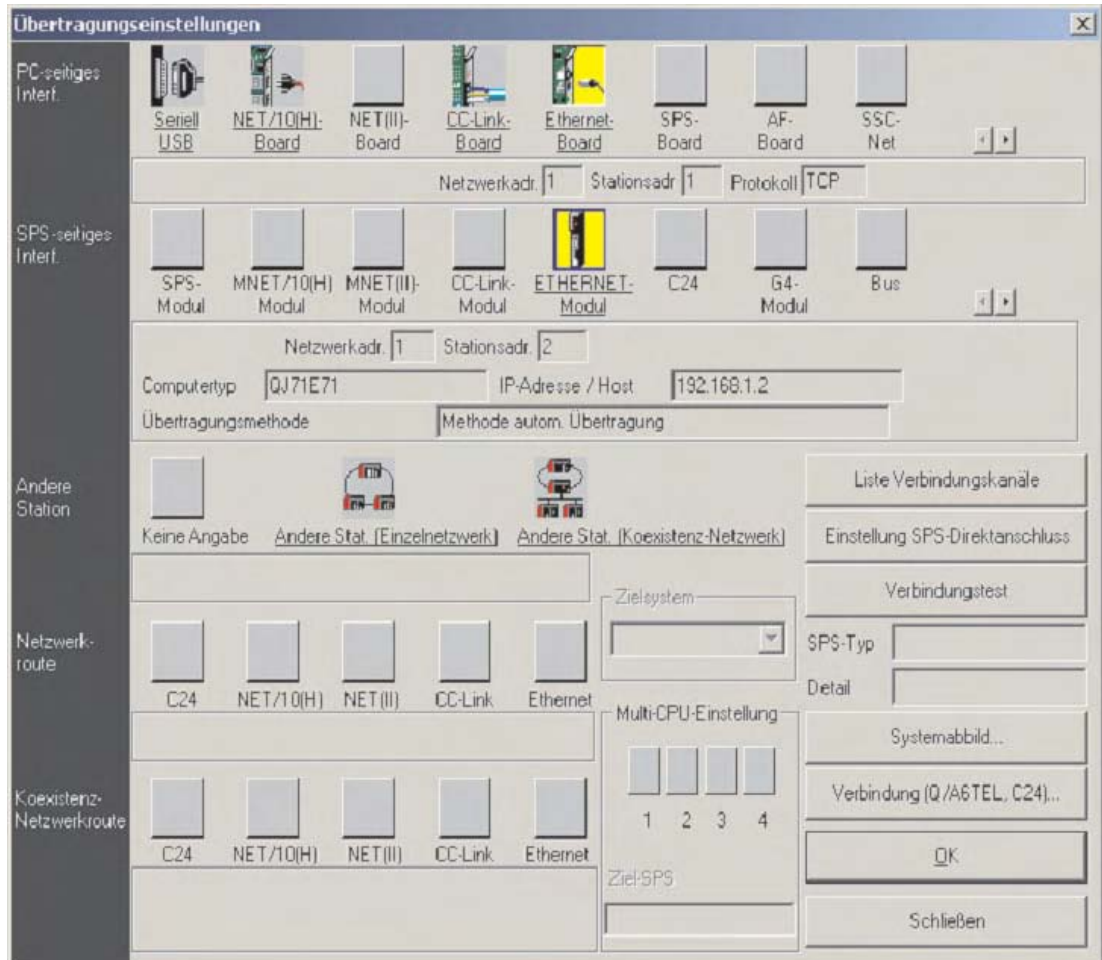
- ④ Klicken Sie, so wie oben gezeigt, in der Zeile **SPS-seitiges Interf.** doppelt auf **ETHERNET-Modul** unter **PLC side I/F**. Dadurch wird ein Dialogfenster geöffnet, in dem Einstellungen zum verwendeten ETHERNET-Modul vorgenommen werden können. Tragen Sie hier dieselben Daten ein, die auch als Netzwerk-Parameter gewählt wurden (siehe Schritte ⑥ und ⑦ in Abschnitt 18.1.1).



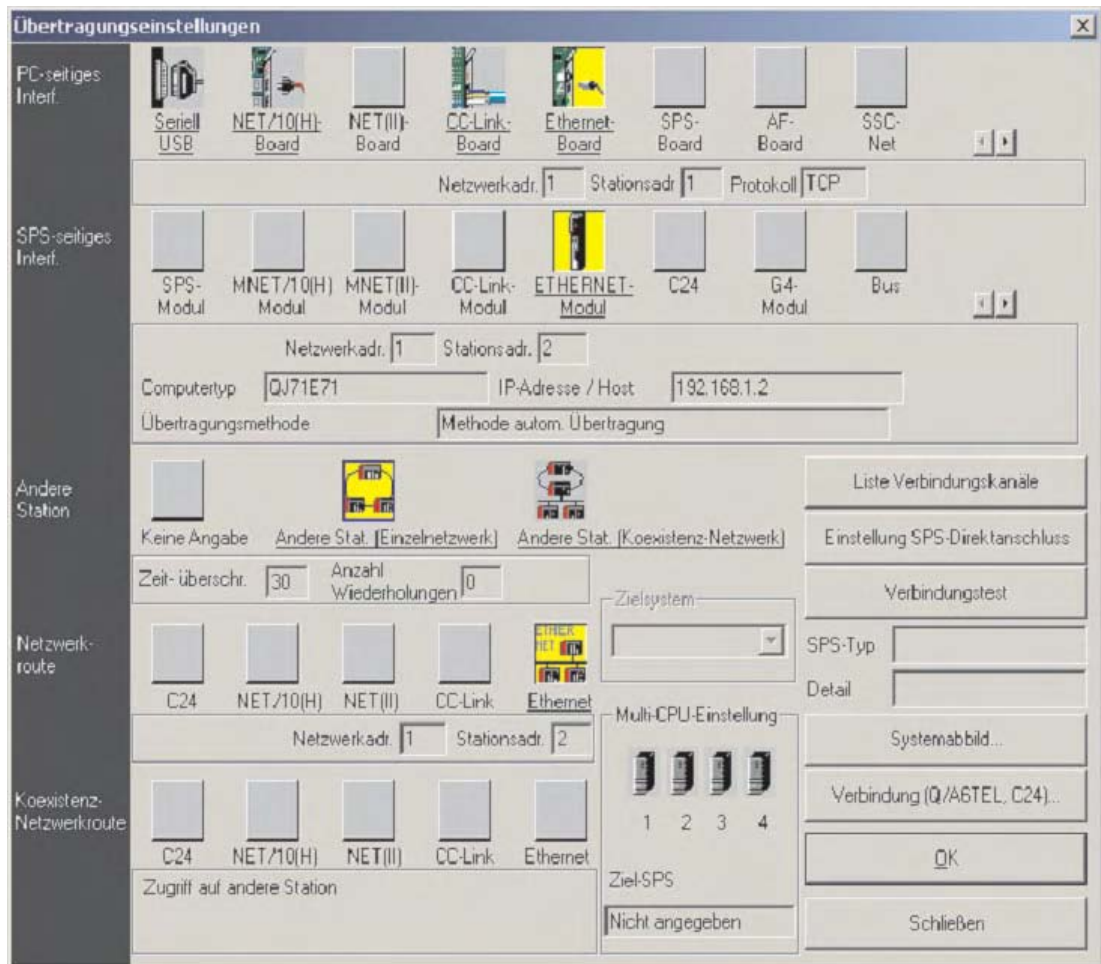
HINWEIS

Es muss keine Port-Nummer angegeben werden, weil die Programmier-Software einen Port verwendet, der für das MELSOFT-Protokoll reserviert ist.

- ⑤ Schließen Sie dieses Dialogfenster nach der Einstellung durch einen Klick auf **OK**.
- ⑥ Klicken Sie dann auf **Andere Station**.

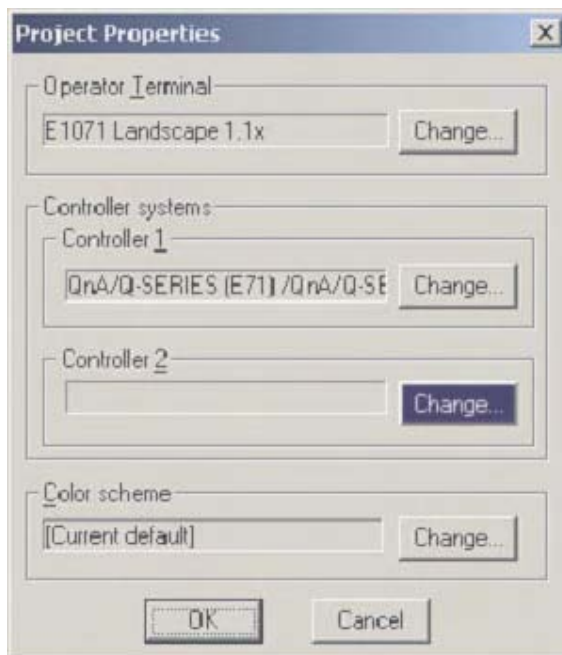


- ⑦ Dadurch werden die Einstellungen abgeschlossen. Das Dialogfenster Übertragungseinstellungen sollte nun so aussehen, wie unten gezeigt. Klicken Sie zum Prüfen der Einstellungen und der Kommunikation auf **Verbindungstest**. Klicken Sie nach dem erfolgreichen Test auf **OK**.

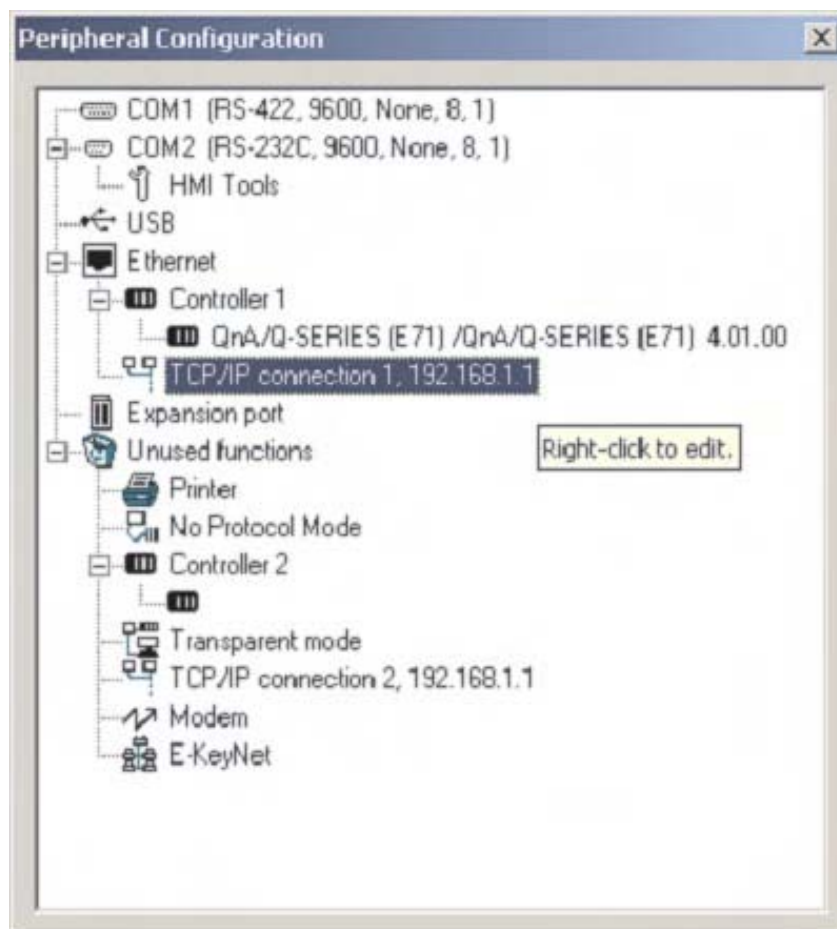


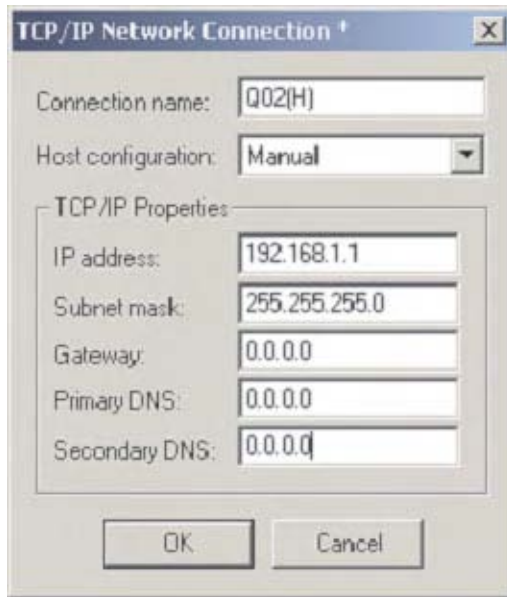
18.4 Einstellung des Bediengeräts

- ① Im E-Designer-Projekt für dieses Beispiel müssen die folgenden Einstellungen vorgenommen werden:

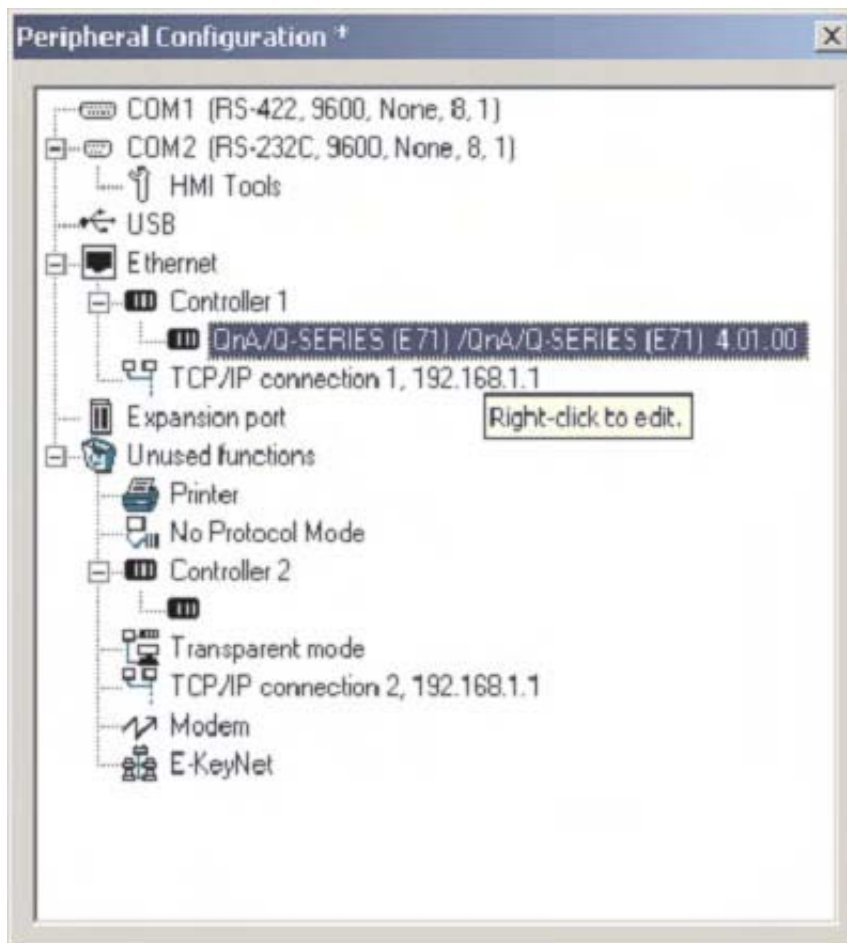


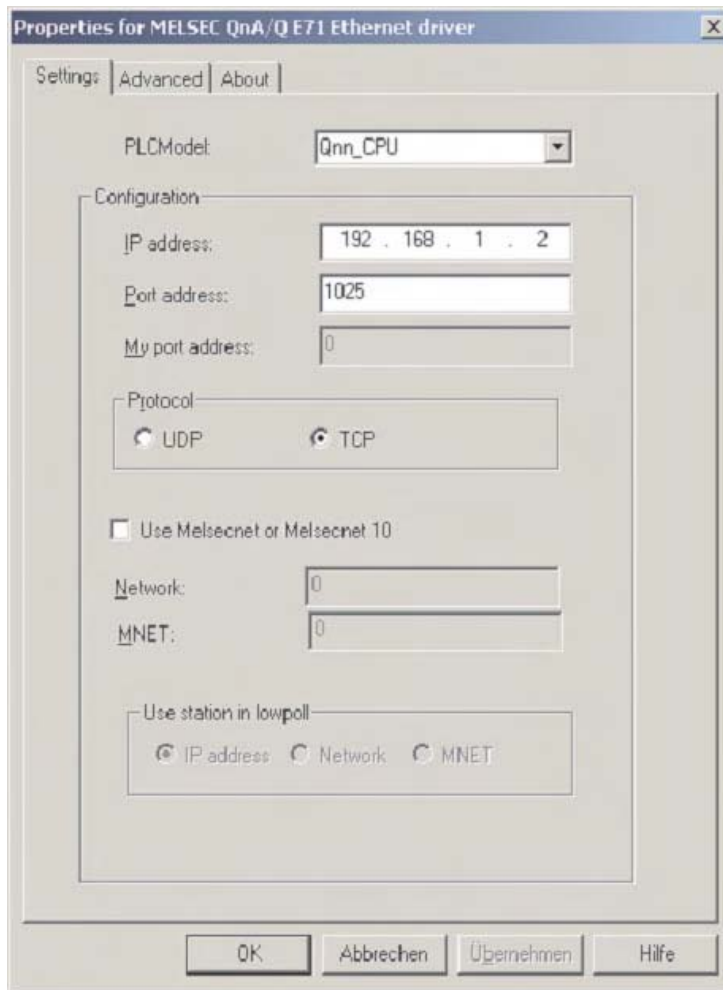
- ② Öffnen Sie als Nächstes die Option Peripherals im Menü System und konfigurieren Sie für das Bediengerät, so wie hier gezeigt, eine TCP/IP-Verbindung.





