

MELFA

Industrieroboter

Bedienungsanleitung

CRn-500 Ethernet- Schnittstelle

Bedienungsanleitung
CRn-500 Ethernet-Schnittstelle
Artikel-Nr.: 165774

Version	Änderungen/Ergänzungen/Korrekturen
A 11/2005 pdp-gb	—

Zu diesem Handbuch

Die in diesem Handbuch vorliegenden Texte, Abbildungen, Diagramme und Beispiele dienen ausschließlich der Erläuterung zur Installation, Bedienung und zum Betrieb der in diesem Handbuch beschriebenen Schnittstellenkarte.

Sollten sich Fragen bezüglich Installation und Betrieb der in diesem Handbuch beschriebenen Geräte ergeben, zögern Sie nicht, Ihr zuständiges Verkaufsbüro oder einen Ihrer Vertriebspartner (siehe Umschlagseite) zu kontaktieren.

Aktuelle Informationen sowie Antworten auf häufig gestellte Fragen erhalten Sie über das Internet: <http://www.mitsubishi-automation.de>.

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

© 11/2005

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 Roboter nebst Zubehör dürfen nur von einer anerkannt ausgebildeten Elektrofachkraft, die mit den Sicherheitsstandards der Automatisierungstechnik vertraut ist, durchgeführt 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 Ethernet-Schnittstellenkarte ist nur für die Einsatzbereiche vorgesehen, die in diesem Handbuch beschrieben sind. Achten Sie auf die Einhaltung aller im Handbuch angegebenen Kenndaten.

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.



ACHTUNG:

Im Lieferumfang des Roboters ist ein Sicherheitstechnisches Handbuch enthalten. Dieses Handbuch behandelt alle sicherheitsrelevanten Details zu Aufstellung, Inbetriebnahme und Wartung. Vor einer Aufstellung, Inbetriebnahme oder der Durchführung anderer Arbeiten mit oder am Roboter ist dieses Handbuch unbedingt durchzuarbeiten. Alle darin aufgeführten Angaben sind zwingend zu beachten!

Sollte dieses Handbuch nicht im Lieferumfang enthalten sein, wenden Sie sich bitte umgehend an Ihren Mitsubishi-Vertriebspartner.

Darüber hinaus müssen folgende Vorschriften (ohne Anspruch auf Vollständigkeit) beachtet werden:

- VDE-Vorschriften
- Brandverhütungsvorschriften
- Unfallverhütungsvorschriften

Erläuterung zu den Gefahrenhinweisen

In diesem Handbuch befinden sich Hinweise, die für den sachgerechten sicheren Umgang mit dem Roboter wichtig sind.

Die einzelnen Hinweise haben folgende Bedeutung:



GEFAHR:

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



ACHTUNG:

Bedeutet eine Warnung vor möglichen Beschädigungen der Roboters, seiner Peripherie oder anderen Sachwerten, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

Inhalt

1	Einführung	
1.1	Allgemein	1-1
1.2	Begriffserklärung	1-2
1.3	Bedienungshinweise	1-3
1.3.1	Lieferumfang	1-3
1.4	Modulbeschreibung	1-4
1.4.1	Funktionen der Ethernet-Schnittstellenkarte	1-4
1.5	Überprüfung der Software-Version	1-6
2	Vor der Inbetriebnahme	
2.1	Flussdiagramm zur Inbetriebnahme	2-1
2.2	Auswahl 10Base-T/10Base-5	2-2
2.3	Installation	2-3
2.4	Anschluss	2-5
2.4.1	Installation des Ferritkerns	2-5
2.4.2	Anschluss des Ethernet-Kabels	2-6
2.5	Parameter	2-7
2.5.1	Übersicht der Parameter	2-7
2.5.2	Parameterbeschreibung	2-9
2.5.3	Beispiel zur Parametereinstellung bei Verwendung der PC-Support-Software	2-12
2.5.4	Beispiel zur Parametereinstellung im Data-Link-Betrieb (Das Steuergerät ist der Server.)	2-13
2.5.5	Beispiel zur Parametereinstellung im Data-Link-Betrieb (Das Steuergerät ist der Client.)	2-15
2.5.6	Beispiel zur Parametereinstellung im externen Echtzeit-Steuerungs-Betrieb	2-17
2.6	Anschlussprüfung	2-19
2.6.1	Überprüfung der Verbindung mit dem PING-Befehl	2-19

3	Betrieb	
3.1	Kommunikationsfunktion des Steuergeräts	3-1
3.1.1	Verbinden von Steuergerät und Personalcomputer	3-2
3.1.2	Netzwerkeinstellungen des Personalcomputers	3-2
3.1.3	Parametereinstellungen des Roboter-Steuergerätes.	3-2
3.1.4	Start der PC-Setup-Software	3-3
3.1.5	Kommunikation	3-4
3.1.6	Beenden der Kommunikation	3-4
3.2	Data-Link-Funktion	3-5
3.2.1	Verbinden von Steuergerät und Personalcomputer	3-5
3.2.2	Netzwerkeinstellungen des Personalcomputers	3-6
3.2.3	Einstellung der Parameter des Steuergeräts	3-6
3.2.4	Start des Beispielprogramms	3-7
3.2.5	Kommunikation	3-9
3.2.6	Beenden der Kommunikation	3-9
3.3	Externe Echtzeit-Steuerfunktion	3-10
3.3.1	Verbinden von Steuergerät und Personalcomputer	3-10
3.3.2	Netzwerkeinstellungen des Personalcomputers	3-10
3.3.3	Einstellung der Parameter des Steuergeräts	3-11
3.3.4	Start des Beispielprogramms	3-12
3.3.5	Verfahren des Roboters	3-14
3.3.6	Beenden der Kommunikation	3-15
4	Funktionen	
4.1	Kommunikationsfunktion des Steuergerätes	4-1
4.2	Data-Link-Funktion	4-3
4.2.1	MELFA-BASIC-IV-Funktionalitäten.	4-5
4.3	Externe Echtzeit-Steuerfunktion.	4-10
4.3.1	Erläuterung der Befehle	4-12
4.3.2	Datenpakete	4-15
5	Fehlerdiagnose	
5.1	Übersicht der Fehlermeldungen	5-1
A	Anhang	
A.1	Beispielprogramme.	A-1
A.1.1	Beispielprogramm zum Data-Link-Betrieb	A-1
A.1.2	Beispielprogramm zur externen Echtzeit-Steuerung.	A-8

1 Einführung

1.1 Allgemein

Dieses Handbuch informiert über die Handhabung und die technischen Daten der Ethernet-Schnittstellenkarte HR533.

Die Funktionen der Ethernet-Schnittstellenkarte sind von der Software-Version des Roboter-Steuergeräts abhängig. Folgende Tabelle zeigt die nutzbaren Funktionen in Abhängigkeit der Software-Version.

Software-Version	Kommunikationsfunktion des Steuergeräts	Data-Link-Funktion (Server)	Data-Link-Funktion (Server/Client)	Externe Echtzeit-Steuerfunktion
A□, B□, C□, D□, E1	Keine Funktion der Ethernet-Schnittstellkarte			
E2 bis E4	✓	✓	—	—
F□, G□, H1 bis H6	✓	✓	—	✓
H7 und höher	✓	✓	✓	✓

Tab. 1-1: Funktionen in Abhängigkeit der Software-Version

- ✓: Funktion nutzbar
- : Funktion nicht nutzbar

1.2 Begriffserklärung

- **Ethernet-Schnittstellenkarte**
Die Ethernet-Schnittstellenkarte ist eine Option, die die Anbindung des Roboter-Steuergeräts CRn500 an ein Ethernet-Netzwerk ermöglicht.
- **Netzwerk-Computer**
Der Netzwerk-Computer ist ein handelsüblicher, netzwerkfähiger PC zur Kommunikation mit dem Roboter-Steuergerät über die Ethernet-Schnittstellenkarte. Als Betriebssystem kann Windows 95/Windows 98/Me/WindowsNT4.0 Workstation/Windows 2000 oder Windows XP verwendet werden.
- **10Base-5/10Base-T**
Die Verbindungskabel 10Base-5/10Base-T sind standardisierte Ethernet-Kabel. Das 10Base-5-Kabel erlaubt die Vernetzung von sogenannten Transceivern, die über ein Transceiver-Kabel an die 10Base-5-Leitung angeschlossen sind. 10Base-T-Verbindungen verwenden paarig verdrehte Leitungen (Twisted Pair) zum Aufbau eines sternförmigen Netzwerks mit einem Hub (Verteiler) als Mittelpunkt. Der Anschluss an einen Hub erfolgt über ein Straight-Through-Kabel, während bei der direkten Verbindung zweier Einheiten ein Crossover-Kabel verwendet wird. Das 10Base-T-Kabel ist das Standard-Ethernet-Kabel, da es einfach zu verdrahten, kostengünstiger und allgemein erhältlich ist.
- **MELFA-BASIC IV/MOVEMASTER-COMMAND**
MELFA-BASIC IV und MOVEMASTER-COMMAND sind zwei Roboter-Programmiersprachen. Die Steuergeräte CRn-500 erlauben eine Programmierung in beiden Sprachen. MELFA-BASIC IV ist eine hoch entwickelte Programmiersprache zur Erstellung komplexer Roboterprogramme. Sie ist an die herkömmliche BASIC-Programmiersprache angelehnt. Die MOVEMASTER-COMMAND-Programmiersprache wurde zur Programmierung der Roboter der RV-EN-Serie verwendet.

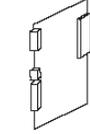
HINWEIS

Die MOVEMASTER-COMMAND-Programmiersprache kann nur mit einigen Robotermodellen verwendet werden (RV-1A/RV-2AJ usw.). In Abhängigkeit des Robotermodells ist nur eine Programmierung in MELFA-BASIC IV möglich. Welche Programmiersprache Sie für Ihren Roboter verwenden können, finden Sie im Technischen Handbuch Ihres Roboters. Werksseitig ist die Programmiersprache MELFA-BASIC IV voreingestellt. Eine Änderung der Programmiersprache erfolgt mit Hilfe des Parameters RLNG (siehe Technisches Handbuch).

1.3 Bedienungshinweise

1.3.1 Lieferumfang

Überprüfen Sie den Packungsinhalt auf Vollständigkeit. Im Lieferumfang sind folgende Komponenten enthalten:

Lieferumfang	Bezeichnung	Anzahl
	Bedienungsanleitung Art.-Nr. 165774	1
	Ethernet-Schnittstellenkarte HR533	1
	Ferritkern —	1

Tab. 1-2: Lieferumfang

Zusätzlich zum Robotersystem werden noch die folgenden Komponenten benötigt. (Diese sind separate Geräte, die vom Benutzer bereit gestellt werden müssen.)

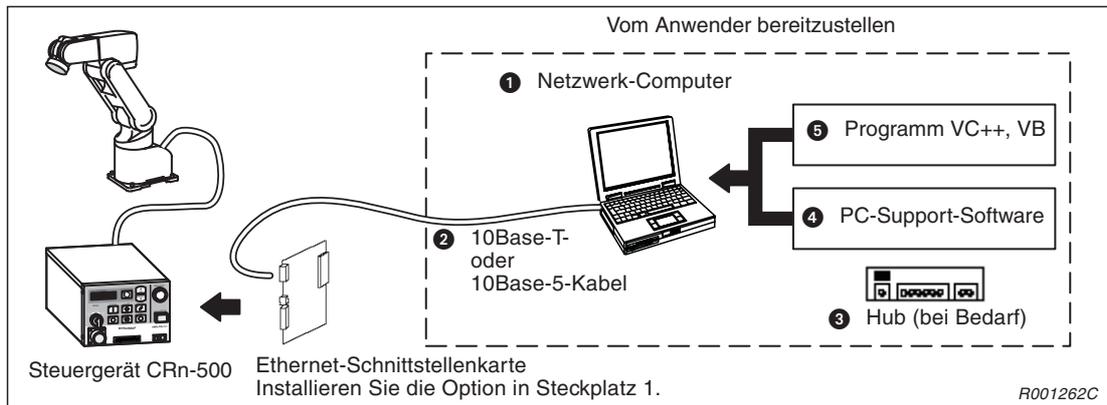


Abb. 1-1: Weitere Komponenten des Netzwerks

Nr.	Bezeichnung	Beschreibung	Anzahl
1	Netzwerk-Computer (mit Ethernet-Netzwerkkarte)	PC mit Betriebssystem Windows 95/98/Me/WindowsNT4.0 Workstation/Windows 2000 oder Windows XP PC mit Unterstützung des TCP/IP-Protokolls für den Netzwerkbetrieb wie z. B. LinuxOS (Betrieb nicht geprüft).	≥ 1
2	Ethernet-Kabel (Wählen Sie in Abhängigkeit der Anforderungen ein Straight-Through- oder ein Crossover-Kabel.)	10Base-T oder 10Base-5	≥ 1
3	Hub (bei Einsatz in einem LAN)	Handelsübliches Produkt	1
4	Programmierwerkzeug für NARC-Steuergeräte	Optionale PC-Support-Software	1
5	Auf Windows basierende Software zur Netzwerkprogrammierung.	Handelsübliches Produkt, z. B. von Microsoft Visual C++5.0/6.0 usw.	1

Tab. 1-3: Weitere Komponenten des Netzwerks

1.4 Modulbeschreibung

1.4.1 Funktionen der Ethernet-Schnittstellenkarte

Die Ethernet-Schnittstellenkarte verfügt über folgende Funktionen:

- Möglichkeit zum Anschluss von 10Base-T- oder 10Base-5-Leitungen
- Unterstützung des TCP/IP-Protokolls zur Kommunikation mit einem PC über das Ethernet
- Jedes Steuergerät, das an das Ethernet angebunden werden soll, benötigt eine Ethernet-Schnittstellenkarte. Die Schnittstellenkarte muss in Steckplatz 1 installiert werden.
- Beispielprogramme (basierend auf Microsoft Visual Basic, Version 5.0) für den Personalcomputer finden Sie im Anhang A.
Die Data-Link-Funktion ermöglicht über die Befehle OPEN, INPUT# und PRINT# das Versenden und den Empfang von Variablen des Personalcomputers und des Roboter-Steuergeräts (Zeichen und numerische Variable).
Für die Ergebnisse, die mittels des Beispielprogramms durch die Anwendung erzeugt werden, ist Mitsubishi nicht verantwortlich.
- Im Folgenden werden die drei auf das Ethernet bezogenen Funktionen der Schnittstellenkarte HR533 erläutert. Eine detaillierte Beschreibung der Funktionen finden Sie in Kap. 4.

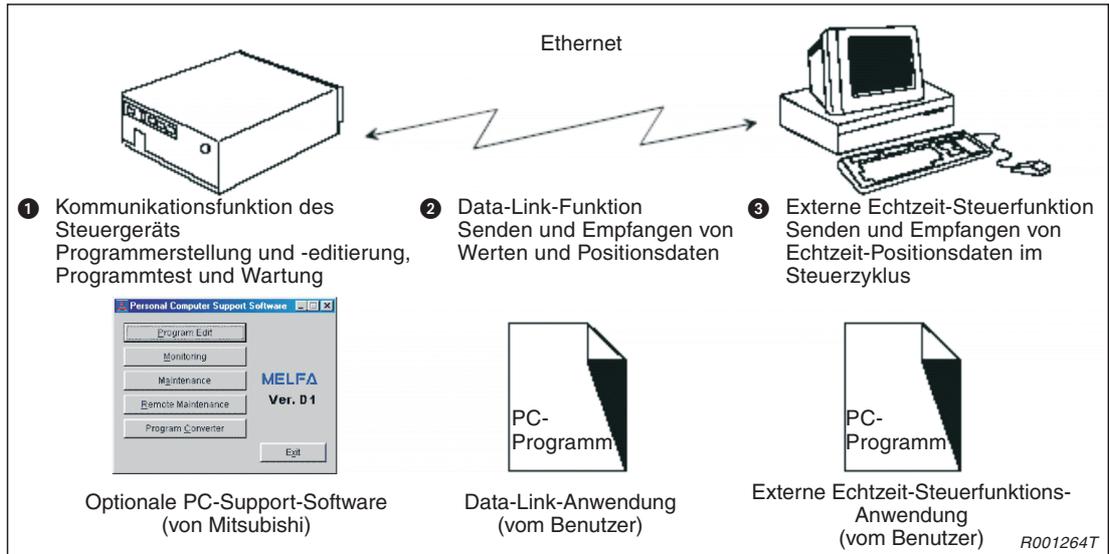


Abb. 1-2: Funktionen der Ethernet-Schnittstellenkarte

Nr.	Funktion	Bemerkung	Siehe
1	Kommunikationsfunktion des Steuergeräts Der Datenaustausch mit dem Steuergerät erfolgt über das Ethernet (Programm laden/speichern, Statusüberwachung usw.). Die optionale PC-Support-Software ist als Anwendungsprogramm erhältlich.	Eine Kommunikation mit bis zu 16 Teilnehmern ist möglich.	Kap. 1 (allgemein) Kap. 2 (allgemein) Abschn. 3.1 Abschn. 4.1 Abschn. 5.1
2	Data-Link-Funktion Der Austausch von Werten und Positionsdaten zwischen dem Steuergerät und dem Personalcomputer erfolgt mit Hilfe der Programmiersprache MELFA-BASIC IV (Befehle OPEN/PRINT/INPUT)	Durch das Öffnen unterschiedlicher Kommunikationsleitungen ist eine Kommunikation mit bis zu 8 Clients möglich.	Kap. 1 (allgemein) Kap. 2 (allgemein) Abschn. 3.2 Abschn. 4.2 Abschn. 5.1 Anhang A.1.1
3	Externe Echtzeit-Steuerfunktion Positionsdaten können als Gelenk-, XYZ-Daten oder als Motorimpulse vom Steuergerät abgerufen und verarbeitet werden. Weiterhin ist eine Überwachung der Ein- und Ausgangssignale und eine gleichzeitige Ausgabe von Signalen möglich. Die Steuerung wird durch den Befehl MXT (MELFA-BASIC IV oder MOVEMASTER-COMMAND) gestartet. Die Funktion kann nur mit den folgenden Roboterarmen verwendet werden:	Zur Steuerung des Roboters muss der Benutzer eine Anwendungsprogramm für den Personalcomputer erstellen. Die Kommunikation erfolgt eins zu eins.	Kap. 1 (allgemein) Kap. 2 (allgemein) Abschn. 3.3 Abschn. 4.3 Abschn. 5.1 Abschn. A.1.2

Tab. 1-4: Übersicht der Funktionen der Ethernet-Schnittstellenkarte

HINWEISE

- Der Personalcomputer, der mit den Steuergeräten kommuniziert, muss am selben Netzwerk angeschlossen sein.
- Eine Datenübertragung über Firewalls (vom Internet) oder Gateways (von anderen Netzwerken o. Ä.) ist nicht möglich.
- Führen Sie die Kommunikation zum Betrieb über einen Server (z. B. HTTP-Server), der am selben Netzwerk angeschlossen ist, aus. Beachten Sie beim Betrieb des Systems insbesondere alle sicherheitsrelevanten Aspekte.

1.5 Überprüfung der Software-Version

Die Ethernet-Funktionen der Schnittstellenkarte sind ab Software-Version E2 nutzbar. Ab Version E5 steht die externe Echtzeit-Steuerfunktion zur Verfügung. Überprüfen Sie daher die Software-Version Ihres Roboter-Steuergeräts, bevor Sie die Ethernet-Schnittstellenkarte einsetzen.

Bei den Software-Versionen A□, B□, C□ oder E1 stehen auch bei installierter Schnittstellenkarte keine Ethernet-Funktionen zur Verfügung.

Überprüfung der Software-Version mit der Teaching Box

Stellen Sie den [EENABLE/DISABLE]-Schalter der Teaching Box auf „DISABLE“ und schalten Sie anschließend die Spannungsversorgung des Roboter-Steuergeräts ein. In Zeile ② der folgenden Tabelle ist das Display der Teaching Box bei Anzeige der Software-Version dargestellt.

Nr.	Display-Darstellung	Beschreibung
①	<pre>P28TB Ver.B2 Robot System Teaching Box ■ JUST A MINUTE ■</pre>	<p>Nach dem Einschalten wird zuerst die Software-Version der Teaching Box angezeigt. Software-Version der Teaching Box: hier z. B. B2.</p>
②	<pre>CRn-5xx Ver.H7 RV-1A Copyright (C) 1999 ■ Press a key ■</pre>	<p>Nach einigen Sekunden wird die Software-Version des Roboter-Steuergeräts angezeigt. Software-Version des Roboter-Steuergeräts: H7 Version A wird z. B. durch Ver.A□ angezeigt (□: 1 oder höher)</p>

Tab. 1-5: Überprüfung der Software-Version

2 Vor der Inbetriebnahme

2.1 Flussdiagramm zur Inbetriebnahme

Gehen Sie zur Inbetriebnahme der Ethernet-Schnittstellenkarte wie in folgendem Flussdiagramm beschrieben vor.

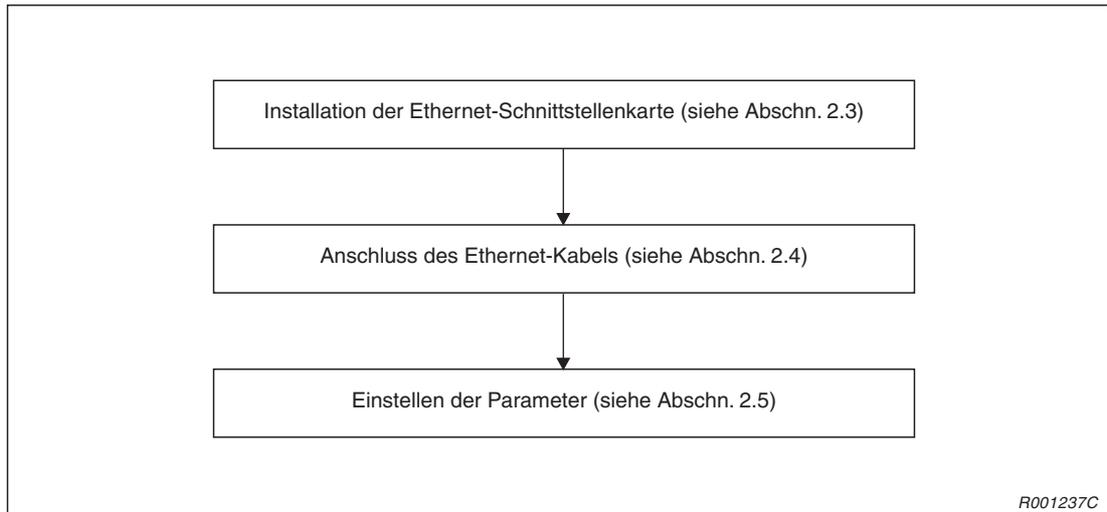


Abb. 2-1: Flussdiagramm zur Inbetriebnahme

2.2 Auswahl 10Base-T/10Base-5

Stellen Sie den Umschalter SW1 auf der Schnittstellenkarte entsprechend dem von Ihnen verwendeten Kabel in die richtige Position.

Verwenden Sie ein 10Base-T-Kabel, stellen Sie den Schalter in die Position „BASET“ (untere Schalterstellung). Verwenden Sie ein 10Base-5-Kabel, stellen Sie den Schalter in die Position „BASE5“ (obere Schalterstellung).

Der Schalter SW1 befindet sich oben rechts auf der Ethernet-Schnittstellenkarte (siehe folgende Abbildung).

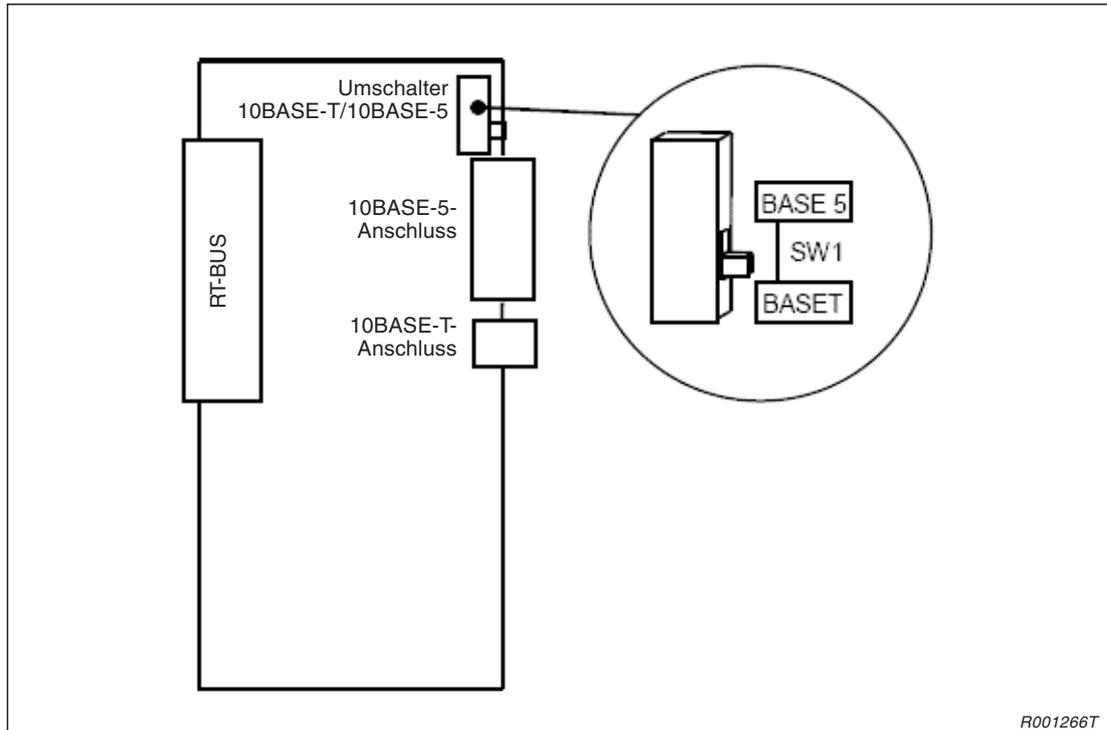


Abb. 2-2: 10Base-T/10BASE-5-Umschalter

2.3 Installation

Die Ethernet-Schnittstellenkarte wird auf die Steuerplatine (R6CPU oder RZ322) des Steuergerätes CR2/CR2A/CR2B/CR3 oder beim Steuergerät CR1 im Erweiterungsmodul (CR1-EB3) montiert. Informationen zum Einbau in das Erweiterungsmodul entnehmen Sie bitte dem technischen Handbuch des entsprechenden Roboterarms.

