Betriebsanleitung

XCx 300 / 500 / 540

Kompaktsteuerungen für SPS- und CNC-Anwendungen

Version 08/06
Artikel-Nr. R4.322.2130.0 (322 383 82)

schleicher

schleicher

Zweck der Betriebsanleitung

Die vorliegende Betriebsanleitung dient als Anleitung zur Projektierung, Programmierung, Installation, Inbetriebnahme, Betrieb und Wartung der XCx 300, XCx 500 und XCx 540.
Die Betriebsanleitung enthält Projektierungs-, Programmier-, Bedienungshinweise und technische Daten der XCx 300, XCx 500 und XCx 540.

Zielgruppe

Die Betriebsanleitung ist für geschulte Fachkräfte ausgelegt. Es werden besondere Anforderungen an die Auswahl und Ausbildung des Personals gestellt, die mit dem Automatisierungssystem umgehen. Als Personen kommen z.B. Elektrofachkräfte und Elektroingenieure in Frage, die entsprechend geschult sind (siehe auch Sicherheitshinweise "Personalauswahl und -qualifikation").

Vorgängerversionen

01/02 06/02 08/02 11/02 01/03 04/03 01/04 08/04 04/05 06/05

Softwarestände

Betriebssystem: XCx Version 06.26/0
Programmiersoftware: MULTIPROG 4.0 Build 197
OPC-Server: ProConOS OPC-Server 2.0 Desktop Build 42
MWT-AddOn's: Version 06.26/0 für MULTIPROG 3.x / 4.x
Schleicher-Dialog: Version 06.26/0

Bezugsmöglichkeiten für Betriebsanleitungen

Alle Betriebsanleitungen können kostenlos vom Internet: *http:\\www.schleicher-electronic.com* geladen, oder unter Angabe der Artikel-Nr. bestellt werden bei:
SCHLEICHER Electronic GmbH & Co. KG, Pichelswerderstraße 3-5, D-13597 Berlin

Weitere Betriebsanleitungen, auf die Bezug genommen wird

Bezeichnung	Artikel-Nr.
Inbetriebnahmehinweise für Feldbussysteme	R4.322.1600.0 (322 152 48)
Commissioning Field Bus Systems	R4.322.1610.0 (322 152 49)
CNC-Programmierung XCx und ProNumeric deutsch	R4.322.2080.0 (322 381 61)
CNC Programming XCx and ProNumeric englisch	R4.322.2090.0 (322 381 62)
RIO Erweiterungsmodule deutsch	R4.322.1720.0 (322 154 14)
RIO Expansion Modules englisch	R4.322.1730.0 (322 154 15)

Copyright by

SCHLEICHER Electronic GmbH & Co. KG, Pichelswerderstraße 3-5, D-13597 Berlin
Telefon 030 33005 – 330, Telefax 030 33005 – 305, Hotline 030 33005 - 304
Internet *http://www.schleicher-electronic.com*