- ③ Tragen Sie dann für Controller 1 (das ist die Ziel-SPS) dieselben Einstellungen ein, die auch in den Netzwerk-Parametern dieser Steuerung eingestellt wurden.





Die dezimale Port-Nummer 1025 des ETHERNET-Moduls entspricht der hexadezimalen Zahl 401. Dieser hexadezimale Wert wurde in den Verbindungseinstellungen des ETHERNET-Moduls als Port-Nummer eingetragen (siehe Schritt ⑩ im Abschnitt 18.1.1).

- ④ Klicken Sie auf **OK**, um die Einstellungen zu übernehmen. Übertragen Sie diese Parameter in das Bediengerät.

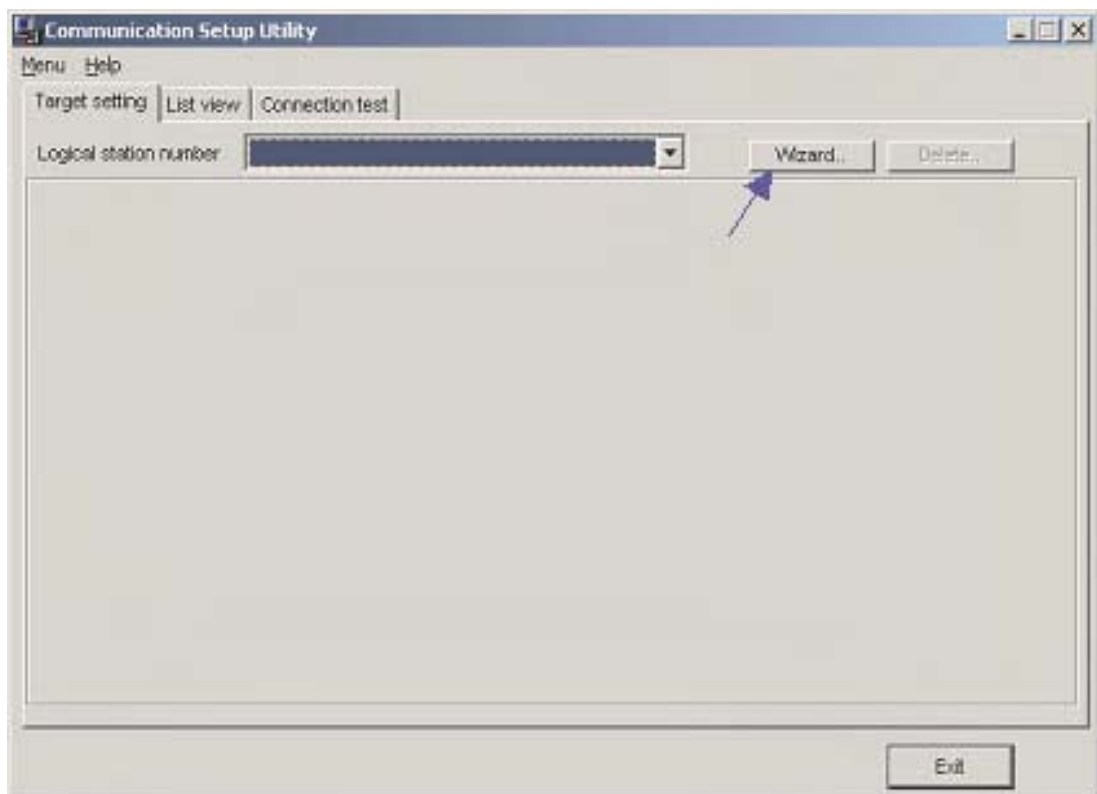
18.5 Kommunikation über MX Component

MX Component dient zur Konfiguration der Kommunikation zwischen einem PC und einer SPS. Vorkenntnisse über Kommunikationsprotokolle oder Module müssen dazu nicht vorhanden sein. MX Component ist ein leistungsfähiges und anwenderfreundliches Werkzeug, mit dem Ihre Mitsubishi-SPS sehr einfach mit Personal Computern verbunden werden kann.

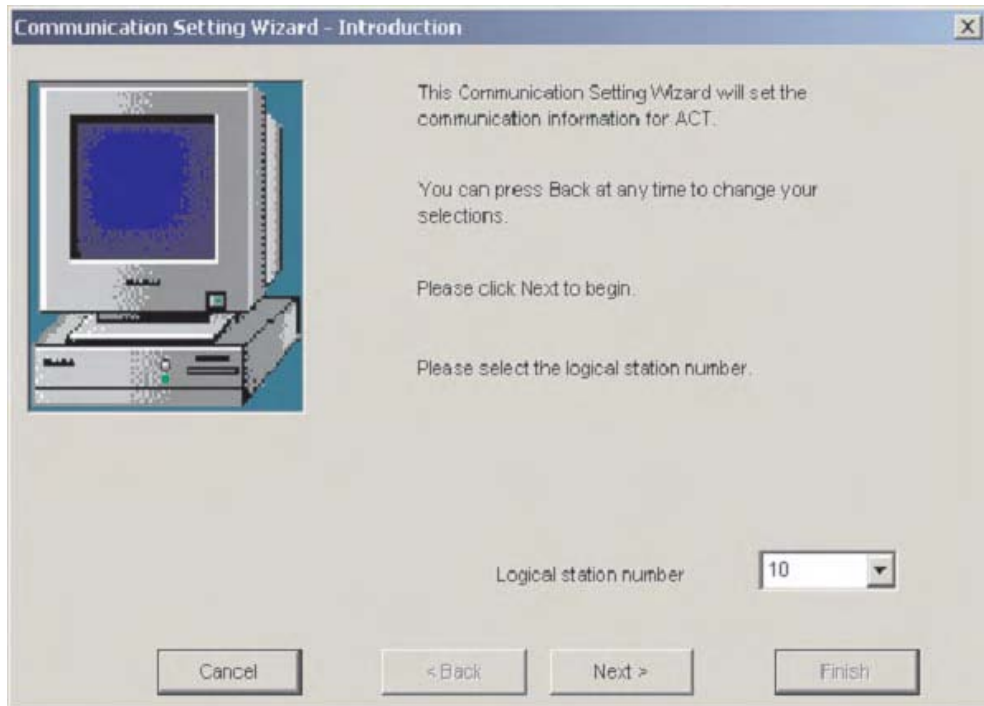
MX Component unterstützt die Kommunikation über die serielle Schnittstelle der CPU, über serielle Schnittstellenmodule (RS232C, RS422) und über Netzwerke (ETHERNET, CC-Link und MELSECNET).

Die folgende Abbildungen zeigen, wie einfach es ist, mit MX Component eine Kommunikation zwischen einem PC und einer SPS einzurichten.

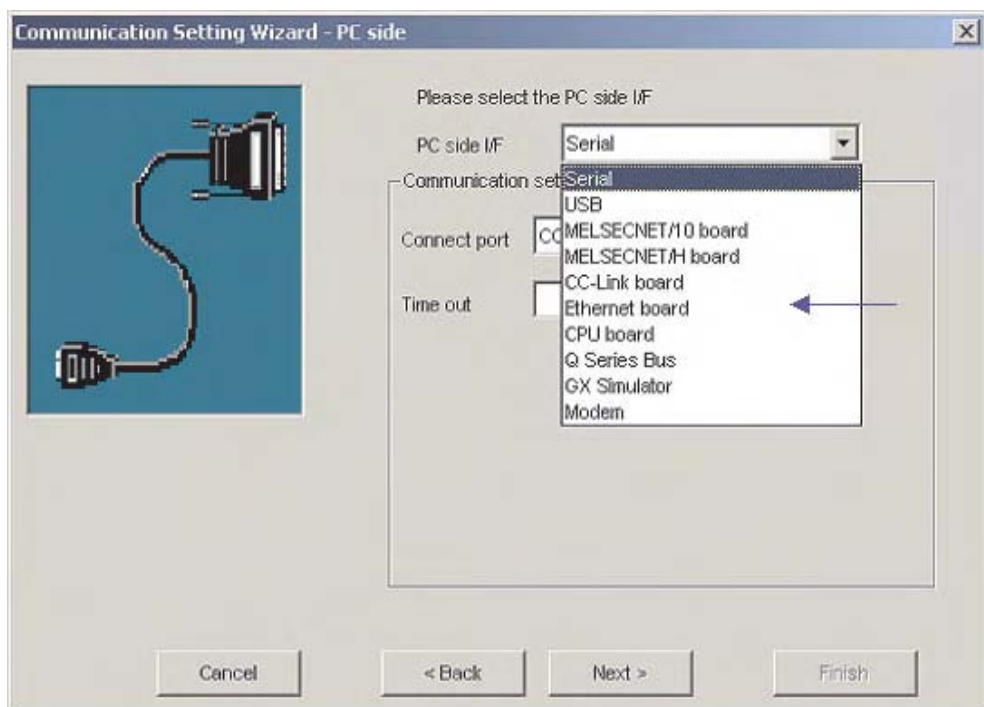
- ① Starten Sie die **Communication Setting Utility** und klicken Sie auf **Wizard**.



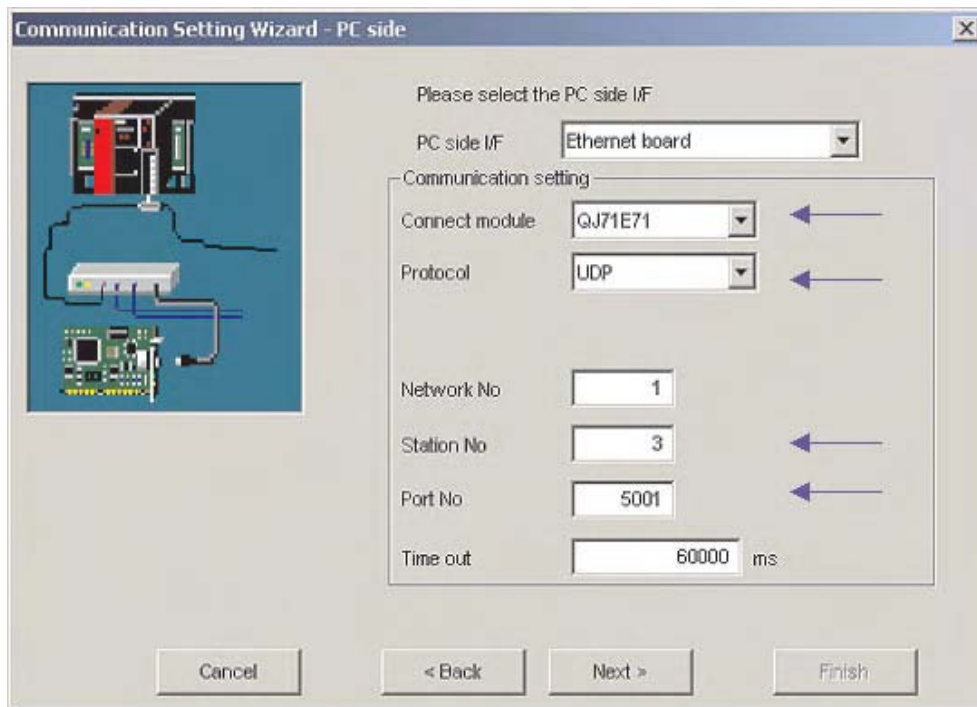
- ② Als Erstes müssen Sie die **Logical station number** angeben.



- ③ Dann konfigurieren Sie die Kommunikationseinstellungen (**Communication Settings**) am PC.



- ④ Wählen Sie das UDP-Protokoll und die Port-Nr. 5001.



Communication Setting Wizard - PC side

Please select the PC side I/F

PC side I/F: Ethernet board

Communication setting

Connect module: GJ71E71

Protocol: UDP

Network No: 1

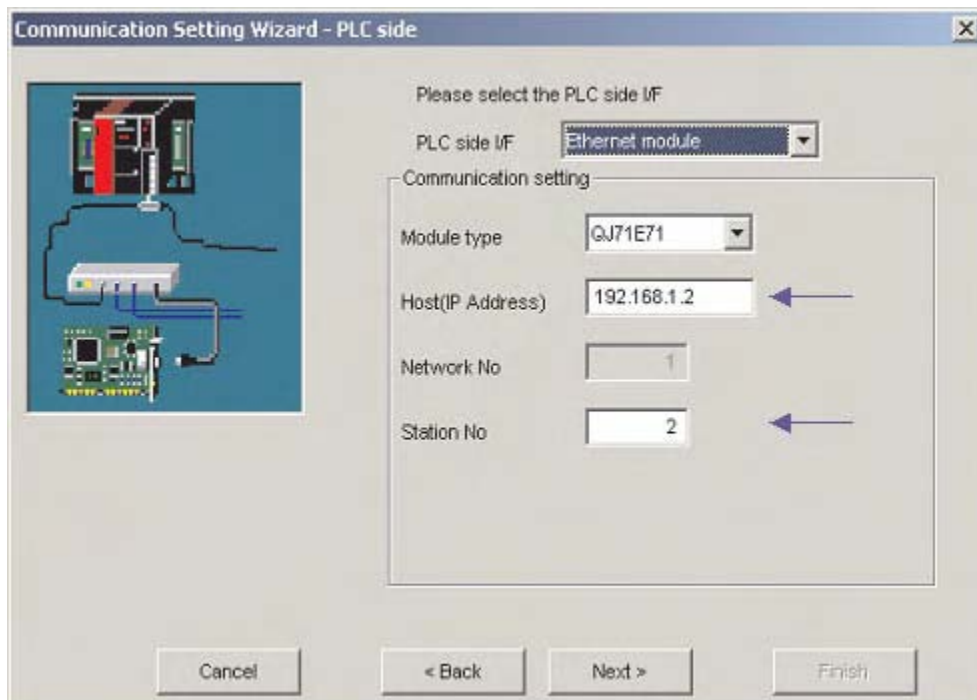
Station No: 3

Port No: 5001

Time out: 60000 ms

Buttons: Cancel, < Back, Next >, Finish

- ⑤ Geben Sie dann die IP-Adresse des ETHERNET-Moduls und die Stations-Nr. ein. Hier werden dieselben Werte eingestellt wie in den Netzwerk-Parametern innerhalb des GX IEC Developers (siehe Abschnitt 18.1.1).