HINWEIS

In diesem Handbuch wird der Einbau der Schnittstellenkarte in die Steuergeräte CR2 und CR3 gezeigt. Die Vorgehensweise zum Einbau in ein anderes Steuergerät finden Sie im technischen Handbuch des entsprechenden Roboterarms.

Bei der Installation auf der Steuerplatine gehen Sie wie folgt vor:

**ACHTUNG:**

Trennen Sie die Netzzuleitung vom Stromnetz, bevor Sie die Gehäuseabdeckung entfernen! Schließen Sie die Spannungsversorgung erst nach Wiederbefestigung der Abdeckung wieder an das Stromnetz an.

- ① Schalten Sie den Netzschalter des Steuergerätes aus. Trennen Sie die Netzzuleitung vom Stromnetz.
- ② Entfernen Sie die obere Gehäuseabdeckung ①. Dazu lösen Sie die Schraube ② auf der Rückseite des Steuergerätes. Ziehen Sie die Abdeckung ca. 100 mm nach hinten, um sie anschließend hochheben zu können.
- ③ Lösen Sie die vier Schrauben ③ der Rückabdeckung des Steuergerätes und entfernen Sie die Rückabdeckung.
- ④ Lösen Sie die drei M3-Schrauben ④ der Steuerplattenabdeckung ⑤ und entfernen Sie die Abdeckung.
- ⑤ Stecken Sie die Ethernet-Schnittstellenkarte ⑥ in den Steckplatz 1 (OPT1).
- ⑥ Montieren Sie die Abdeckung ⑤ der Steuerplatine. Die Abdeckung besitzt zwei Führungsschienen. Die Schnittstellenkarte muss auf die Führungsschienen aufgesetzt werden.
- ⑦ Befestigen Sie die Steuerplattenabdeckung ⑤ mit den entsprechenden Schrauben ④.
- ⑧ Montieren Sie die Rückabdeckung des Steuergeräts mit vier Schrauben ③.
- ⑨ Montieren Sie die Gehäuseabdeckung ①. Achten Sie darauf, dass keine Kabel eingeklemmt sind.
- ⑩ Schalten Sie die Versorgungsspannung wieder ein.

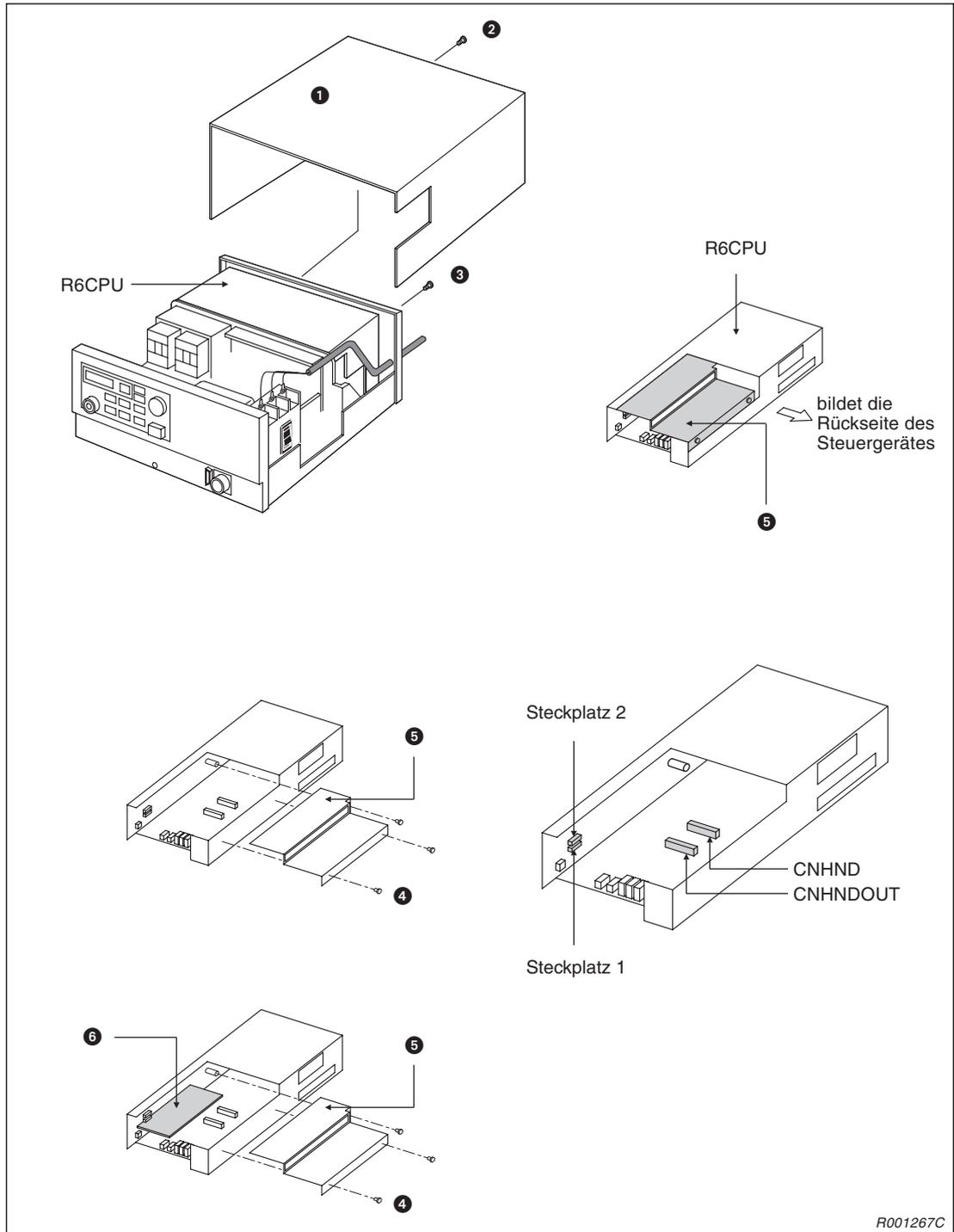


Abb. 2-3 Einbau der Ethernet-Schnittstellenkarte in Steckplatz 1

2.4 Anschluss

2.4.1 Installation des Ferritkerns

Der Ferritkern dient zur Unterdrückung hochfrequenter Störungen auf der Datenleitung. Der Ferritkern muss zwischen dem Roboter-Steuergerät und der nächsten Ethernet-Einheit am Ethernet-Kabel installiert werden. Dabei sollte der Abstand zum Roboter-Steuergerät etwa 30 cm betragen (siehe folgende Abbildung). Wird der Ferritkern nicht den Angaben entsprechend installiert, können Störungen auf der Datenleitung zu Fehlfunktionen führen.

Ist das Netzwerk starken Störeinwirkungen ausgesetzt, verwenden Sie ein abgeschirmtes Ethernet-Kabel.

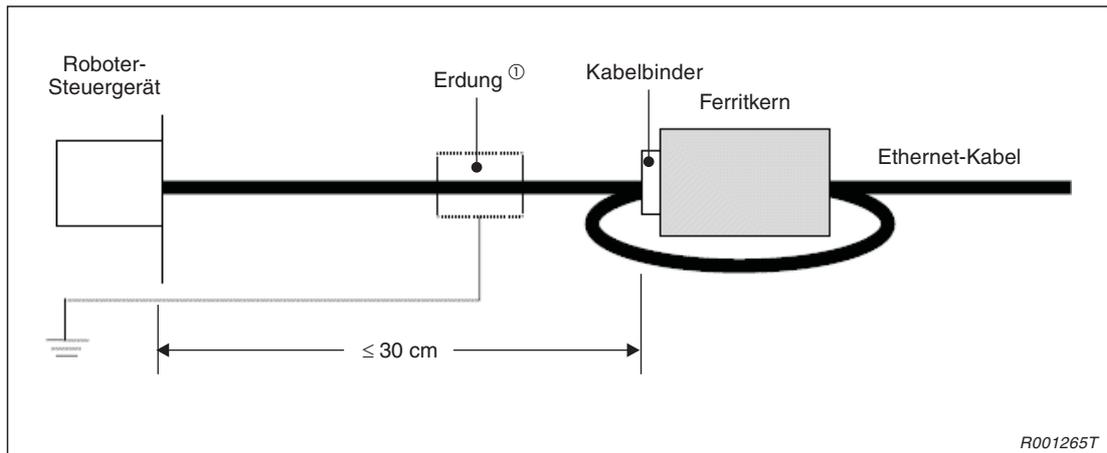


Abb. 2-4: Installation des Ferritkerns

- ① Erden Sie die Abschirmung des Ethernet-Kabels beim Betrieb des Systems in einer Umgebung mit großen Störeinwirkungen. Entfernen Sie dazu einen Teil der Isolierung des Ethernet-Kabels und verbinden Sie die freigelegte Abschirmung durch eine P-förmige Metallschelle mit der Erde des Steuergerätes.

2.4.2 Anschluss des Ethernet-Kabels

Verbinden Sie das Ethernet-Kabel mit dem 10BASE-T- oder dem 10BASE-5-Anschluss.

Der Anschluss an einen Hub erfolgt über ein Straight-Through-Kabel, während bei der direkten Verbindung von Personalcomputer und Steuergerät ein Crossover-Kabel verwendet wird.

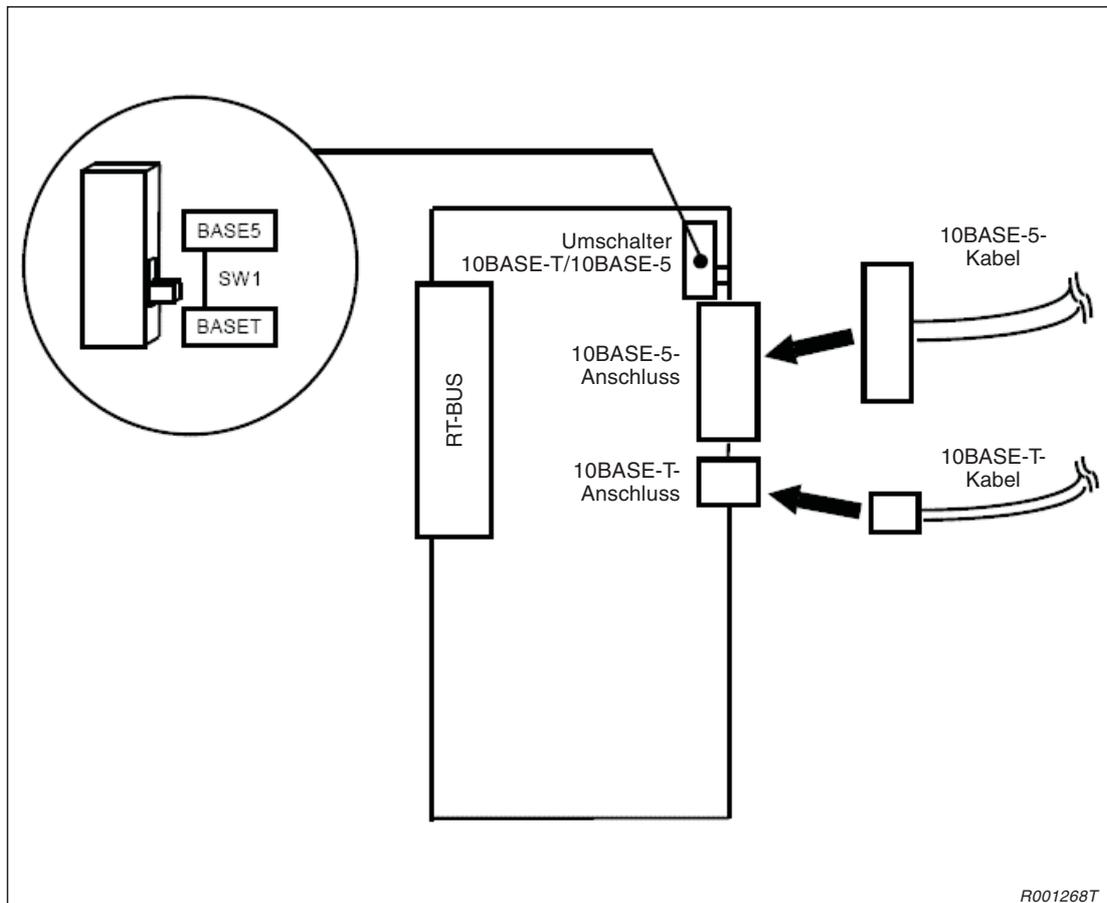


Abb. 2-5: Anschluss des Ethernet-Kabels

2.5 Parameter

Vor dem Betrieb des Netzwerks müssen folgende Parameter des Roboter-Steuergeräts eingestellt werden. Die Methode zur Einstellung der Parameter finden Sie in der Bedienungs- und Programmieranleitung der Roboter-Steuergeräte.

HINWEIS

Die Änderung einer Parametereinstellung wird erst nach Aus- und Wiedereinschalten der Spannungsversorgung des Steuergerätes aktiv.

2.5.1 Übersicht der Parameter

Eine detaillierte Beschreibung der Parameter finden Sie auf den nächsten Seiten.

Parameter	Beschreibung	Anzahl der Elemente	Werks-einstellung	Kommuni-kations-funktion des Steuer-geräts	Data-Link-Funktion	Echtzeit-Steuer-funktion
NETIP	IP-Adresse des Steuergerätes	Zeichenkette 1	192.168.0.1	✓	✓	✓
NETMSK	Subnetzmaskierung	Zeichenkette 1	255.255.255.0	✓	✓	✓
NETPORT	Schnittstellenummer Bereich 0–32767 Für Funktionserweiterung (reserviert) ----- Entspricht OPT11–19 von COMDEV (OPT11) (OPT12) (OPT13) (OPT14) (OPT15) (OPT16) (OPT17) (OPT18) (OPT19)	Numerischer Wert 10	10000 10001 10002 10003 10004 10005 10006 10007 10008 10009	✓	✓	✓
CPRCE11 CPRCE12 CPRCE13 CPRCE14 CPRCE15 CPRCE16 CPRCE17 CPRCE18 CPRCE19	Protokoll 0: kein Protokoll, 1: Protokoll, 2: Data-Link (1: Protokoll ist ohne Funktion) Für Funktionserweiterung (reserviert) ----- Entspricht OPT11–19 von COMDEV (OPT11) (OPT12) (OPT13) (OPT14) (OPT15) (OPT16) (OPT17) (OPT18) (OPT19)	Numerischer Wert 9	0 0 0 0 0 0 0 0 0 0	—	✓	—
COMDEV	Festlegung des Geräts an COM1 bis 8 Festlegung des Geräts an COM1: Festlegung des Geräts an COM2: Festlegung des Geräts an COM3: Festlegung des Geräts an COM4: Festlegung des Geräts an COM5: Festlegung des Geräts an COM6: Festlegung des Geräts an COM7: Festlegung des Geräts an COM8: Bei gewählter Data-Link-Funktion ist eine Einstellung notwendig. OPT11 bis OPT19 werden zugewiesen. Die RS232-Schnittstelle des Steuergeräts ist COM1 zugewiesen.	Zeichenkette 8	RS232C RS232C RS232C RS232C RS232C RS232C RS232C RS232C	—	✓	—

Tab. 2-1: Parameterübersicht (1)

Parameter	Beschreibung	Anzahl der Elemente	Werks-einstellung	Kommuni-kations-funktion des Steuer-geräts	Data-Link-Funktion	Echtzeit-Steuer-funktion
NETMODE (ab Software-Version H7)	Festlegung des Servers (1: Server, 0: Client) (OPT11) (OPT12) (OPT13) (OPT14) (OPT15) (OPT16) (OPT17) (OPT18) (OPT19)	Numerischer Wert 9	1 1 1 1 1 1 1 1 1	—	✓	—
NETHSTIP (ab Software-Version H7)	IP-Adresse des Servers (OPT11) (OPT12) (OPT13) (OPT14) (OPT15) (OPT16) (OPT17) (OPT18) (OPT19)	Zeichenkette 9	192.168.0.2 192.168.0.3 192.168.0.4 192.168.0.5 192.168.0.6 192.168.0.7 192.168.0.8 192.168.0.9 192.168.0.10	—	✓	—
MXTCOM1	IP-Adresse für die externe Echtzeit-Steuerfunktion Adresse für den Kommunikationspunkt Nummer 1	Wert 1	192.168.0.2	—	—	✓ Bei Programmierung in MOVE-MASTER-COMMAND
MXTCOM2	Adresse für den Kommunikationspunkt Nummer 2	Wert 1	192.168.0.3	—	—	
MXTCOM3	Adresse für den Kommunikationspunkt Nummer 3	Wert 1	192.168.0.4	—	—	
MXTTOUT	Zeit zur Ausführung eines externen Echtzeit-Steuerbefehls (Einstellwert mal 7,1 ms; wird durch die Einstellung von „-1“ deaktiviert)	Wert 1 (0-32767)	-1	—	—	✓

Tab. 2-1: Parameterübersicht (2)

2.5.2 Parameterbeschreibung

- **NETIP (IP-Adresse des Roboter-Steuergerätes)**

Dieser Parameter dient zur Einstellung der IP-Adresse des Roboter-Steuergerätes. Die IP-Adresse ist mit einer Postanschrift vergleichbar.
Die IP-Adresse besteht aus vier Zahlen im Bereich zwischen 0 und 255, die durch einen Punkt getrennt sind (z. B. 192.168.0.1 oder 10.97.11.31).
Ist das Steuergerät direkt mit dem Netzwerk-Computer verbunden, kann ein frei wählbarer Wert vorgegeben werden. Bei Anschluss an ein LAN (Local Area Network) ist die IP-Adresse entsprechend den Vorgaben des LAN-Systemadministrators einzustellen.
Bei einer Überlappung der IP-Adressbereiche ist kein einwandfreier Betrieb möglich. Vermeiden Sie daher ein Überlappen der IP-Adressbereiche.
Der Personalcomputer, der mit dem Steuergerät kommuniziert, muss am gleichen Netzwerk angeschlossen sein.
- **NETMSK (Subnetmaske)**

Die Subnetmaske des Roboter-Steuergerätes dient zur Festlegung von untergeordneten Netzwerken.
Die Subnetmaske besteht aus vier Zahlen im Bereich zwischen 0 und 255, die durch einen Punkt getrennt sind (z. B. 255.255.255.0 oder 255.255.0.0).
In der Regel kann hier ein frei wählbarer Wert vorgegeben werden. Bei Anschluss an ein LAN (Local Area Network) ist die Subnetmaske entsprechend den Vorgaben des LAN-Systemadministrators einzustellen.
- **NETPORT (Schnittstellennummer)**

Dieser Parameter dient zur Einstellung der Schnittstellennummer des Roboter-Steuergerätes. Die Schnittstellennummer ist mit dem Namen eines Postempfängers vergleichbar.
Die Schnittstellennummer besteht aus 9 Elementen. Das erste Element legt die Echtzeit-Steuerung fest. Die Elemente 2 bis 9 werden für die Support-Software oder für die Data-Link-Funktion verwendet.
In der Regel muss die Werkseinstellung nicht verändert werden. Achten Sie darauf, dass keine Schnittstellennummer mehrfach vergeben wird.
- **CPRCE11 bis 19 (Protokoll)**

Bei Verwendung der Data-Link-Funktion ist eine Einstellung notwendig. Durch den Parameter wird das Kommunikationsprotokoll (Protokoll) festgelegt. Es sind drei Einstellungen möglich: kein Protokoll, Protokoll oder Data-Link.

 - 0: kein Protokoll (Das Protokoll wird über die PC-Support-Software vorgegeben.)
 - 1: Protokoll: reserviert (Vermeiden Sie eine fehlerhafte Einstellung des Wertes, da die Einstellung keine Funktion hat.)
 - 2: Data-Link (Es werden die Kommunikationsbefehle OPEN/INPUT/PRINT verwendet.)

- **COMDEV** (Festlegung der Geräte an den Schnittstellen COM1 bis 8)
Bei Verwendung der Data-Link-Funktion ist eine Einstellung notwendig.
Der Parameter dient bei Verwendung des OPEN-Befehls der Zuweisung der Kommunikationsleitungen an die Schnittstellen COM1 bis COM8. Stellen Sie die Parameter nur ein, wenn Sie über die Parameter CPRCE11 bis 19 die Data-Link-Funktion gewählt haben. Die Einstellwerte der Ethernet-Schnittstellenkarte entsprechen den Schnittstellennummern des Parameters NETPORT.
Im Folgenden werden die Elementnummern der Parameter NETPORT(n) und COMDEV(n) durch „n“ gekennzeichnet.

n	Mit Parameter COMDEV(n) eingestellte Gerätebezeichnung	Schnittstellenummer
1	OPT11	Die Schnittstellenummer ist durch NETPORT(2) festgelegt.
2	OPT12	Die Schnittstellenummer ist durch NETPORT(3) festgelegt.
3	OPT13	Die Schnittstellenummer ist durch NETPORT(4) festgelegt.
4	OPT14	Die Schnittstellenummer ist durch NETPORT(5) festgelegt.
5	OPT15	Die Schnittstellenummer ist durch NETPORT(6) festgelegt.
6	OPT16	Die Schnittstellenummer ist durch NETPORT(7) festgelegt.
7	OPT17	Die Schnittstellenummer ist durch NETPORT(8) festgelegt.
8	OPT18	Die Schnittstellenummer ist durch NETPORT(9) festgelegt.
9	OPT19	Die Schnittstellenummer ist durch NETPORT(10) festgelegt.

Tab. 2-2: Gerätebezeichnung und Schnittstellenummer

Beispiel ▽

Soll die mit der Einstellung NETPORT(3) festgelegte Schnittstellenummer im Data-Link-Betrieb mit der Schnittstelle COM3: verbunden werden, müssen folgende Einstellungen vorgenommen werden:

COMDEV(3) = OPT13

OPT13 wird als 3. Element von COMDEV festgelegt.

CPRCE13 = 2

Einstellung der Data-Link-Funktion



- **NETMODE** (Serverfestlegung) (ab Software-Version H7)
Bei Verwendung der Data-Link-Funktion ist eine Einstellung notwendig.
Stellen Sie im Data-Link-Betrieb die TCP/IP-Kommunikation des Roboter-Steuergerätes als Server oder Client ein. Eine Einstellung muss in Abhängigkeit der mit dem Steuergerät verbundenen Geräte erfolgen.
Diese Funktion steht ab Software-Version H7 des Roboter-Steuergerätes zur Verfügung. In älteren Software-Versionen arbeitet das Steuergerät ausschließlich als Server.
- **NETHSTIP** (IP-Adresse des Servers) (ab Software-Version H7)
Eine Einstellung ist im Data-Link-Betrieb bei Einsatz des Roboter-Steuergerätes als Client notwendig.
Stellen Sie in diesem Parameter die IP-Adresse des Servers ein, mit dem das Roboter-Steuergerät im Data-Link-Betrieb kommuniziert.
Stellen Sie den Parameter ein, wenn das Roboter-Steuergerät über die Server-Festlegung im Parameter NETMODE als Client festgelegt wurde.

- MXTCOM1 bis 3 (IP-Adresse der Kommunikation im externen Echtzeit-Steuerungs-Betrieb)
Die Einstellung des Parameters ist nur im externen Echtzeit-Steuerungs-Betrieb bei Verwendung der MOVEMASTER-COMMAND-Programmiersprache notwendig. Legen Sie in diesem Parameter die Ziel-IP-Adresse des Personalcomputers fest, mit dem das Roboter-Steuergerät kommuniziert.
- MXTOUT (Zeit zur Ausführung eines externen Echtzeit-Steuerbefehls)
Dieser Parameter dient zur Einstellung der Zeit, die bei Kommunikation mit dem Steuergerät zur Ausführung eines externen Echtzeit-Steuerbefehls zur Verfügung steht. Stellen Sie die Zeit als ein Vielfaches des Steuerzyklus von 7,1 ms ein.
Bei Ausführung eines externen Echtzeit-Steuerbefehls wird die Zeit, in der das Steuergerät keine Daten vom Personalcomputer empfängt, erfasst. Übersteigt die Zeit den in Parameter MXTOUT eingestellten Wert, stoppt der Betrieb und es erfolgt die Ausgabe der Fehlermeldung #7820. In der Werkseinstellung (-1) ist die Erfassung der Zeit zur Ausführung eines externen Echtzeit-Steuerbefehls deaktiviert.

Beispiel ▾

Soll nach einer kommunikationsfreien Zeit von etwa 7 s die Ausgabe der Fehlermeldung #7820 erfolgen, stellen Sie einen Wert von 1000 ein.



2.5.3 Beispiel zur Parametereinstellung bei Verwendung der PC-Support-Software

Stellen Sie die Parameter des Roboter-Steuergerätes und des Netzwerks zur Kommunikation mit dem Personalcomputer ein.

Parameter	Einstellung
IP-Adresse des Roboter-Steuergerätes	192.168.0.1
IP-Adresse des Personalcomputers	192.168.0.2
Schnittstellenummer des Roboter-Steuergerätes	10001

Tab. 2-3: Bedingungen

Stellen Sie die Parameter des Roboter-Steuergerätes wie folgt ein. Bei Verwendung der Werkseinstellung bleiben die Parameter unverändert.

Einzustellender Parameter	Vor und nach der Änderung	Einstellung
NETIP	Vorher	192.168.0.1
	Nachher	192.168.0.1 (Werkseinstellung bleibt)
NETPORT	Vorher	10001
	Nachher	10001 (Werkseinstellung bleibt)

Tab. 2-4: Verwendete Einstellungen

Stellen Sie als nächstes im Menü „Netzwerkverbindungen“ die IP-Adresse des Personalcomputers 192.168.0.2 ein.