Änderungen und Irrtum vorbehalten

Inhaltsverzeichnis

1 Übersicht ... 8
2 Installation ... 13
2.1 XCx ... 13
2.1.1 Montagelage ... 13
2.1.2 Montageabstände ... 13
2.1.3 Montage der XCx 500 und 540 ... 14
2.1.4 Montage der XCx 300 auf der Hutschiene ... 16
2.1.5 Anschluss der Spannungsversorgung ... 18
2.2 XRIO ... 20
2.2.1 Montagelage ... 20
2.2.2 Montageabmaße und -abstände ... 20
2.2.3 Hutschienenmontage ... 21
2.2.4 Verbindung der Module untereinander ... 22
2.2.5 Federkraftklemmen des XRIO-Buskopplers ... 23
2.2.6 Federkraftklemmen der RIO-Erweiterungsmodule ... 24
2.2.7 Anschlussquerschnitte und Abisolierlänge RIO ... 25
2.2.8 Anschluss der Spannungsversorgung für XRIO ... 26
2.2.9 Buskabel XRIO ... 28
3 Anzeigen, Anschlüsse und Bedienelemente ... 29
3.1 Übersicht ... 29
3.2 LED-Anzeigen ... 30
3.3 Compact Flash ... 32
3.4 X7 PRG RS232-Schnittstelle für Programmiergeräte ... 32
3.5 X6 ETH RJ 45 EtherNet-Anschluss ... 32
3.6 X5 CANopen Feldbusanschluss ... 33
3.7 X4 X RIO Anschluss ... 33
3.8 X3 USB-(Universal Serial Bus) Anschluss ... 33
3.9 X1 RS422-Schnittstelle für den Anschluss von Bedien- und Anzeigegeräten ... 34
3.10 X2 Anschluss für Versorgungsspannung, direkte E/A und BUSY-Kontakt ... 34
3.11 Betriebsartenschalter ... 35
3.12 RESET-Taster ... 35
4 Schnelleinführung ... 36
4.1 Inbetriebnahme der Kommunikation ... 36
4.1.1 Kommunikationsschema ... 36
4.1.2 Inbetriebnahme der seriellen Verbindung über die RS232-Schnittstelle ... 36
4.2 Inbetriebnahme der Ethernet-Schnittstelle ... 38
4.2.1 Auslesen der aktuellen IP-Adresse aus der XCx ... 38
4.2.2 Einstellen der IP-Adresse des Service-PC ... 39
4.2.3 Ethernet-Verbindung herstellen und prüfen ... 39
4.3 Ändern der IP-Adresse der XCx ... 40
4.4 Installation der Programmiersoftware MULTIPROG, OPC-Server, AddOns ... 41
4.4.1 Installation von MULTIPROG ... 41
4.4.2 MULTIPROG starten und ein neues Projekt öffnen ... 47
4.4.3 Ethernet-Verbindung in MULTIPROG einstellen ... 49
4.4.4 Ein Projekt Kompilieren und zur XCx senden ... 50
4.4.5 Einfügen der Koppelspeicher-Datentypen und -Variablen in das Projekt ... 52
4.5 Zugriff auf die I/O Ebene mit XRIO ... 57
4.5.1 Anschlussprinzip XRIO ... 57
4.5.2 Beispiel einer XRIO-Konfiguration ... 57
4.5.3 Einlesen der XRIO-Konfiguration ... 58
4.5.4 Der erste I/O-Zugriff durch das SPS-Programm ... 62
4.5.5 Diagnose ... 63
4.6 CANopen für dezentrale I/O ... 64
4.6.1 Spezifikationen ... 64
4.6.2 Anschlussprinzip und Verkabelung ... 65
4.6.3 Einstellungen am I/O-Modul RIO 8 I/O CANopen ... 66
4.6.4 Deklaration des I/O-Treibers für CANopen ... 66
4.6.5 Deklaration von Netzwerkvariablen in MULTIPROG ... 68
4.6.6 Konfiguration des CANopen-Netzwerkes mit ProCANopen ... 69