Communication Setting Wizard - PLC side

Please select the PLC side I/F

PLC side I/F: Ethernet module

Communication setting

Module type: GJ71E71

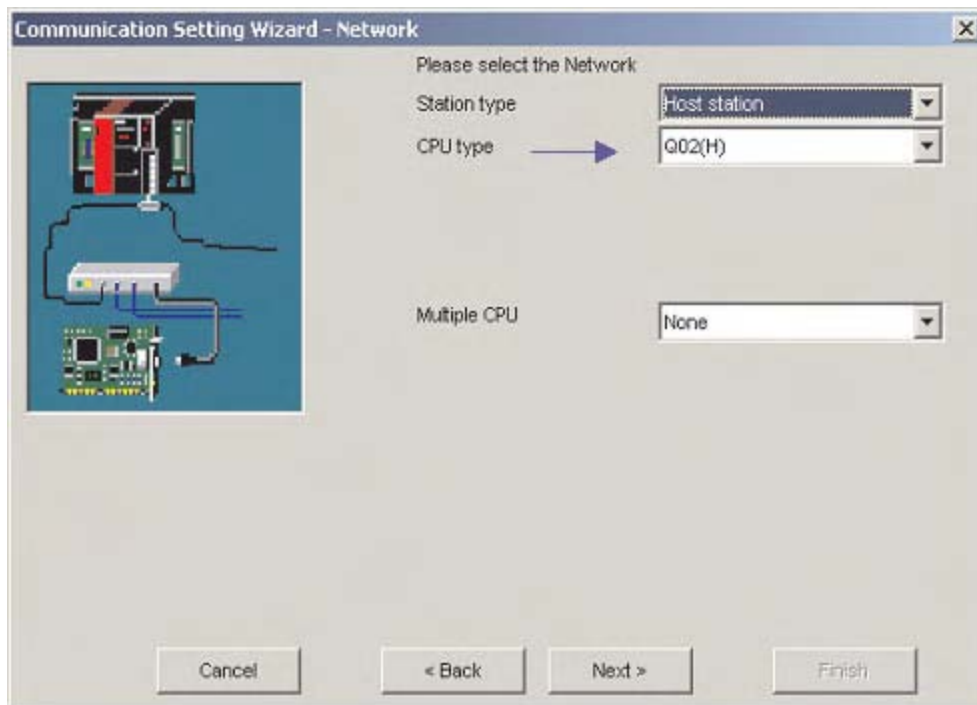
Host (IP Address): 192.168.1.2

Network No: 1

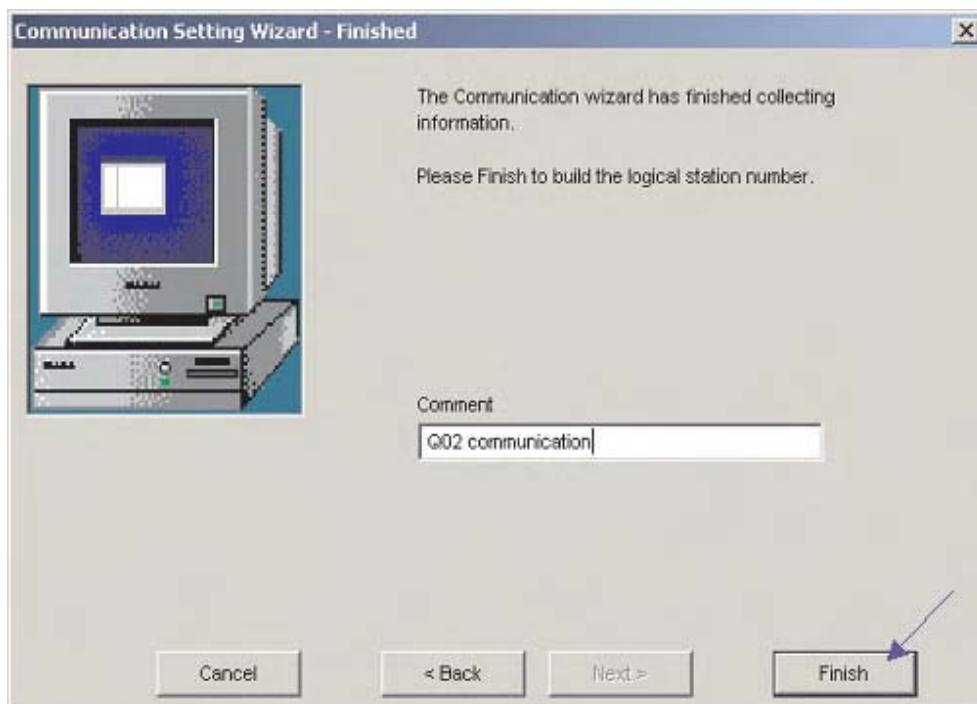
Station No: 2

Buttons: Cancel, < Back, Next >, Finish

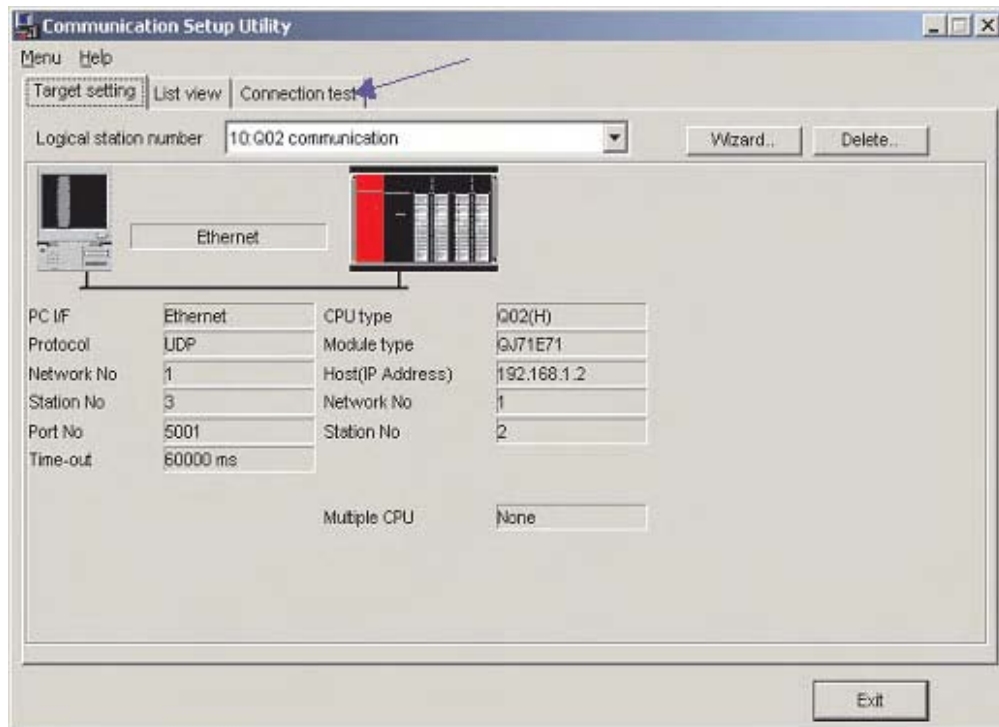
- ⑥ Wählen Sie den richtigen CPU-Typ.



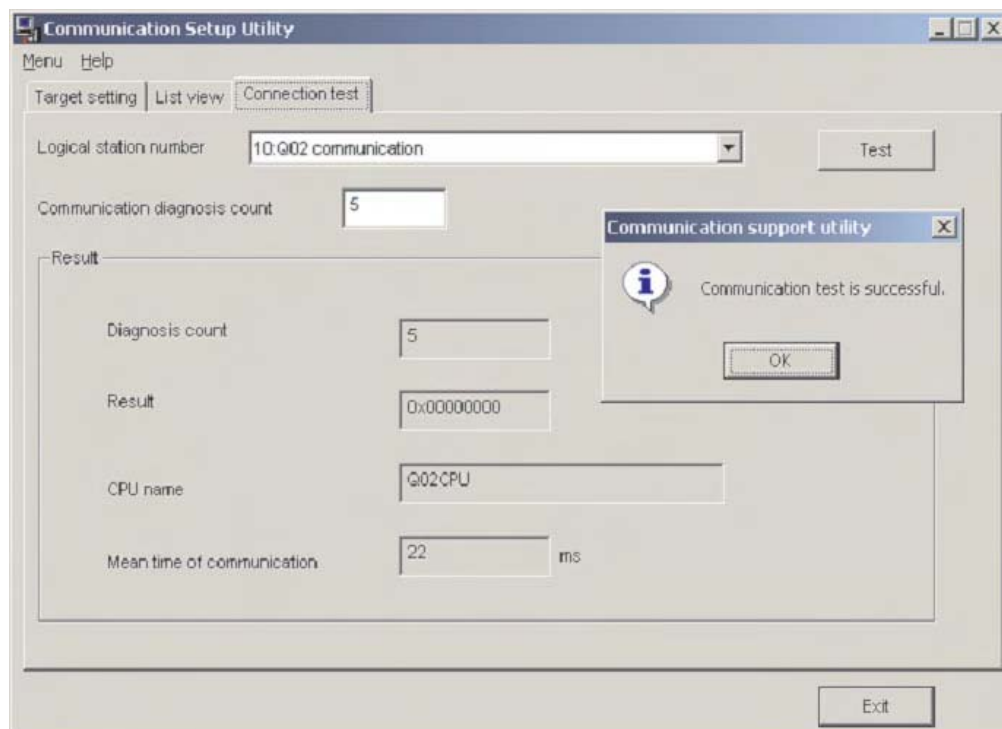
- ⑦ Zum Schluss wählen Sie einen Namen für diese Konfiguration und klicken auf **Finish**.



Damit sind alle Einstellungen zur Kommunikation festgelegt. Klicken Sie auf die Karteikarte **Connection test**, um die Verbindung zu prüfen.



Wählen Sie die Station (**Logical station number**), für die der Test ausgeführt werden soll. Im Feld **Diagnosis count** wird angezeigt, wie oft eine Verbindung erfolgreich aufgebaut werden konnte. Unter **Result** wird das Ergebnis des Test angezeigt. Falls ein Fehler aufgetreten ist, wird dessen Fehlercode angezeigt.



Nach der Konfiguration eines Kommunikationspfads haben Sie mit Microsoft Programmier-Software wie z. B. Visual Basic oder C++, Schreib- und Lesezugriff auf alle Operanden der SPS.

A Anhang

A.1 Diagnosemerker (SM)

Diagnosemerker (Operandenkennzeichen: SM) sind interne Merker, deren Anwendung in der SPS festgelegt ist. Aus diesem Grund können sie nicht wie andere Merker in Ablaufprogrammen verwendet werden. Einige Diagnosemerker können jedoch zur Steuerung der CPU ein- oder ausgeschaltet werden.

In diesem Abschnitt werden nicht alle Diagnosemerker beschrieben, sondern nur die, die am häufigsten verwendet werden.

HINWEISE

Die Diagnosemerker SM1200 bis SM1255 werden bei einer QnA-CPU verwendet. Bei einer SPS des MELSEC System Q sind diese Merker nicht belegt.

Die Diagnosemerker ab SM 1500 sind für die Q4AR-CPU reserviert.

Die Überschriften der Tabellen auf den nächsten Seiten haben die folgenden Bedeutungen:

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)	<p>Gibt Aufschluss darüber, ob der Diagnosemerker vom System oder vom Benutzer gesetzt wurde.</p> <p><Gesetzt vom></p> <p>S: Durch das System gesetzt</p> <p>B: Durch den Benutzer gesetzt (im Ablaufprogramm oder Prüfmodus eines Peripheriegerätes)</p> <p>S/B: Durch das System und den Benutzer gesetzt</p> <p>Wird nur gezeigt, wenn die Einstellung durch das System vorgenommen wurde.</p> <p><Wenn gesetzt></p> <p>END-Verarbeitung: Wird während jeder END-Verarbeitung gesetzt.</p> <p>Initialisierung: Wird nur während der Initialisierung gesetzt (beim Einschalten des Netzteils oder beim Umschalten der CPU vom STOP- in den RUN-Modus)</p> <p>Zustandsänderung: Wird nur nach Auftreten einer Zustandsänderung gesetzt</p> <p>Fehler: Wird nur nach Auftreten eines Fehlers gesetzt.</p> <p>Anweisungsausführung: Wird gesetzt, wenn die Anweisung ausgeführt wird.</p> <p>Anforderung: Wird nur gesetzt, wenn eine Benutzeranforderung ansteht (durch SM, etc.)</p>
A CPU M9[] [] []	<p>Zeigt Diagnosemerker M9 [] [] [] korrespondierend zur A-CPU. (Änderung und Schreibweise, wenn es inhaltliche Änderungen gab.)</p> <p>Wird mit „Neu“ gekennzeichnet, wenn der Diagnosemerker der Q-CPU neu hinzugefügt wurde.</p>
Gültig für:	<p>Gibt an, für welche CPU dieser Diagnosermerker zur Verfügung steht:</p> <p>●: Gilt für alle CPU-Typen</p> <p>Q-CPU: Gilt nur für die CPU-Module des MELSEC System Q</p> <p>QnA-CPU: Gilt für die CPUs der QnA-Reihe und die Q2AS-CPU</p> <p>CPU-Typ: Gilt nur für diese CPU (z. B. Q4AR-CPU)</p> <p>Rem: Gültig für dezentrale MELSECNET/H E/A-Module</p>

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	Selbstdiagnosefehler	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 Fehlerinformationen	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 zurücksetzen	AUS -> EIN: Fehler löschen	Mit diesem Merker kann ein Fehler zurückgesetzt werden.	B	Neu	
SM51	Niedrige Batteriespannung (Latch-Merker)	AUS: Normal EIN: Spannung zu niedrig	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 Batteriespannung	AUS: Normal EIN: Spannung zu niedrig	Gleiche Funktion wie SM51, der Merker wird jedoch zurückgesetzt, wenn die Batteriespannung wieder normal ist.	S (Fehler)	M9006	
SM53	Spannungsabfall bei der Versorgungsspannung	AUS: Normal EIN: Spannung zu niedrig Das Rücksetzen erfolgt mit dem Aus- und Wiedereinschalten der Versorgungsspannung.	Die Eingangsspannung des Wechselspannungsnetzteils ist für weniger als 20 ms abgefallen.	S (Fehler)	M9005	●
			Die Eingangsspannung des Netzteils mit Gleichspannungseingang ist für weniger als 10 ms abgefallen.			Q CPU
			Die Eingangsspannung des Netzteils mit Gleichspannungseingang ist für weniger als 1 ms abgefallen.			QnA-CPU
SM54	Fehler im MELSECNET/ MINI	AUS: Normal ON: 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	Verarbeitungsfehler	AUS: Normal ON: Fehler	Der Merker wird gesetzt, wenn ein Verarbeitungsfehler aufgetreten ist. Der Merker bleibt auch nach Wegfall der Störung gesetzt.	S (Fehler)	M9011	●
SM60	Sicherung defekt	AUS: Normal ON: 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	●
SM61	Vergleichsfehler E/A-Module	AUS: Normal ON: Fehler	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 dezentralen E/A-Station ausgeführt.	S (Fehler)	M9002	● Rem
SM62	Fehlermerkerkennung	AUS: Keine Kennung ON: Kennung	Wird gesetzt, wenn mindestens ein Fehlermerker F gesetzt wurde	S (Anweisungsausführung)	M9009	●



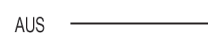
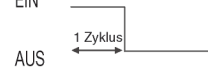
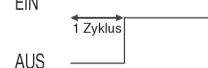



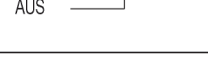
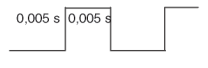
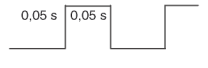


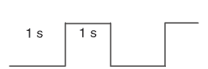
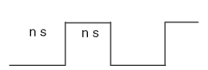

Adresse	Name	Bedeutung	Beschreibung	Gesetzt vom (wenn gesetzt)	A CPU M9[] [] []	Gültig für:	
SM80	Mittels CHK-Anweisung erkannter Fehler	AUS: Normal EIN: Fehler erkannt	Wird gesetzt, wenn ein Fehler mittels CHK-Anweisung erkannt wurde.	S (Anweisungsausführung)	Neu		
SM90	Start des WDT (Watchdog-Timer) 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	QnA CPU, Q CPU (außer Q00J, Q00 und Q01CPU)
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			New	
SM98			Korrespondierend zu SD98			New	
SM99			Korrespondierend zu SD99			New	

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, Q01CPU)
SM203	Kennung des STOP-Status	STOP-Status	Wird gesetzt, wenn die CPU stoppt.	S (Zustandsänderung)	M9042	●
SM204	Kennung des Pause-Status	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, 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 Q01 CPU
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: Kein Fehler EIN: Fehler	Der Merker ist gesetzt, wenn ein Fehler in den Werten der Uhrdaten, die in SD210 bis SD213 gespeichert sind, 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

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	Q02, Q02H, Q06H, Q12H, Q25H CPU ab Ver. B
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	

Systemtakte

Adresse	Name	Bedeutung	Beschreibung	Gesetzt vom (wenn gesetzt)	A-CPU M9[][][]	Gültig für:
SM400	Immer EIN	EIN  AUS	Dieser Sondermerker ist immer gesetzt (EIN).	S (END-Verarbeitung)	M9036	●
SM401	Immer AUS	EIN  AUS 	Dieser Sondermerker ist immer zurückgesetzt (AUS).	S (END-Verarbeitung)	M9037	
SM402	EIN nur für einen Programmzyklus nach RUN	EIN  AUS	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	EIN  AUS 	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	● außer Q00J-, Q00- und Q01CPU
SM404	EIN nur für einen Programmzyklus nach RUN	EIN  AUS	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	
SM405	AUS nur für einen Programmzyklus nach RUN	EIN  AUS 	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 Abschalten 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	Q-CPU, außer Q00J-, Q00- und Q01CPU
SM415	2 x n ms Takt		Wechselt zwischen EIN und AUS entsprechend der in SD415 eingestellten Anzahl von Millisekunden.	S (Zustandsänderung)	Neu	

Systemtakte (Fortsetzung)

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					

A.2 Übereinstimmungen zwischen Sonder- und Diagnosemerkern

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.