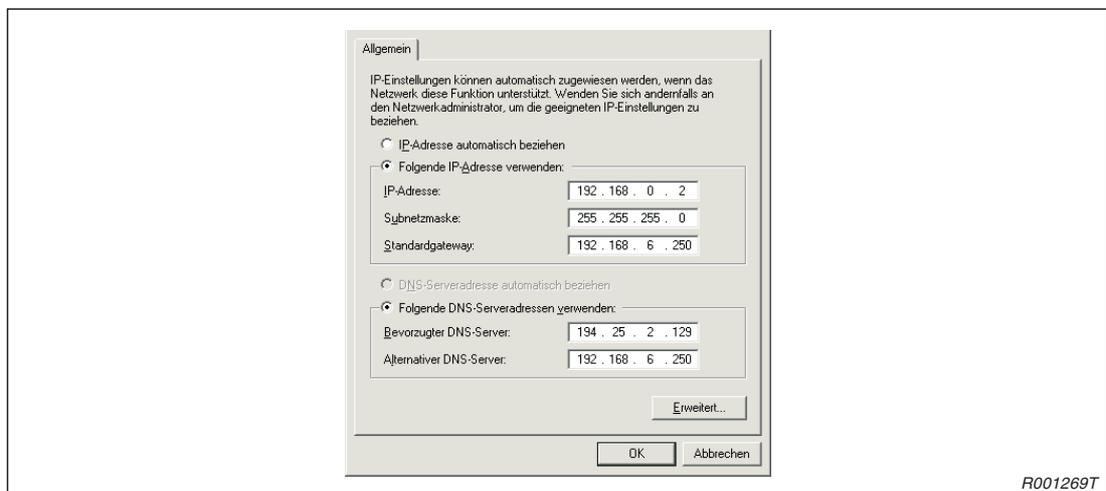


Abb. 2-6: Einstellung der IP-Adresse des Personalcomputers

Die IP-Adresse des Personalcomputers wird in Windows im Menü „Netzwerkverbindungen“ eingestellt. Die Menüs unterscheiden sich in Abhängigkeit der Windows-Version. Weitere Hinweise entnehmen Sie dem Handbuch Ihrer Windows-Version. Die Abbildung oben zeigt das Menü in Windows XP.

Hinweise zur Anwendung der PC-Support-Software entnehmen Sie dem Handbuch der PC-Support-Software.

2.5.4 Beispiel zur Parametereinstellung im Data-Link-Betrieb (Das Steuergerät ist der Server.)

Dieses Beispiel zeigt die Einstellung der Parameter, wenn das Steuergerät im Data-Link-Betrieb als Server dient.

Parameter	Einstellung
IP-Adresse des Roboter-Steuergerätes	192.168.0.1
IP-Adresse des Personalcomputers	192.168.0.2
Schnittstellenummer des Roboter-Steuergerätes	10003
Nummer der Kommunikationsschnittstelle (Für MELFA-BASIC IV) OPEN-Befehl COM-Nr. (Für MOVEMASTER-COMMAND) OPN-Befehl Kommunikationsleitung	COM3: 3

Tab. 2-5: Bedingungen

Stellen Sie die Parameter des Roboter-Steuergerätes wie folgt ein. Bei Verwendung der Werkseinstellung bleiben die Parameter unverändert.

Einzustellender Parameter	Vor und nach der Änderung	Einstellung
NETIP	Vorher	192.168.0.1
	Nachher	192.168.0.1 (Werkseinstellung bleibt)
NETPORT	Vorher	10000, 10001, 10002, 10003, 10004, 10005, 10006, 10007, 10008, 10009
	Nachher	10000, 10001, 10002, 10003, 10004, 10005, 10006, 10007, 10008, 10009 (Werkseinstellung bleibt)
CPRCE13	Vorher	0
	Nachher	2
COMDEV	Vorher	RS232, , , , , ,
	Nachher	RS232, , OPT13, , , , ,

Tab. 2-6: Verwendete Einstellungen

Stellen Sie als nächstes im Menü „Netzwerkverbindungen“ die IP-Adresse des Personalcomputers 192.168.0.2 ein.

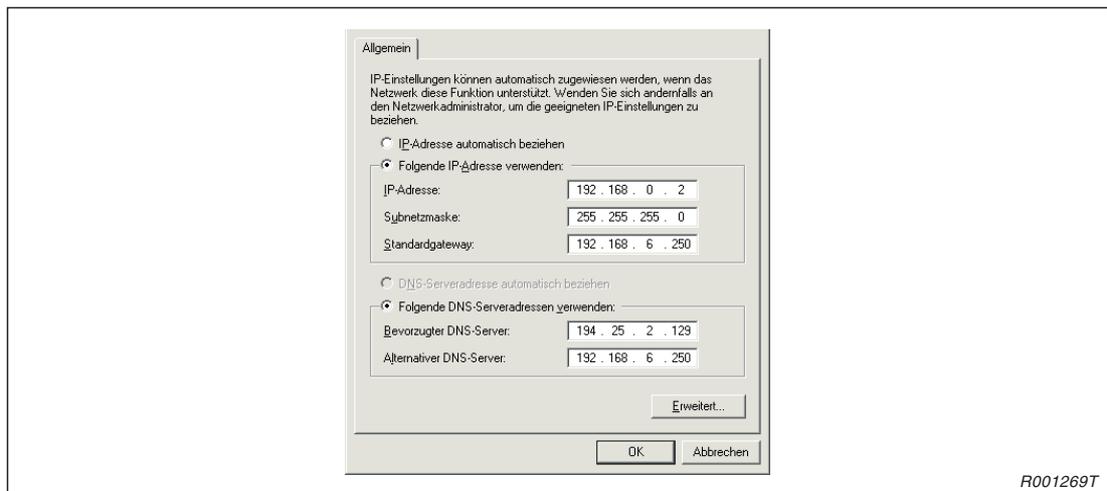


Abb. 2-7: Einstellung der IP-Adresse des Personalcomputers

Die IP-Adresse des Personalcomputers wird in Windows im Menü „Netzwerkverbindungen“ eingestellt. Die Menüs unterscheiden sich in Abhängigkeit der Windows-Version. Weitere Hinweise entnehmen Sie dem Handbuch Ihrer Windows-Version. Die Abbildung oben zeigt das Menü in Windows XP.

Hinweise zur Anwendung der PC-Support-Software entnehmen Sie dem Handbuch der PC-Support-Software.

2.5.5 Beispiel zur Parametereinstellung im Data-Link-Betrieb (Das Steuergerät ist der Client.)

Dieses Beispiel zeigt die Einstellung der Parameter, wenn das Steuergerät im Data-Link-Betrieb als Client dient.

Parameter	Einstellung
IP-Adresse des Roboter-Steuergerätes	192.168.0.1
IP-Adresse des Personalcomputers	192.168.0.2
Schnittstellenummer des Roboter-Steuergerätes	10003
Nummer der Kommunikationsschnittstelle (Für MELFA-BASIC IV) OPEN-Befehl COM-Nr. (Für MOVEMASTER-COMMAND) OPN-Befehl Kommunikationsleitung	COM3: 3

Tab. 2-7: Bedingungen

Stellen Sie die Parameter des Roboter-Steuergerätes wie folgt ein. Bei Verwendung der Werkseinstellung bleiben die Parameter unverändert.

Einzustellender Parameter	Vor und nach der Änderung	Einstellung
NETIP	Vorher	192.168.0.1
	Nachher	192.168.0.1 (Werkseinstellung bleibt)
NETPORT	Vorher	10000, 10001, 10002, 10003, 10004, 10005, 10006, 10007, 10008, 10009
	Nachher	10000, 10001, 10002, 10003, 10004, 10005, 10006, 10007, 10008, 10009 (Werkseinstellung bleibt)
CPRCE13	Vorher	0
	Nachher	2
COMDEV	Vorher	RS232, , , , , ,
	Nachher	RS232, , OPT13, , , , ,
NETMODE	Vorher	1, 1, 1, 1, 1, 1, 1, 1, 1
	Nachher	1, 1, 0, 1, 1, 1, 1, 1, 1
NETHSTIP	Vorher	192.168.0.2, 192.168.0.3, 192.168.0.4, 192.168.0.5, 192.168.0.6, 192.168.0.7, 192.168.0.8, 192.168.0.9, 192.168.0.10
	Nachher	192.168.0.2, 192.168.0.3, <u>192.168.0.2</u> , 192.168.0.5, 192.168.0.6, 192.168.0.7, 192.168.0.8, 192.168.0.9, 192.168.0.10

Tab. 2-8: Verwendete Einstellungen

Stellen Sie als nächstes im Menü „Netzwerkverbindungen“ die IP-Adresse des Personalcomputers 192.168.0.2 ein.

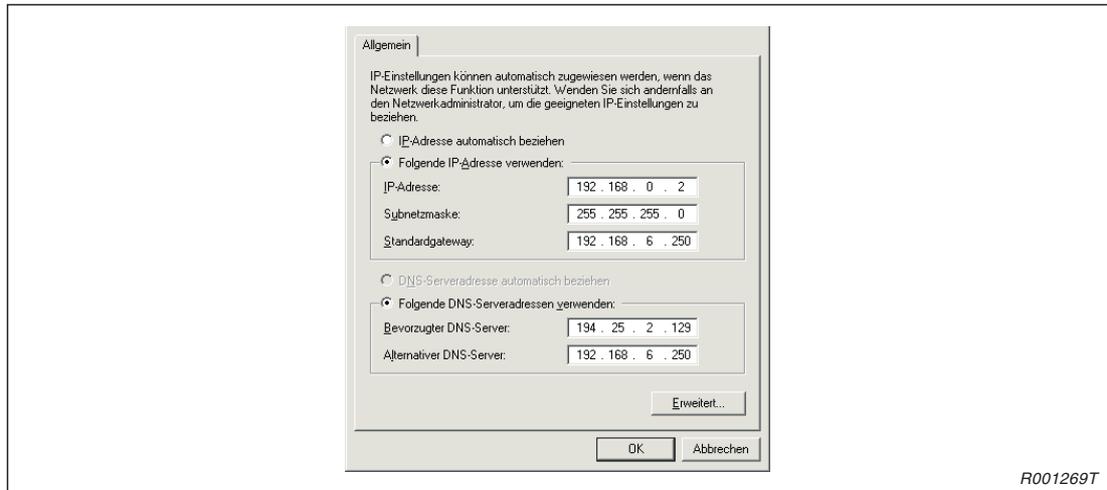


Abb. 2-8: Einstellung der IP-Adresse des Personalcomputers

Die IP-Adresse des Personalcomputers wird in Windows im Menü „Netzwerkverbindungen“ eingestellt. Die Menüs unterscheiden sich in Abhängigkeit der Windows-Version. Weitere Hinweise entnehmen Sie dem Handbuch Ihrer Windows-Version. Die Abbildung oben zeigt das Menü in Windows XP.

Hinweise zur Anwendung der PC-Support-Software entnehmen Sie dem Handbuch der PC-Support-Software.

2.5.6 Beispiel zur Parametereinstellung im externen Echtzeit-Steuerungs-Betrieb

Dieses Beispiel zeigt die Einstellung der Parameter im externen Echtzeit-Steuerungs-Betrieb des Roboter-Steuergeräts.

Parameter	Einstellung
IP-Adresse des Roboter-Steuergeräts	192.168.0.1
IP-Adresse des Personalcomputers	192.168.0.2
Schnittstellenummer des Roboter-Steuergeräts	10000

Tab. 2-9: Bedingungen

Stellen Sie die Parameter des Roboter-Steuergeräts wie folgt ein. Bei Verwendung der Werkseinstellung bleiben die Parameter unverändert.

Einzustellender Parameter	Vor und nach der Änderung	Einstellung
NETIP	Vorher	192.168.0.1
	Nachher	192.168.0.1 (Werkseinstellung bleibt)
NETPORT	Vorher	10000, 10001, 10002, 10003, 10004, 10005, 10006, 10007, 10008, 10009
	Nachher	10000, 10001, 10002, 10003, 10004, 10005, 10006, 10007, 10008, 10009 (Werkseinstellung bleibt)
MXTTOUT	Vorher	-1
	Nachher	-1 (Werkseinstellung bleibt)
MXTCOM1 ^①	Vorher	192.168.0.2
	Nachher	192.168.0.2 (Werkseinstellung bleibt)

Tab. 2-10: Verwendete Einstellungen

^① Der Parameter wird nur bei der Programmiersprache MOVEMASTER-COMMAND verwendet.

Stellen Sie als nächstes im Menü „Netzwerkverbindungen“ die IP-Adresse des Personalcomputers 192.168.0.2 ein.

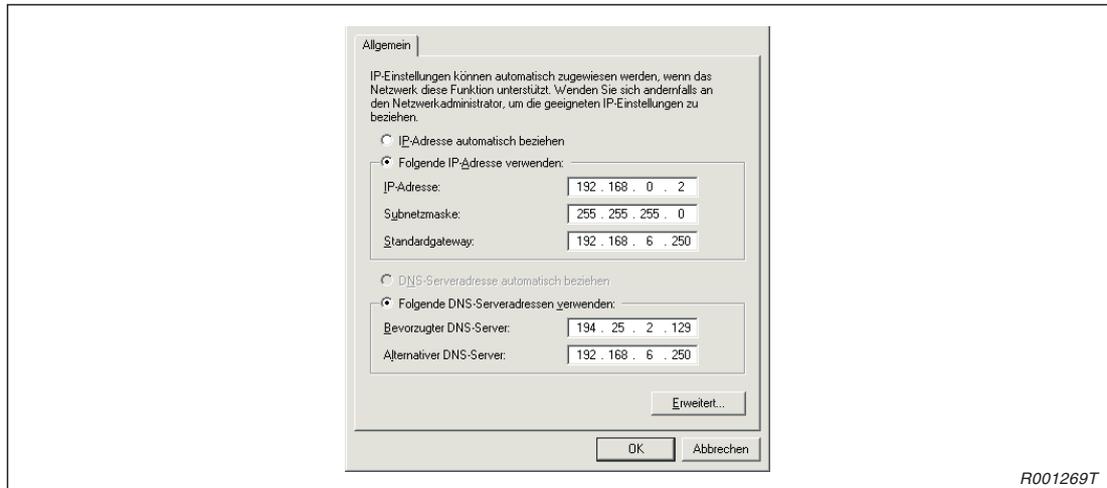


Abb. 2-9: Einstellung der IP-Adresse des Personalcomputers

Die IP-Adresse des Personalcomputers wird in Windows im Menü „Netzwerkverbindungen“ eingestellt. Die Menüs unterscheiden sich in Abhängigkeit der Windows-Version. Weitere Hinweise entnehmen Sie dem Handbuch Ihrer Windows-Version. Die Abbildung oben zeigt das Menü in Windows XP.

Hinweise zur Anwendung der PC-Support-Software entnehmen Sie dem Handbuch der PC-Support-Software.

2.6 Anschlussprüfung

- Ist die Teaching Box korrekt angeschlossen?
- Sind das Roboter-Steuergerät und der Personalcomputer korrekt über das Ethernet-Kabel verbunden (siehe Abschn. 2.4.2)?
- Wird das richtige Ethernet-Kabel verwendet?
(Der Anschluss an einen Hub erfolgt über ein Straight-Through-Kabel, während bei der direkten Verbindung von Personalcomputer und Steuergerät ein Crossover-Kabel verwendet wird.)
- Sind die Parameter des Roboter-Steuergerätes richtig eingestellt (siehe Abschn. 2.5)?
- Ist die Spannungsversorgung des Roboter-Steuergerätes nach der Einstellung der Parameter aus- und wieder eingeschaltet worden?

2.6.1 Überprüfung der Verbindung mit dem PING-Befehl

Folgende Abbildung zeigt die Überprüfung der Verbindung mit dem PING-Befehl. Starten Sie dazu die MS-DOS-Eingabeaufforderung und geben Sie den PING-Befehl mit der IP-Adresse des Roboter-Steuergerätes ein. Ist die Verbindung fehlerfrei erscheint die Meldung „Antwort von ...“. Bei einem Kommunikationsfehler erscheint die Meldung „Zeitüberschreitung der Anforderung“.

```
C:\>ping 10.97.11.31 ← IP-Adresse des Roboter-Steuergerätes

Ping wird ausgeführt für 10.97.11.31 mit 32 Bytes Daten:

Antwort von 10.97.11.31: Bytes=32 Zeit<1ms TTL=128

Ping-Statistik für 10.97.11.31:
    Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0 (0% Verlust),
    Ca. Zeitangaben in Millisek.:
        Minimum = 0ms, Maximum = 0ms, Mittelwert = 0ms

C:\>
```

Abb. 2-10: Überprüfung der Verbindung mit dem PING-Befehl

3 Betrieb

In diesem Kapitel werden die drei grundlegenden Funktionen der Ethernet-Schnittstellenkarte für ein System erklärt, in dem das Roboter-Steuergerät und der Netzwerk-Computer über ein Crossover-Kabel direkt miteinander verbunden sind.

- Kommunikationsfunktion (siehe Abschn. 3.1)
- Data-Link-Funktion (siehe Abschn. 3.2)
- externe Echtzeit-Steuerungs-Funktion (siehe Abschn. 3.3)

3.1 Kommunikationsfunktion des Steuergeräts

In diesem Abschnitt wird die Vorgehensweise zur Kommunikation des Roboter-Steuergeräts mit dem Personalcomputers erläutert.

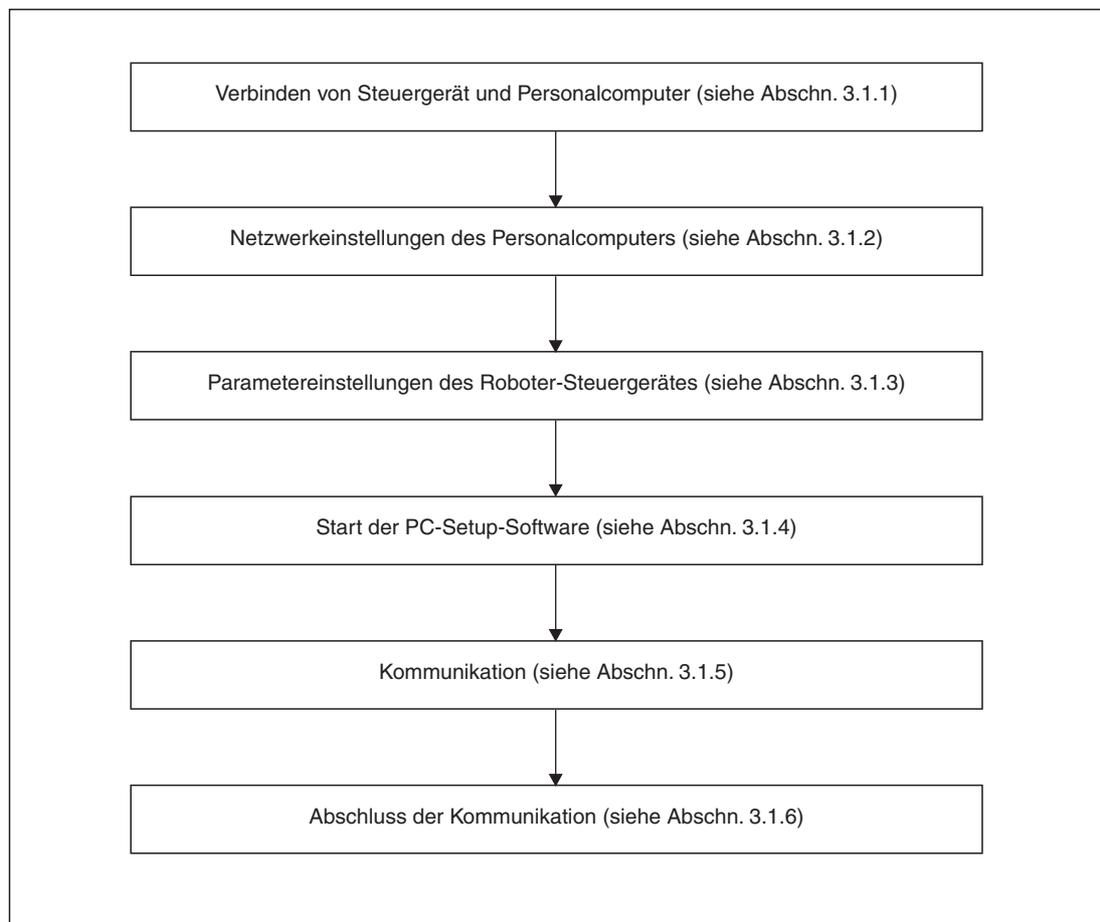


Abb. 3-1: Kommunikation zwischen Roboter-Steuergerät und Personalcomputer

3.1.1 Verbinden von Steuergerät und Personalcomputer

Verbinden Sie das Roboter-Steuergerät und den Personalcomputer über ein 10Base-T-Crossover-Kabel (siehe auch Abschn. 2.4.2).

3.1.2 Netzwerkeinstellungen des Personalcomputers

Nehmen Sie die Netzwerkeinstellungen des Personalcomputers wie in Abschn. 2.5.3 beschrieben vor.

3.1.3 Parametereinstellungen des Roboter-Steuergerätes

Schalten Sie die Spannungsversorgung des Steuergerätes ein und setzen Sie die Parameter wie nachfolgend beschrieben. Bei Verwendung der Werkseinstellungen müssen die Parameter nicht eingestellt werden.

Einzustellender Parameter	Vor und nach der Änderung	Einstellung
NETIP	Vorher	192.168.0.1
	Nachher	192.168.0.1 (Werkseinstellung bleibt)
NETPORT	Vorher	10000, 10001, 10002, 10003, 10004, 10005, 10006, 10007, 10008, 10009
	Nachher	10000, 10001, 10002, 10003, 10004, 10005, 10006, 10007, 10008, 10009 (Werkseinstellung bleibt)

Abb. 3-2: Parametereinstellungen des Roboter-Steuergerätes

HINWEISE

Die Änderung einer Parametereinstellung wird erst nach Aus- und Wiedereinschalten der Spannungsversorgung des Steuergerätes aktiv.

Die detaillierte Vorgehensweise zum Einstellen der Parameter finden Sie in der Bedienungs- und Programmieranleitung der Roboter-Steuergeräte.

3.1.4 Start der PC-Setup-Software

Starten Sie die PC-Support-Software und nehmen Sie die Kommunikationseinstellungen vor.

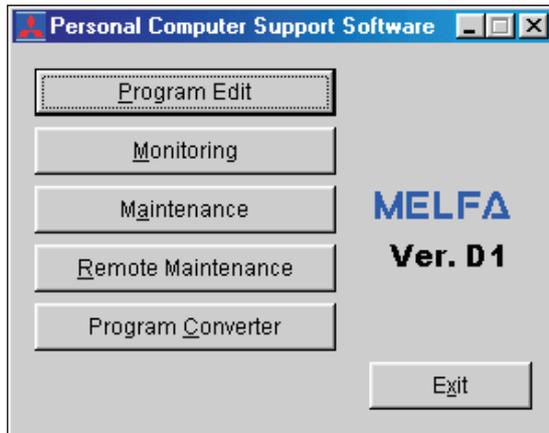


Abb. 3-3:
PC-Support-Software

R001272T

Wählen Sie das TCP/IP-Protokoll und stellen Sie die IP-Adresse 192.168.0.1 ein.

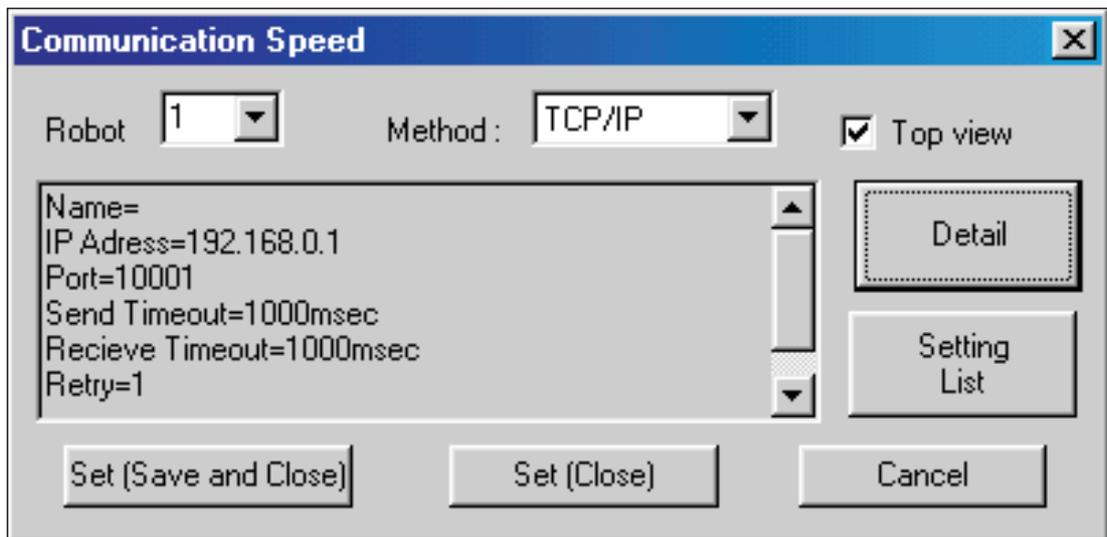


Abb. 3-4: Kommunikationseinstellungen

HINWEIS

Eine detaillierte Beschreibung der PC-Support-Software finden Sie im Handbuch der Software.

3.1.5 Kommunikation

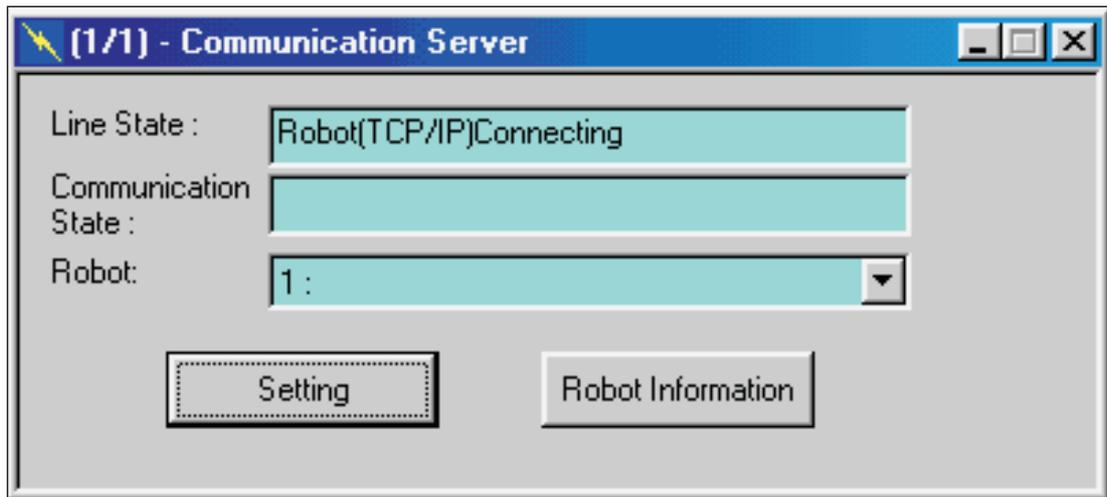


Abb. 3-5: Kommunikation mit der PC-Support-Software

Die mit dem TCP/IP-Protokoll über die Ethernet-Schnittstelle ausgeführte Kommunikation entspricht der Kommunikation über eine RS232C-Schnittstelle. Eine detaillierte Beschreibung der PC-Support-Software finden Sie im Handbuch der Software.