4.6.7 Installation von ProCANopen ... 70
4.6.8 Einbindung von ProCANopen in MULTIPROG ... 71
4.6.9 Erste Verbindungen mit ProCANopen ... 72
4.7 Die Web-Server Funktionen der XCx ... 76
4.7.1 Allgemeine Funktionen und Konzept ... 76
4.7.2 Schleicher spezifisches Applet ... 76
4.7.3 Deklaration von Variablen zur Visualisierung ... 77
4.7.4 Applikationsbeispiel ... 77
4.7.5 Browser / Komponenten ... 77
4.8 Allgemeine Hinweise zur Inbetriebnahme ... 78
4.8.1 Steuerungsanlauf mit leerer Puffer-Batterie (batterieloser Betrieb) ... 78
4.8.2 Grundinitialisierung ... 79
5 Bedienung der XCx mit dem Schleicher-Dialog ... 80
5.1 Übersicht ... 80
5.2 Installation auf dem PC und Vorbereitung des OPC-Servers ... 81
5.2.1 Installation des OPC-Servers ... 81
5.2.2 Installation des Schleicher-Dialoges ... 81
5.2.3 Deklaration der OPC-Variablen im SPS-Projekt mit MULTIPROG ... 82
5.2.4 Konfiguration und Test des OPC-Servers ... 83
5.3 Allgemeiner Aufbau der Bedienoberfläche ... 87
5.4 Schleicher-Dialog der XCS ... 90
5.5 Schleicher-Dialog der XCN ... 92
5.6 Aufruf Activ-Error-Buffer und Log-Book ... 97
6 XRIO ... 98
6.1 XRIO-Regeln ... 99
6.2 Erkennung der XRIO-Konfiguration durch das Betriebssystem der XCx ... 100
6.2.1 Die Datei sysinfo.txt Abschnitt XRIO ... 101
6.3 Adressierung der I/O-Module ... 103
6.4 XRIO Flags im Koppelspeicher Gültigkeit der Prozessdaten ... 103
6.5 Der XRIO-Treiber ... 104
6.6 Berechnung der "Estimated Transmission Time" ... 105
6.7 Erweiterungsmodule aus dem System RIO ... 106
6.7.1 Übersicht ... 106
6.7.2 Tabellen der Modultypen und Modulklassen ... 109
6.8 Erweiterungsmodul XRIO ... 111
6.8.1 Anzeigen und Anschlüsse ... 111
6.8.2 Technische Daten ... 111
7 Feldbus CANopen ... 112
7.1 Grundlagen ... 112
7.1.1 Process- Data- Objects (PDO) ... 113
7.1.2 Service-Data-Objects (SDO) ... 113
7.1.3 Nodeguarding ... 113
7.1.4 Lifeguarding ... 114
7.2 CANopen spezifische SPS- Adressen ... 115
7.2.1 Parametrier- und Diagnosedaten ... 115
7.2.2 Netzvariablen ... 115
7.2.3 Zugriff auf Netzvariablen und I/O-Konfiguration ... 116
7.2.4 Zugriff auf direkte Netzvariablen ... 118
7.2.5 Zugriff auf Parametrier- und Diagnosefunktionen ... 118
7.3 SDO Funktionsbausteine ... 119
7.4 Die Parametrier- und Diagnosefunktionen ... 120
7.4.1 Funktion 1 CANopen Knotennummer einstellen und auslesen ... 121
7.4.2 Funktion 2 Datenübertragungsrate auslesen und einstellen ... 121
7.4.3 Funktion 7 CANopen- Fehlernummer auslesen ... 122
7.4.4 Funktion 8 CANopen- Firmware-Version auslesen ... 122
7.4.5 Funktion 20 CANopen- Hochlaufverzögerung einstellen ... 122
7.5 Fehlermeldungen ... 123
7.6 Die CANopen-spezifische Firmware-Bibliothek in MULTIPROG ... 124
7.6.1 Inhalt der Bibliothek CANopen_V001 ... 124
7.7 Erweiterungsmodul XF-CAN ... 125
7.7.1 Anzeigen und Anschlüsse ... 125