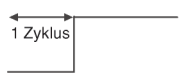
HINWEISE

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 äquivalenter 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.

A-CPU Sondermerker	Diagnosemerker nach der Umstellung	Äquivalente System Q-/QnA-Diagnosemerker	Name	Bedeutung	Gültig 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	

A-CPU Sondermerker	Diagnosemerker nach der Umstellung	Äquivalente System Q-/QnA-Diagnosemerker	Name	Bedeutung	Gültig 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 Kommunikationsanforderung	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	EIN ————— AUS	
M9037	SM1037		Ständig AUS	EIN AUS —————	
M9038	SM1038		EIN für 1 Zyklus nur nach RUN	EIN AUS	

A-CPU Sondermerker	Diagnosemerker nach der Umstellung	Äquivalente System Q-/QnA-Diagnosemerker	Name	Bedeutung	Gültig 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	Kennung 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 Gleich der Ausführung der ST RA-Anweisung 1 0 Gleich 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		

A-CPU Sondermerker	Diagnosemerker nach der Umstellung	Äquivalente System Q-/QnA-Diagnosemerker	Name	Bedeutung	Gültig 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/Stop 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)		

A-CPU Sondermerker	Diagnosemerker nach der Umstellung	Äquivalente System Q-/QnA-Diagnosemerker	Name	Bedeutung	Gültig 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: Ausgänge 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. und 3. 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	

A-CPU Sondermerker	Diagnosemerker nach der Umstellung	Äquivalente System Q-/QnA-Diagnosemerker	Name	Bedeutung	Gültig 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 Rückwärtsschleife	
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.3 Diagnoseregister (SD)

Die Diagnoseregister (Operandenkennzeichen: SD) sind interne Register mit einer festgelegten Aufgabe innerhalb der SPS. Aus diesem Grund ist es nicht möglich, diese Register in der gleichen Weise im Ablaufprogrammen zu nutzen wie normale Register. Zur Steuerung der CPU können jedoch in einige der Diagnoseregister Daten eingetragen werden.

Die in den Diagnoseregistern gespeicherten Daten werden im binären Format abgespeichert, es sei denn, es wird ein anderes Format gefordert.

In diesem Abschnitt werden nicht alle Diagnoseregister beschreiben, sondern nur die, die am häufigsten verwendet werden.

HINWEISE

Die Diagnoseregister SD1200 bis SD1255 werden von den CPUs der QnA-Serie verwendet. Bei einer CPU des MELSEC System Q sind die Sonderregister nicht belegt.

Die Diagnoseregister ab SD 1500 sind für die Q4AR-CPU reserviert.

Die Überschriften der Tabellen auf den nächsten Seiten haben die folgenden Bedeutungen:

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)	<p>Gibt Aufschluss darüber, ob das Diagnoseregister vom System oder vom Benutzer beschrieben wurde.</p> <p><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.</p> <p><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- 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.)</p>
Korrespondierende A-CPU Register D9 [] [] []	<p>Zeigt Diagnoseregister D9 [] [] [] korrespondierend zur A-CPU. (Änderung und Schreibweise, wenn es inhaltliche Änderungen gab.) Wird mit „Neu“ gekennzeichnet, wenn das Diagnoseregister der Q-CPU neu hinzugefügt wurde.</p>
Gültig für:	<p>Gibt an, für welche CPU dieses Diagnoseregister zur Verfügung steht:</p> <ul style="list-style-type: none"> ●: Gilt für alle CPU-Typen Q-CPU: Gilt nur für die CPU-Module des MELSEC System Q QnA-CPU: 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

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 Fehlerereignisinformationen.	S (Fehler)	D9008 Geändertes Format							
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 <table border="1" style="width: 100%; text-align: center;"> <tr> <td style="width: 33%;">b15</td> <td style="width: 33%;">b8 b7</td> <td style="width: 33%;">b0</td> </tr> <tr> <td>Jahr (0 bis 99)</td> <td>Monat (1 bis 12)</td> <td></td> </tr> </table>	b15	b8 b7	b0	Jahr (0 bis 99)	Monat (1 bis 12)		S (Fehler)	Neu	
b15			b8 b7	b0								
Jahr (0 bis 99)			Monat (1 bis 12)									
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 <table border="1" style="width: 100%; text-align: center;"> <tr> <td style="width: 33%;">b15</td> <td style="width: 33%;">b8 b7</td> <td style="width: 33%;">b0</td> </tr> <tr> <td>Tag (1 bis 31)</td> <td>Stunde (0 bis 23)</td> <td></td> </tr> </table>	b15	b8 b7	b0	Tag (1 bis 31)	Stunde (0 bis 23)						
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 <table border="1" style="width: 100%; text-align: center;"> <tr> <td style="width: 33%;">b15</td> <td style="width: 33%;">b8 b7</td> <td style="width: 33%;">b0</td> </tr> <tr> <td>Minute (0 bis 59)</td> <td>Sekunde (0 bis 59)</td> <td></td> </tr> </table>	b15	b8 b7	b0	Minute (0 bis 59)	Sekunde (0 bis 59)						
b15	b8 b7	b0										
Minute (0 bis 59)	Sekunde (0 bis 59)											
SD4	Kategorien der Fehlerinformationen	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 <table border="1" style="width: 100%; text-align: center;"> <tr> <td style="width: 33%;">b15</td> <td style="width: 33%;">b8 b7</td> <td style="width: 33%;">b0</td> </tr> <tr> <td>Spezifische Fehlerinformationen</td> <td>Allgemeine Fehlerinformationen</td> <td></td> </tr> </table> <p>Die Kategorie-Codes der allgemeinen Fehlerinformation werden wie folgt gespeichert:</p> <ul style="list-style-type: none"> 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) <p>Die Kategorie-Codes der spezifischen Fehlerinformation werden wie folgt gespeichert:</p> <ul style="list-style-type: none"> 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 	b15	b8 b7	b0	Spezifische Fehlerinformationen	Allgemeine Fehlerinformationen		S (Fehler)	Neu	
b15	b8 b7	b0										
Spezifische Fehlerinformationen	Allgemeine Fehlerinformationen											

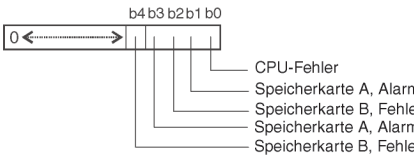
●
Rem

Adresse	Name	Bedeutung	Beschreibung	Gesetzt von (wenn gesetzt)	A-CPU-Register D9 [][][][]	Gültig für:																																																																																																																								
SD5	Allgemeine Fehlerinformation		<p>Die zu den Fehlercodes (SD0) korrespondierenden allgemeinen Informationen werden hier gespeichert. Die folgenden 5 Arten von Informationen werden gespeichert:</p> <p>(1) Stations-/Modulnummer</p> <table border="1"> <thead> <tr> <th>Nummer</th> <th>Bedeutung</th> </tr> </thead> <tbody> <tr> <td>SD5</td> <td>Stations-/Modulnummer</td> </tr> <tr> <td>SD6</td> <td>E/A-Nummer</td> </tr> <tr> <td>SD7</td> <td rowspan="8">Frei</td> </tr> <tr> <td>SD8</td> </tr> <tr> <td>SD9</td> </tr> <tr> <td>SD10</td> </tr> <tr> <td>SD11</td> </tr> <tr> <td>SD12</td> </tr> <tr> <td>SD13</td> </tr> <tr> <td>SD14</td> </tr> <tr> <td>SD15</td> </tr> </tbody> </table> <p>(2) File-Name/Laufwerksname</p> <table border="1"> <thead> <tr> <th>Nummer</th> <th>Bedeutung</th> </tr> </thead> <tbody> <tr> <td>SD5</td> <td>Laufwerk</td> </tr> <tr> <td>SD6</td> <td rowspan="3">File-Name (ASCII-Code: 8 Zeichen)</td> </tr> <tr> <td>SD7</td> </tr> <tr> <td>SD8</td> </tr> <tr> <td>SD9</td> <td>Erweiterung</td> </tr> <tr> <td>SD10</td> <td>2E_H(.)</td> </tr> <tr> <td>SD11</td> <td>(ASCII-Code: 3 Zeichen)</td> </tr> <tr> <td>SD12</td> <td rowspan="4">Frei</td> </tr> <tr> <td>SD13</td> </tr> <tr> <td>SD14</td> </tr> <tr> <td>SD15</td> </tr> </tbody> </table> <p>Beispiel: File-Name = ABCDEFGH.IJK</p> <table border="1"> <tr> <td>b15</td> <td>b0</td> </tr> <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> <p>(3) Zeit (eingestellter Wert)</p> <table border="1"> <thead> <tr> <th>Nummer</th> <th>Bedeutung</th> </tr> </thead> <tbody> <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</td> <td rowspan="8">Frei</td> </tr> <tr> <td>SD8</td> </tr> <tr> <td>SD9</td> </tr> <tr> <td>SD10</td> </tr> <tr> <td>SD11</td> </tr> <tr> <td>SD12</td> </tr> <tr> <td>SD13</td> </tr> <tr> <td>SD14</td> </tr> <tr> <td>SD15</td> </tr> </tbody> </table> <p>(4) Lokalisierung des Programmfehlers</p> <table border="1"> <thead> <tr> <th>Nummer</th> <th>Bedeutung</th> </tr> </thead> <tbody> <tr> <td>SD5</td> <td rowspan="4">File-Name (ASCII-Code: 8 Zeichen)</td> </tr> <tr> <td>SD6</td> </tr> <tr> <td>SD7</td> </tr> <tr> <td>SD8</td> </tr> <tr> <td>SD9</td> <td>Erweiterung</td> </tr> <tr> <td>SD10</td> <td>2E_H(.)</td> </tr> <tr> <td>SD11</td> <td>(ASCII-Code: 3 Zeichen)</td> </tr> <tr> <td>SD12</td> <td>Muster*</td> </tr> <tr> <td>SD13</td> <td>Block-Nr.</td> </tr> <tr> <td>SD14</td> <td>Schritt-/Transitions-Nr.</td> </tr> <tr> <td>SD15</td> <td>Ablaufschritt-Nr. (L)</td> </tr> <tr> <td></td> <td>Ablaufschritt-Nr. (H)</td> </tr> </tbody> </table> <p>*Belegung des Musters:</p> <table border="1"> <tr> <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> <td>0</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </table> <p>← (Bit Nr.)</p> <p>nicht verwendet</p> <p>AS-Block vorhanden (1) / nicht vorhanden (0) AS-Schritt vorhanden (1) / nicht vorhanden (0) AS-Transition vorhanden (1) / nicht vorhanden (0)</p>	Nummer	Bedeutung	SD5	Stations-/Modulnummer	SD6	E/A-Nummer	SD7	Frei	SD8	SD9	SD10	SD11	SD12	SD13	SD14	SD15	Nummer	Bedeutung	SD5	Laufwerk	SD6	File-Name (ASCII-Code: 8 Zeichen)	SD7	SD8	SD9	Erweiterung	SD10	2E _H (.)	SD11	(ASCII-Code: 3 Zeichen)	SD12	Frei	SD13	SD14	SD15	b15	b0	B	A	D	C	F	E	H	G	I	.	K	J	Nummer	Bedeutung	SD5	Zeit: 1 µs-Schritte (0 – 999 µs)	SD6	Zeit: 1 ms-Schritte (0 – 65535 ms)	SD7	Frei	SD8	SD9	SD10	SD11	SD12	SD13	SD14	SD15	Nummer	Bedeutung	SD5	File-Name (ASCII-Code: 8 Zeichen)	SD6	SD7	SD8	SD9	Erweiterung	SD10	2E _H (.)	SD11	(ASCII-Code: 3 Zeichen)	SD12	Muster*	SD13	Block-Nr.	SD14	Schritt-/Transitions-Nr.	SD15	Ablaufschritt-Nr. (L)		Ablaufschritt-Nr. (H)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	S (Fehler)	Neu	●
Nummer				Bedeutung																																																																																																																										
SD5				Stations-/Modulnummer																																																																																																																										
SD6				E/A-Nummer																																																																																																																										
SD7				Frei																																																																																																																										
SD8																																																																																																																														
SD9																																																																																																																														
SD10																																																																																																																														
SD11																																																																																																																														
SD12																																																																																																																														
SD13																																																																																																																														
SD14																																																																																																																														
SD15																																																																																																																														
Nummer				Bedeutung																																																																																																																										
SD5				Laufwerk																																																																																																																										
SD6	File-Name (ASCII-Code: 8 Zeichen)																																																																																																																													
SD7																																																																																																																														
SD8																																																																																																																														
SD9	Erweiterung																																																																																																																													
SD10	2E _H (.)																																																																																																																													
SD11	(ASCII-Code: 3 Zeichen)																																																																																																																													
SD12	Frei																																																																																																																													
SD13																																																																																																																														
SD14																																																																																																																														
SD15																																																																																																																														
b15	b0																																																																																																																													
B	A																																																																																																																													
D	C																																																																																																																													
F	E																																																																																																																													
H	G																																																																																																																													
I	.																																																																																																																													
K	J																																																																																																																													
Nummer	Bedeutung																																																																																																																													
SD5	Zeit: 1 µs-Schritte (0 – 999 µs)																																																																																																																													
SD6	Zeit: 1 ms-Schritte (0 – 65535 ms)																																																																																																																													
SD7	Frei																																																																																																																													
SD8																																																																																																																														
SD9																																																																																																																														
SD10																																																																																																																														
SD11																																																																																																																														
SD12																																																																																																																														
SD13																																																																																																																														
SD14																																																																																																																														
SD15																																																																																																																														
Nummer	Bedeutung																																																																																																																													
SD5	File-Name (ASCII-Code: 8 Zeichen)																																																																																																																													
SD6																																																																																																																														
SD7																																																																																																																														
SD8																																																																																																																														
SD9	Erweiterung																																																																																																																													
SD10	2E _H (.)																																																																																																																													
SD11	(ASCII-Code: 3 Zeichen)																																																																																																																													
SD12	Muster*																																																																																																																													
SD13	Block-Nr.																																																																																																																													
SD14	Schritt-/Transitions-Nr.																																																																																																																													
SD15	Ablaufschritt-Nr. (L)																																																																																																																													
	Ablaufschritt-Nr. (H)																																																																																																																													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																																																																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																																																																																																															