HINWEISE

┆ Erfolgt kein Verbindungsaufbau, überprüfen Sie die in Abschn. 2.6 aufgeführten Punkte.

┆ Beim Aus- und Wiedereinschalten der Spannungsversorgung des Roboter-Steuergerätes wird die Verbindung unterbrochen und die Kommunikation abgebrochen. Beenden Sie in diese Fall die Software auf dem Personalcomputer und starten Sie diese neu.

3.1.6 Beenden der Kommunikation

- ① Betätigen Sie die Schaltfläche „Fenster schließen “, um das Fenster zu schließen.
- ② Bestätigen Sie die Abfrage „Okay to quit communication server“ mit „Ja/Yes“, um die Kommunikation zu beenden.

3.2 Data-Link-Funktion

In diesem Abschnitt werden die Schritte zum Starten des in Anhang A.1.1 dargestellten Beispielprogramms zur Ausführung des Data-Link-Betriebs erläutert. Die Kommunikation erfolgt durch die direkte Verbindung des Steuergerätes mit dem Personalcomputer über ein Crossover-Kabel.

Ab Software-Version H7 kann das Steuergerät auch als Client definiert werden.

Protokoll	Software-Version	
	H6 oder älter	Ab H7
TCP/IP-Verbindung	Steuergerät = ausschließliche Server Personalcomputer = ausschließliche Client	Steuergerät = Server Personalcomputer = Client
		Steuergerät = Client Personalcomputer = Server

Tab. 3-1: Festlegung von Server und Client in Abhängigkeit der Software-Version

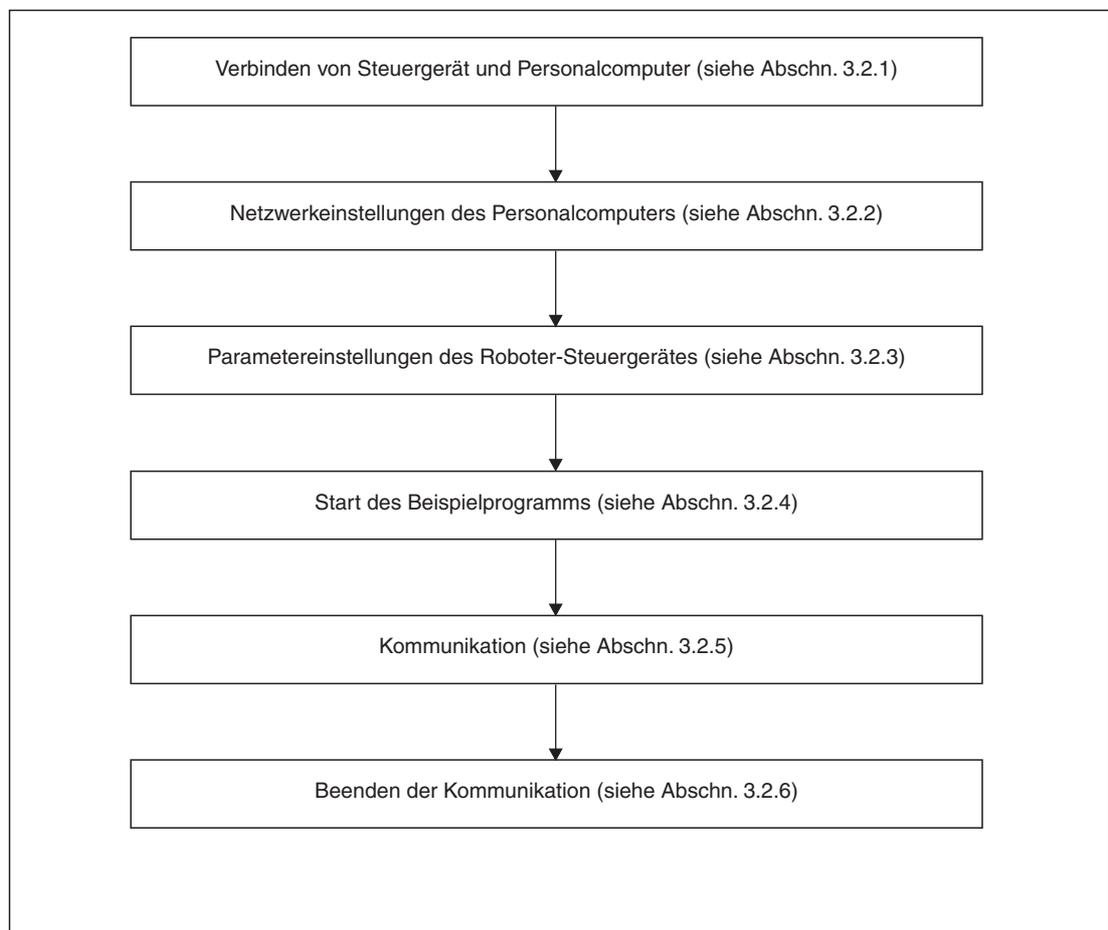


Abb. 3-6: Kommunikation zwischen Roboter-Steuergerät und Personalcomputer

3.2.1 Verbinden von Steuergerät und Personalcomputer

Verbinden Sie das Roboter-Steuergerät und den Personalcomputer über ein 10Base-T-Crossover-Kabel (siehe auch Abschn. 2.4).

3.2.2 Netzwerkeinstellungen des Personalcomputers

Nehmen Sie die Netzwerkeinstellungen entsprechend der Systemkonfiguration auf eine der beiden Arten vor. (Das Steuergerät ist entweder Server oder Client.)

- Parametereinstellung im Data-Link-Betrieb, wenn das Steuergerät der Server ist (siehe Abschn. 2.5.4)
- Parametereinstellung im Data-Link-Betrieb, wenn das Steuergerät der Client ist (siehe Abschn. 2.5.5)

3.2.3 Einstellung der Parameter des Steuergeräts

In Abhängigkeit davon, ob das Steuergerät in der TCP/IP-Verbindung als Server oder als Client arbeitet, sind die Einstellungen der Parameter unterschiedlich.

Schalten Sie die Spannungsversorgung des Steuergerätes ein und setzen Sie die Parameter wie unten beschrieben. Die Parameter NETIP und NETPORT müssen bei Verwendung der Werkseinstellungen nicht eingestellt werden. Schalten Sie die Spannungsversorgung nach der Einstellung der Parameter aus und wieder ein, um die neuen Werte zu aktivieren. Eine detaillierte Beschreibung zur Einstellung der Parameter finden Sie in der Bedienungs- und Programmieranleitung der Roboter-Steuergeräte.

Einzustellender Parameter	Vor und nach der Änderung	Einstellung
NETIP	Vorher	192.168.0.1
	Nachher	192.168.0.1 (Werkseinstellung bleibt)
NETPORT	Vorher	10000, 10001, 10002, 10003, 10004, 10005, 10006, 10007, 10008, 10009
	Nachher	10000, 10001, 10002, 10003, 10004, 10005, 10006, 10007, 10008, 10009 (Werkseinstellung bleibt)
CPCRE13	Vorher	0
	Nachher	2
COMDEV	Vorher	RS232, , , , , ,
	Nachher	RS232, , OPT13, , , , ,

Tab. 3-2: Einstellung der Parameter des Steuergeräts (Steuergerät = Server)

Einzustellender Parameter	Vor und nach der Änderung	Einstellung
NETIP	Vorher	192.168.0.1
	Nachher	192.168.0.1 (Werkseinstellung bleibt)
NETPORT	Vorher	10000, 10001, 10002, 10003, 10004, 10005, 10006, 10007, 10008, 10009
	Nachher	10000, 10001, 10002, 10003, 10004, 10005, 10006, 10007, 10008, 10009 (Werkseinstellung bleibt)
CPCRE13	Vorher	0
	Nachher	2
COMDEV	Vorher	RS232, , , , , ,
	Nachher	RS232, , OPT13, , , , ,
NETMODE	Vorher	1, 1, 1, 1, 1, 1, 1, 1, 1
	Nachher	1, 1, 0, 1, 1, 1, 1, 1, 1
NETHSTIP	Vorher	192.168.0.2, 192.168.0.3, 192.168.0.4, 192.168.0.5, 192.168.0.6, 192.168.0.7, 192.168.0.8, 192.168.0.9, 192.168.0.10
	Nachher	192.168.0.2, 192.168.0.3, <u>192.168.0.2</u> , 192.168.0.5, 192.168.0.6, 192.168.0.7, 192.168.0.8, 192.168.0.9, 192.168.0.10

Tab. 3-3: Einstellung der Parameter des Steuergeräts (Steuergerät = Client)

3.2.4 Start des Beispielprogramms

Das Beispielprogramm zeigt beispielhaft den Aufbau einer Datenverbindung zwischen dem Steuergerät und dem Personalcomputer. Als Schnittstelle wird COM3 verwendet.

- ① Erstellen Sie das Roboterprogramm mit der Teaching Box oder der PC-Support-Software und speichern Sie es unter einem eindeutigen Programmnamen ab. Als Programmiersprache kann MELFA-BASIC IV oder MOVEMASTER-COMMAND verwendet werden. Werksseitig ist die Programmiersprache MELFA-BASIC IV voreingestellt. Die Auswahl der Programmiersprache erfolgt mit Hilfe des Parameters RLNG. Eine detaillierte Beschreibung finden Sie in der Bedienungs- und Programmieranleitung zu den Robotersteuergerten.

Die Programmierung in MOVEMASTER-COMMAND ist nur bei einigen Robotermodellen möglich (RV-1A, RV-2AJ usw.). Andere Robotermodelle sind nur in MELFA-BASIC IV programmierbar.

Roboterprogramm

Beispiel in MELFA-BASIC IV

10	OPEN "COM3:" AS #1	Öffnet COM3 als Datei Nr. 1
20	PRINT #1,"START"	Sendet die Zeichenkette START
30	INPUT #1,DATA	Wartet auf den Empfang eines Werts zur Übertragung in die Variable DATA
40	IF DATA<0 THEN GOTO 70	Sprung zur Zeile 70 und Ende des Programms, falls der Wert in DATA negativ ist
50	PRINT #1,"DATA=";DATA	Gibt die Zeichenkette DATA= und den Wert von DATA aus
60	GOTO 30	Sprung zur Zeile 30 und erneuter Schleifendurchlauf
70	PRINT #1,"END"	Gibt die Zeichenkette END aus
80	END	Programmende

Beispiel in MOVEMASTER-COMMAND

10	OPN 1,3	Öffnet COM3 als Datei Nr. 1
20	SC \$1,"START"	Zeichenkette OK in Zeichenkette Nummer 1 laden
25	CR \$1,1	Sendet die Zeichenkette START
30	INP 1,1,0	Wartet auf den Empfang eines Werts in Zähler 1
40	CP 1	Daten vom Zähler 1 in das interne Register laden
45	SM 0,70	Sprung zur Zeile 70 und Ende des Programms, falls der Wert negativ ist
50	CR 1,1	Gibt den Wert des Zählers 1 aus
60	GT 30	Sprung zur Zeile 30 und erneuter Schleifendurchlauf
70	SC \$1,"END"	Zeichenkette END in Zeichenkette Nummer 1 laden
75	CR \$1,1	Gibt die Zeichenkette END aus
80	ED	Programmende

- ② Starten Sie das Datenübertragungsprogramm des Personalcomputers. Das Programm finden Sie in Anhang A.1.1. Speichern Sie das Programm unter dem Namen sample.exe ab.
- Starten Sie den Windows-Explorer und doppelklicken Sie auf die Datei sample.exe. Stellen Sie die IP-Adresse und die Schnittstellenummer ein. Aktivieren Sie das Kontrollfeld „Connection“ und öffnen Sie den Kommunikationskanal zum Roboter-Steuergerät. Ist die Schaltfläche „Send“ nicht aktiviert, überprüfen Sie ob die IP-Adresse mit der Einstellung des Parameters NETIP des Steuergerätes übereinstimmt.
- Ist die Schaltfläche „Send“ noch immer nicht aktiviert, überprüfen Sie die Anschlüsse (siehe Abschn. 2.6) oder starten Sie das Steuergerät und die Datei sample.exe erneut.

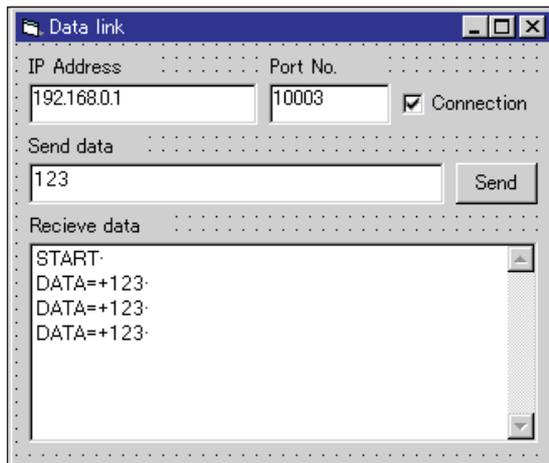


Abb. 3-7:
Das Menü Data-Link

R001275T

- ③ Starten Sie das Roboterprogramm. Betätigen Sie die START-Taste auf dem Bedienfeld des Steuergerätes und starten Sie das Roboterprogramm.

3.3 Externe Echtzeit-Steuerfunktion

In diesem Abschnitt werden die Schritte zum Starten des in Anhang A.1.2 dargestellten Beispielprogramms zur Ausführung der externen Echtzeit-Steuerfunktion erläutert. Die Kommunikation erfolgt durch die direkte Verbindung des Steuergerätes mit dem Personalcomputer über ein Crossover-Kabel.

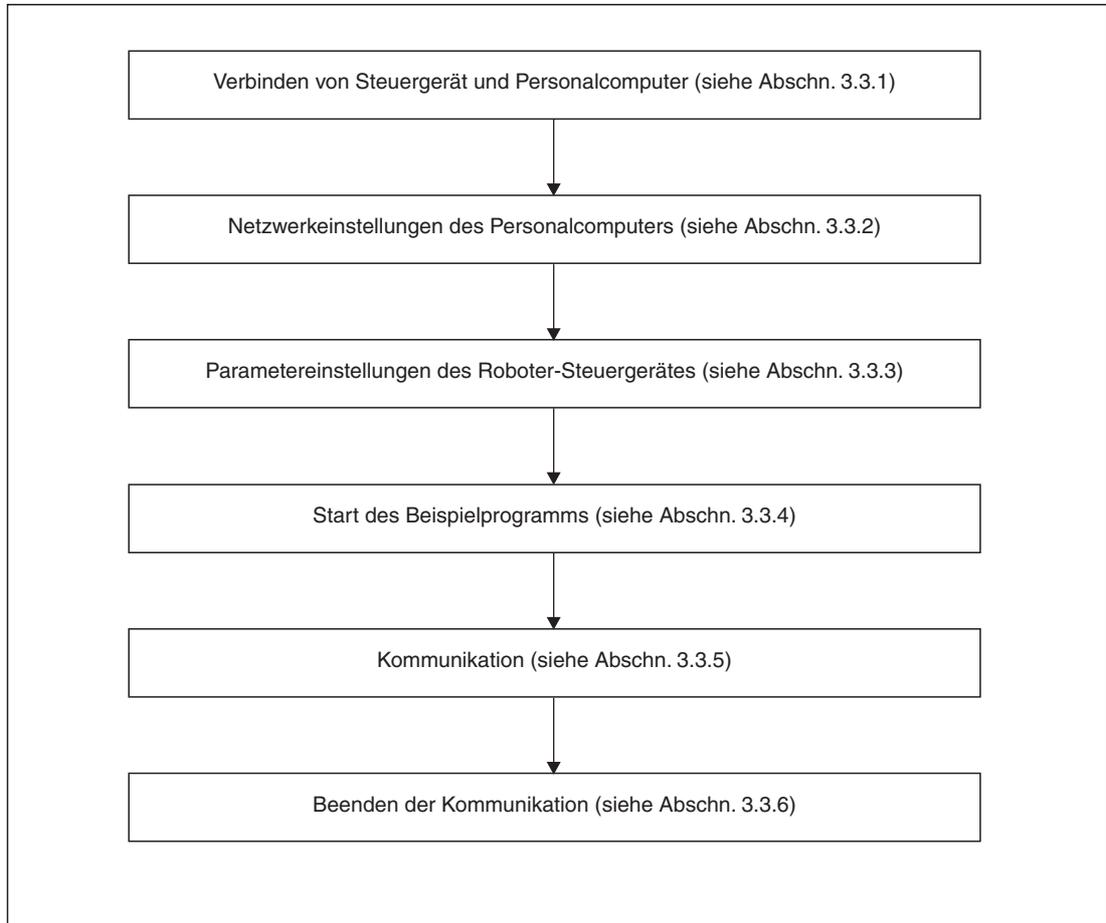


Abb. 3-8: Kommunikation zwischen Roboter-Steuergerät und Personalcomputer

3.3.1 Verbinden von Steuergerät und Personalcomputer

Verbinden Sie das Roboter-Steuergerät und den Personalcomputer über ein 10Base-T-Crossover-Kabel (siehe auch Abschn. 2.4).

3.3.2 Netzwerkeinstellungen des Personalcomputers

Nehmen Sie die Netzwerkeinstellungen des Personalcomputers wie in Abschn. 2.5.5 beschrieben vor.

3.3.3 Einstellung der Parameter des Steuergeräts

Schalten Sie die Spannungsversorgung des Steuergerätes ein und setzen Sie die Parameter wie nachfolgend beschrieben. Bei Verwendung der Werkseinstellungen müssen die Parameter nicht eingestellt werden. Schalten Sie die Spannungsversorgung nach der Einstellung der Parameter aus und wieder ein, um die neuen Werte zu aktivieren. Eine detaillierte Beschreibung zur Einstellung der Parameter finden Sie in der Bedienungs- und Programmieranleitung der Roboter-Steuergeräte.

Einzustellender Parameter	Vor und nach der Änderung	Einstellung
NETIP	Vorher	192.168.0.1
	Nachher	192.168.0.1 (Werkseinstellung bleibt)
NETPORT	Vorher	10000, 10001, 10002, 10003, 10004, 10005, 10006, 10007, 10008, 10009
	Nachher	10000, 10001, 10002, 10003, 10004, 10005, 10006, 10007, 10008, 10009 (Werkseinstellung bleibt)
MXTTOUT	Vorher	-1
	Nachher	-1 (Werkseinstellung bleibt)
MXTCOM1 ^①	Vorher	192.168.0.2
	Nachher	192.168.0.2 (Werkseinstellung bleibt)

Tab. 3-4: Einstellung der Parameter des Steuergeräts

- ^① Der Parameter wird nur bei der Programmiersprache MOVEMASTER-COMMAND verwendet.

3.3.4 Start des Beispielprogramms

Das Beispielprogramm zeigt den Aufbau einer Echtzeit-Kommunikation zwischen dem Steuergerät und dem Personalcomputer. Die XYZ-Koordinaten der X-Achse oder die Gelenkdaten der J1-Achse werden vom Personalcomputer aus gesteuert.

- ① Erstellen Sie das Roboterprogramm mit der PC-Support-Software und speichern Sie es unter einem eindeutigen Programmnamen ab. Als Programmiersprache kann MELFA-BASIC IV oder MOVEMASTER-COMMAND verwendet werden. Werksseitig ist die Programmiersprache MELFA-BASIC IV voreingestellt. Die Auswahl der Programmiersprache erfolgt mit Hilfe des Parameters RLNG. Eine detaillierte Beschreibung finden Sie in der Bedienungs- und Programmieranleitung zu den Robotersteuergeräten. Die Programmierung in MOVEMASTER-COMMAND ist nur bei einigen Robotermodellen möglich (RV-1A, RV-2AJ usw.). Andere Robotermodelle sind nur in MELFA-BASIC IV programmierbar.

Roboterprogramm

Beispiel in MELFA-BASIC IV

10	OPEN "ENET: 192.168.0.2" AS #1	Öffnet die Kommunikationsleitung zur IP-Adresse des Personalcomputers als Ethernet-Verbindung in Datei Nr. 1
20	MOV P1	Grundposition P1 anfahren (Teachen Sie eine frei gewählte Position als P1.) Die XYZ-Koordinaten der aktuellen Position werden vom Steuergerät an den Personalcomputer übertragen.
30	MXT1,0	Bewegung entsprechend den Befehlsdaten aus Datei Nr. 1
40	MOV P1	Fahre die Grundposition P1 nach Beendigung des externen Steuermodus mit Gelenkinterpolation an
50	HLT	Roboter stoppen
60	END	Programmende

Beispiel in MOVEMASTER-COMMAND

10	MO 1	Grundposition P1 anfahren (Teachen Sie eine frei gewählte Position als P1.)
20	MXT 1,0	Bewegung entsprechend den Befehlsdaten vom Kommunikationsziel Nummer 1 Empfang der XYZ-Koordinaten vom Personalcomputer
30	MO 1	Fahre die Grundposition 1 nach Beendigung des externen Steuermodus mit Gelenkinterpolation an
40	HLT	Roboter stoppen
50	ED	Programmende

- ② Starten Sie das Roboterprogramm.
Betätigen Sie die START-Taste auf dem Bedienfeld des Steuergerätes und starten Sie das Roboterprogramm. Der Roboter fährt die Grundposition P1 an. Die externe Echtzeit-Steuerung erfolgt über den Befehl MXT.
- ③ Starten Sie das Beispielprogramm des Personalcomputers zur Ausführung der externen Echtzeit-Steuerfunktion.
Das Programm finden Sie in Anhang A.1.2. Speichern Sie das Programm unter dem Namen sample.exe ab.
Starten Sie den Windows-Explorer und doppelklicken Sie auf die Datei sample.exe.

3.3.5 Verfahren des Roboters

Geben Sie nach der Eingabeaufforderung des Beispielprogramms folgende Daten ein:

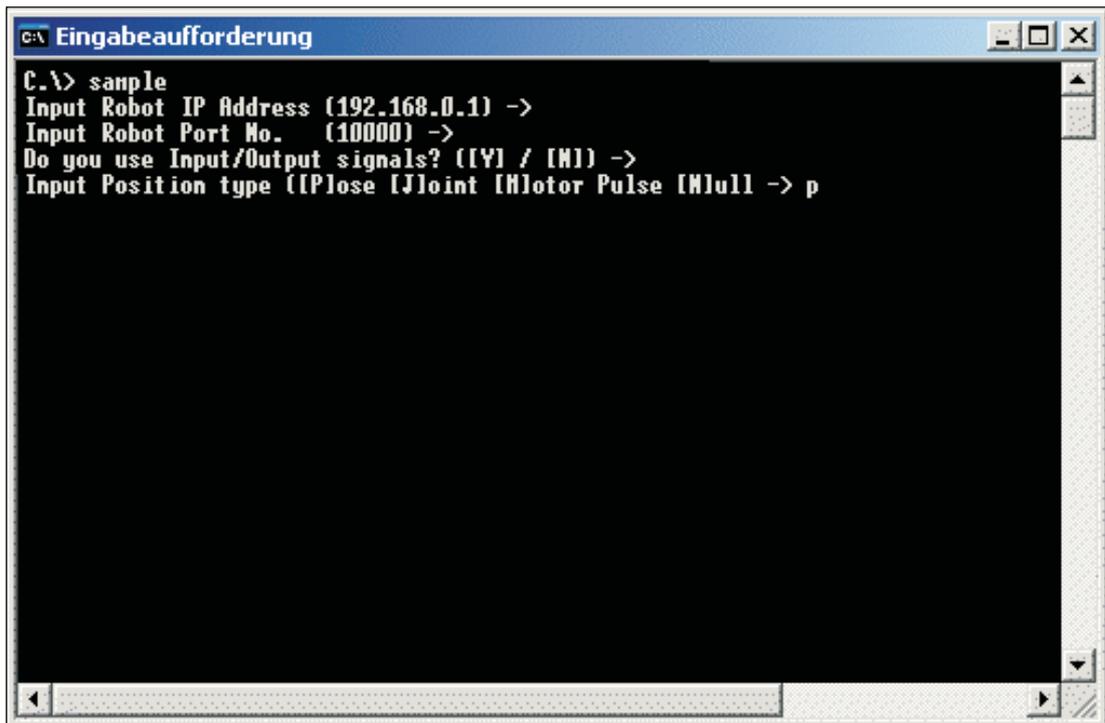
- IP-Adresse (192.168.0.1) des Knotenpunkts für das Roboter-Steuergerät
- Schnittstellennummer (10001)
- Befehlsdatentyp
- Datentyp der Überwachungsdaten (ab Software-Version H7)

Wählen Sie den Befehlsdatentyp entsprechend dem Argument des MXT-Befehls im Roboterprogramm. In folgender Tabelle sind die Tastenfunktionen aufgeführt. Detaillierte Informationen entnehmen Sie dem Beispielprogramm.

Taste	Funktion
Z oder X	Steuerung der Roboterbewegung
C	Der Befehlswert wird auf „0“ gesetzt und der Roboter stoppt.
D	Bei jeder Betätigung der MOVE-Taste ändert sich die Anzeige/Überwachungsdaten werden ausgeblendet.
ENTER	Beenden des MXT-Befehls

Tab. 3-5: Tastenfunktionen

Wird die Anzahl der Anweisungen zu groß oder versucht, den Roboter außerhalb seines Bewegungsbereichs zu bewegen, erfolgt eine Fehlermeldung und der Roboter stoppt. Starten Sie in diesem Fall das Programm erneut.



```

C:\> sample
Input Robot IP Address (192.168.0.1) ->
Input Robot Port No. (10000) ->
Do you use Input/Output signals? ([Y] / [N]) ->
Input Position type ([P]lose [J]oint [M]otor Pulse [W]ull -> p
  
```

Abb. 3-9: Dateneingabe

HINWEISE

Erfolgt kein Verbindungsaufbau, überprüfen Sie die in Abschn. 2.6 aufgeführten Punkte oder starten Sie das Steuergerät und die Datei sample.exe erneut.