7.7.2 Technische Daten ... 125
8 Das Multi-Task-System ... 126
8.1 Übersicht ... 126
8.2 Anwender-Tasks ... 127
8.2.1 Zyklische Tasks ... 127
8.2.2 Ereignis-Tasks ... 128
8.2.3 System-Tasks ... 129
8.2.4 Default- Task ... 131
8.3 Anwender-Task-Information ... 132
8.4 Task-Prioritäten ... 134
8.5 Tasks und Watchdogs ... 135
8.6 Tasks einfügen und Programme zuweisen ... 136
9 Die SPS ... 138
9.1 Programmierung ... 138
9.2 SPS-Betriebszustände und Startverhalten ... 138
9.2.1 Betriebszustände ... 138
9.2.2 Wechseln der Betriebszuständen mit MULTIPROG ... 139
9.2.3 Startverhalten der SPS nach dem Einschalten der Versorgungsspannung ... 140
9.3 Systemvariablen ... 140
9.4 Bibliotheken und Funktionsbausteine in MULTIPROG ... 141
9.4.1 Bibliothek CANopen_Vxxx ... 143
9.4.2 Bibliothek CFB_Vxxx ... 143
9.4.3 Bibliothek CNC_Vxxx ... 144
9.4.4 Bibliothek Date_Time ... 144
9.4.5 Bibliothek MMI ... 145
9.4.6 Bibliothek PLC_Vxxx ... 145
9.4.7 Bibliothek Profibus_Vxxx ... 146
9.4.8 Bibliothek Serial ... 146
9.4.9 Bibliothek SchleicherLib_Vxxx ... 146
9.4.10 Bibliothek XCx7_Vxxx ... 146
9.5 Das SPS-Betriebssystem ProConOS ... 147
9.5.1 Die Initialisierungsdatei ProConOS.INI ... 147
10 Die CNC ... 148
10.1 Inbetriebnahme und Programmierung ... 148
10.2 Der SPS / CNC Koppelspeicher ... 148
10.3 Wichtige Hinweise zum Multi-Task-System und CNC ... 149
11 SLM-Antriebsmodul XP-SLM ... 150
11.1 Anzeigen und Anschlüsse ... 151
11.2 Belegung der Buchsen am Modul (Draufsicht) ... 152
11.3 Technische Daten SLM Antriebsmodul XP-SLM ... 154
11.4 Schnittstellen ... 155
11.5 Applikationsbeispiel ... 157
11.6 Verdrahtungs-Beispiel mit MultiAx ... 158
11.7 Verdrahtungs-Beispiel mit M'Ax ... 160
12 SERCOS-Modul XP-SRC ... 162
12.1 Anzeigen und Anschlüsse ... 163
12.2 Technische Daten ... 163
12.3 Applikationsbeispiel ... 164
12.4 Kommunikationsaufbau über den SERCOS-Ring ... 165
13 PROFIBUS-DP Module ... 166
13.1 PROFIBUS-DP Master XF-DPM ... 166
13.1.1 Anzeigen und Anschlüsse XF-DPM ... 167
13.1.2 Technische Daten XF-DPM ... 167
13.2 PROFIBUS-DP Slave XF-DPS ... 168
13.2.1 Anzeigen und Anschlüsse XF-DPS ... 169
13.2.2 Technische Daten XF-DPS ... 169
13.3 Die PROFIBUS-DP Konfigurationsdaten ... 170
14 Analog-Achsen mit MC_ANALOG und RIO A10-10 - ein Beispiel ... 173
14.1 Beispiel / Voraussetzung ... 173
14.2 Schritt 1: XRIO-Konfiguration ... 173
14.3 Schritt 2: Achsinterface RIO A10-10 ... 174

14.4 Schritt 3: Lageregel- und IPO Abtastrate ... 174
14.5 Schritt 4:Task-Struktur ... 174
14.6 Schritt 5: RIO A10-10 und MC_ANALOG ... 175
14.7 Datentyp TypRioA10_10 ... 177
14.8 Schritt 6: RIO A10-10 und CNC ... 179
14.9 Anmerkungen ... 180
15 Fehlermeldungen ... 181
15.1 Aufbau der Fehlermeldungen ... 181
16 Technische Daten ... 183
17 Abmessungen ... 185
18 Anhang ... 186
18.1 Warenzeichenvermerke ... 186
19 Sicherheitshinweise ... 187
19.1 Bestimmungsgemäße Verwendung ... 187
19.2 Personalauswahl und -qualifikation ... 187
19.3 Projektierung, Programmierung, Installation, Inbetriebnahme und Betrieb ... 188
19.4 Wartung und Instandhaltung ... 188
19.5 Gefahren durch elektrische Energie ... 188
19.6 Umgang mit verbrauchten Batterien ... 188
20 Abbildungsverzeichnis und Index ... 189

Darstellungskonventionen

Sicherheits- und Handhabungshinweise werden in dieser Programmieranleitung durch besondere Kennzeichnungen hervorgehoben:

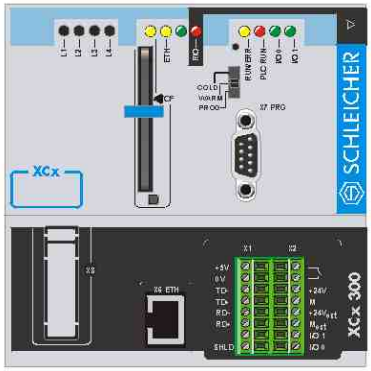
Bedeutet, daß Personen, das Automatisierungssystem oder eine Sache beschädigt werden kann, wenn die entsprechenden Hinweise nicht eingehalten werden.