Bedeutung der Erweiterungen der Dateinamen

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

Adresse	Name	Bedeutung	Beschreibung	Gesetzt von (wenn gesetzt)	A-CPU-Register D9 [][][]	Gültig für:																																									
SD16	Allgemeine Fehlerinformation		<p>Die zu den Fehlercodes (SD0) korrespondierenden allgemeinen Informationen werden hier gespeichert. Die folgenden 6 Arten von Informationen werden hier gespeichert:</p> <p>(1) File-Name/Laufwerksname Beispiel: File-Name = ABCDEFHG.IJK</p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Nummer</th> <th>Bedeutung</th> </tr> </thead> <tbody> <tr> <td>SD16</td> <td>Laufwerk</td> </tr> <tr> <td>SD17</td> <td rowspan="2">File-Name (ASCII-Code: 8 Zeichen)</td> </tr> <tr> <td>SD18</td> </tr> <tr> <td>SD19</td> <td rowspan="2">Erweiterung 2E_H(.)</td> </tr> <tr> <td>SD20</td> </tr> <tr> <td>SD21</td> <td rowspan="2">Erweiterung 2E_H(.)</td> </tr> <tr> <td>SD22</td> </tr> <tr> <td>SD23</td> <td rowspan="4">Frei</td> </tr> <tr> <td>SD24</td> </tr> <tr> <td>SD25</td> </tr> <tr> <td>SD26</td> </tr> </tbody> </table> <div style="margin-left: 20px; margin-top: 10px;"> <table border="1"> <tr> <td style="width: 20px;">b15</td> <td style="width: 20px;">b0</td> </tr> <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> </div>	Nummer	Bedeutung	SD16	Laufwerk	SD17	File-Name (ASCII-Code: 8 Zeichen)	SD18	SD19	Erweiterung 2E_H(.)	SD20	SD21	Erweiterung 2E_H(.)	SD22	SD23	Frei	SD24	SD25	SD26	b15	b0	B	A	D	C	F	E	H	G	I	.	K	J	S (Fehler)	Neu	●									
Nummer				Bedeutung																																											
SD16				Laufwerk																																											
SD17				File-Name (ASCII-Code: 8 Zeichen)																																											
SD18																																															
SD19				Erweiterung 2E_H(.)																																											
SD20																																															
SD21				Erweiterung 2E_H(.)																																											
SD22																																															
SD23				Frei																																											
SD24																																															
SD25																																															
SD26																																															
b15	b0																																														
B	A																																														
D	C																																														
F	E																																														
H	G																																														
I	.																																														
K	J																																														
SD17																																															
SD18																																															
SD19																																															
SD20																																															
SD21																																															
SD22																																															
SD23																																															
SD24																																															
SD25																																															
SD26																																															
			<p>(2) Zeit (eingestellter Wert)</p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Nummer</th> <th>Bedeutung</th> </tr> </thead> <tbody> <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</td> <td rowspan="10">Frei</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> </tbody> </table>	Nummer	Bedeutung	SD16	Zeit: 1 µs-Schritte (0 – 999 µs)	SD17	Zeit: 1 ms-Schritte (0 – 65535 ms)	SD18	Frei	SD19	SD20	SD21	SD22	SD23	SD24	SD25	SD26																												
Nummer	Bedeutung																																														
SD16	Zeit: 1 µs-Schritte (0 – 999 µs)																																														
SD17	Zeit: 1 ms-Schritte (0 – 65535 ms)																																														
SD18	Frei																																														
SD19																																															
SD20																																															
SD21																																															
SD22																																															
SD23																																															
SD24																																															
SD25																																															
SD26																																															
				<p>(3) Lokalisierung des Programmfehlers</p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Nummer</th> <th>Bedeutung</th> </tr> </thead> <tbody> <tr> <td>SD16</td> <td rowspan="4">File-Name (ASCII-Code: 8 Zeichen)</td> </tr> <tr> <td>SD17</td> </tr> <tr> <td>SD18</td> </tr> <tr> <td>SD19</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> </tbody> </table> <p>*Belegung des Musters:</p> <table border="1" style="margin-left: 20px;"> <tr> <td style="width: 15px;">15</td><td style="width: 15px;">14</td><td style="width: 15px;">--</td><td style="width: 15px;">--</td><td style="width: 15px;">4</td><td style="width: 15px;">3</td><td style="width: 15px;">2</td><td style="width: 15px;">1</td><td style="width: 15px;">0</td><td style="width: 15px;">← (Bit Nr.)</td> </tr> <tr> <td>0</td><td>0</td><td>--</td><td>--</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td> </tr> </table> <p style="margin-left: 20px;"> nicht verwendet AS-Block vorhanden (1) / nicht vorhanden (0) AS-Schritt vorhanden (1) / nicht vorhanden (0) AS-Transition vorhanden (1) / nicht vorhanden (0) </p>	Nummer	Bedeutung	SD16	File-Name (ASCII-Code: 8 Zeichen)	SD17	SD18	SD19	SD20	Erweiterung 2E_H(.)	SD21	(ASCII-Code: 3 Zeichen)	SD22	Muster*	SD23	Block-Nr.	SD24	Schritt-/Transitions-Nr.	SD25	Ablaufschritt-Nr. (L)	SD26	Ablaufschritt-Nr. (H)	15	14	--	--	4	3	2	1	0	← (Bit Nr.)	0	0	--	--	0	0	1	1	1	1		
Nummer	Bedeutung																																														
SD16	File-Name (ASCII-Code: 8 Zeichen)																																														
SD17																																															
SD18																																															
SD19																																															
SD20	Erweiterung 2E_H(.)																																														
SD21	(ASCII-Code: 3 Zeichen)																																														
SD22	Muster*																																														
SD23	Block-Nr.																																														
SD24	Schritt-/Transitions-Nr.																																														
SD25	Ablaufschritt-Nr. (L)																																														
SD26	Ablaufschritt-Nr. (H)																																														
15	14	--	--	4	3	2	1	0	← (Bit Nr.)																																						
0	0	--	--	0	0	1	1	1	1																																						

Adresse	Name	Bedeutung	Beschreibung	Gesetzt von (wenn gesetzt)	A-CPU-Register D9 [] [] []	Gültig für:				
SD16	Spezifische Fehlerinformationen		(4) Parameter-Nr (5). Fehlermerker- Nr./ CHK-Anweisungs- Fehlfunktions-Nr.	S (Fehler)	Neu	●				
SD17										
SD18										
SD19										
SD20										
SD21										
SD22										
SD23										
SD24										
SD25										
SD26										
SD26							(6) Parametrierfehler bei Sondermodulen (Nur bei CPUs des MELSEC System Q)	S (Fehler)	Neu	●
SD16			Parameter-Nr.				SD16			
SD17	Nicht belegt	SD17	Nicht belegt							
SD18		SD18								
SD19		SD19								
SD20		SD20								
SD21		SD21								
SD22		SD22								
SD23		SD23								
SD24		SD24								
SD25		SD25								
SD26		SD26								
SD26				(6) Parametrierfehler bei Sondermodulen (Nur bei CPUs des MELSEC System Q)	S (Fehler)	Neu	●			
SD16	Parameter-Nr.	SD17	Fehlercode für Sondermodul							
SD18	Nicht belegt	SD18	Nicht belegt							
SD19		SD19								
SD20		SD20								
SD21		SD21								
SD22		SD22								
SD23		SD23								
SD24		SD24								
SD25		SD25								
SD26		SD26								
SD26				(6) Parametrierfehler bei Sondermodulen (Nur bei CPUs des MELSEC System Q)	S (Fehler)	Neu	●			
SD16	Parameter-Nr.	SD17	Fehlercode für Sondermodul							
SD18	Nicht belegt	SD18	Nicht belegt							
SD19		SD19								
SD20		SD20								
SD21		SD21								
SD22		SD22								
SD23		SD23								
SD24		SD24								
SD25		SD25								
SD26		SD26								
SD50	Rücksetzen eines Fehlers	Fehlernummer des zurückgesetzten Fehlers	Speichert die Fehlernummer des zurückgesetzten Fehlers	B	Neu					
SD51	Batteriespannung zu niedrig (Latch-Merker)	Bit-Muster, das anzeigt, wo die Batteriespannung abgefallen ist	Die entsprechenden Bits werden gesetzt, wenn die Batteriespannung sinkt. Dieses Bit bleibt gesetzt, auch wenn die Batteriespannung wieder ihren Normalwert erreicht hat. 	S (Fehler)	Neu					
SD52	Batteriespannung niedrig	Bit-Muster, das anzeigt, wo die Batteriespannung abgefallen ist	Funktionweise wie in SD51 beschrieben (siehe oben) Dieses Bit wird rückgesetzt, nachdem die Batterie ihren Normalwert erreicht hat.	S (Fehler)	Neu					
SD53	Abfall der Versorgungsspannung	Häufigkeit der Spannungsabfälle	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.	S (Fehler)	D9005	● Rem				

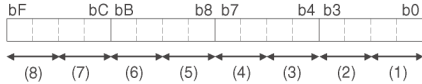
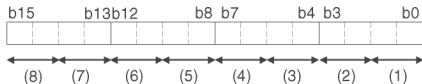

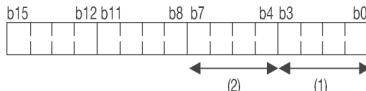
Adresse	Name	Bedeutung	Beschreibung	Gesetzt von (wenn gesetzt)	A-CPU-Register D9 [] [] []	Gültig für:
SD54	MINI-Link-Fehler	Status der Fehlerkennung	<p>(1) Das relevante Stations-Bit wird gesetzt, wenn eine der Kopfadressen eines installierten MINI (-S3)-Modules $X(n+0)/X(n+20)$, $X(n+6)/(n+26)$, $X(n+7)/(n+27)$ oder $X(n+8)/X(n+28)$ gesetzt wird.</p> <p>(2) Das relevante Bit wird gesetzt, wenn die Kommunikation zwischen den installierten MINI (-S3)-Modulen und der CPU nicht möglich ist.</p>	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 (Anweisungsausführung)	D9009	●
SD63	Anzahl der Fehlermerker		Speichert die Anzahl der Fehlermerker.	S (Anweisungsausführung)	D9124	

Adresse	Name	Bedeutung	Beschreibung	Gesetzt von (wenn gesetzt)	A-CPU-Register D9 [][][]	Gültig für:
SD64	Tabelle der erkannten Fehlermerker-nummer	Nummer der erkannten Fehlermerker	<p>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.</p> <p>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.</p> <p>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.</p> <pre> SET SET SET SET SET SET SET SET SET SET SET SET SET SET SET SET F50 F25 F19 F25 F15 F70 F65 F38 F110F151F210 LEDR ↳ SD62 0 5099 SD63 0 1 2 3 2 3 4 5 6 7 8 9 8 SD64 0 5099 SD65 0 0 0 25 99 99 99 99 99 99 99 99 99 99 99 15 SD66 0 0 0 99 0 15 15 15 15 15 15 15 15 15 15 15 SD67 0 0 0 0 0 0 0 70 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 0 38 38 38 38 110 SD70 0 0 0 0 0 0 0 0 0 0 0 110 110 110 151 SD71 0 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 0 0 210 0 SD73 0 0 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 0 0 SD75 0 0 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 0 0 SD77 0 0 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 0 0 SD79 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 </pre>	S (Anweisungsausführung)	D9125	●
SD65					D9126	
SD66					D9127	
SD67					D9128	
SD68					D9129	
SD69					D9130	
SD70					D9131	
SD71					D9132	
SD72					Neu	
SD74					Neu	
SD75					Neu	
SD76					Neu	
SD77					Neu	
SD78					Neu	
SD79					Neu	
SD80					Fehlercode der CHK-Anweisung	

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	<p>Fehlermerkernummer, die gesetzt wird, wenn eine falsche WDT-Schritt-Überwachungszeit eingegeben oder ein WDT-Zeitüberschreitungsfehler ansteht.</p> <div style="text-align: center;"> </div> <p>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.</p>	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			Neu	
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.	S (Beim Einschalten der Versorgungsspannung oder nach einem Reset)		Neu	Q00J-Q00- und Q01CPU
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

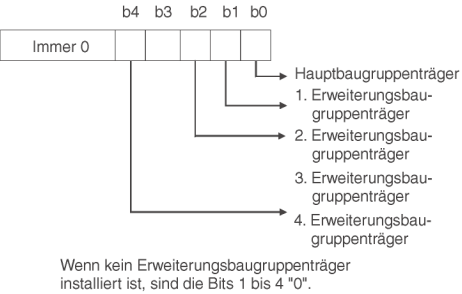
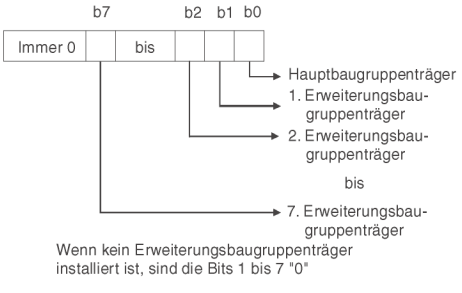
Adresse	Name	Bedeutung	Beschreibung	Gesetzt von (wenn gesetzt)	ACPU-Register D9 [] [] []	Gültig für:
SD130	Module mit defekter Sicherung	Das Bit-Muster (16 Bit) zeigt die Module mit defekter Sicherung an 0 : Keine defekte Sicherung 1 : Defekte Sicherung vorhanden	Die Anzahl der Ausgangsmodule mit defekter Sicherung wird als Bit-Muster von 16 Bit gespeichert. (Wenn die Modulnummer in den Parametern gesetzt ist, wird diese Nummer gespeichert.)	S (Fehler)	Neu	Q00J-, Q00- und Q01CPU
SD131			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.			
SD132						
SD133						
SD134						
SD135						
SD136						
SD137	<p>b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0</p> <p>SD130 0 0 0 1 (Y1C0) 0 0 0 1 (Y1B0) 0 0 0 0 0 0 0 0</p> <p>SD131 1 (Y1F0) 0 0 0 0 0 1 (Y1A) 0 0 0 0 0 0 0 0 0 0</p> <p>SD137 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 (Y1F30) 0 0 0 0</p> <p>Defekte Sicherung beim Modul mit der Adresse Y1F80.</p>					
SD150	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.)	S (Fehler)	Neu	Q00J-, Q00- und Q01CPU
SD151			Es werden auch E/A-Modulinformationen erkannt			
SD152						
SD153						
SD154						
SD155						
SD156						
SD157	<p>b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0</p> <p>SD150 0 0 0 0 0 0 0 0 1 (XY80) 0 0 0 0 0 0 0 0 1 (XY0)</p> <p>SD151 0 0 0 0 0 0 0 0 1 (XY190) 0 0 0 0 0 0 0 0 0 0</p> <p>SD157 0 0 0 0 0 1 (XYFB0) 0 0 0 0 0 0 0 0 0 0 0 0</p> <p>Anzeige des E/A-Moduls mit Vergleichsfehler</p>					

Adresse	Name	Bedeutung	Beschreibung	Gesetzt von (wenn gesetzt)	A-CPU-Register D9 [] [] []	Gültig für:
SD200	Schaltzustand	Zustand des CPU-Betriebsartenschalters	<p>Der Zustand des Betriebsartenschalters wird im folgenden Format gespeichert:</p> <p>(1) Betriebsart Immer 1: STOP</p>	S (Kontinuierlich)	Neu	Remote
			<p>Der Zustand des Betriebsartenschalters wird im folgenden Format gespeichert:</p> <p>(1) Betriebsart (0): RUN (1): STOP</p> <p>(2) Speicherkartenschal- ter Immer AUS</p>		Neu	Q00J-, Q00- und Q01CPU
			<p>Der Zustand des Betriebsartenschalters wird im folgenden Format gespeichert:</p> <p>(1) Betriebsart (0): RUN (1): STOP (2): L.CLR</p> <p>(2) Speicherkartenschal- ter b4: Speicherkarte A b5: Speicherkarte B 0: AUS, 1: EIN</p> <p>(3) DIP-Schalter Die Bits b8 bis b11 ent- sprechen den Schaltern SW1 bis SW5. 0: AUS, 1: EIN</p>	S (END-Verarbeitung)	Neu	Q-CPU außer Q00J- Q00- und Q01CPU
			<p>Der Zustand des CPU-Schlüsselschalters wird im folgenden Format gespeichert</p> <p>(1) Betriebsart (0): RUN (1): STOP (2): L.CLR</p> <p>(2) Speicherkartenschal- ter b4: Speicherkarte A b5: Speicherkarte B 0: AUS, 1: EIN</p> <p>(3) DIP-Schalter Die Bits b8 bis b11 ent- sprechen den Schaltern SW1 bis SW5. 0: AUS, 1: EIN</p>	S (END-Verarbeitung)	Neu	QnA-CPU

Adresse	Name	Bedeutung	Beschreibung	Gesetzt von (wenn gesetzt)	A-CPU-Register D9 [] [] [] []	Gültig für:
SD201	LED-Zustand	Zustand der CPU-LED-Anzeige	<p>Die unten aufgeführten Informationen beziehen sich auf die LED-Anzeigen der CPU.</p>  <p>(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</p> <p>Bei einer Q00J-, Q00- oder Q01CPU stehen nur die Bereiche 1 und 2 zur Verfügung.</p>	S (Zustandsänderung)	Neu	Q-CPU
			<p>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</p>  <p>(1) : RUN (5) : BOOT (2) : ERROR (6) : Card A (Speicherkarte A) (3) : USER (7) : Card B (Speicherkarte B) (4) : BAT.ALARM (8) : Frei</p>	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		<p>Der Verarbeitungszustand wird wie unten angegeben gespeichert</p>  <p>(1) Verarbeitungszustand von dezentralen E/A-Modulen Immer 2: STOP</p>	S (Kontinuierlich)	Neu	Remote
			<p>Der CPU-Verarbeitungsstatus wird wie unten angegeben gespeichert.</p>  <div style="border: 1px solid black; padding: 5px;"> <p>(1) : Verarbeitungs-zustand der CPU 0: RUN 1: STEP-RUN (nicht bei Q00J-, Q00- und Q01CPU) 2: STOP 3: PAUSE</p> <p>(2) : STOP/PAUSE verursacht durch 0: Betriebsartenschalter 1: Remote-Kontakt 2: Peripheriegerät, Computerverbindung oder andere Remote-Quellen 3: interne Programmanweisungen 4: Fehler</p> <p>Hinweis: Es wird nur der zuerst aufgetretene Fehler angezeigt.</p> </div>	S (END-Verarbeitung)	D9015 (Geändertes Format)	●