Beim Aus- und Wiedereinschalten der Spannungsversorgung des Roboter-Steuergerätes wird die Verbindung unterbrochen und die Kommunikation abgebrochen. Beenden Sie in diese Fall die Software auf dem Personalcomputer und starten Sie diese neu.

3.3.6 Beenden der Kommunikation

- ① Rufen Sie den zyklischen Betrieb durch Betätigung der END-Taste auf dem Bedienfeld des Steuergeräts auf.
- ② Beenden Sie das Beispielprogramm auf dem Personalcomputer.
Durch Betätigung der ENTER-Taste beenden Sie den MXT-Befehl. Der Roboter kehrt zu seiner Ausgangsposition zurück und das Roboterprogramm wird unterbrochen. Das Beispielprogramm wird ebenfalls beendet.
- ③ Schalten Sie die Spannungsversorgung des Steuergerätes aus.

4 Funktionen

Dieses Kapitel beschreibt die Funktionen der Ethernet-Schnittstellenkarte.

4.1 Kommunikationsfunktion des Steuergerätes

Die vom Personalcomputer über das Netzwerk ausgeführte Kommunikation entspricht der Kommunikation der PC-Support-Software über eine RS232C-Schnittstelle.

Die PC-Support-Software unterstützt alle Funktionen wie das Laden und Speichern sowie die Zustandüberwachung usw. des Roboterprogramms. Die Kommunikation über das Ethernet erfolgt mit höherer Übertragungsgeschwindigkeit als über eine RS232C-Schnittstelle.

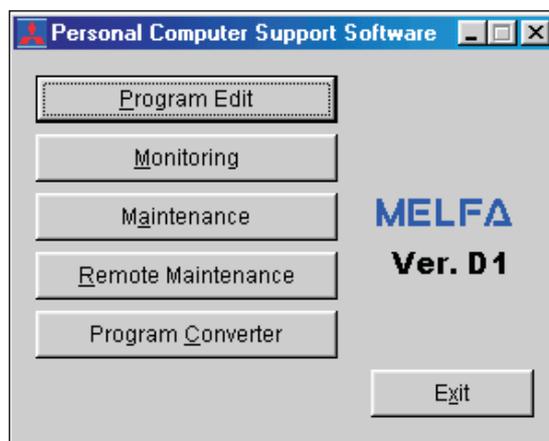


Abb. 4-1:
PC-Support-Software

R001272T

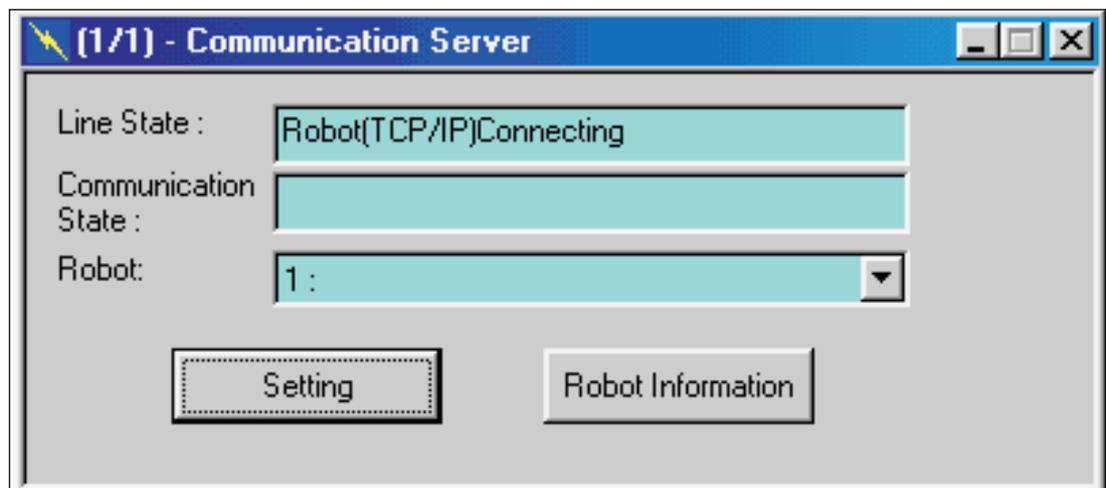


Abb. 4-2: Kommunikation mit der PC-Support-Software

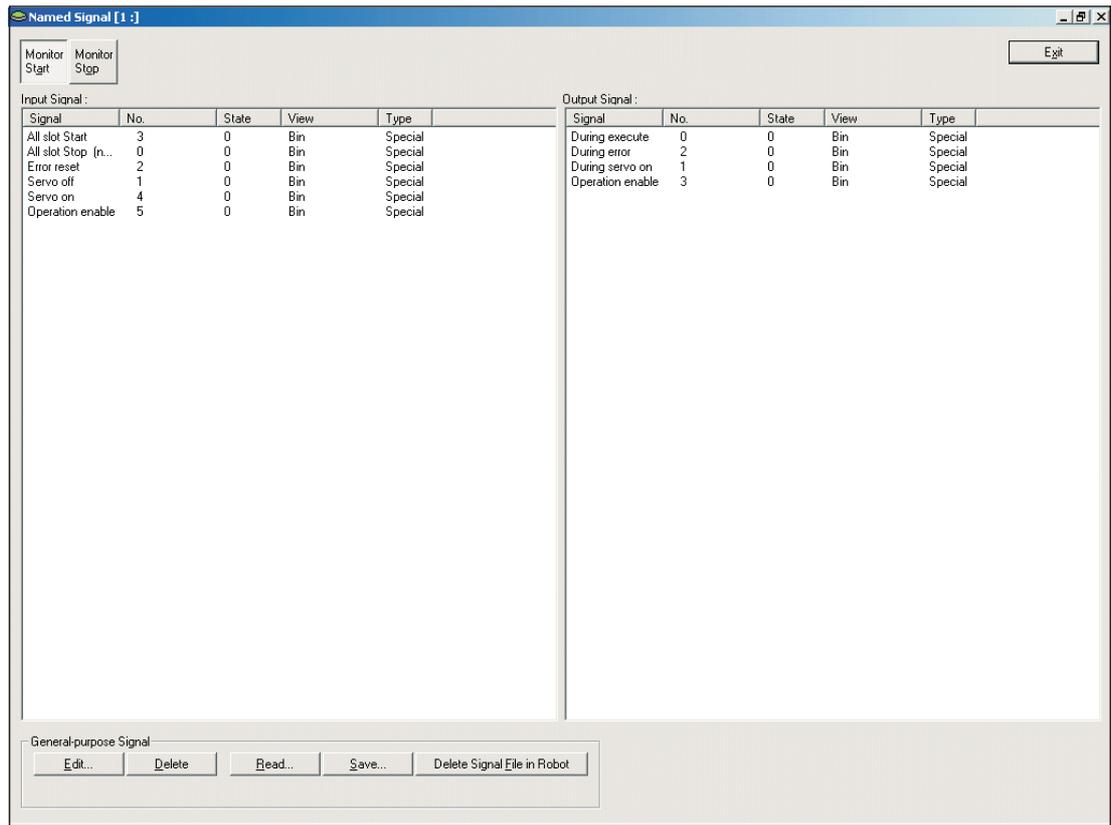


Abb. 4-3: Überwachungsmenü

4.2 Data-Link-Funktion

Wie bei der Kommunikation über eine RS232C-Schnittstelle können die Befehle OPEN, PRINT und INPUT der Roboter-Programmiersprache auch bei der Kommunikation über das Ethernet verwendet werden. Eine detaillierte Beschreibung der Programmiersprache finden Sie in der Bedienungs- und Programmieranleitung der Roboter-Steuergeräte.

Beispiel ▾

In diesem Beispiel wird die Schnittstelle Nummer 10003 als Kommunikationskanal definiert und als Datei #1 geöffnet. Dazu muss das Element 3 des Parameters COMDEV auf OPT13 und der Parameter NETPORT auf 10003 gesetzt werden.

100 OPEN "COM3:" AS #1	Öffnet COM3 als Datei Nr. 1
110 INPUT #1, C1\$	Einlesen
120 PRINT #1, "Reply", C1\$	Ausgabe
130 CLOSE #1	Kommunikationskanal schließen
140 HLT	Stopp

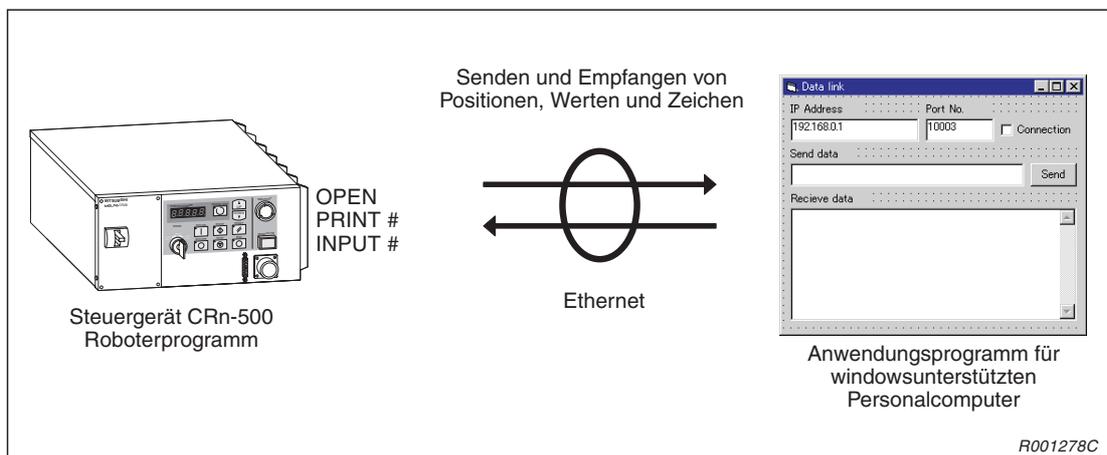


Abb. 4-4: Schema der Ethernet-Kommunikation

Die Data-Link-Funktion des Ethernets kann auf zwei verschiedene Arten genutzt werden:

- Das Roboter-Steuergerät wird als Server eingesetzt.
- Das Roboter-Steuergerät wird als Client eingesetzt.

Der Einsatz des Roboter-Steuergeräts als Client ist erst ab Software-Version H7 möglich. Folgende Abbildung zeigt die unterschiedlichen Konfigurationen eines Systems.

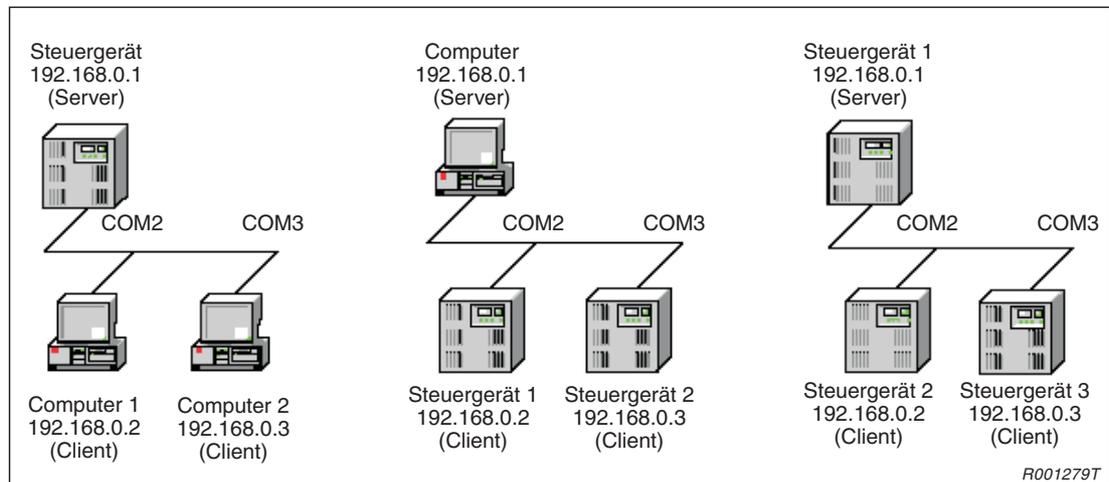


Abb. 4-5: Aufbau eines Ethernet-Systems

HINWEIS

An einen Kommunikationskanal COMn können nicht mehr als zwei Clients angeschlossen werden. Verwenden Sie bei Anschluss von mehr als zwei Clients weitere Kommunikationskanäle.

4.2.1 MELFA-BASIC-IV-Funktionalitäten

In diesem Abschnitt werden die Funktionalitäten der Roboter-Programmiersprache MELFA-BASIC IV erläutert. Die hier beschriebenen Funktionalitäten stehen ab Software-Version H7 des Roboter-Steuergerätes zur Verfügung. In Version H6 oder älteren Versionen können sie nicht verwendet werden. Weitere Informationen über die Kommunikationsbefehle OPEN, INPUT # und PRINT # finden Sie in der Bedienungs- und Programmieranleitung der Steuergeräte.

M_OPEN

Funktion: Dateistatus lesen

Die Variable zeigt an, ob eine Datei geöffnet ist oder nicht.

Eingabeformat

Ab Software-Version H7:

```
<Numerische Variable> = M_OPEN [( <Dateinummer> )]
```

<Numerische Variable>

Legt eine numerische Variable fest

<Dateinummer>

Gibt die Nummer eines mit dem OPEN-Befehl geöffneten Kommunikationskanals als Konstante im Bereich zwischen 1 und 8 an (Standardwert: 1)
Bei Angabe einer Dateinummer größer gleich 9 erfolgt eine Fehlermeldung.

Programmbeispiel

10	Client-Programm -----	
100	M1=0	
110	M_TIMER(1)=0	Timer auf „0“ zurücksetzen
120	OPEN "COM2:" AS #1	Kommunikationskanal öffnen
130	IF M_TIMER(1)>10000.0 THEN 240	Sprung nach 10 s
140	IF M_OPEN(1)<>1 THEN GOTO 120	Wiederholschleife bei fehlender Verbindung
145	DEF ACT 1,M_OPEN(1)=0 GOSUB 300	Überwacht den Betriebszustand des Servers über einen Interrupt
146	ACT 1=1	Überwachung starten
150	M1=M1+1	
160	IF M1<10 THEN C1\$="MELFA" ELSE C1\$="END"	Lädt die Zeichenkette „END“ in die Variable C1\$, nachdem 9-mal die Zeichenkette „MELFA“ geladen wurde
170	PRINT #1,C1\$	Überträgt die Zeichenkette
180	INPUT #1,C2\$	Empfang einer Zeichenkette
190	IF C1\$="END" THEN 210	Nach Senden der Zeichenkette „END“ Sprung zur Zeile 210
200	GOTO 150	Sprung zur Zeile 150
210	CLOSE #1	Schließt den Kommunikationskanal
220	HLT	Programm stoppen
230	END	Programmende
240	ERROR 9100	Ausgabe des Fehlers 9100, falls keine Verbindung zum Server aufgebaut werden kann
250	CLOSE #1	
260	HLT	
270	END	
280	ERROR 9101	Ausgabe des Fehlers 9101, falls der Server während der Verarbeitung nicht betriebsbereit ist
290	CLOSE #1	
300	HLT	
310	END	

Erläuterung

Die Variable M_OPEN wird gemeinsam mit dem OPEN-Befehl verwendet. Folgende Tabelle zeigt die Bedeutung und Werte für die im OPEN-Befehl festgelegten Dateien.

Dateityp	Beschreibung		Variablenwert
Datei	Die Variable zeigt an, ob eine Datei geöffnet ist oder nicht. Nach Ausführung des Befehls OPEN ist der Variablenwert „1“.		1: bereits geöffnet -1: nicht definierte Dateinummer (nicht geöffnet)
Kommunikationskanal RS232C	Die Variable zeigt den Status der RS232C-Kommunikationsleitung auf Seiten des Kommunikationspartners an. Es wird der Zustand des CTS-Signals angezeigt. Erfasst wird der ausgeschaltete Zustand und eine Kabelunterbrechung (Die Funktion kann nur nach Freigabe des RTS-Signals des Kommunikationspartners und bei Verwendung des Mitsubishi-Kabels RV-CAB genutzt werden.)		1: bereits verbunden (CTS-Signal EIN) 0: nicht verbunden (CTS-Signal AUS) -1: nicht definierte Dateinummer (nicht geöffnet)
Kommunikationskanal Ethernet	Zeigt eine Verbindung mit dem Kommunikationspartner an.	Bei Festlegung als Server	1: Client bereits verbunden 0: Client nicht verbunden -1: nicht definierte Dateinummer
		Bei Festlegung als Client	1: bereits mit dem Server verbunden (Verbindung aufgebaut) 0: nicht mit dem Server verbunden (Kein Verbindungsaufbau. Entspricht einem nicht betriebsbereiten Server, nachdem der Kommunikationskanal geöffnet wurde.) -1: nicht definierte Dateinummer (bei nicht geöffneter Datei oder bei geöffneter Datei und nicht betriebsbereitem Server)

Tab. 4-1: Variablenwert und Bedeutung

Steht in Beziehung zu folgenden Befehlen:

OPEN

Steht in Beziehung zu folgenden Parametern:

COMDEV, CPRCE□□, NETMODE

C_COM

Funktion: Parameter des Kommunikationskanals setzen

Die Variable legt die Parameter des Kommunikationskanals fest, der durch den OPEN-Befehl geöffnet wird. Er wird beim häufigen Wechsel des Kommunikationsziels verwendet. Die Variable ist eine Zeichenkettentyp und wird ausschließlich für einen Client mit installierter Ethernet-Schnittstellenkarte verwendet.

Eingabeformat

Ab Software-Version H7:

```
C_COM <Nummer des Kommunikationskanals> = "ETH: <IP-Adresse
                                         des Servers>
                                         [, <Schnittstellen-
                                         nummer>]"
```

ETH:	Zeigt das Ethernet als Kommunikationsziel an.
<Nummer des Kommunikationskanals>	Legt eine numerische Variable fest
<Dateinummer>	Gibt die Nummer eines mit dem OPEN-Befehl geöffneten Kommunikationskanals als Konstante im Bereich zwischen 1 und 8 an (Der Typ des Kommunikationskanals wird im Parameter COMDEV festgelegt.)
<IP-Adresse des Servers>	Gibt die IP-Adresse des Servers an (kann weggelassen werden)
<Schnittstellennummer>	Schnittstellennummer des Servers (Erfolgt keine Angabe wird der im Parameter NETPORT gesetzte Wert verwendet.)

Programmbeispiel

In einem Steckplatz des Steuergeräts ist eine Ethernetschnittstelle installiert und als drittes Element ist im Parameter COMDEV „OPT12“ gesetzt.

100 C_COM(2)="ETH:192.168.0.10,10010"	Einstellung der IP-Adresse des Ziel-Servers für die Kommunikation über COM2
110 OPEN "COM2:" AS #1	Öffnet den Kommunikationskanal zur Zieladresse 192.168.0.10 und zur Schnittstellenummer 10010.
120 IF M_OPEN(1)<>1 THEN 110	Wiederholschleife, falls kein Verbindungsaufbau zum Server möglich ist
130 PRINT #1, "HELLO"	Senden einer Zeichenkette
140 INPUT #1, C1\$	Empfang einer Zeichenkette
150 CLOSE #1	Kommunikationskanal schließen
160 C_COM(2)="ETH:192.168.0.11,10011"	Einstellung der IP-Adresse des Ziel-Servers für die Kommunikation über COM2
170 OPEN "COM2:" AS #1	Öffnet den Kommunikationskanal zur Zieladresse 192.168.0.11 und zur Schnittstellenummer 10011.
180 IF M_OPEN(1)<>1 THEN 170	Wiederholschleife, falls kein Verbindungsaufbau zum Server möglich ist
190 PRINT #1, C1\$	Senden einer Zeichenkette
200 INPUT #1, C2\$	Empfang einer Zeichenkette
210 CLOSE #1	Kommunikationskanal schließen
220 HLT	Programm stoppen
230 END	Programmende

Erläuterung

- Ist der Kommunikationspartner über die Parameter NETHSTIP und NETPORT bereits festgelegt worden und soll nicht geändert werden, ist eine Einstellung der Variablen nicht notwendig.
- Die Funktion steht nur für einen Client im Data-Link-Betrieb über die Ethernet-Schnittstelle zur Verfügung.
- Die Variable muss vor Ausführung des OPEN-Befehls gesetzt werden, da sie die Parameter für den Befehl festlegt.
- Beim Einschalten der Spannungsversorgung werden die in den Parametern NETSHIP und NETPORT eingestellten Werte verwendet. Die Verarbeitung der Variablen verändert die Werte bis zum Ausschalten der Spannungsversorgung. Beim Einschalten sind wieder die in den Parametern eingestellten Werte gültig.
- Bei einer Verarbeitung der Variablen nach Ausführung des OPEN-Befehls ändert sich der OPEN-Status nicht. In diesem Fall muss vor der Ausführung des OPEN-Befehls der Kommunikationskanal zuerst mit dem CLOSE-Befehl geschlossen werden.
- Ist die Syntax fehlerhaft, erfolgt bei Ausführung des Programms (nicht bei seiner Editierung) die Ausgabe einer Fehlermeldung.

Steht in Beziehung zu folgenden Parametern:

NETHSTIP, NETPORT

4.3 Externe Echtzeit-Steuerfunktion

Die Bewegungssteuerung des Roboters kann in Echtzeit-Zyklen Positionsbefehle wiederherstellen und die vorgegebene Position anfahren. Gleichzeitig kann eine Überwachung der Ein- und Ausgangssignale sowie deren Ausgabe erfolgen. Die Ausführung einer Echtzeit-Kommunikation (Befehle/Überwachung) ist über den MXT-Befehl möglich.

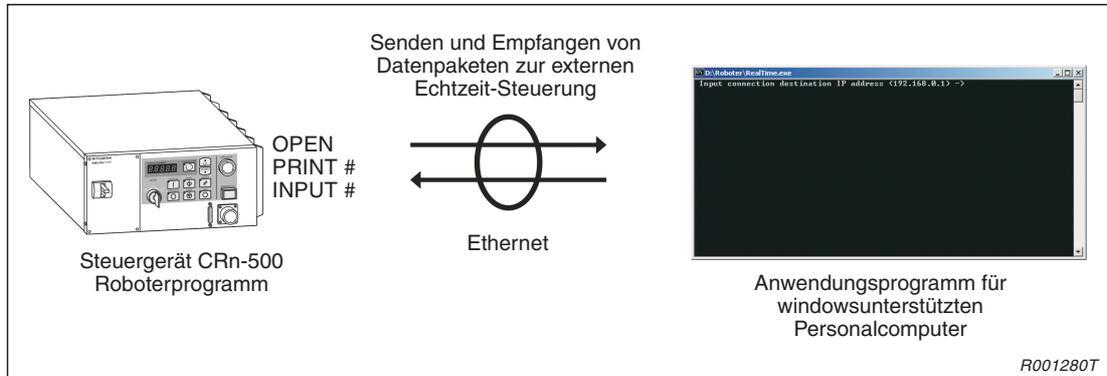


Abb. 4-6: Echtzeit-Kommunikation

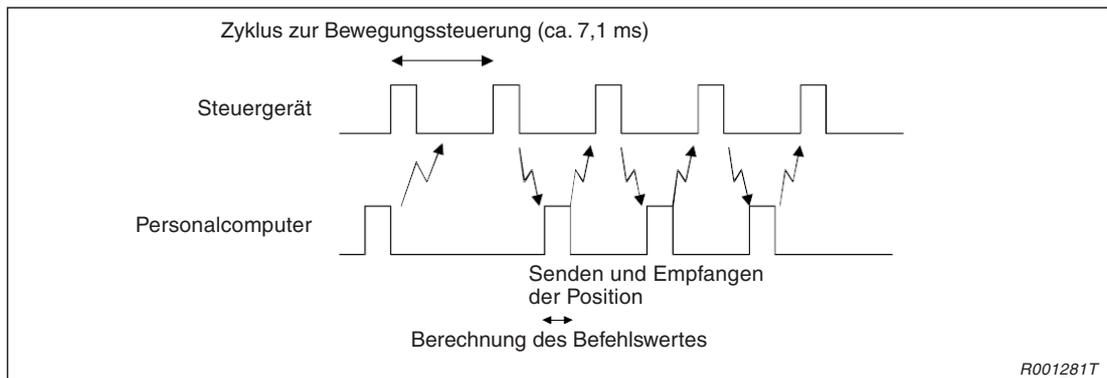


Abb. 4-7: Zyklus zur Bewegungssteuerung

Folgende Tabelle zeigt die Datentypen der Positionierbefehle, die während eines Zyklus zur Bewegungssteuerung für die Vorgabe der Sollposition vom Personalcomputer an das Steuergerät gesendet werden, und die Datentypen zur Überwachung des Roboters. Weitere Informationen über die Kommunikationsdaten finden Sie im Abschn. 4.3.1 und Abschn. 4.3.2.