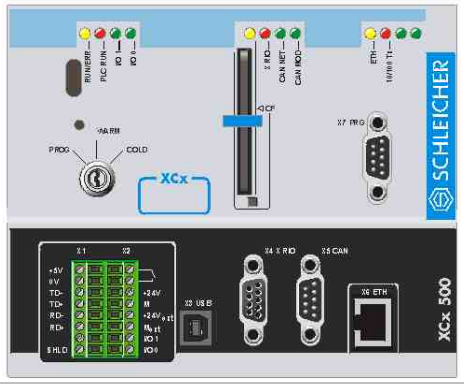
Hebt eine wichtige Information hervor, die die Handhabung des Automatisierungssystems oder den jeweiligen Teil der Betriebsanleitung betrifft.

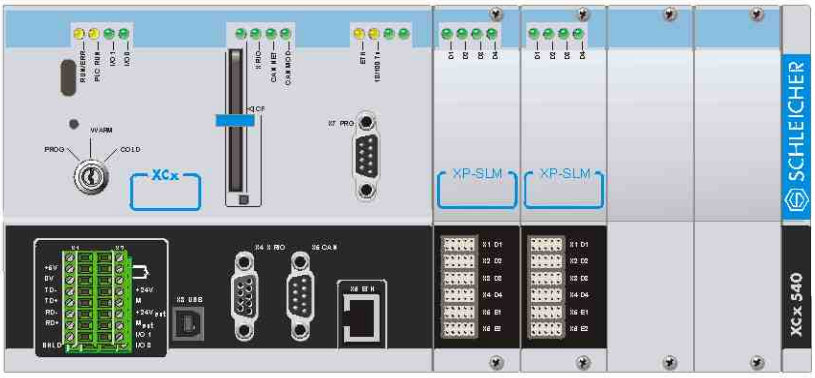
Weitere Objekte werden folgendermaßen dargestellt.

Objekt	Beispiel
Dateinamen	HANDBUCH.DOC
Menüs / Menüpunkte	*Einfügen / Graphik / Aus Datei*
Pfade / Verzeichnisse	*C:\Windows\System*
Hyperlinks	http://www.schleicher-electronic.com
Programmlisten	MaxTsdr_9.6 = 60 MaxTsdr_93.75 = 60
Tasten	<Esc> <Enter> (nacheinander drücken) <Ctrl+Alt+Del> (gleichzeitig drücken)
Bezeichner der Konfigurationsdaten	Q34 und Q.054
Namen der Koppelspeicher-Variablen	`cncMem.sysSect[n].flgN2P.bM345Act`

1 Übersicht

Economy-Variante XCx 300	
Mit SPS-Betriebssystem und optional mit CNC-Betriebssystem bis 4 NC-Achsen, sowie mit der Möglichkeit einen Erweiterungsmodul zu verwenden	

Basisvariante XCx 500	
Mit leistungsfähigem SPS-Betriebssystem	

Leistungsvariante XCx 540	
Mit SPS-Betriebssystem und leistungsfähigem CNC-Betriebssystem, sowie mit der Möglichkeit Erweiterungsmodule zu verwenden	

Bei allen Varianten ist es möglich digitale und analoge E/A in flexibler Konfiguration zu realisieren. Dazu werden Module aus dem Schleicher-System RIO (Remote Input Output) angeschlossen.

Alle Varianten verfügen über eine Vielzahl moderner, genormter Schnittstellen, die den Einsatz der XCx flexibel machen.