Adresse	Name	Bedeutung	Beschreibung	Gesetzt von (wenn gesetzt)	A-CPU-Register D9 [] [] []	Gültig für:																				
SD206	Ausführungsart des Operandentest	Anzeige des ausgeführten Operandentest	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)	S (Anforderung)	Neu	Remote																				
SD207	Anzeigepriorität der ERR-LED	Priorität 1 bis 4	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. <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>B15</td> <td>B12 B11</td> <td>B8 B7</td> <td>B4 B3</td> <td>B0</td> </tr> <tr> <td>SD207</td> <td>Priorität 4</td> <td>Priorität 3</td> <td>Priorität 2</td> <td>Priorität 1</td> </tr> <tr> <td>SD208</td> <td>Priorität 8</td> <td>Priorität 7</td> <td>Priorität 6</td> <td>Priorität 5</td> </tr> <tr> <td>SD209</td> <td colspan="2" style="border: none;"></td> <td>Priorität 10</td> <td>Priorität 9</td> </tr> </table> Werkseinstellung: (4321 _H) (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.	B15	B12 B11	B8 B7	B4 B3	B0	SD207	Priorität 4	Priorität 3	Priorität 2	Priorität 1	SD208	Priorität 8	Priorität 7	Priorität 6	Priorität 5	SD209			Priorität 10	Priorität 9	B	D9038	● außer Q00J-, Q00- und Q01CPU
B15		B12 B11		B8 B7	B4 B3	B0																				
SD207		Priorität 4		Priorität 3	Priorität 2	Priorität 1																				
SD208	Priorität 8	Priorität 7	Priorität 6	Priorität 5																						
SD209			Priorität 10	Priorität 9																						
SD208	Priorität 5 bis 8	D9039 (Geändertes Format)																								
SD209	Priorität 9 bis 10	Neu																								
SD210	Uhr-Daten	Uhr-Daten (Jahr, Monat)	Das Jahr (letzten 2 Stellen) und der Monat werden im BCD-Code in Register SD210 gespeichert: <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>b15</td> <td>b12 b11</td> <td>b8 b7</td> <td>b4 b3</td> <td>b0</td> </tr> <tr> <td colspan="4" style="border: none;">Jahr</td> <td style="border: none;">Monat</td> </tr> </table> Beispiel: Juli 1993 = 9307	b15	b12 b11	b8 b7	b4 b3	b0	Jahr				Monat	S/B (Anforderung)	D9025	● Rem										
b15		b12 b11	b8 b7	b4 b3	b0																					
Jahr				Monat																						
SD211	Uhr-Daten (Tag, Stunde)	Der Tag und die Stunden werden im BCD-Code in Register SD211 gespeichert: <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>b15</td> <td>b12 b11</td> <td>b8 b7</td> <td>b4 b3</td> <td>b0</td> </tr> <tr> <td colspan="4" style="border: none;">Tag</td> <td style="border: none;">Stunde</td> </tr> </table> Beispiel: 31., 10 Uhr = 3110	b15	b12 b11	b8 b7	b4 b3	b0	Tag				Stunde	D9026													
b15	b12 b11	b8 b7	b4 b3	b0																						
Tag				Stunde																						
SD212	Uhr-Daten (Minute, Sekunde)	Die Minuten und die Sekunden werden in Register SD212 BCD-codiert gespeichert. <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>b15</td> <td>b12 b11</td> <td>b8 b7</td> <td>b4 b3</td> <td>b0</td> </tr> <tr> <td colspan="4" style="border: none;">Minute</td> <td style="border: none;">Sekunde</td> </tr> </table> Beispiel: 35 min, 48s = 3548	b15	b12 b11	b8 b7	b4 b3	b0	Minute				Sekunde	D9027													
b15	b12 b11	b8 b7	b4 b3	b0																						
Minute				Sekunde																						

Adresse	Name	Bedeutung	Beschreibung	Gesetzt von (wenn gesetzt)	A-CPU-Register D9 [] [] []	Gültig für:																								
SD213	Uhr-Daten	Uhr-Daten (Wochentag)	<p>Der Wochentag wird im BCD-Code in Register SD213 gespeichert.</p> <table border="1"> <tr><th>Wochentag</th></tr> <tr><td>0</td><td>Sonntag</td></tr> <tr><td>1</td><td>Montag</td></tr> <tr><td>2</td><td>Dienstag</td></tr> <tr><td>3</td><td>Mittwoch</td></tr> <tr><td>4</td><td>Donnerstag</td></tr> <tr><td>5</td><td>Freitag</td></tr> <tr><td>6</td><td>Samstag</td></tr> </table>	Wochentag	0	Sonntag	1	Montag	2	Dienstag	3	Mittwoch	4	Donnerstag	5	Freitag	6	Samstag	S/B (Anforderung)	D9028	Q-CPU Rem									
			Wochentag																											
0	Sonntag																													
1	Montag																													
2	Dienstag																													
3	Mittwoch																													
4	Donnerstag																													
5	Freitag																													
6	Samstag																													
<p>Der Wochentag wird im BCD-Code in Register SD213 gespeichert.</p> <table border="1"> <tr><th>Wochentag</th></tr> <tr><td>0</td><td>Sonntag</td></tr> <tr><td>1</td><td>Montag</td></tr> <tr><td>2</td><td>Dienstag</td></tr> <tr><td>3</td><td>Mittwoch</td></tr> <tr><td>4</td><td>Donnerstag</td></tr> <tr><td>5</td><td>Freitag</td></tr> <tr><td>6</td><td>Samstag</td></tr> </table>	Wochentag	0	Sonntag	1	Montag	2	Dienstag	3	Mittwoch	4	Donnerstag	5	Freitag	6	Samstag	S/B (Anforderung)	QnA-CPU													
Wochentag																														
0	Sonntag																													
1	Montag																													
2	Dienstag																													
3	Mittwoch																													
4	Donnerstag																													
5	Freitag																													
6	Samstag																													
SD220	Daten des LED-Displays	Anzeigedaten des Displays	<p>Die ASCII-Daten (16 Zeichen) des LED-Displays werden in den unten aufgeführten Registern gespeichert.</p> <table border="1"> <tr> <td>SD220</td> <td>15. Zeichen von rechts</td> <td>16. Zeichen von rechts</td> </tr> <tr> <td>SD221</td> <td>13. Zeichen von rechts</td> <td>14. Zeichen von rechts</td> </tr> <tr> <td>SD222</td> <td>11. Zeichen von rechts</td> <td>12. Zeichen von rechts</td> </tr> <tr> <td>SD223</td> <td>9. Zeichen von rechts</td> <td>10. Zeichen von rechts</td> </tr> <tr> <td>SD224</td> <td>7. Zeichen von rechts</td> <td>8. Zeichen von rechts</td> </tr> <tr> <td>SD225</td> <td>5. Zeichen von rechts</td> <td>6. Zeichen von rechts</td> </tr> <tr> <td>SD226</td> <td>3. Zeichen von rechts</td> <td>4. Zeichen von rechts</td> </tr> <tr> <td>SD227</td> <td>1. Zeichen von rechts</td> <td>2. Zeichen von rechts</td> </tr> </table>	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	S (Zustandsänderung)	Neu	●
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																												
SD240	Betriebsart des Baugruppenträgers	0: Automatischer Betrieb 1: Detaillierter Betrieb	Dieses Register dient zur Speicherung der Betriebsart des Baugruppenträgers.	S (Initialisierung)	Neu	Q-CPU Rem																								
SD241	Anzahl der Erweiterungsbaugruppenträger	0: Nur Hauptbaugruppenträger 1 bis 7: Anzahl der Erweiterungsbaugruppenträger	In diesem Register wird die Anzahl der installierten Erweiterungsbaugruppenträger gespeichert.	S (Initialisierung)	Neu																									

Adresse	Name	Bedeutung	Beschreibung	Gesetzt von (Wenn gesetzt)	A-CPU-Register D9 [] [] []	Gültig für:										
SD242	Unterscheidung zwischen A- und Q-Baugruppenträger	0: Baugruppenträger vom Typ QA [] [] B ist installiert (Betriebsart A) 1: Baugruppenträger vom Typ Q [] [] B ist installiert (Betriebsart Q)		S (Initialisierung)	Neu	Q00J-, Q00- und Q01CPU										
SD243	Anzahl der Steckplätze auf den Baugruppenträgern	Anzahl der Steckplätze auf den Baugruppenträgern Bei einer Q00J-, Q00- oder Q01CPU sind die Positionen für den 5. bis 7. EBT mit Null belegt.		S (Initialisierung)	Neu	Q02-Q06H-Q12H-Q25H-CPU; Rem										
SD244			<table border="1" data-bbox="582 1041 1045 1131"> <thead> <tr> <th>bF</th> <th>bC bB</th> <th>b8 b7</th> <th>b4 b3</th> <th>b0</th> </tr> </thead> <tbody> <tr> <td>SM243</td> <td>3. EBT</td> <td>2. EBT</td> <td>1. EBT</td> <td>HBT</td> </tr> <tr> <td>SM244</td> <td>7. EBT</td> <td>6. EBT</td> <td>5. EBT</td> <td>4. EBT</td> </tr> </tbody> </table> <p>Die Anzahl der Steckplätze wird für den Hauptbaugruppen (HBT) und die Erweiterungsbaugruppenträger (EBT) in den entsprechenden Bereichen abgelegt.</p>				bF	bC bB	b8 b7	b4 b3	b0	SM243	3. EBT	2. EBT	1. EBT	HBT
bF	bC bB	b8 b7	b4 b3	b0												
SM243	3. EBT	2. EBT	1. EBT	HBT												
SM244	7. EBT	6. EBT	5. EBT	4. EBT												
SD250	Maximum an E/As geladen	Maximale Anzahl an E/As geladen	Wird SM250 gesetzt, werden zu den oberen beiden Stellen der letzten geladenen E/A-Modul-Adresse der Wert 1 addiert und als Binärwert gespeichert.	S (END-Verarbeitung)	Neu	●										
SD251	Adresse eines auszuwechselnden E/A-Moduls	Startadresse des E/A-Moduls	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.	B	D9094	Q2A(S1)-, Q3A-Q4A-Q4AR-CPU										
SD253	Übertragungsgeschwindigkeit für RS422	0: 9600 Bit/s 1: 19,2 kBit/s 2: 38,4 kBit/s	Das Register speichert den Wert für die Übertragungsgeschwindigkeit der RS422-Schnittstelle.	S (Bei Änderung)	Neu	QnA-CPU										

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		E/A-Adresse	E/A-Adresse des ersten installierten Moduls im MELSECNET/10.			
SD256		Netzwerknummer	Netzwerkadresse des ersten installierten Moduls im MELSECNET/10.			
SD257		Gruppennummer	Gruppennummer des ersten installierten Moduls im MELSECNET/10.			
SD258		Stationsnummer	Stationsnummer des ersten installierten Moduls im MELSECNET/10.			
SD259		Standby-Information	Sind Standby-Stationen vorhanden, wird die Modulnummer der Standby-Station gespeichert (1 bis 4).			
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	<p>(1) Wenn Xn0 des installierten CC-Link eingeschaltet wird, wird das der Station zugeordnete Bit gesetzt.</p> <p>(2) Wenn entweder Xn1 oder XnF des installierten CC-Link ausgeschaltet wird, wird das der Station zugeordnete Bit gesetzt.</p> <p>(3) Die Bits in diesem Bereich werden gesetzt, wenn die CPU nicht mit dem installierten CC-Link kommunizieren kann.</p>	S (Bei Fehler)	Neu	Q-CPU Rem
			<p>(1) Wenn Xn0 des installierten CC-Link eingeschaltet wird, wird das der Station zugeordnete Bit gesetzt.</p> <p>(2) Wenn entweder Xn1 oder XnF des installierten CC-Link ausgeschaltet wird, wird das der Station zugeordnete Bit gesetzt.</p> <p>(3) Die Bits in diesem Bereich werden gesetzt, wenn die CPU nicht mit dem installierten CC-Link kommunizieren kann.</p>			

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						
SD294		Anzahl der Adressen des Operanden B	Aktuell eingestellte Anzahl der Adressen der Operanden B						
SD295		Anzahl der Adressen des Operanden F	Aktuell eingestellte Anzahl der Adressen der Operanden F						
SD296		Anzahl der Adressen des Operanden SB	Aktuell eingestellte Anzahl der Adressen der Operanden SB						
SD297		Anzahl der Adressen des Operanden V	Aktuell eingestellte Anzahl der Adressen der Operanden V						
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 Kommunikation 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			

System-Takte / 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 Programmzyklen niedriger Verarbeitungsgeschwindigkeit	Zählt die Anzahl der Programmzyklen niedriger Verarbeitungsgeschwindigkeit	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

A.3.1 Programmzyklusinformationen

Adresse	Name	Bedeutung	Beschreibung	Gesetzt von (wenn gesetzt)	A-CPU-Register D9 [] [] []	Gültig für:
SD500	Programmnummer 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	Programmnummer 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- s-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 s)			Neu	
SD522	Zeit des Initialisierungszyklus	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 s)	Speichert die Zeit des ersten Programmzyklus (in 1- s-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 100 s)	Speichert die minimale Programmzykluszeit (in 100- s-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 s)	Speichert die maximale Programmzykluszeit (in 100- s-Schritten) mit Ausnahme des ersten Zyklus. Bereich von 000 bis 900		Neu	
SD528	Zykluszeit für Programme des Ausführungsmodus „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 s)	Speichert die aktuelle Zykluszeit des Programms vom Typ „Low Speed Execution“ (in 100- s-Schritten). Bereich von 000 bis 900			
SD532	Minimale Zykluszeit für Programme des Ausführungsmodus „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	● außer Q00J-, Q00- und Q01CPU
SD533		Minimale Zykluszeit (Einheit 100 s)	Speichert die minimale Programmzykluszeit des Programms vom Typ „Low Speed Execution“ (in 100- s-Schritten). Bereich von 000 bis 900			
SD534	Maximale Zykluszeit für Programme des Ausführungsmodus „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	● außer Q00J-, Q00- und Q01CPU
SD535		Maximale Zykluszeit (Einheit 100 s)	Speichert die maximale Programmzykluszeit des Programms vom Typ „Low Speed Execution“ (in 100- s-Schritten) mit Ausnahme des 1. Zyklus. Bereich von 000 bis 900			

Programmzyklusinformationen (Fortsetzung)

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 s)	Speichert die Zeit vom Ende des letzten Programmzyklus bis zum Anfang des nächsten Zyklus (in 100- s-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 s)	Speichert die Wartezeit bei gesetzter konstanter Zykluszeit (in 100- s-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 s)	Speichert die kumulierte Ausführungszeit des Programms vom Typ „Low Speed Execution“ (in 100- s-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 s)	Speichert die Ausführungszeit des Programms vom Typ „Low Speed Execution“ (in 100- s-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 s)	Speichert die Ausführungszeit des Programms vom Typ „Scan Execution“ (in 100 s 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 s)	Ist SM551 gesetzt, wird das Intervall gespeichert, nach dem das in SD550 benannte Modul gewartet wird (in 1- s-Schritten). Bereich von 000 bis 900			