Datentyp des Positionierbefehls	Datentyp der Überwachung
1. Kartesische Koordinaten	1. Kartesische Koordinaten
2. Gelenkdaten	2. Gelenkdaten
3. Motorimpulse	3. Motorimpulse
	4. Kartesische Koordinaten (Befehlswert nach Filterung)
	5. Gelenkdaten (Befehlswert nach Filterung)
	6. Motorimpulse (nach Filterung)
	7. Kartesische Koordinaten (Encoder-Impulse) ①
	8. Gelenkdaten (Encoder-Impulse) ①
	9. Motorimpulse (Encoder-Impulse) ①
	10. Aktueller Befehl (%) ①
	11. Aktuelle Rückmeldung (%) ①

Tab. 4-2: Datentypen der Positionierbefehle und der Überwachung

① Die Datentypen stehen ab Software-Version H7 zur Verfügung

Flussdiagramm der externen Echtzeit-Steuerfunktion

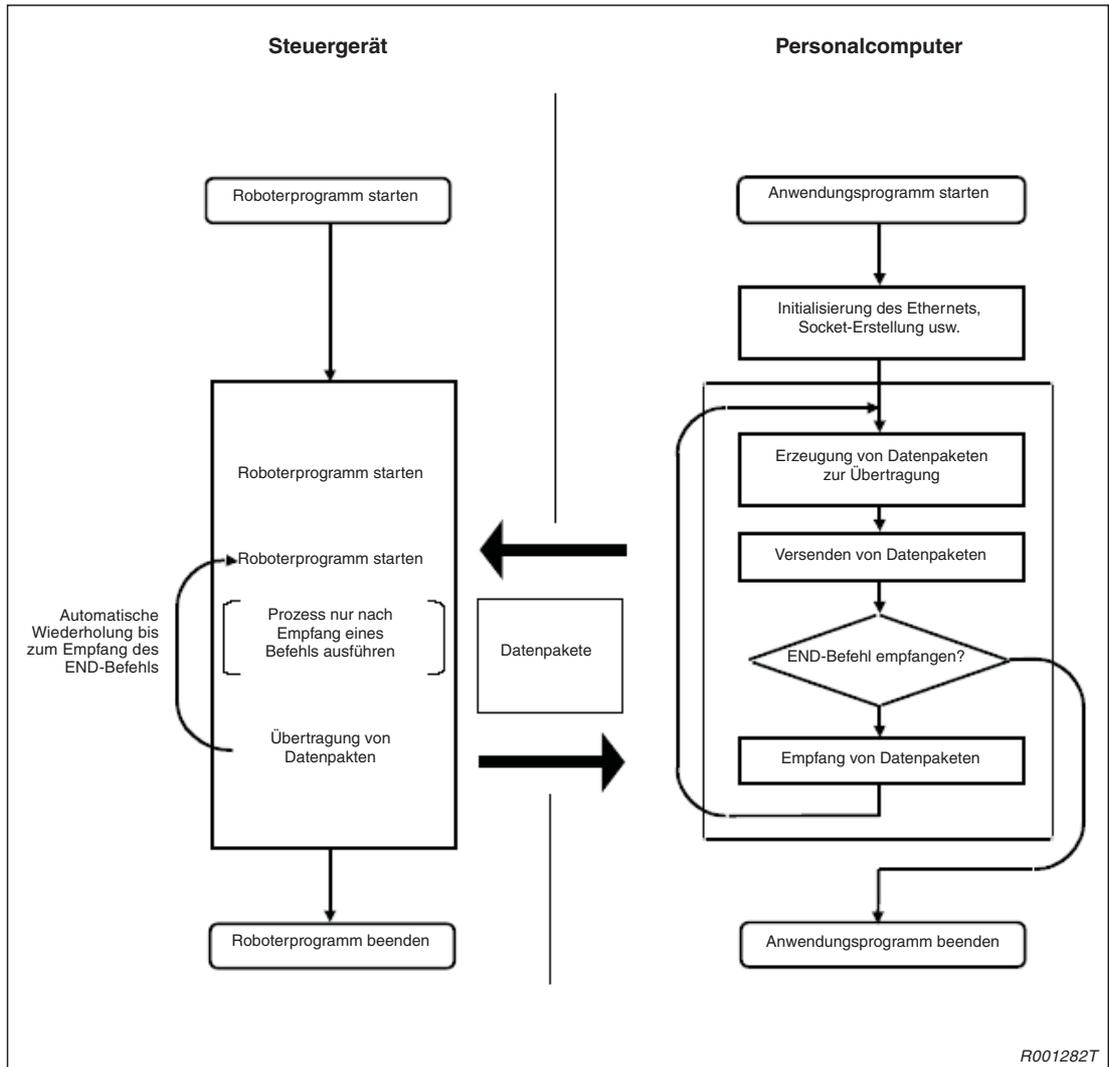


Abb. 4-8: Flussdiagramm der externen Echtzeit-Steuerfunktion

4.3.1 Erläuterung der Befehle

Die externe Echtzeit-Steuerfunktion kann entweder in MELFA-BASIC IV oder MOVEMASTER-COMMAND ausgeführt werden. Die Bedeutung der Befehlsparameter ist in beiden Programmiersprachen unterschiedlich.

Detaillierte Informationen über den Aufbau der Kommunikationsdaten, die mit dieser Funktion verwendet werden, finden Sie im Abschn. 4.3.2.

MXT (Move External)

Funktion: Bewegung des Roboters durch externe Steuerung

Die Absolutwert-Positionsdaten einer externen Steuerung werden in jedem Zyklus zur Bewegungssteuerung des Steuergerätes (7,1 ms) wieder hergestellt und der Roboter wird direkt verfahren.

Eingabeformat

MELFA-BASIC IV

```
MXT <Dateinummer>, <Datenyp der Positions-Antwortdaten>
      [, <Zeitkonstante des Filters>]
```

MOVEMASTER-COMMAND

```
MXT <Nummer des Kommunikationsziels>,
     <Datenyp der Positions-Antwortdaten>
     [, <Zeitkonstante des Filters>]
```

Für MELFA-BASIC IV

<Dateinummer>

Gibt die Nummer eines mit dem OPEN-Befehl geöffneten Kommunikationskanals im Bereich zwischen 1 und 8 an

Ist der Kommunikationskanal nicht über den OPEN-Befehl geöffnet worden, ist keine Kommunikation möglich und es erfolgt eine Fehlermeldung.

Daten, die von einer anderen Datenquelle als dem Kommunikationsziel empfangen wurden, werden ignoriert.

Für MOVEMASTER-COMMAND

<Nummer des Kommunikationsziels> Gibt die Nummer eines in den Parametern MXTCOM1 bis MXTCOM3 festgelegten Kommunikationsziels im Bereich zwischen 1 und 3 an
Legen Sie das Kommunikationsziel als IP-Adresse in den Parametern MXTCOM1 bis MXTCOM3 fest.
Erfolgt keine Festlegung, ist keine Kommunikation möglich.
Daten, die von einer anderen Datenquelle als dem Kommunikationsziel empfangen wurden, werden ignoriert.
Setzen Sie zum Beispiel zur Festlegung der IP-Adresse des Personalcomputers als Kommunikationsziel Nummer 1 Parameter MXTCOM1 auf 192.168.0.2.

Für MELFA-BASIC und MOVEMASTER-COMMAND

<Datentyp der Positions-Antwortdaten> Festlegung des Datentyps der Positionsdaten, die vom Personalcomputer empfangen werden
Die Festlegung kann als XYZ- oder Gelenkdaten oder als Motorimpulse erfolgen.
0: XYZ-Koordinaten
1: Gelenkdaten
2: Motorimpulse

<Zeitkonstante des Filters> Festlegung des Filterzeitkonstanten in ms
Bei einer Einstellung auf „0“ ist das Filter deaktiviert.
(Standardwert: 0)
Beim Empfang werden die Positionsdaten gefiltert, der Befehlswert wird geglättet und an den Servo ausgegeben.

Programmbeispiel**MELFA-BASIC IV**

10	OPEN "ENET: 192.168.0.2" AS #1	Öffne Ethernet-Kommunikationskanal zum Kommunikationsziel als Datei Nummer 1
20	MOV P1	Position P1 anfahren
30	MXT1,1,50	Verfahrbewegung mit externer Echtzeit-Steuerung und Filterzeitkonstante von 50 ms ausführen
40	MOV P1	Position P1 anfahren
50	HLT	Programm stoppen

MOVEMASTER-COMMAND

Stellen Sie die Ziel-IP-Adresse 192.168.0.2 des Personalcomputers für die Ethernet-Kommunikation in Parameter MXTCOM1 ein.

10	MO 1	Position 1 anfahren
20	MXT 1,1,50	Verfahrbewegung mit externer Echtzeit-Steuerung und Filterzeitkonstante von 50 ms ausführen
30	MO 1	Position 1 anfahren
40	HLT	Programm stoppen

Erläuterung

- Bei Ausführung des MXT-Befehls kann der Positionierungsbefehl zur Steuerung der Roboterbewegung von dem am Netzwerk angeschlossenen Personalcomputer zurückgeladen werden (Eins-zu-eins-Kommunikation).
- Ein Positionierbefehl kann während eines Zyklus (7,1 ms) zurückgeladen und ausgeführt werden.
- Ausführung des MXT-Befehls
 - Bei Ausführung des Befehls wird beim Steuergerät der Empfang von Befehlswerten freigegeben.
 - Empfängt das Steuergerät einen Befehlswert vom Personalcomputer, gibt es den empfangenen Befehlswert innerhalb des nächsten Steuerzyklus an den Servomotor aus.
 - Nach Absenden des Befehls zum Servo, wird die Information des Steuergeräts, wie z. B. die aktuelle Position, vom Steuergerät an den Personalcomputer übertragen.
 - Eine Antwort des Steuergeräts an den Personalcomputer erfolgt nur, wenn vom Personalcomputer ein Befehlswert an das Steuergerät übertragen wird.
 - Werden keine Daten empfangen, bleibt die aktuelle Position erhalten.
 - Beim Empfang des END-Befehls zum Abschluss der externen Echtzeit-Steuerfunktion wird der MXT-Befehl beendet.
 - Wird der Betrieb über das Bedienfeld durch ein externes Signal gestoppt, werden der MXT-Befehl und der Sende-/Empfangsvorgang bis zum nächsten Neustart unterbrochen.
- Die Zeit zur Ausführung eines externen Echtzeit-Steuerbefehls wird in Parameter MXTTOUT festgelegt.
- Ein frei wählbares (Kopfbit, Bitbreite) Ein-/Ausgangssignal kann zeitgleich mit den Positionsdaten gesendet oder empfangen werden.
- Zur Steuerung der Roboterbewegung innerhalb des Steuerzyklus muss der Personalcomputer über eine ausreichend hohe Rechengeschwindigkeit verfügen. Es wird ein Rechner mit Windows NT/2000 mit mindestens einem Pentium-II-Prozessor und 450 MHz empfohlen.

4.3.2 Datenpakete

Dieser Abschnitt erläutert den Aufbau des Datenpakets, das bei der externen Echtzeit-Steuerung verwendet wird. Das Datenpaket dient auch zur Befehlsvorgabe für die Positionierung und zur Überwachung. Der Inhalt des Datenpakets, das vom Personalcomputer zum Steuergerät (Befehlsvorgabe) übertragen wird, weicht vom Inhalt des Datenpakets vom Steuergerät zum Personalcomputer (Überwachung) ab.

In folgender Tabelle und in Abschn. A.1.2 wird der Aufbau der Datenpakete zur Kommunikation beschrieben. Der in der Tabelle verwendete Datentyp basiert auf der Programmiersprache C. Weiterhin sind die Datenpaket in Abhängigkeit des Software-Version des Steuergerätes in ein Datenpaket 1 und ein Datenpaket 2 unterteilt. Wählen Sie das Datenpaket entsprechend Ihrer Software-Version. Detaillierte Informationen zur Prüfung der Software-Version finden Sie in Abschn. 1.5.

Datenpaket 1 (für Software-Version H6 oder älter)

Bezeichnung	Datentyp	Beschreibung
Befehl	Unsigned short (2 Byte)	Definiert die Gültigkeit eines externen Echtzeit-Befehls und das Ende. 0 // Externer Echtzeit-Befehl ungültig 1 // Externer Echtzeit-Befehl gültig 255 // Externer Echtzeit-Befehl beendet
Festlegung des Datentyps zum Senden SendType	Unsigned short (2 Byte)	1. Festlegung des Datentyps für die vom Personalcomputer gesendeten Positionsdaten beim Senden von Daten (Befehle) vom Personalcomputer zum Steuergerät 0 // keine Daten 1 // XYZ-Daten 2 // Gelenk-Daten 3 // Motorimpulse 2. Festlegung des Datentyps für die vom Steuergerät gesendeten Positionsdaten beim Empfang von Antwortdaten (Überwachung) des Personalcomputers vom Steuergerät 0 // keine Daten 1 // XYZ-Daten 2 // Gelenk-Daten 3 // Motorimpulse 4 // XYZ-Daten (Position nach Filterung) 5 // Gelenk-Daten (Position nach Filterung) 6 // Motorimpulse (Position nach Filterung)
Festlegung des Datentyps zum Empfang RecvType	Unsigned short (2 Byte)	1. Festlegung des Datentyps für die vom Steuergerät gesendeten Antwortdaten beim Senden von Daten (Befehle) vom Personalcomputer zum Steuergerät 0 // keine Daten 1 // XYZ-Daten 2 // Gelenk-Daten 3 // Motorimpulse 4 // XYZ-Daten (Position nach Filterung) 5 // Gelenk-Daten (Position nach Filterung) 6 // Motorimpulse (Position nach Filterung) 2. Für den Empfang von Daten (Überwachung) des Personalcomputers vom Steuergerät ist kein Datentyp definiert.
Reserviert reserve	Unsigned short (2 Byte)	Wird nicht verwendet

Tab. 4-3: Datenpaket 1 (1)

Bezeichnung	Datentyp	Beschreibung
Positionsdaten Pos/jnt/pls	POSE, JOINT oder PULSE (40 Byte) (Details der Datenstruktur finden Sie unter strdef.h im Beispiel- programm)	<ol style="list-style-type: none"> 1. Festlegung der vom Personalcomputer gesendeten Positionsdaten beim Senden von Daten (Befehle) vom Personalcomputer zum Steuergerät Wählen Sie hier denselben Datentyp wie bei der Festlegung des Datentyps zum Senden. 2. Festlegung der vom Steuergerät gesendeten Positionsdaten beim Empfang von Antwortdaten (Überwachung) des Personalcomputers vom Steuergerät <p>Die Daten für 1. und 2. sind gleich.</p> <pre>POSE // XYZ-Daten [mm/s] JOINT // Gelenk-Daten [mm/rad] PULSE // Motorimpulse [Impulse]</pre>
Festlegung des Datentyps der Ein-/Ausgangssignaldaten beim Senden SendIOType	Unsigned short (2 Byte)	<ol style="list-style-type: none"> 1. Festlegung des Datentyps der vom Personalcomputer gesendeten Ein-/Ausgangssignaldaten beim Senden von Daten (Befehle) vom Personalcomputer zum Steuergerät Stellen Sie hier „keine Daten“ ein, falls Sie die Funktion nicht verwenden. 2. Festlegung des vom Steuergerät gesendeten Datentyps der Ein-/Ausgangssignaldaten beim Empfang von Antwortdaten (Überwachung) des Personalcomputers vom Steuergerät <p>Die Daten für 1. und 2. sind gleich.</p> <pre>0 // keine Daten 1 // Ausgangssignal 2 // Eingangssignal</pre>
Festlegung des Datentyps der Ein-/Ausgangssignaldaten beim Empfang RecvIOType	Unsigned short (2 Byte)	<ol style="list-style-type: none"> 1. Festlegung des vom Steuergerät gesendeten Datentyps der Ein-/Ausgangssignaldaten beim Senden von Daten (Befehle) vom Personalcomputer zum Steuergerät Stellen Sie hier „keine Daten“ ein, falls Sie die Funktion nicht verwenden. 2. Für den Empfang von Daten (Überwachung) des Personalcomputers vom Steuergerät ist kein Datentyp definiert. <pre>0 // keine Daten 1 // Ausgangssignal 2 // Eingangssignal</pre>
Ein-/Ausgangssignaldaten BitTop BitMask IoData	Unsigned short Unsigned short Unsigned short (2 Byte × 3)	<ol style="list-style-type: none"> 1. Festlegung der vom Personalcomputer gesendeten Ausgangssignaldaten beim Senden von Daten (Befehle) vom Personalcomputer zum Steuergerät 2. Festlegung der vom Steuergerät gesendeten Ein-/Ausgangssignaldaten beim Empfang von Antwortdaten (Überwachung) des Personalcomputers vom Steuergerät <p>Die Daten für 1. und 2. sind gleich.</p> <pre>BitTop // Kopfbitnummer des Ein- oder Ausgangssignals BitMask // Festlegung der Bitmaske (nur für Befehlsvorgabe) PULSE // Wert der Ein- und Ausgangssignaldaten (für Überwachung) Wert der Ausgangssignaldaten (für Befehlsvorgabe) Die Daten sind 16-Bit-Daten.</pre>

Tab. 4-3: Datenpaket 1 (2)

Bezeichnung	Datentyp	Beschreibung
Wert des Zählers zur Zeitüberwachung Tcount	Unsigned short (2 Byte)	<ol style="list-style-type: none"> 1. Für das Senden von Daten (Befehle) vom Personalcomputer an das Steuergerät ist der Wert nicht definiert. 2. Ist der Parameter MXTTOUT beim Empfang von Daten (Überwachung) des Personalcomputers vom Steuergerät auf einen anderen Wert als „-1“ eingestellt, erfasst der Zähler die Anzahl der erfolglosen Kommunikationsversuche mit dem Steuergerät. Nach Überschreiten des Maximalwertes beginnt der Zähler erneut bei „0“. Bei Ausführung des MXT-Befehls wird der Zähler auf „0“ gesetzt.
Wert des Zählers für Kommunikationsdaten Ccount	Unsigned long (4 Byte)	<ol style="list-style-type: none"> 1. Für das Senden von Daten (Befehle) vom Personalcomputer an das Steuergerät ist der Wert nicht definiert. 2. Beim Empfang von Daten (Überwachung) des Personalcomputers vom Steuergerät erfasst der Zähler die Anzahl der Übertragungen. Nach Überschreiten des Maximalwertes beginnt der Zähler erneut bei „0“. Bei Ausführung des MXT-Befehls wird der Zähler auf „0“ gesetzt.

Tab. 4-3: Datenpaket 1 (3)

Datenpaket 2 (ab Software-Version H7)

Bezeichnung	Datentyp	Beschreibung
Befehl	Unsigned short (2 Byte)	Definiert die Gültigkeit eines externen Echtzeit-Befehls und das Ende. 0 // Externer Echtzeit-Befehl ungültig 1 // Externer Echtzeit-Befehl gültig 255 // Externer Echtzeit-Befehl beendet
Festlegung des Datentyps zum Senden SendType	Unsigned short (2 Byte)	1. Festlegung des Datentyps für die vom Personalcomputer gesendeten Positionsdaten beim Senden von Daten (Befehle) vom Personalcomputer zum Steuergerät 0 // keine Daten 1 // XYZ-Daten 2 // Gelenk-Daten 3 // Motorimpulse 2. Festlegung des Datentyps für die vom Steuergerät gesendeten Positionsdaten beim Empfang von Antwortdaten (Überwachung) des Personalcomputers vom Steuergerät 0 // keine Daten 1 // XYZ-Daten 2 // Gelenk-Daten 3 // Motorimpulse 4 // XYZ-Daten (Position nach Filterung) 5 // Gelenk-Daten (Position nach Filterung) 6 // Motorimpulse (Position nach Filterung) 7 // XYZ-Daten (Encoderwert) 8 // Gelenk-Daten (Encoder-Impulse) 9 // Motorimpulse (Encoder-Impulse) 10 // Aktueller Befehl [%] 11 // Aktuelle Rückmeldung [%] Die Daten entsprechen denen des Empfangs-Datentyps RecvType.
Festlegung des Datentyps zum Empfang RecvType	Unsigned short (2 Byte)	1. Festlegung des Datentyps für die vom Steuergerät gesendeten Antwortdaten beim Senden von Daten (Befehle) vom Personalcomputer zum Steuergerät 0 // keine Daten 1 // XYZ-Daten 2 // Gelenk-Daten 3 // Motorimpulse 4 // XYZ-Daten (Position nach Filterung) 5 // Gelenk-Daten (Position nach Filterung) 6 // Motorimpulse (Position nach Filterung) 7 // XYZ-Daten (Encoderwert) 8 // Gelenk-Daten (Encoder-Impulse) 9 // Motorimpulse (Encoder-Impulse) 10 // Aktueller Befehl [%] 11 // Aktuelle Rückmeldung [%] 2. Festlegung des vom Steuergerät gesendeten Datentyps der Positionsdaten beim Empfang von Antwortdaten (Überwachung) des Personalcomputers vom Steuergerät 0 // keine Daten 1 // XYZ-Daten 2 // Gelenk-Daten 3 // Motorimpulse 4 // XYZ-Daten (Position nach Filterung) 5 // Gelenk-Daten (Position nach Filterung) 6 // Motorimpulse (Position nach Filterung) 7 // XYZ-Daten (Encoderwert) 8 // Gelenk-Daten (Encoder-Impulse) 9 // Motorimpulse (Encoder-Impulse) 10 // Aktueller Befehl [%] 11 // Aktuelle Rückmeldung [%] Die Daten entsprechen denen des Sende-Datentyps SendType.
Reserviert reserve	Unsigned short (2 Byte)	Wird nicht verwendet

Tab. 4-4: Datenpaket 2 (1)

Bezeichnung	Datentyp	Beschreibung
Positionsdaten Pos/jnt/pls	POSE, JOINT oder PULSE (40 Byte) (Details der Datenstruktur finden Sie unter strdef.h im Beispiel- programm)	<ol style="list-style-type: none"> 1. Festlegung der vom Personalcomputer gesendeten Positionsdaten beim Senden von Daten (Befehle) vom Personalcomputer zum Steuergerät Wählen Sie hier denselben Datentyp wie bei der Festlegung des Datentyps zum Senden. 2. Festlegung der vom Steuergerät gesendeten Positionsdaten beim Empfang von Antwortdaten (Überwachung) des Personalcomputers vom Steuergerät Der Datentyp ist unter SendType (= RecvType) aufgeführt. Die Daten für Befehle und Überwachung sind gleich. POSE // XYZ-Daten [mm/s] JOINT // Gelenk-Daten [mm/rad] PULSE // Motorimpulse [Impulse]
Festlegung des Datentyps der Ein-/Ausgangssignaldaten beim Senden SendIOType	Unsigned short (2 Byte)	<ol style="list-style-type: none"> 1. Festlegung des Datentyps der vom Personalcomputer gesendeten Ein-/Ausgangssignaldaten beim Senden von Daten (Befehle) vom Personalcomputer zum Steuergerät Stellen Sie hier „keine Daten“ ein, falls Sie die Funktion nicht verwenden. 2. Festlegung des vom Steuergerät gesendeten Datentyps der Ein-/Ausgangssignaldaten beim Empfang von Antwortdaten (Überwachung) des Personalcomputers vom Steuergerät Die Daten für 1. und 2. sind gleich. 0 // keine Daten 1 // Ausgangssignal 2 // Eingangssignal
Festlegung des Datentyps der Ein-/Ausgangssignaldaten beim Empfang RecvIOType	Unsigned short (2 Byte)	<ol style="list-style-type: none"> 1. Festlegung des vom Steuergerät gesendeten Datentyps der Ein-/Ausgangssignaldaten beim Senden von Daten (Befehle) vom Personalcomputer zum Steuergerät Stellen Sie hier „keine Daten“ ein, falls Sie die Funktion nicht verwenden. 0 // keine Daten 1 // Ausgangssignal 2 // Eingangssignal 2. Für den Empfang von Daten (Überwachung) des Personalcomputers vom Steuergerät ist kein Datentyp definiert.
Ein-/Ausgangssignaldaten BitTop BitMask IoData	Unsigned short Unsigned short Unsigned short (2 Byte × 3)	<ol style="list-style-type: none"> 1. Festlegung der vom Personalcomputer gesendeten Ausgangssignaldaten beim Senden von Daten (Befehle) vom Personalcomputer zum Steuergerät 2. Festlegung der vom Steuergerät gesendeten Ein-/Ausgangssignaldaten beim Empfang von Antwortdaten (Überwachung) des Personalcomputers vom Steuergerät Die Daten für 1. und 2. sind gleich. BitTop // Kopfbitnummer des Ein- oder Ausgangssignals BitMask // Festlegung der Bitmaske (nur für Befehlsvorgabe) PULSE // Wert der Ein- und Ausgangssignaldaten (für Überwachung) Wert der Ausgangssignaldaten (für Befehlsvorgabe) Die Daten sind 16-Bit-Daten.
Wert des Zählers zur Zeitüberwachung Tcount	Unsigned short (2 Byte)	<ol style="list-style-type: none"> 1. Für das Senden von Daten (Befehle) vom Personalcomputer an das Steuergerät ist der Wert nicht definiert. 2. Ist der Parameter MXTTOUT beim Empfang von Daten (Überwachung) des Personalcomputers vom Steuergerät auf einen anderen Wert als „-1“ eingestellt, erfasst der Zähler die Anzahl der erfolglosen Kommunikationsversuche mit dem Steuergerät. Nach Überschreiten des Maximalwertes beginnt der Zähler erneut bei „0“. Bei Ausführung des MXT-Befehls wird der Zähler auf „0“ gesetzt.

Tab. 4-4: Datenpaket 2 (2)

Bezeichnung	Datentyp	Beschreibung
Wert des Zählers für Kommunikationsdaten Ccount	Unsigned long (4 Byte)	<ol style="list-style-type: none"> 1. Für das Senden von Daten (Befehle) vom Personalcomputer an das Steuergerät ist der Wert nicht definiert. 2. Beim Empfang von Daten (Überwachung) des Personalcomputers vom Steuergerät erfasst der Zähler die Anzahl der Übertragungen. Nach Überschreiten des Maximalwertes beginnt der Zähler erneut bei „0“. Bei Ausführung des MXT-Befehls wird der Zähler auf „0“ gesetzt.
Erweiterte Festlegung des Datentyps zum Empfang 1 RecvType1	Unsigned short (2 Byte)	Entspricht der Festlegung des Datentyps zum Empfang (RecvType) Verwenden Sie den Datentyp nicht für Befehle.
Reserviert 1 reserve1	Unsigned short (2 Byte)	Wird nicht verwendet
Erweiterte Daten 1 pos/jnt/pls	POSE, JOINT oder PULSE (40 Byte)	Entspricht den Daten von pos/jnt/pls Verwenden Sie die Daten nicht für Befehle.
Erweiterte Festlegung des Datentyps zum Empfang 2 RecvType2	Unsigned short (2 Byte)	Entspricht der Festlegung des Datentyps zum Empfang (RecvType) Verwenden Sie den Datentyp nicht für Befehle.
Reserviert 2 reserve2	Unsigned short (2 Byte)	Wird nicht verwendet
Erweiterte Daten 2 pos/jnt/pls	POSE, JOINT oder PULSE (40 Byte)	Entspricht den Daten von pos/jnt/pls Verwenden Sie die Daten nicht für Befehle.
Erweiterte Festlegung des Datentyps zum Empfang 3 RecvType3	Unsigned short (2 Byte)	Entspricht der Festlegung des Datentyps zum Empfang (RecvType) Verwenden Sie den Datentyp nicht für Befehle.
Reserviert 3 reserve3	Unsigned short (2 Byte)	Wird nicht verwendet
Erweiterte Daten 3 pos/jnt/pls	POSE, JOINT oder PULSE (40 Byte)	Entspricht den Daten von pos/jnt/pls Verwenden Sie die Daten nicht für Befehle.