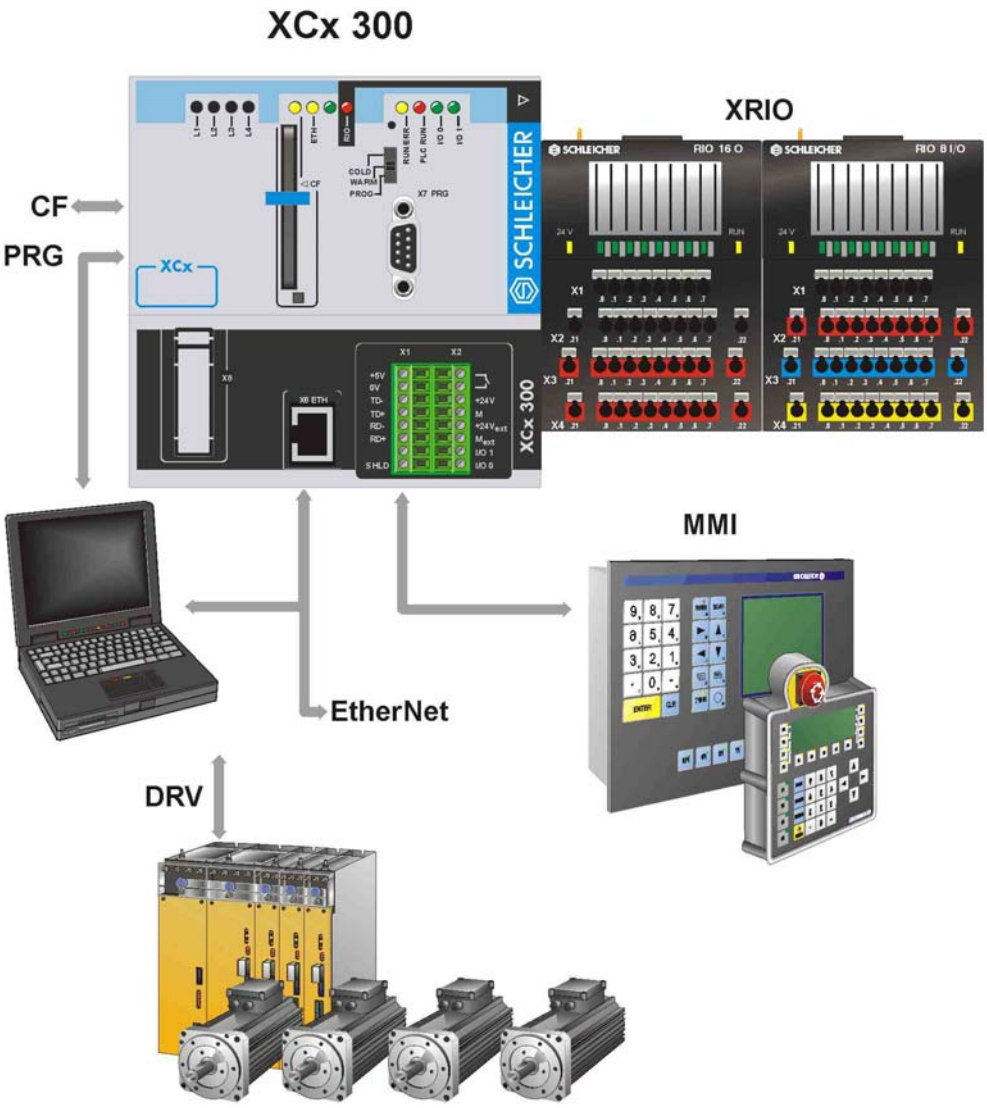


Abbildung 1: Übersicht XCx 300

CF	CompactFlash für die komfortable Speicherung der Anwenderprogramme und Firmware
MMI	Bedien- und Visualisierungsgeräte an der Bediengeräteschnittstelle RS 422
XRIO	Direkte Ankopplung von RIO-Erweiterungsmodulen für die flexible Realisierung von digitalen und analogen E/A, Zählern und Positonierung von NC-Achsen.
EtherNet	RJ-45-Anschluß für Vernetzung und Programmierung über TCP/IP
PRG	Diagnose- oder Programmiergeäte-Schnittstelle RS 232
DRV	Anschluss digitale Servoantriebe (optional über zusätzliche Steckkarte)