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	● außer Q00J-, Q00- und Q01CPU																
SD604	Nutzungsbedingungen der Speicherkarte A		Die Nutzungsbedingungen der Speicherkarte A werden als Bit-Muster gespeichert (EIN wenn in Benutzung). Bedeutung dieses Bit-Musters: <table border="1" style="width: 100%;"> <tr> <td>b0 : BOOT-Vorgang (QBT)</td> <td>b8 :</td> </tr> <tr> <td>b1 : Parameter (QPA)</td> <td>b9 : CPU- Fehlerprotokoll (QFD)</td> </tr> <tr> <td>b2 : Operandenkommentare (QCD)</td> <td>bA : Trace der Ablaufsprache (QTS)</td> </tr> <tr> <td>b3 : Startwert von Operanden (QDI)</td> <td>bB : Lokale Variable (QDL)</td> </tr> <tr> <td>b4 : File-R (QDR)</td> <td>bC :</td> </tr> <tr> <td>b5 : Trace (QTS)</td> <td>bD :</td> </tr> <tr> <td>b6 :</td> <td>bE :</td> </tr> <tr> <td>b7 :</td> <td>bF :</td> </tr> </table>	b0 : BOOT-Vorgang (QBT)	b8 :	b1 : Parameter (QPA)	b9 : CPU- Fehlerprotokoll (QFD)	b2 : Operandenkommentare (QCD)	bA : Trace der Ablaufsprache (QTS)	b3 : Startwert von Operanden (QDI)	bB : Lokale Variable (QDL)	b4 : File-R (QDR)	bC :	b5 : Trace (QTS)	bD :	b6 :	bE :	b7 :	bF :	S (Zustandsänderung)	Neu	Q-CPU außer Q00J-, Q00- und Q01CPU
			b0 : BOOT-Vorgang (QBT)	b8 :																		
b1 : Parameter (QPA)	b9 : CPU- Fehlerprotokoll (QFD)																					
b2 : Operandenkommentare (QCD)	bA : Trace der Ablaufsprache (QTS)																					
b3 : Startwert von Operanden (QDI)	bB : Lokale Variable (QDL)																					
b4 : File-R (QDR)	bC :																					
b5 : Trace (QTS)	bD :																					
b6 :	bE :																					
b7 :	bF :																					
Die Nutzungsbedingungen der Speicherkarte A werden als Bit-Muster gespeichert (EIN wenn in Benutzung). Bedeutung dieses Bit-Musters: <table border="1" style="width: 100%;"> <tr> <td>b0 : BOOT-Vorgang (QBT)</td> <td>b8 : Simulationsdaten</td> </tr> <tr> <td>b1 : Parameter (QPA)</td> <td>b9 : CPU- Fehlerprotokoll (QFD)</td> </tr> <tr> <td>b2 : Operandenkommentare (QCD)</td> <td>b10 : Trace der Ablaufsprache (QTS)</td> </tr> <tr> <td>b3 : Startwert von Operanden (QDI)</td> <td>b11 : Lokale Variable (QDL)</td> </tr> <tr> <td>b4 : File-R (QDR)</td> <td>b12 :</td> </tr> <tr> <td>b5 : Trace (QTS)</td> <td>b13 :</td> </tr> <tr> <td>b6 : Status-Latch (QTL)</td> <td>b14 :</td> </tr> <tr> <td>b7 : Programm-Trace (QTP)</td> <td>b15 :</td> </tr> </table>	b0 : BOOT-Vorgang (QBT)	b8 : Simulationsdaten	b1 : Parameter (QPA)	b9 : CPU- Fehlerprotokoll (QFD)	b2 : Operandenkommentare (QCD)	b10 : Trace der Ablaufsprache (QTS)	b3 : Startwert von Operanden (QDI)	b11 : Lokale Variable (QDL)	b4 : File-R (QDR)	b12 :	b5 : Trace (QTS)	b13 :	b6 : Status-Latch (QTL)	b14 :	b7 : Programm-Trace (QTP)	b15 :	S (Zustandsänderung)	Neu	QnA-CPU			
b0 : BOOT-Vorgang (QBT)	b8 : Simulationsdaten																					
b1 : Parameter (QPA)	b9 : CPU- Fehlerprotokoll (QFD)																					
b2 : Operandenkommentare (QCD)	b10 : Trace der Ablaufsprache (QTS)																					
b3 : Startwert von Operanden (QDI)	b11 : Lokale Variable (QDL)																					
b4 : File-R (QDR)	b12 :																					
b5 : Trace (QTS)	b13 :																					
b6 : Status-Latch (QTL)	b14 :																					
b7 : Programm-Trace (QTP)	b15 :																					

Speicherkarten

Adresse	Name	Bedeutung	Beschreibung	Gesetzt von (wenn gesetzt)	A-CPU-Register D9 [] [] []	Gültig für:																		
SD620	Typ der Speicherkarte B		<p>Zeigt den Speichertyp der installierten Speicherkarte B an.</p> <p>Der Wert für Laufwerk 4 ist wegen des eingebauten Flash-ROMs fest auf „3“ eingestellt.</p>	S (Initialisierung)	Neu	Q-CPU																		
			<p>Zeigt den Speichertyp der installierten Speicherkarte B an.</p>	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		<p>Die Nutzungsbedingungen der Laufwerke 3 und 4 werden als Bit-Muster gespeichert (EIN wenn in Benutzung). Die Bedeutung dieses Bit-Musters wird unten beschrieben.</p> <table border="1"> <tr> <td>b0: BOOT-Vorgang (QBT)</td> <td>b8 :</td> </tr> <tr> <td>b1: Parameter (QPA)</td> <td>b9 : CPU- Fehlerprotokoll (QFD)</td> </tr> <tr> <td>b2: Operandenkommentare (QCD)</td> <td>bA : Trace der</td> </tr> <tr> <td>b3: Startwert von Operanden (QDI)</td> <td>Ablaufsprache (QTS)</td> </tr> <tr> <td>b4: File-R (QDR)</td> <td>bB : Lokale Variable (QDL)</td> </tr> <tr> <td>b5: Trace (QTS)</td> <td>bC :</td> </tr> <tr> <td>b6 :</td> <td>bD :</td> </tr> <tr> <td>b7 :</td> <td>bE :</td> </tr> <tr> <td></td> <td>bF :</td> </tr> </table>	b0: BOOT-Vorgang (QBT)	b8 :	b1: Parameter (QPA)	b9 : CPU- Fehlerprotokoll (QFD)	b2: Operandenkommentare (QCD)	bA : Trace der	b3: Startwert von Operanden (QDI)	Ablaufsprache (QTS)	b4: File-R (QDR)	bB : Lokale Variable (QDL)	b5: Trace (QTS)	bC :	b6 :	bD :	b7 :	bE :		bF :	S (Zustandsänderung)	Neu	Q-CPU außer Q00J-, Q00- und Q01CPU
	b0: BOOT-Vorgang (QBT)	b8 :																						
b1: Parameter (QPA)	b9 : CPU- Fehlerprotokoll (QFD)																							
b2: Operandenkommentare (QCD)	bA : Trace der																							
b3: Startwert von Operanden (QDI)	Ablaufsprache (QTS)																							
b4: File-R (QDR)	bB : Lokale Variable (QDL)																							
b5: Trace (QTS)	bC :																							
b6 :	bD :																							
b7 :	bE :																							
	bF :																							
Nutzungsbedingungen der Speicherkarte B		<p>Die Nutzungsbedingungen der Speicherkarte B werden als Bit-Muster gespeichert (EIN wenn in Benutzung). Die Bedeutung dieses Bit-Musters wird unten beschrieben.</p> <table border="1"> <tr> <td>b0: BOOT-Vorgang (QBT)</td> <td>b8 : Simulationsdaten</td> </tr> <tr> <td>b1: Parameter (QPA)</td> <td>b9 : CPU- Fehlerprotokoll (QFD)</td> </tr> <tr> <td>b2: Operandenkommentare (QCD)</td> <td>b10: Trace der Ablauf-</td> </tr> <tr> <td>b3: Startwert von Operanden (QDI)</td> <td>sprache (QTS)</td> </tr> <tr> <td>b4: File-R (QDR)</td> <td>b11: Lokale Variable (QDL)</td> </tr> <tr> <td>b5: Trace (QTS)</td> <td>b12:</td> </tr> <tr> <td>b6: Status-Latch (QTL)</td> <td>b13:</td> </tr> <tr> <td>b7: Programm-Trace (QTP)</td> <td>b14:</td> </tr> <tr> <td></td> <td>b15:</td> </tr> </table>	b0: BOOT-Vorgang (QBT)	b8 : Simulationsdaten	b1: Parameter (QPA)	b9 : CPU- Fehlerprotokoll (QFD)	b2: Operandenkommentare (QCD)	b10: Trace der Ablauf-	b3: Startwert von Operanden (QDI)	sprache (QTS)	b4: File-R (QDR)	b11: Lokale Variable (QDL)	b5: Trace (QTS)	b12:	b6: Status-Latch (QTL)	b13:	b7: Programm-Trace (QTP)	b14:		b15:	S (Zustandsänderung)	Neu	Q2A(S1)-Q3A-, Q4A-, Q4AR-CPU	
b0: BOOT-Vorgang (QBT)	b8 : Simulationsdaten																							
b1: Parameter (QPA)	b9 : CPU- Fehlerprotokoll (QFD)																							
b2: Operandenkommentare (QCD)	b10: Trace der Ablauf-																							
b3: Startwert von Operanden (QDI)	sprache (QTS)																							
b4: File-R (QDR)	b11: Lokale Variable (QDL)																							
b5: Trace (QTS)	b12:																							
b6: Status-Latch (QTL)	b13:																							
b7: Programm-Trace (QTP)	b14:																							
	b15:																							

File-Register

Adresse	Name	Bedeutung	Beschreibung	Gesetzt von (wenn gesetzt)	A-CPU-Register D9 [] [] []	Gültig für:																	
SD640	Laufwerk des File-Registers	Laufwerksnummer	Speichert die Nummer des Laufwerks, das vom File-Register benutzt wird.	S (Zustandsänderung)	Neu																		
SD641	File-Register File-Name		Speichert die durch Parameter oder QCDSET-Anweisung angegebenen File-Register und File-Name (mit Erweiterung) als ASCII-Code.	S (Zustandsänderung)	Neu	●																	
SD642																							
SD643																							
SD644																							
SD645																							
SD646																							
							<table border="1"> <tr> <td style="text-align: center;">b15</td> <td style="text-align: center;">b8 b7</td> <td style="text-align: center;">b0</td> </tr> <tr> <td>SD641</td> <td>2. Zeichen</td> <td>1. Zeichen</td> </tr> <tr> <td>SD642</td> <td>4. Zeichen</td> <td>3. Zeichen</td> </tr> <tr> <td>SD643</td> <td>6. Zeichen</td> <td>5. Zeichen</td> </tr> <tr> <td>SD644</td> <td>8. Zeichen</td> <td>7. Zeichen</td> </tr> <tr> <td>SD645</td> <td>1. Zeichen der Erweiterung</td> <td>2EH (.)</td> </tr> <tr> <td>SD646</td> <td>3. Zeichen der Erweiterung</td> <td>2. Zeichen der Erweiterung</td> </tr> </table>	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
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																					
SD647	Kapazität der File-Register		Datenkapazität des aktuell ausgewählten File-Registers in 1-k-Wort-Einheiten	S (Zustandsänderung)	Neu																		
SD648	Blocknummer des File-Registers		Speichert die aktuell ausgewählte File-Register-Blocknummer.	S (Zustandsänderung)	D9035																		
SD650	Kommentarlaufwerk		Speichert die durch Parameter oder QCDSET-Anweisung angegebene Laufwerksnummer des Kommentarlaufwerks.	S (Zustandsänderung)	Neu																		
SD651	Name des Kommentar-Files		Speichert den durch Parameter oder die QCDSET-Anweisung angegebenen File-Namen (mit Erweiterung) im ASCII-Code.	S (Zustandsänderung)	Neu	●																	
SD652																							
SD653																							
SD654																							
SD655																							
SD656																							
							<table border="1"> <tr> <td style="text-align: center;">b15</td> <td style="text-align: center;">b8 b7</td> <td style="text-align: center;">b0</td> </tr> <tr> <td>SD651</td> <td>2. Zeichen</td> <td>1. Zeichen</td> </tr> <tr> <td>SD652</td> <td>4. Zeichen</td> <td>3. Zeichen</td> </tr> <tr> <td>SD653</td> <td>6. Zeichen</td> <td>5. Zeichen</td> </tr> <tr> <td>SD654</td> <td>8. Zeichen</td> <td>7. Zeichen</td> </tr> <tr> <td>SD655</td> <td>1. Zeichen der Erweiterung</td> <td>2EH (.)</td> </tr> <tr> <td>SD656</td> <td>3. Zeichen der Erweiterung</td> <td>2. Zeichen der Erweiterung</td> </tr> </table>	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
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																					
SD660	Für den Boot-Vorgang benanntes File	Laufwerksnummer, auf dem sich das für den Boot-Vorgang benannte File befindet	Speichert die Nummer des Laufwerks, auf dem sich das File befindet, das für den (*.QBT) Bootvorgang benannt wurde.	S (Initialisierung)	Neu	● außer Q00J-, Q00- und Q01CPU																	
SD661		Name des Files, das für den Boot-Vorgang benannt ist		Speichert den File-Namen des Files, das für den Boot-Vorgang benannt ist (*.QBT).	S (Initialisierung)		Neu																
SD662																							
SD663																							
SD664																							
SD665																							
SD666																							
		<table border="1"> <tr> <td style="text-align: center;">b15</td> <td style="text-align: center;">b8 b7</td> <td style="text-align: center;">b0</td> </tr> <tr> <td>SD661</td> <td>2. Zeichen</td> <td>1. Zeichen</td> </tr> <tr> <td>SD662</td> <td>4. Zeichen</td> <td>3. Zeichen</td> </tr> <tr> <td>SD663</td> <td>6. Zeichen</td> <td>5. Zeichen</td> </tr> <tr> <td>SD664</td> <td>8. Zeichen</td> <td>7. Zeichen</td> </tr> <tr> <td>SD665</td> <td>1. Zeichen der Erweiterung</td> <td>2EH (.)</td> </tr> <tr> <td>SD666</td> <td>3. Zeichen der Erweiterung</td> <td>2. Zeichen der Erweiterung</td> </tr> </table>	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
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																					

Anweisungsbezogene Register

Adresse	Name	Bedeutung	Beschreibung	Gesetzt von (wenn gesetzt)	A-CPU-Register D9 [] [] []	Gültig für:							
SD705	Bit-Schema		Während der Blockverarbeitung 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							
SD706													
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	Bit-Schema der IMASK-Anweisung	Bit-Schema	Bei Verwendung der IMASK-Anweisung wird das Bit-Schema genutzt.	S (während der Ausführung)	Neu	●							
SD716													
SD717			<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">b15</td> <td style="text-align: center;">.....</td> <td style="text-align: center;">b0</td> </tr> <tr> <td>SD715</td> <td> 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0</td> <td></td> </tr> <tr> <td>SD716</td> <td> 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16</td> <td></td> </tr> <tr> <td>SD717</td> <td> 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 32</td> <td></td> </tr> </table>				b15	b0	SD715	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		SD716
b15	b0											
SD715	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0												
SD716	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16												
SD717	47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 32												
SD718	Akkumulator		Diese Register emulieren die Akkumulatoren der MELSEC A-Serie.	S/B	Neu								
SD719													
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							

Index

A

Ablaufsprache	
Abschluss-Schritt	15 - 2
Initialisierungs-Schritt	15 - 2
Transition	15 - 2
Übersicht	3 - 14
Analog-Ausgangsmodule	2 - 38
Analog-Eingangsmodule	2 - 38
Anweisungsliste	3 - 11
Anzeigemodus (bei Operanden editieren)	9 - 3
Array	
Programmierung	12 - 1
Übersicht	3 - 15
Ausgangsmodule	
Relais	2 - 31
Transistor (minusschaltend)	2 - 36
Transistor (plusschaltend)	2 - 34
Transistor-Ausgangsmodule	2 - 34
Triac-Ausgangsmodule	2 - 32
Übersicht	2 - 30
Auto Connect	4 - 20

B

Baugruppenträger	2 - 6
Betriebseinstellungen (ETHERNET)	18 - 4

C

CC-Link-Module	2 - 42
Counter	
Operandenadressen	3 - 20
programmieren	4 - 26
CPU-Module	
Anzahl der Operanden	2 - 14
Batterie	2 - 19
LEDs	2 - 15
RUN/STOP-Schalter	2 - 17
Speicherkarten	2 - 19
Systemschalter	2 - 17
Technische Daten	2 - 13

D

Datentypen	3 - 15
Debug (Menü)	
Operanden editieren	9 - 1
Registrierung/Abbruch erzwungener Ein-/Ausgänge	8 - 1
SPS-Diagnosen	4 - 52
DeviceNet-Modul	2 - 43
Diagnosemerker	
Fehlerdiagnose	A - 2
Systeminformationen	A - 3
Systemtakte	A - 5
Vergleichstabelle	A - 7
Diagnoseregister	
anweisungsbezogene	A - 36
Fehlerdiagnose	A - 14
File-Register	A - 35
integrierte Uhr der CPU	A - 25
Programmzyklusinformationen	A - 31
Speicherkarten	A - 33
System-Takte	A - 30
Dokumentation	
Druckoptionen	4 - 53
Netzwerk-Kommentar	4 - 34
Netzwerk-Label	4 - 34