Tab. 4-4: Datenpaket 2 (3)

5 Fehlerdiagnose

Folgender Abschnitt zeigt die auf den Ethernet-Betrieb bezogenen Fehlermeldungen. Weitere Fehlermeldungen finden Sie in der Bedienungs- und Programmieranleitung der Steuergeräte.

5.1 Übersicht der Fehlermeldungen

Fehlercode	Bedeutung	Ursache	Gegenmaßnahme
H7800	Es wurden mehrere Ethernet-Schnittstellenkarten installiert.	Es darf nur eine Ethernet-Schnittstellenkarte installiert werden.	Eine Ethernet-Schnittstellenkarte installieren
	Fehler bei der Initialisierung der Ethernet-Schnittstellenkarte	Ethernet-Schnittstellenkarte defekt	Ethernet-Schnittstellenkarte austauschen
H7810	Fehlerhafte Einstellung eines Parameters der Ethernet-Schnittstelle	Ein Parameter ist fehlerhaft eingestellt oder falsch geschrieben	Einstellung korrigieren
H7820	Zeitüberschreitung bei Ausführung des MXT-Befehls	Die im Parameter MXTTOUT eingestellte Zeit wurde überschritten	Wert des Parameters MXTTOUT vergrößern
H7830	Keine Ethernet-Schnittstellenkarte installiert	Keine Ethernet-Schnittstellenkarte installiert	Ethernet-Schnittstellenkarte installieren
H7840	Empfangene Daten bei Ausführung des Befehls MXT ungültig	Befehlsparameter und Datentyp stimmen nicht überein	Befehlsparameter und zu übertragene Datenpaket prüfen

Tab. 5-1: Fehlermeldungen

A Anhang

A.1 Beispielprogramme

In diesem Abschnitt wird der Aufbau des Beispielprogramms für den Betrieb der Ethernet-Schnittstellenkarte beschrieben.

A.1.1 Beispielprogramm zum Data-Link-Betrieb

Dieser Abschnitt beschreibt das in Visual-Basic 5.0/6.0 von Microsoft programmierte Beispielprogramm zum Data-Link-Betrieb. Folgende Schritte geben einen kurzen Überblick über die Vorgehensweise zur Erstellung des Programms. Eine detaillierte Beschreibung der Programmiersprache Visual Basic sowie die Erstellung von Anwendungen finden Sie im Handbuch der Software Visual Basic.

- ① Vorbereitung der Winsock-Steuerung
- ② Erstellung einer Form
- ③ Erstellung des Programms Form1.frm

Das Programm steht in zwei Versionen zur Verfügung. Wählen Sie die Version, die zu Ihrer Anwendung passt.

- Programm für Clients
(Bei Einsatz des Personalcomputers als Client und des Steuergerätes als Server)
- Programm für Server
(Bei Einsatz des Personalcomputers als Server und des Steuergerätes als Client)

Die oben aufgeführten Schritte ① und ② sind für beide Programmversionen dieselben.

Als Visual-Basic-Version muss die Professional- oder die Enterprise-Edition verwendet werden. Die Visual-Basic-Learning-Edition kann nicht verwendet werden, da sie die Winsock-Steuerung (Winsock = Windows-Socket) nicht unterstützt.

Vorbereitung der Winsock-Steuerung

Öffnen Sie zur Aktivierung der Winsock-Steuerung Visual-Basic. Klicken Sie auf „Component“ im Menü „Project“. Aktivieren Sie die Winsock-Steuerung. Winsock wird der Werkzeugleiste hinzugefügt.

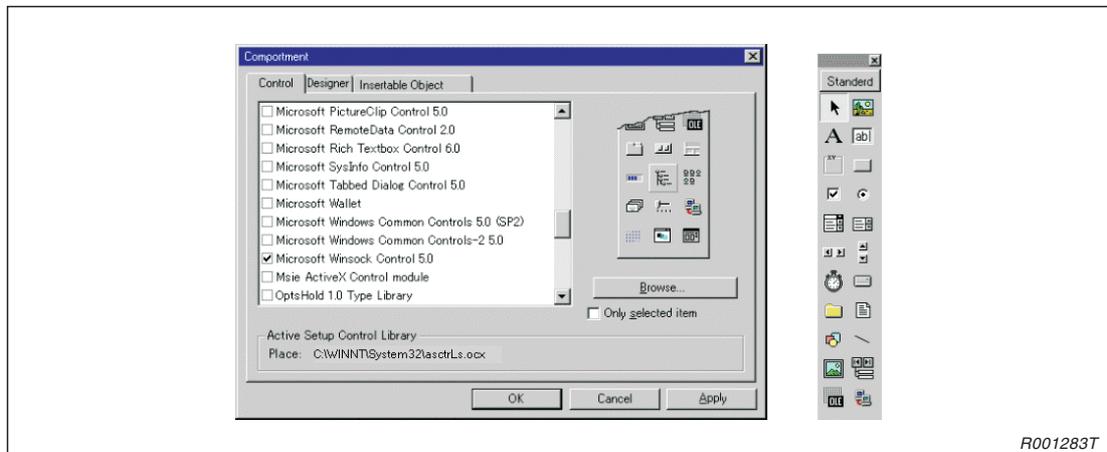


Abb. A-1: Aktivierung der Winsock-Steuerung

Erstellung einer Form

Die Form umfasst 4 Textfelder, 1 Befehlsschaltfläche, 1 Kontrollfeld und 1 Winsocket-Steuerung. Die Properties (Eigenschaften) müssen auf folgende Werte gesetzt werden.

Geänderte Properties		
Objekt	Property	Einstellwert
Form1	Caption	Data link
Command1	Caption	Send
	Enabled	False
Text1	Text	192.168.0.1
Text2	Text	10003
Text3	Text	
Text4	MultiLine	True
	ScrollBars	2-Vertical
Check1	Caption	Connection

Tab. A-1: Einstellungen der Properties

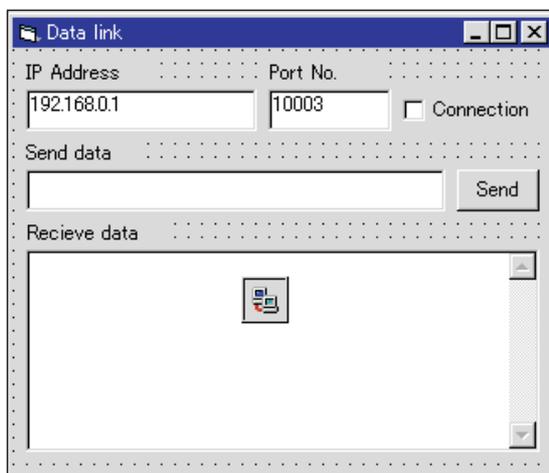


Abb. A-2: Die Form

R001285T

Programm Form1.frm

VERSION 5.00

Object = "{248DD890-BB45-11CF-9ABC-0080C7E7B78D}#1.0#0"; "MSWINSCK.OCX"

```

Begin VB.Form Form1                                'Fenstereinstellungen      Von hier ↓
  Caption      =      "Data link"
  ClientHeight =      3795
  ClientLeft   =      60
  ClientTop    =      345
  ClientWidth  =      4800
  LinkTopic    =      "Form1"
  ScaleHeight  =      3795
  ScaleWidth   =      4800
  StartUpPosition =      3                Von Windows vorgegebener Wert
Begin MSWinsockLib.Winsock Winsock1
  Left         =      2040
  Top          =      2040
  _ExtentX     =      741
  _ExtentY     =      741
End
Begin VB.CommandButton Command1
  Caption      =      "Send"
  Enabled      =      0                'Unwahr
  Height       =      375
  Left         =      3960
  TabIndex     =      6
  Top          =      1080
  Width        =      735
End
Begin VB.CheckBox Check1
  Caption      =      "Connection"
  Height       =      375
  Left         =      3960
  TabIndex     =      4
  Top          =      360
  Width        =      735
End
Begin VB.TextBox Text4
  Height       =      1815
  Left         =      120
  MultiLine   =      -1                'Wahr
  ScrollBars   =      2                'Vertikal
  TabIndex     =      7
  Top          =      1800
  Width        =      4575
End
Begin VB.TextBox Text3
  Height       =      375
  Left         =      120
  TabIndex     =      5
  Top          =      1080
  Width        =      3735
End
Begin VB.TextBox Text2
  Height       =      375
  Left         =      2280
  TabIndex     =      3
  Text         =      "10003"

```

```

    Top          =      360
    Width        =      1575
End
Begin VB.TextBox Text1
    Height       =      375
    Left        =      120
    TabIndex    =      2
    Text        =      "192.168.0.1"
    Top         =      360
    Width       =      2055
End
Begin VB.Label Label4
    Caption      =      "Receive data"
    Height      =      195
    Left        =      120
    TabIndex    =      9
    Top         =      1560
    Width       =      975
End
Begin VB.Label Label3
    Caption      =      "Send data"
    Height      =      195
    Left        =      120
    TabIndex    =      8
    Top         =      840
    Width       =      975
End
Begin VB.Label Label2
    Caption      =      "Port No."
    Height      =      195
    Left        =      2280
    TabIndex    =      1
    Top         =      120
    Width       =      975
End
Begin VB.Label Label1
    Caption      =      "IT address"
    Height      =      255
    Left        =      120
    TabIndex    =      0
    Top         =      120
    Width       =      1095
End
End
'Fenstereinstellungen
Bis hier ↑

Attribute VB_Name          =      "Form1"
Attribute VB_GlobalNameSpace =      False
Attribute VB_Creatable     =      False
Attribute VB_PredeclaredId =      True
Attribute VB_Exposed       =      False

```

**Programm für Clients
(Bei Einsatz des Personalcomputers als Client und des Steuergerätes als Server)**

```

Option Explicit
Dim RecvData() As Byte

Private Sub Check1_Click()      'Ausführung bei Betätigung der Check-Schaltfläche
If Check1.Value Then
    Winsock1.RemoteHost = Text1.Text
    Winsock1.RemotePort = Text2.Text
    Winsock1.Connect
Else
    Winsock1.Close
End If
End Sub

Private Sub Winsock1_Connect() 'Ausführung, wenn das Netzwerk verbunden
                             werden kann
    Command1.Enabled = True
End Sub

Private Sub Winsock1_Close()  'Ausführung bei geschlossenem Netzwerk
    Check1.Value = False
End Sub

Private Sub Command1_Click()  'Ausführung bei Betätigung der Schaltfläche
                             „Transmission“
    Winsock1.SendData (Text3.Text)
End Sub

Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)  'Ausführung beim
                                                           Datenempfang
    If bytesTotal > 0 Then
        ReDim RecvData(bytesTotal - 1)
        Call Winsock1.GetData(RecvData, , bytesTotal)
        Text4.SelStart = Len(Text4.Text)
        Text4.SelText = StrConv(RecvData, vbUnicode)
    End If
End Sub

Private Sub Winsock1_Error(ByVal Number As Integer, _
    Description As String, ByVal Scode As Long, _
    ByVal Source As String, ByVal HelpFile As String, _
    ByVal HelpContext As Long, CancelDisplay As Boolean)
    'Ausführung bei einem Fehler in Windows
    Socket

    Check1.Value = False
    Command1.Enabled = False
    Winsock1.Close
    MsgBox "Error:" & Number & "(" & Description & ")"
End Sub

```

**Programm für Clients
(Bei Einsatz des Personalcomputers als Server und des Steuergerätes als Client)**

```

Option Explicit
Dim RecvData() As Byte

Private Sub Form_Load()
    Text1.Enabled = False          'Editierung der IP-Adresse sperren
End Sub

Private Sub Check1_Click()        'Ausführung bei Betätigung der Check-Schaltfläche
    If Check1.Value Then
        Text1.Text = Winsock1.LocalIP
        Winsock1.LocalPort = Text2.Text
        Winsock1.Listen
    Else
        Command1.Enabled = False
        Winsock1.Close
    End If
End Sub

Private Sub Winsock1_Connect()    'Ausführung, wenn das Netzwerk verbunden
    Command1.Enabled = True       werden kann
End Sub

Private Sub Winsock1_Close()      'Ausführung bei geschlossenem Netzwerk
    Check1.Value = False
End Sub

Private Sub Command1_Click()      'Ausführung bei Betätigung der Schaltfläche
    Winsock1.SendData (Text3.Text) „Transmission“
End Sub

Private Sub Winsock1_ConnectionRequest(ByVal requestID As Long) 'Ausführung bei
    If Winsock1.State = sckClosed Then Winsock1.Close           einer Verbindungs-
    Winsock1.Accept requestID                                     anforderung
    Command1.Enabled = True
End Sub

Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)      'Ausführung beim
    If bytesTotal > 0 Then                                       Datenempfang
        ReDim RecvData(bytesTotal - 1)
        Call Winsock1.GetData(RecvData, , bytesTotal)
        Text4.SelStart = Len(Text4.Text)
        Text4.SelText = StrConv(RecvData, vbUnicode)
        Text4.Text = Text4.Text & vbCrLf
    End If
End Sub

Private Sub Winsock1_Error(ByVal Number As Integer, _
    Description As String, ByVal Scode As Long, _
    ByVal Source As String, ByVal HelpFile As String, _
    ByVal HelpContext As Long, CancelDisplay As Boolean)        'Ausführung bei einem
                                                                Fehler in Windows
                                                                Socket

    Check1.Value = False
    Command1.Enabled = False
    Winsock1.Close
    MsgBox "Error:" & Number & "(" & Description & ")"
End Sub

```

- Beziehung der im Befehl OPEN verwendeten Datei der Kommunikationsleitung COMn: und dem Parameter COMDEV [COMDEV (1), (2), (3), (4), (5), (6), (7), (8)]

Name der Datei für den Kommunikationskanal	COMDEV
COM1:	(1)
COM2:	(2)
COM3:	(3)
COM4:	(4)
COM5:	(5)
COM6:	(6)
COM7:	(7)
COM8:	(8)

Tab. A-2:

Beziehung zwischen der Datei COMn und dem Parameter COMDEV

- Gerätebezeichnung, die dem Parameter COMDEV zugewiesen ist und Parameter zur Einstellung des Protokolls

OPT11 bis OPT19 sind den Einstellungen (1) bis (8) zugewiesen. Der Parameter CPRCE ist für den Data-Link-Betrieb auf „2“ zu setzen.

Schnittstellenummer ^①	Gerätebezeichnung	Protokoll	
	Einstellung COMDEV	CPRCE□□	Einstellwert
10001	OPT11	CPRCE11	2
10002	OPT12	CPRCE12	2
10003	OPT13	CPRCE13	2
10004	OPT14	CPRCE14	2
10005	OPT15	CPRCE15	2
10006	OPT16	CPRCE16	2
10007	OPT17	CPRCE17	2
10008	OPT18	CPRCE18	2
10009	OPT19	CPRCE19	2

Tab. A-3: Gerätebezeichnung und Protokollauswahl

- ^① Die Schnittstellenummer kann über den Parameter NETPORT geändert werden.

A.1.2 Beispielprogramm zur externen Echtzeit-Steuerung

Dieser Abschnitt beschreibt das in Visual C++ 5.0/6.0 von Microsoft programmierte Beispielprogramm zum Data-Link-Betrieb. Folgende Schritte geben einen kurzen Überblick über die Vorgehensweise zur Erstellung des Programms. Eine detaillierte Beschreibung der Programmiersprache Visual C sowie die Erstellung von Anwendungen finden Sie im Handbuch der Software Visual C.

- ① Erstellung eines neuen Projekts
- ② Erstellung des Beispielprogramms `sample.cpp/strdef.h`

Erstellung eines neuen Projekts

Starten Sie Visual C zur Erstellung des neuen Projekts. Fügen Sie das Programm als Win32 Console Application hinzu.

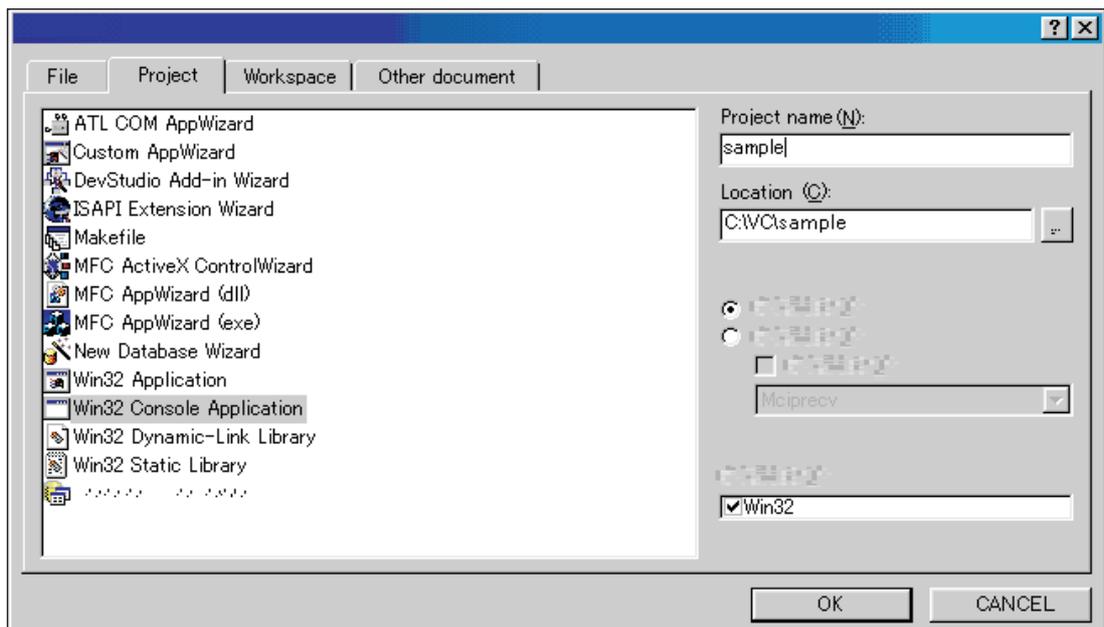


Abb. A-3: Erstellung eines neuen Projekts

Fügen Sie unter Projekteinstellungen `wsock32.lib` zu der Bibliothek des Objekts hinzu.

Erstellung des Beispielprogramms `sample.cpp/strdef.h`

Die Header-Datei `strdef.h` und die Quelldatei `sample.cpp` sind neu zu erstellen. In Abhängigkeit der Software des Steuergeräts können die Kommentare in die Header-Datei `strdef.h` eingefügt werden (siehe `strdef.h`).

Detaillierte Informationen zur Prüfung der Software-Version finden Sie in Abschn. 1.5.

HINWEIS

Hinweis zur Kompilierung:

Verwenden Sie das Setup der Compiler-Option Alignment des Structure-Members mit den 8 Bytes der Grundeinstellung. Bei einer Neuerstellung eines Projekts in Visual C++ ist die Verwendung des Setup mit der Grundeinstellung problemlos. Weitere Hinweise finden Sie im Handbuch der Software Visual C++.

Header-Datei strdef.f

```

/*****
// Beispielprogramm zur externen Echtzeit-Steuerung
// Header-Datei zur Definition der Datenstruktur für das Datenpaket
/*****
// strdef.h

// Führen Sie ab Software-Version H7 die Zeile „define VER_H7“ aus.
// Kennzeichnen Sie die Zeile bei der Version H6 oder älteren Versionen als Kommentar
// (keine Verarbeitung)
#define VER_H7

/*****
/*      Gelenk-Koordinatensystem (nicht verwendete Achsen auf „0“ setzen)  */
/*      Eine detaillierte Beschreibung jedes Roboters finden              */
/*      Sie im Technischen Handbuch des Roboters.                        */
/*****
typedef struct{
    float    j1;                // Winkel J1-Achse (Radiant)
    float    j2;                // Winkel J2-Achse (Radiant)
    float    j3;                // Winkel J3-Achse (Radiant)
    float    j4;                // Winkel J4-Achse (Radiant)
    float    j5;                // Winkel J5-Achse (Radiant)
    float    j6;                // Winkel J6-Achse (Radiant)
    float    j7;                // Zusatzachse 1 (Winkel J7-Achse) (Radiant)
    float    j8;                // Zusatzachse 2 (Winkel J8-Achse) (Radiant)
} JOINT;

/*****
/*      XYZ-Koordinatensystem (nicht verwendete Achsen auf „0“ setzen)  */
/*      Eine detaillierte Beschreibung jedes Roboters finden              */
/*      Sie im Technischen Handbuch des Roboters.                        */
/*****
typedef struct{
    float    x;                // X-Achsen-Koordinate (mm)
    float    y;                // Y-Achsen-Koordinate (mm)
    float    z;                // Z-Achsen-Koordinate (mm)
    float    a;                // A-Achsen-Koordinate (Radiant)
    float    b;                // B-Achsen-Koordinate (Radiant)
    float    c;                // C-Achsen-Koordinate (Radiant)
    float    l1;               // Zusatzachse 1 (mm oder Radiant)
    float    l2;               // Zusatzachse 2 (mm oder Radiant)
} WORLD;

typedef struct{
    WORLD    w;
    unsigned int sflg1;        // Stellungsmerker 1
    unsigned int sflg2;        // Stellungsmerker 2
} POSE;

/*****
/*      Impulskoordinatensystem (nicht verwendete Achsen auf „0“ setzen)  */
/*      Die Koordinaten jedes Gelenks                                    */
/*      sind über Motorimpulse festgelegt.                              */
/*****
typedef struct{
    long     p1;                // Achse Motor 1
    long     p2;                // Achse Motor 2

```

```

    long    p3;           // Achse Motor 3
    long    p4;           // Achse Motor 4
    long    p5;           // Achse Motor 5
    long    p6;           // Achse Motor 6
    long    p7;           // Zusatzachse 1 (Achse Motor 7)
    long    p8;           // Zusatzachse 2 (Achse Motor 8)
} PULSE;

/*****
/* Datenpaket zur Echtzeit-Steuerung */
*****/

typedef struct enet_rtcmd_str {
unsigned short Command;    // Befehl
#define MXT_CMD_NULL      0    // Externer Echtzeit-Befehl gesperrt
#define MXT_CMD_MOVE     1    // Externer Echtzeit-Befehl freigegeben
#define MXT_CMD_END      255  // Externer Echtzeit-Befehl Ende

    unsigned short SendType; // Festlegung Befehlsdatentyp
    unsigned short RecvType; // Festlegung Überwachungsdatentyp
    //////////////// Befehl- oder Überwachungsdatentyp ///

#define MXT_TYP_NULL 0    // Keine Daten
    // Für Befehl und Überwachung ////////////////

#define MXT_TYP_POSE 1    // XYZ-Daten
#define MXT_TYP_JOINT 2  // Gelenkdaten
#define MXT_TYP_PULSE 3  // Impulsdaten
    //////////////// Für auf die Position bezogene Größen ///

#define MXT_TYP_FPOSE 4  // XYZ-Daten (nach Filterung)
#define MXT_TYP_FJOINT 5 // Gelenkdaten (nach Filterung)
#define MXT_TYP_FPULSE 6 // Impulsdaten (nach Filterung)
#define MXT_TYP_FB_POSE 7 // XYZ-Daten (Encoderimpulse) <H7A>
#define MXT_TYP_FB_JOINT 8 // Gelenkdaten (Encoderimpulse) <H7A>
#define MXT_TYP_FB_PULSE 9 // Impulsdaten (Encoderimpulse) <H7A>
    // Für auf Strom bezogene Größen////////// <H7A>

#define MXT_TYP_CMDCUR 10 // Strombefehl <H7A>
#define MXT_TYP_FBKCUR 11 // Stromrückführung <H7A>

    unsigned short reserve; // Reserviert
    union rtdata { // Befehlsdaten
        POSE pos; // XYZ-Daten [mm/rad]
        JOINT jnt; // Gelenkdaten [rad]
        PULSE pls; // Impulsdaten [Impulse]
        long lng1[8]; // Ganze Zahlen [% / dimensionslos]
    } dat;

    unsigned short SendIOType; // Datentypfestlegung der E/A-Signaldaten beim Senden
    unsigned short RecvIOType; // Datentypfestlegung der E/A-Signaldaten beim Empfang
#define MXT_IO_NULL 0 // Keine Daten
#define MXT_IO_OUT 1 // Ausgangssignal
#define MXT_IO_IN 2 // Eingangssignal

    unsigned short BitTop; // Kopfbitnummer
    unsigned short BitMask; // Festlegung Bitmaske zur Übertragung (0x000–0xffff)
    unsigned short IoData; // Ein- und Ausgangssignaldaten (0x0000–0xffff)

    unsigned short TCount; // Wert des Zählers zur Zeitüberwachung
    unsigned long CCount; // Wert des Zählers für Kommunikationsdaten

```

```

#ifdef VER_H7
    unsigned short RecvType1; // Festlegung des Datentyps zum Empfang 1
    unsigned short reserve1; // Reserviert 1
    union rtdata1 {          // Überwachungsdaten 1
        POSE pos1;          // XYZ-Daten [mm/rad]
        JOINT jnt1;         // Gelenkdaten [mm/rad]
        PULSE pls1;         // Impulsdaten [mm/rad]
        long lng1[8];        // Ganze Zahl [%/dimensionslos]
    } dat1;
    unsigned short RecvType2; // Festlegung des Datentyps zum Empfang 2
    unsigned short reserve2; // Reserviert 2
    union rtdata2 {          // Überwachungsdaten 2
        POSE pos2;          // XYZ-Daten [mm/rad]
        JOINT jnt2;         // Gelenkdaten [mm/rad]
        PULSE pls2;         // Impulsdaten [mm/rad] oder ganze Zahl [%/dimensionslos]
        long lng2[8];        // Ganze Zahl [%/dimensionslos]
    } dat2;
    unsigned short RecvType3; // Festlegung des Datentyps zum Empfang 3
    unsigned short reserve3; // Reserviert 3
    union rtdata3 {          // Überwachungsdaten 3
        POSE pos3;          // XYZ-Daten [mm/rad] .
        JOINT jnt3;         // Gelenkdaten [mm/rad] .
        PULSE pls3;         // Impulsdaten [mm/rad] oder ganze Zahl [%/dimensionslos]
        long lng3[8];        // Ganze Zahl [%/dimensionslos]
    } dat3;
#endif

} MXTCMD;