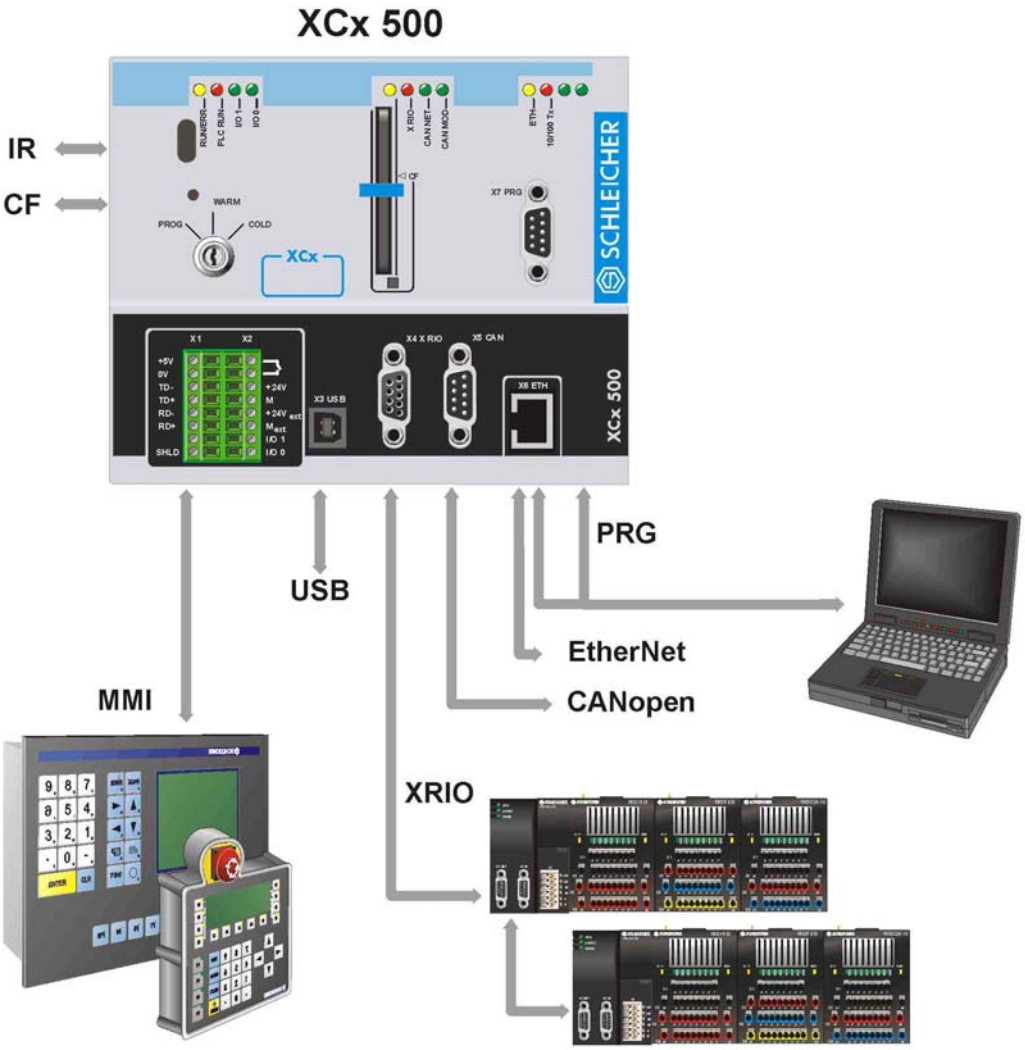


Abbildung 2: Übersicht XCx 500

IR	IRDa-Infrarotschnittstelle für Diagnosezwecke
CF	CompactFlash für die komfortable Speicherung der Anwenderprogramme und Firmware
MMI	Bedien- und Visualisierungsgeräte an der Bediengeräteschnittstelle RS 422
USB	USB-Schnittstelle als zusätzliche serielle Programmiergeräte-Schnittstelle
XRIO	Ankopplung von mehreren RIO-Busknoten für die flexible Realisierung von direkten digitalen und analogen E/A, Zählern und Positonierung von NC-Achsen.
CANopen	CANopen Feldbus-Anschluß
EtherNet	RJ-45-Anschluß für Vernetzung und Programmierung über TCP/IP
PRG	Diagnose- oder Programmiergeäte-Schnittstelle RS 232