E

Eigenschaften	
einer Task	4 - 32
Eingangsmodule	
für minusschaltende Geber	2 - 27
für plusschaltende Geber	2 - 25
für Wechselspannungen	2 - 28
Sink	2 - 27
Source	2 - 25
EN-Eingang	6 - 24
ENO-Ausgang	6 - 24
Entry-Data-Monitor	7 - 1
Ergebnistyp	
einer Funktion	6 - 12
ETHERNET	
Einstellungen	18 - 3
-Modul	2 - 41

Extras (Menü)

- Optionen (Monitoren zeigen) 4 - 48
- Optionen (Monitormodus) 4 - 47
- Optionen (Querverweise) 4 - 49

F

Funktion

- Ergebnistyp 6 - 12
- kopieren 6 - 11
- neu anlegen 6 - 2
- programmieren 6 - 2
- Vergleich mit Funktionsbaustein 6 - 1

Funktionsbaustein

- Ausführungsarten 6 - 23
- im Programm aufrufen 4 - 26
- Instanzen 4 - 26
- Instanzen monitoren 7 - 13
- Instanzen-Namen zuweisen 6 - 17
- programmieren 6 - 15
- Variablen zuweisen 6 - 19
- Vergleich mit Funktion 6 - 1

Funktionsbausteinsprache 3 - 13

G

Geführter Editor 4 - 37

Gleitkommazahl

- als Ergebnistyp einer Funktion 6 - 12
- Übersicht 3 - 15

Globale Variablen

- Definition 3 - 6
- deklarieren 4 - 9
- Liste 3 - 6
- prüfen 4 - 12

Globale Variablenliste

- Angaben zu Variablen 4 - 10
- erweitern 4 - 27

GVL

- siehe Globale Variablenliste

H

Header-Monitor 7 - 8

High-Speed-Zählermodule 2 - 39

I

IEC61131-3 3 - 1

Instanz 6 - 17

K

Kommentar

- für Programm-Netzwerke 4 - 34
- kopieren 4 - 35
- löschen 4 - 35

Kompilieren 4 - 36

Kontaktplan

- Funktionsbaustein aufrufen 4 - 18
- Geführter Editor 4 - 37
- programmieren 4 - 14
- Übersicht 3 - 13

L

Lichtschranken 2 - 24

Lokale Variablen

- Definition 3 - 6
- Liste 3 - 6
- neu definieren 4 - 19

M

Makro-Code-Abarbeitung 6 - 24

MELSECNET-Module 2 - 42

MMI 2 - 2

Monitormodus 4 - 45

N

Näherungsschalter 2 - 24

Netzteile

- Auswahlkriterien 2 - 11
- Technische Daten 2 - 10

Netzwerk Nr. (ETHERNET-Parameter) . . . 18 - 4

Netzwerk-Kommentar 4 - 34

Netzwerk-Label 4 - 34

Netzwerkparameter 18 - 2

Netzwerk-Parameter 18 - 2

O

Online (Menü)

- Entry-Data-Monitor 7 - 1
- Header-Monitor 7 - 8
- Monitor starten 4 - 46
- Speicher formatieren 4 - 42
- Übertragungseinstellungen 4 - 39

Online-Programmänderung 10 - 4

Operanden

- kennzeichen 2 - 48

Operanden editieren (Debug-Menü) 9 - 1

P

PLCopen	3 - 1
POE	
Body	3 - 5
Definition	3 - 2
einer Task zuweisen	4 - 31
Header	4 - 13
POE-Pool	
Definition	3 - 4
Ports (Übertragungseinstellungen)	4 - 39
Positioniermodule	2 - 40
PROFIBUS-Module	2 - 43
Programm	
Ausführung beobachten	4 - 45
in SPS übertragen	4 - 43
prüfen	4 - 24
Querverweisliste	4 - 49
Projekt (Menü)	
Druckoptionen	4 - 53
Online-Programmänderung	10 - 4
Passwortvergabe	14 - 1
Querverweisliste	4 - 49
Sicherheitsstufe ändern	14 - 2
Prozessabbildverfahren	2 - 46

Q

Q64TCRT	2 - 39
Q64TCRTBW	2 - 39
Q64TCTT	2 - 39
Q64TCTTBW	2 - 39
QD51	2 - 41
QD62	2 - 39
QD75	2 - 40
QJ71BR11	2 - 42
QJ71C24	2 - 40
QJ71DN91	2 - 43
QJ71E71	2 - 41
QJ71LP21	2 - 42
QJ71PB92D	2 - 43
QJ71PB93D	2 - 43
QJ71WS96	2 - 44
Querverweisliste	4 - 49

R

Relais	
Ausgangsmodule	2 - 31

S

SCADA	2 - 2
Schütze	
Vergleich mit SPS	2 - 1
SDT	3 - 17
sieheStrukturierter Datentyp	
Sicherheitsstufe ändern	14 - 2
Sink	
Ausgang	2 - 36
Eingang	2 - 23
Source	
Ausgang	2 - 34
Eingang	2 - 23
Speicherkarten	2 - 19
Speicherprogrammierbare Steuerung	
sieheSPS	
SPS	
Diagnosefunktionen	4 - 52
geschichtliche Entwicklung	2 - 1
Systemkonfiguration	2 - 4
Vergleich mit Schützsteuerung	2 - 1
Steuerungsanweisung	3 - 11
Strukturierter Datentyp	
Anwendungsbeispiel	11 - 1
Übersicht	3 - 17
Strukturierter Text	
Programmierung	17 - 1
Übersicht	3 - 12
System-Label	3 - 10
Systemvariablen	3 - 9

T

Task	
Attribute	4 - 32
Definition	3 - 3
Eigenschaften	4 - 32
neu anlegen	4 - 30
POE zuweisen	4 - 31
Pool	3 - 7
Task-Pool	
Definition	3 - 7
Timer	
Operandenadressen	3 - 20
programmieren	4 - 28

U

- Übertragungseinstellungen
 - für Programmtransfer 4 - 39
 - für Zugriff über ETHERNET 18 - 9

V

- Variablen
 - aus dem Header der POE wählen 4 - 16
 - einer Anweisung zuweisen 4 - 19
 - Global (Definition) 3 - 6
 - siehe auch Globale Variablen
 - Lokal (Definition) 3 - 6
 - siehe auch Lokale Variablen
- Verbindungs-Einstellungen 18 - 6
- Verbindungs-Modus 4 - 20
- Verbindungstest 4 - 40

W

- Web-Server-Modul 2 - 44

Global Partner. Local Friend.

EUROPE

MITSUBISHI ELECTRIC EUROPE B.V.
German Branch
Gothaer Straße 8
D-40880 Ratingen
Phone: +49 (0) 2102 / 486-0
Fax: +49 (0) 2102 / 486-1120
e mail: megfamail@meg.mee.com

FRANCE

MITSUBISHI ELECTRIC EUROPE B.V.
French Branch
25, Boulevard des Bouvets
F-92741 Nanterre Cedex
Phone: +33 1 55 68 55 68
Fax: +33 1 55 68 56 85
email: factory.automation@fra.mee.com

IRELAND

MITSUBISHI ELECTRIC EUROPE B.V.
Irish Branch
Westgate Business Park, Ballymount
IRL-Dublin 24
Phone: +353 (0) 1 / 419 88 00
Fax: +353 (0) 1 / 419 88 90
e mail: sales.info@meir.mee.com

ITALY

MITSUBISHI ELECTRIC EUROPE B.V.
Italian Branch
Via Paracelso 12
I-20041 Agrate Brianza (MI)
Phone: +39 039 6053 1
Fax: +39 039 6053 312
e mail: factory.automation@it.mee.com

SPAIN

MITSUBISHI ELECTRIC EUROPE B.V.
Spanish Branch
Carretera de Rubí 76-80
E-08190 Sant Cugat del Vallés
Phone: +34 9 3 / 565 3160
Fax: +34 9 3 / 589 1579
e mail: industrial@sp.mee.com

UK

MITSUBISHI ELECTRIC EUROPE B.V.
UK Branch
Travellers Lane
GB-Hatfield Herts. AL10 8 XB
Phone: +44 (0) 1707 / 27 61 00
Fax: +44 (0) 1707 / 27 86 95
e mail: automation@meuk.mee.com

JAPAN

MITSUBISHI ELECTRIC CORPORATION
Office Tower "Z" 14 F
8-12, 1 chome, Harumi Chuo-Ku
Tokyo 104-6212
Phone: +81 3 6221 6060
Fax: +81 3 6221 6075

USA

MITSUBISHI ELECTRIC AUTOMATION
500 Corporate Woods Parkway
Vernon Hills, IL 60061
Phone: +1 847 / 478 21 00
Fax: +1 847 / 478 22 83

AUSTRIA

GEVA
Wiener Straße 89
AT-2500 Baden
Phone: +43 (0) 2252 / 85 55 20
Fax: +43 (0) 2252 / 488 60
e mail: office@geva.at

BELARUS

TEHNIKON
Oktjabrskaya 16/5, Ap 704
BY-220030 Minsk
Phone: +375 (0)17 / 210 4626
Fax: +375 (0)17 / 210 4626
e mail: tehnikon@belsonet.net

BELGIUM

Koning & Hartman B.V.
Researchpark Zellik, Pontbeeklaan 43
BE-1731 Brussels
Phone: +32 (0)2 / 467 17 51
Fax: +32 (0)2 / 467 17 45
e mail: info@koninghartman.com

BULGARIA

AKHNATON
Andrej Ljapchev Lbv. Pvd 21 4
BG-1756 Sofia
Phone: +359 (0) 2 / 97 44 05 8
Fax: +359 (0) 2 / 97 44 06 1
e mail: —

CZECH REPUBLIC

AutoCont
Control Systems s.r.o.
Nemocnici 12
CZ-702 00 Ostrava 2
Phone: +420 59 / 6152 111
Fax: +420 59 / 6152 562
e mail: consys@autocont.cz

DENMARK

louis poulsen
industri & automation
Geminivej 32
DK-2670 Greve
Phone: +45 (0) 70 / 10 15 35
Fax: +45 (0) 43 / 95 95 91
e mail: lpia@lpmail.com

ESTONIA

UTU Elektrotehnika AS
Pärnu mnt. 160i
EE-11317 Tallinn
Phone: +372 (0) 6 / 51 72 80
Fax: +372 (0) 6 / 51 72 88
e mail: utu@utu.ee

FINLAND

Beijer Electronics OY
Ansatie 6a
FIN-01740 Vantaa
Phone: +358 (0) 9 / 886 77 500
Fax: +358 (0) 9 / 886 77 555
e mail: info@beijer.fi

GREECE

UTECO A.B.E.E.
5, Mavrogenous Str.
GR-18542 Piraeus
Phone: +302 (0) 10 / 42 10 050
Fax: +302 (0) 10 / 42 12 033
e mail: sales@uteco.gr

HUNGARY

Meltrade Ltd.
Fertő Utca 14.
HU-1107 Budapest
Phone: +36 (0)1 / 431-9726
Fax: +36 (0)1 / 431-9727
e mail: office@meltrade.hu

ISRAEL

TEXEL Electronics Ltd.
Box 6272
IL-42160 Netanya
Phone: +972 (0) 9 / 863 08 91
Fax: +972 (0) 9 / 885 24 30
e mail: texel_me@netvision.net.il

KAZAKHSTAN

Kazpromautomatics Ltd.
2, Scladskaya Str.
KAZ-470046 Karaganda
Phone: +7 3212 50 11 50
Fax: +7 3212 50 11 50
e mail: info@kpkaz.com

LATVIA

SIA POWEL
Lienes iela 28
LV-1009 Riga
Phone: +371 784 / 22 80
Fax: +371 784 / 22 81
e mail: utu@utu.lv

LITHUANIA

UAB UTU POWEL
Savanoriu pr. 187
LT-2053 Vilnius
Phone: +370 (0) 52323-101
Fax: +370 (0) 52322-980
e mail: powel@utu.lt

MOLDOVA

INTEHSIS SRL
Cuza-Voda 36/1-81
MD-2061 Chisinau
Phone: +373 (0)2 / 562 263
Fax: +373 (0)2 / 562 263
e mail: intehsis@mdl.net

NETHERLANDS

Koning & Hartman B.V.
Donauweg 2 B
NL-1000 AK Amsterdam
Phone: +31 (0)20 / 587 76 00
Fax: +31 (0)20 / 587 76 05
e mail: info@koninghartman.com

NORWAY

Beijer Electronics A/S
Teglverksveien 1
N-3002 Drammen
Phone: +47 (0) 32 / 24 30 00
Fax: +47 (0) 32 / 84 85 77
e mail: info@beijer.no

POLAND

MPL Technology Sp.z o.o.
ul. Sliczna 36
PL-31-444 Kraków
Phone: +48 (0) 12 / 632 28 85
Fax: +48 (0) 12 / 632 47 82
e mail: krakow@mpl.pl

ROMANIA

Sirius Trading & Services srl
Str. Biharia No. 67-77
RO-013981 Bucuresti 1
Phone: +40 (0) 21 / 201 1146
Fax: +40 (0) 21 / 201 1148
e mail: sirius@siriustrading.ro

RUSSIA

Avtomatika Sever Ltd.
Lva Tolstogo Str. 7, Off. 311
RU-197376 St Petersburg
Phone: +7 812 1183 238
Fax: +7 812 1183 239
e mail: as@avtsev.spb.ru

RUSSIA

Consys
Promyshlennaya St. 42
RU-198099 St Petersburg
Phone: +7 812 325 3653
Fax: +7 812 147 2055
e mail: consys@consys.spb.ru

RUSSIA

Electrotechnical Systems Siberia
Shetinkina St. 33, Office 116
RU-630088 Novosibirsk
Phone: +7 3832 / 119598
Fax: +7 3832 / 119598
e mail: info@eltechsystems.ru

RUSSIA

Elektrostyle
Poslannikov Per., 9, Str. 1
RU-107005 Moscow
Phone: +7 095 542 4323
Fax: +7 095 956 7526
e mail: info@estl.ru

RUSSIA

Elektrostyle
Krasnij Prospekt 220-1, Office No. 312
RU-630049 Novosibirsk
Phone: +7 3832 / 106618
Fax: +7 3832 / 106626
e mail: info@estl.ru

RUSSIA

ICOS
Industrial Computer Systems Zao
Ryazanskij Prospekt, 8A, Off. 100
RU-109428 Moscow
Phone: +7 095 232 0207
Fax: +7 095 232 0327
e mail: mail@icos.ru

RUSSIA

NPP Uralelektra
Sverdlova 11A
RU-620027 Ekaterinburg
Phone: +7 34 32 / 532745
Fax: +7 34 32 / 532745
e mail: elektra@etel.ru

RUSSIA

STC Drive Technique
Poslannikov Per., 9, Str. 1
RU-107005 Moscow
Phone: +7 095 790 7210
Fax: +7 095 790 7212
e mail: info@privod.ru

SERBIA AND MONTENEGRO

INEA SR d.o.o.
Karadjordjeva 12/260
SCG-113000 Smederevo
Phone: +381 (0)26/ 617 - 163
Fax: +381 (0)26/ 617 - 163
e mail: inea_sr@verat.net

SLOVAKIA

AutoCont Control s.r.o.
Radlinského 47
SK-02601 Dolný Kubín
Phone: +421 435868 210
Fax: +421 435868 210
e mail: info@autocontcontrol.sk

SLOVENIA

INEA d.o.o.
Stegne 11
SI-1000 Ljubljana
Phone: +386 (0) 1-513 8100
Fax: +386 (0) 1-513 8170
e mail: inea@inea.si

SWEDEN

Beijer Electronics AB
Box 426
S-20124 Malmö
Phone: +46 (0) 40 / 35 86 00
Fax: +46 (0) 40 / 35 86 02
e mail: info@beijer.se

SWITZERLAND

ECONOTEC AG
Postfach 282
CH-8309 Nürensdorf
Phone: +41 (0) 1 / 838 48 11
Fax: +41 (0) 1 / 838 48 12
e mail: info@econotec.ch

SOUTH AFRICA

CBI Ltd.
Private Bag 2016
ZA-1600 Isando
Phone: +27 (0) 11 / 928 2000
Fax: +27 (0) 11 / 392 2354
e mail: cbi@cbi.co.za

TURKEY

GTS
Darulaceze Cad. No. 43 Kat. 2
TR-80270 Okmeydani-Istanbul
Phone: +90 (0) 212 / 320 1640
Fax: +90 (0) 212 / 320 1649
e mail: gts@turk.net

UKRAINE

CSC Automation Ltd.
15, M. Raskova St., Fl. 10, Office 1010
UA-02002 Kiev
Phone: +380 (0) 44 / 494 3355
Fax: +380 (0) 44 / 494 3366
e mail: csc-a@csc-a.kiev.ua