```

Quelldatei sample.cpp

```

// sample.cpp

// Stellen Sie die Definition in der Datei strdef.h in Abhängigkeit der Software-Version des
// Steuergeräts ein. Siehe auch Datei strdef.h.

#include <windows.h>
#include <iostream.h>
#include <winsock.h>
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <math.h>
#include "strdef.h"
#define NO_FLAGS_SET 0
#define MAXBUFLEN 512

INT main(VOID)
{
    WSADATA Data;
    SOCKADDR_IN destSockAddr;
    SOCKET destSocket;
    unsigned long destAddr;
    int status;
    int numsnt;
    int numrcv;
    char sendText[MAXBUFLEN];
    char recvText[MAXBUFLEN];

```

```

char dst_ip_address[MAXBUFLEN];
unsigned short port;
char msg[MAXBUFLEN];
char buf[MAXBUFLEN];
char type, type_mon[4];
unsigned short IOSendType; // Datentypfestlegung der E/A-Signaldaten beim Senden
unsigned short IORecvType; // Datentypfestlegung der E/A-Signaldaten beim Empfang
unsigned short IOBitTop=0;
unsigned short IOBitMask=0xffff;
unsigned short IOBitData=0;

cout << " Eingabe IP-Adresse Verbindungsziel (192.168.0.1) -> ";
cin.getline(dst_ip_address, MAXBUFLEN);
if(dst_ip_address[0]==0) strcpy(dst_ip_address, "192.168.0.1");

cout << " Eingabe Schnittstellennummer Verbindungsziel (10000) -> ";
cin.getline(msg, MAXBUFLEN);
if(msg[0]!=0) port=atoi(msg);
else port=10000;

cout << " Verwenden Sie Ein-/Ausgangssignale?([Y] / [N])-> ";
cin.getline(msg, MAXBUFLEN);
if(msg[0]!=0 && (msg[0]=='Y' || msg[0]=='y')) {
    cout << " Ziel? Eingangssignal/Ausgangssignal([I]nput / [O]utput)-> ";
    cin.getline(msg, MAXBUFLEN);
    switch(msg[0]) {
        case 'O': // Ziel zur Ausgabe eines Signals setzen
        case 'o':
            IOSendType = MXT_IO_OUT;
            IORecvType = MXT_IO_OUT;
            break;
        case 'I': // Ziel zur Eingabe eines Signals setzen
        case 'i':
            IOSendType = MXT_IO_NULL;
            IORecvType = MXT_IO_IN;
            break;
        default:
            IOSendType = MXT_IO_NULL;
            IORecvType = MXT_IO_IN;
            break;
    }
}

cout << " Eingabe Kopfbitnummer (0~32767)-> ";
cin.getline(msg, MAXBUFLEN);
if(msg[0]!=0) IOBitTop = atoi(msg);
else IOBitTop = 0;

if(IOSendType==MXT_IO_OUT) { // Nur für Ausgangssignale
    cout << " Eingabe der Bitmaske zur Ausgabe als Hexadezimalwert (0000~FFFF)-> ";
    cin.getline(msg, MAXBUFLEN);
    if(msg[0]!=0) sscanf(msg,"%4x",&IOBitMask);
    else IOBitMask = 0;
    cout << " Eingabe der Bitdaten zur Ausgabe als Hexadezimalwert (0000~FFFF)-> ";
    cin.getline(msg, MAXBUFLEN);
    if(msg[0]!=0) sscanf(msg,"%4x",&IOBitData);
    else IOBitData = 0;
}
}

cout <<" --- Eingabe des Befehlsdatentyps --- ¥n";
cout <<"[0: kein / 1: XYZ / 2: GELENK / 3: IMPULSE]¥n.";
cout <<" -- Bitte Wert eingeben -- [0] - [3]->";

```

```

    cin.getline(msg, MAXBUFLEN);
    type = atoi(msg);
#ifdef VER_H7
    for(int k=0; k<4; k++) {
        sprintf (msg, " --- Eingabe Datentyp Überwachungsdaten ( %d-th ) --- ¥n", k);
        cout << msg;
        cout << "[0: kein]¥n";
        cout << "[1: XYZ / 2: GELENK / 3: IMPULSE].....Befehlswert ¥n";
        cout << "[4: XYZ / 5: GELENK / 6: IMPULSE].....Befehlswert nach Filterung ¥n";
        cout << "[7: XYZ / 5: GELENK / 6: IMPULSE].....Rückgeführter Wert ¥n";
        cout << "[10: Strom / 11: Rückgeführter Strom].....Elektrischer Strom ¥n";
        cout << "Bitte Wert eingeben [0]~[11] -> ";
        cin.getline(msg, MAXBUFLEN);
        type_mon[k] = atoi(msg);
    }
#else
    type_mon[0]=type;
    type_mon[1]=type_mon[2]=type_mon[3]=0;
#endif
    sprintf(msg, "IP=%s / PORT=%d / Sendetyp=%d / Überwachungstyp0/1/2/3=%d/%d/%d/%d"
, dst_ip_address, port , type , type_mon[0], type_mon[1], type_mon[2], type_mon[3]);
    cout << msg << endl;
    cout << "[Enter]= Ende / [d]= Anzeige Überwachungsdaten";
    cout << "[z/x]= Erhöhe/verringere die ersten durch den Deltawert übertragenen Befehlsdaten. ";
    cout << " Okay? [Enter] / [Ctrl+C] ";
    cin.getline(msg, MAXBUFLEN);

    // Windows Socket DLL Initialisierung
    status=WSAStartup(MAKEWORD(1, 1), &Data);
    if (status != 0)
        cerr << "FEHLER: WSA-Startup fehlgeschlagen" << endl;

    // IP-Adresse, Schnittstelle, etc. einstellen
    memset(&destSockAddr, 0, sizeof(destSockAddr));
    destAddr=inet_addr(dst_ip_address);
    memcpy(&destSockAddr.sin_addr, &destAddr, sizeof(destAddr));
    destSockAddr.sin_port=htons(port);
    destSockAddr.sin_family=AF_INET;

    // Socket-Erstellung
    destSocket=socket(AF_INET, SOCK_DGRAM, 0);
    if (destSocket == INVALID_SOCKET) {
        cerr << "FEHLER: Socketerstellung fehlgeschlagen" << endl;
        status=WSACleanup();
        if (status == SOCKET_ERROR)
            cerr << "FEHLER: WSA-Cleanup fehlgeschlagen" << endl;
        return(1);
    }

    MXTCMD MXTsend;
    MXTCMD MXTrecv;
    JOINT jnt_now;
    POSE pos_now;
    PULSE pls_now;

```

```

unsigned long counter = 0;
int loop = 1;
int disp = 0;
int disp_data = 0;
int ch;
float delta=(float)0.0;
long ratio=1;
int retry;
fd_set SockSet;           // In select verwendete Socket-Gruppe
timeval sTimeOut;       // Einstellung Wartezeit

memset(&MXTsend, 0, sizeof(MXTsend));
memset(&jnt_now, 0, sizeof(JOINT));
memset(&pos_now, 0, sizeof(POSE));
memset(&pls_now, 0, sizeof(PULSE));

while(loop) {

    memset(&MXTsend, 0, sizeof(MXTsend));
    memset(&MXTrecev, 0, sizeof(MXTrecev));

    // Erstellung der Übertragungsdaten
    if(loop==1) { // Nur beim ersten Durchlauf
        MXTsend.Command = MXT_CMD_NULL;
        MXTsend.SendType = MXT_TYP_NULL;
        MXTsend.RecvType = type;
        MXTsend.SendIOType = MXT_IO_NULL;
        MXTsend.RecvIOType = IOSendType;
        MXTsend.CCount = counter = 0;
    }
    else { // Beim zweiten Durchlauf und den folgenden Durchläufen
        MXTsend.Command = MXT_CMD_MOVE;
        MXTsend.SendType = type;
        MXTsend.RecvType = type*_mon[0];
#ifdef VER_H7
        MXTsend.RecvType1= type_mon[1];
        MXTsend.RecvType2= type_mon[2];
        MXTsend.RecvType3= type_mon[3];
#endif
    #endif
        switch(type) {
            case MXT_TYP_JOINT:
                memcpy(&MXTsend.dat.jnt, &jnt_now, sizeof(JOINT));
                MXTsend.dat.jnt.j1 += (float)(delta*ratio*3.141592/180.0);
                break;
            case MXT_TYP_POSE:
                memcpy(&MXTsend.dat.pos, &pos_now, sizeof(POSE));
                MXTsend.dat.pos.w.x += (delta*ratio);
                break;
            case MXT_TYP_PULSE:
                memcpy(&MXTsend.dat.pls, &pls_now, sizeof(PULSE));
                MXTsend.dat.pls.p1 += (long)((delta*ratio)*10);
                break;
            default:
                break;
        }
        MXTsend.SendIOType = IOSendType;
        MXTsend.RecvIOType = IORecvType;

```

```

MXTsend.BitTop = IOBitTop;
MXTsend.BitMask = IOBitMask;
MXTsend.loData = IOBitData;
MXTsend.CCount = counter;
}
// Tastatureingabe
// [Enter]=Ende / [d]= Anzeige der Überwachungsdaten, oder keine /
// [0/1/2/3]= Anzeige der Überwachungsdaten ändern
// [z/x]=Erhöhe/verringe die ersten durch den Deltawert übertragenen Befehlsdaten
while(kbhit()!=0) {
    ch=getch();
    switch(ch) {
    case 0x0d:
        MXTsend.Command = MXT_CMD_END;
        loop = 0;
        break;
    case 'Z':
    case 'z':
        delta += (float)0.1;
        break;
    case 'X':
    case 'x':
        delta -= (float)0.1;
        break;
    case 'C':
    case 'c':
        delta = (float)0.0;
        break;
    case 'd':
        disp = ~disp;
        break;
    case '0': case '1': case '2': case '3':
        disp_data = ch - '0';
        break;
    }
}

memset(sendText, 0, MAXBUFLen);
memcpy(sendText, &MXTsend, sizeof(MXTsend));
if(disp) {
    sprintf(buf, "Senden (%ld):",counter);
    cout << buf << endl;
}
numsent=sendto(destSocket, sendText, sizeof(MXTCMD), NO_FLAGS_SET
                , (LPSOCKADDR) &destSockAddr, sizeof(destSockAddr));
if (numsent != sizeof(MXTCMD)) {
    cerr << "FEHLER: Senden fehlgeschlagen" << endl;
    status=closesocket(destSocket);
    if (status == SOCKET_ERROR)
        cerr << "FEHLER: Socket schließen fehlgeschlagen" << endl;
    status=WSACleanup();
    if (status == SOCKET_ERROR)
        cerr << "FEHLER: WSA-Cleanup fehlgeschlagen" << endl;
    return(1);
}

memset(recvText, 0, MAXBUFLen);

```

```

retry = 1; // Anzahl Empfangsversuche
while(retry) {
    FD_ZERO(&SockSet); // SockSet-Initialisierung
    FD_SET(destSocket, &SockSet); // Socket-Registrierung
    sTimeOut.tv_sec = 1; // Einstellung der Datenübertragungszeit (s)
    sTimeOut.tv_usec = 0; // (µs)
    status = select(0, &SockSet, (fd_set *)NULL, (fd_set *)NULL, &sTimeOut);
    if(status == SOCKET_ERROR) {
        return(1);
    }
    // Empfang in der Datenübertragungszeit
    if((status > 0) && (FD_ISSET(destSocket, &SockSet) != 0)) {
        numrcv=recvfrom(destSocket, recvText, MAXBUFLEN, NO_FLAGS_SET, NULL,
        NULL);
        if (numrcv == SOCKET_ERROR) {
            cerr << "FEHLER: recvfrom fehlgeschlagen" << endl;
            status=closesocket(destSocket);
            if (status == SOCKET_ERROR)
                cerr << "FEHLER: Socket schließen fehlgeschlagen" << endl;
            status=WSACleanup();
            if (status == SOCKET_ERROR)
                cerr << "FEHLER: WSA-Cleanup fehlgeschlagen" << endl;
            return(1);
        }
        memcpy(&MXTrecv, recvText, sizeof(MXTrecv));
        char str[10];
        if(MXTrecv.SendIOType==MXT_IO_IN)
            sprintf(str,"IN%04x", MXTrecv.ioData);
        else if(MXTrecv.SendIOType==MXT_IO_OUT)
            sprintf(str,"OT%04x", MXTrecv.ioData);
        else sprintf(str,"-----");

        int DispType;
        void *DispData;
#ifdef VER_H7
        switch(disp_data) {
            case 0:
                DispType = MXTrecv.RecvType;
                DispData = &MXTrecv.dat;
                break;
            case 1:
                DispType = MXTrecv.RecvType1;
                DispData = &MXTrecv.dat1;
                break;
            case 2:
                DispType = MXTrecv.RecvType2;
                DispData = &MXTrecv.dat2;
                break;
            case 3:
                DispType = MXTrecv.RecvType3;
                DispData = &MXTrecv.dat3;
                break;
            default:
                break;
        }
#else
        DispType = MXTrecv.SendType;
        DispData = &MXTrecv.dat;
#endif
    }
}

```

```

switch(DispType) {
case MXT_TYP_JOINT:
case MXT_TYP_FJOINT:
case MXT_TYP_FB_JOINT:
    if(loop==1) {
        memcpy(&jnt_now, DispData, sizeof(JOINT));
        loop = 2;
    }
    if(DispType) {
        JOINT *j=(JOINT*)DispData;
        sprintf(buf, "Empfang (%ld): TCount=%d Type(GELENK)=%d%n
            %7.2f,%7.2f,%7.2f,%7.2f,%7.2f,%7.2f,%7.2f,%7.2f (%s)"
            ,MXTrecv.CCount,MXTrecv.TCount,DispType
            ,j->j1, j->j2, j->j3 ,j->j4, j->j5, j->j6, j->j7, j->j8, str);
        cout < buf < endl;
    }
    break;
case MXT_TYP_POSE:
case MXT_TYP_FPOSE:
case MXT_TYP_FB_POSE:
    if(loop==1) {
        memcpy(&pos_now, &MXTrecv.dat.pos, sizeof(POSE));
        loop = 2;
    }
    if(DispType) {
        POSE *p=(POSE*)DispData;
        sprintf(buf, "Empfang (%ld): TCount=%d Type(POSITION)=%d%n
            %7.2f,%7.2f,%7.2f,%7.2f,%7.2f,%7.2f, %04x,%04x (%s)"
            ,MXTrecv.CCount,MXTrecv.TCount,DispType
            ,p->w.x, p->w.y, p->w.z, p->w.a, p->w.b, p->w.c
            , p->sflg1, p->sflg2, str);
        cout < buf < endl;
    }
    break;
case MXT_TYP_PULSE:
case MXT_TYP_FPULSE:
case MXT_TYP_FB_PULSE:
case MXT_TYP_CMDCUR:
case MXT_TYP_FBKCUR:
    if(loop==1) {
        memcpy(&pls_now, &MXTrecv.dat.pls, sizeof(PULSE));
        loop = 2;
    }
    if(DispType) {
        PULSE *l=(PULSE*)DispData;
        sprintf(buf, "Empfang (%ld): TCount=%d Type(IMPULS/ANDERE)=%d%n
            %ld,%ld,%ld,%ld,%ld,%ld,%ld,%ld (%s)"
            ,MXTrecv.CCount,MXTrecv.TCount,DispType
            ,l->p1, l->p2, l->p3, l->p4, l->p5, l->p6, l->p7, l->p8, str);
        cout < buf < endl;
    }
    break;
case MXT_TYP_NULL:
    if(loop==1) {
        loop = 2;
    }
    if(DispType) {
        sprintf(buf, "Empfang (%ld): TCount=%d Type(NULL)=%d%n (%s)"

```

```

        ,MXTrecv.CCount,MXTrecv.TCount, DispType, str);
        cout < buf < endl;
    }
    break;
default:
    cout < "Fehlerhafter Datentyp.¥n" < endl;
    break;
}
counter++;           // Aufwärtszählen nur bei erfolgreicher Kommunikation
retry=0;            // Verlasse Empfangsschleife
}
else { // Zeitüberschreitung beim Empfang
    cout < "... Zeitüberschreitung beim Empfang! <[Enter] zum Programmstopp betä-
    tigen>" < endl;
    retry--;         // Anzahl der Versuche herunterzählen
    if(retry==0) loop=0; // Programmende wenn Anzahl der Versuche gleich 0
}
} /* while(retry) */
} /* while(loop) */

// Ende
cout < "/// Ende /// ";
sprintf(buf, "Zähler = %ld", counter);
cout < buf < endl;

// Socket schließen
status=closesocket(destSocket);
if (status == SOCKET_ERROR)
cerr < "FEHLER: Socket schließen fehlgeschlagen" < endl;
status=WSACleanup();
if (status == SOCKET_ERROR)
cerr < "FEHLER: WSA-Cleanup fehlgeschlagen" < endl;

return 0;
}

```


HEADQUARTERS

MITSUBISHI ELECTRIC EUROPE B.V.
 German Branch
 Gothaer Straße 8
D-40880 Ratingen
 Telefon: (02102) 486-0
 Telefax: (02102) 486-1120
 E-Mail: megfamail@meg.mee.com

MITSUBISHI ELECTRIC EUROPE B.V.
 French Branch
 25, Boulevard des Bouvets
F-92741 Nanterre Cedex
 Telefon: +33 1 55 68 55 68
 Telefax: +33 1 55 68 56 85
 E-Mail: factory.automation@fram.mee.com

MITSUBISHI ELECTRIC EUROPE B.V.
 Irish Branch
 Westgate Business Park, Ballymount
IRL-Dublin 24
 Telefon: +353 (0)1 / 419 88 00
 Telefax: +353 (0)1 / 419 88 90
 E-Mail: sales.info@meir.mee.com

MITSUBISHI ELECTRIC EUROPE B.V.
 Italian Branch
 Via Paracelso 12
I-20041 Agrate Brianza (MI)
 Telefon: +39 (0)39 / 60 53 1
 Telefax: +39 (0)39 / 60 53 312
 E-Mail: factory.automation@it.mee.com

MITSUBISHI ELECTRIC EUROPE B.V.
 Spanish Branch
 Carretera de Rubí 76-80
E-08190 Sant Cugat del Vallés (Barcelona)
 Telefon: +34 9 3 / 565 3160
 Telefax: +34 9 3 / 589 1579
 E-Mail: industrial@sp.mee.com

MITSUBISHI ELECTRIC EUROPE B.V.
 UK Branch
 Travellers Lane
GB-Hatfield Herts. AL10 8 XB
 Telefon: +44 (0)1707 276100
 Telefax: +44 (0)1707 278695
 E-Mail: automation@meuk.mee.com

MITSUBISHI ELECTRIC CORPORATION
 Office Tower "Z" 14 F
 8-12,1 chome, Harumi Chuo-Ku
Tokyo 104-6212
 Telefon: +81 3 6221 6060
 Telefax: +81 3 6221 6075

MITSUBISHI ELECTRIC AUTOMATION
 500 Corporate Woods Parkway
Vernon Hills, IL 60061
 Telefon: +1 847 / 478 21 00
 Telefax: +1 847 / 478 22 83

EUROPÄISCHE VERTRETUNGEN

Koning & Hartman b.v. BELGIEN
 Researchpark Zellik
 Pontbeeklaan 43
BE-1731 Brussels
 Telefon: +32 (0)2 / 467 17 51
 Telefax: +32 (0)2 / 467 17 45
 E-Mail: info@koningenhartman.com

Herstad + Piper A/S DÄNEMARK
 Jernholmen 48 C
DK-2650 Hvidovre
 Telefon: +45 (0)36 - 77 40 00
 Telefax: +45 (0)36 - 77 77 40
 E-Mail: mail@herstad-piper.dk

Beijer Electronics OY FINNLAND
 Ansatie 6a
FI-01740 Vantaa
 Telefon: +358 (0)9 / 886 77 500
 Telefax: +358 (0)9 / 886 77 555
 E-Mail: info@beijer.fi

Kouvalias GRIECHENLAND
 Robot + Vision Systems
 25, El. Venizelou Ave
GR-17671 Kallithea
 Telefon: +30 22950 / 42902/3/4
 Telefax: +30 22950 / 42690
 E-Mail: info@kouvalias.com

Koning & Hartman b.v. NIEDERLANDE
 Donauweg 2 B
NL-1000 AK Amsterdam
 Telefon: +31 (0)20 / 587 76 00
 Telefax: +31 (0)20 / 587 76 05
 E-Mail: info@koningenhartman.com

Beijer Electronics AS NORWEGEN
 Teglverksveien 1
NO-3002 Drammen
 Telefon: +47 (0)32 / 24 30 00
 Telefax: +47 (0)32 / 84 85 77
 E-Mail: info@beijer.no

GEVA ÖSTERREICH
 Wiener Straße 89
AT-2500 Baden
 Telefon: +43 (0)2252 / 85 55 20
 Telefax: +43 (0)2252 / 488 60
 E-Mail: office@geva.at

EUROPÄISCHE VERTRETUNGEN

MPL Technology Sp. z o.o. POLEN
 ul. Sliczna 36
PL-31-444 Kraków
 Telefon: +48 (0)12 / 632 28 85
 Telefax: +48 (0)12 / 632 47 82
 E-Mail: krakow@mpl.pl

Beijer Electronics AB SCHWEDEN
 Box 426
S-20124 Malmö
 Telefon: +46 (0)40 / 35 86 00
 Telefax: +46 (0)40 / 35 86 02
 E-Mail: info@beijer.se

ECONOTEC AG SCHWEIZ
 Postfach 282
CH-8309 Nürensdorf
 Telefon: +41 (0)1 / 838 48 11
 Telefax: +41 (0)1 / 838 48 12
 E-Mail: info@econotec.ch

INEA SR d.o.o. SERBIEN & MONTENEGRO
 Karadjordjeva 12/260
SCG-113000 Smederevo
 Telefon: +381 (0)26 / 617 163
 Telefax: +381 (0)26 / 617 163
 E-Mail: vladstoj@yubc.net

AutoCont Control s.r.o. SLOWAKEI
 Radlinského 47
SK-02601 Dolný Kubín
 Telefon: +421 435868 210
 Telefax: +421 435868 210
 E-Mail: info@autocontcontrol.sk

INEA d.o.o. SLOWENIEN
 Stegne 11
SI-1000 Ljubljana
 Telefon: +386 (0)1 513 8100
 Telefax: +386 (0)1 513 8170
 E-Mail: inea@inea.si

AutoCont TSCHECHISCHE REPUBLIK
 Control Systems s.r.o.
 Nemocnici 12
CZ-70200 Ostrava 2
 Telefon: +420 59 / 6152 111
 Telefax: +420 59 / 6152 562
 E-Mail: consys@autocont.cz

GTS TÜRKEI
 Darülaceze Cad. No. 43 Kat. 2
TR-80270 Okmeydani-Istanbul
 Telefon: +90 (0)212 / 320 1640
 Telefax: +90 (0)212 / 320 1649
 E-Mail: gts@turk.net

Axicont Automatika Kft. UNGARN
 Reitter F. U. 132
HU-1131 Budapest
 Telefon: +36 (0)1 / 412-0882
 Telefax: +36 (0)1 / 412-0883
 E-Mail: office@axicont.hu

KUNDEN-TECHNOLOGIE-CENTER DEUTSCHLAND

MITSUBISHI ELECTRIC EUROPE B.V.
 Kunden-Technologie-Center Nord
 Revierstraße 5
D-44379 Dortmund
 Telefon: (0231) 96 70 41-0
 Telefax: (0231) 96 70 41-41

MITSUBISHI ELECTRIC EUROPE B.V.
 Kunden-Technologie-Center
 Süd-West
 Kurze Straße 40
D-70794 Filderstadt
 Telefon: (0711) 77 05 98 0
 Telefax: (0711) 77 05 98 79

MITSUBISHI ELECTRIC EUROPE B.V.
 Kunden-Technologie-Center
 Süd-Ost
 Am Söldnermoos 8
D-85399 Hallbergmoos
 Telefon: (0811) 99 87 40
 Telefax: (0811) 99 87 410

VERTRETUNGEN EURASIEN

ELEKTROSTYLE RUSSLAND
 Poslannikov Per., 9, Str.1
RU-107005 Moscow
 Telefon: +7 095 / 542-4323
 Telefax: +7 095 / 956-7526
 E-Mail: info@estl.ru

ELEKTROSTYLE RUSSLAND
 Krasnij Prospekt 220-1, Office 312
RU-630049 Novosibirsk
 Telefon: +7 3832 / 106618
 Telefax: +7 3832 / 106626
 E-Mail: info@estl.ru

ICOS RUSSLAND
 Industrial Computer Systems Zao
 Ryazanskij Prospekt, 8A, Office 100
RU-109428 Moscow
 Telefon: +7 095 232 0207
 Telefax: +7 095 232 0327
 E-Mail: mail@icos.ru

VERTRETUNG MITTLERER OSTEN

Ilan & Gavish Ltd ISRAEL
 Automation Service
 24 Shenkar St., Kiryat Arie
IL-49001 Petach-Tiqva
 Telefon: +972 (0 3 / 922 18 24
 Telefax: +972 (0 3 / 924 07 61
 E-Mail: iandg@internet-zahav.net

VERTRETUNG AFRIKA

CBI Ltd SÜDAFRIKA
 Private Bag 2016
ZA-1600 Isando
 Telefon: +27 (0 11 / 928 2000
 Telefax: +27 (0 11 / 392 2354
 E-Mail: cbi@cbi.co.za