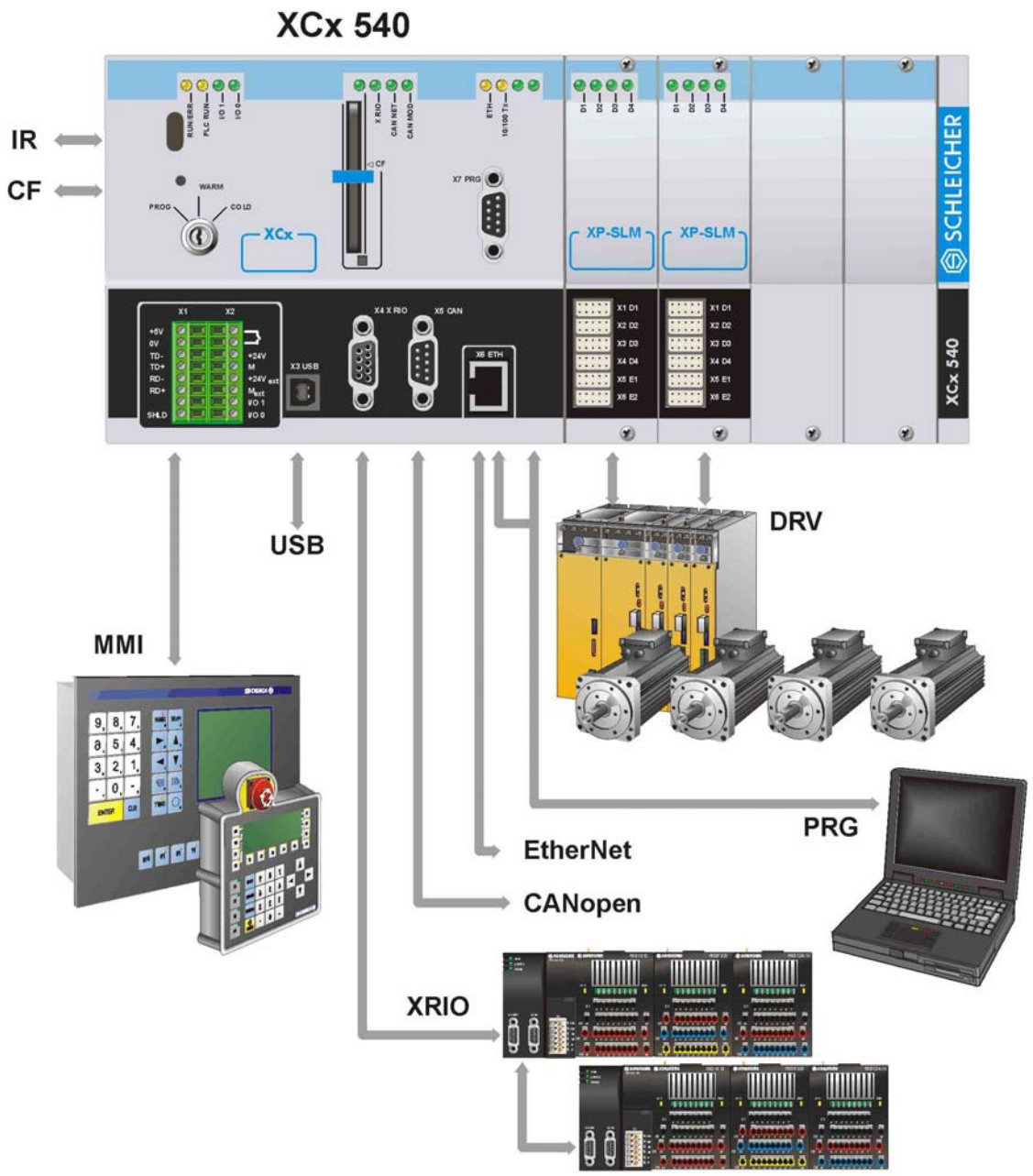


Abbildung 3: Übersicht XCx 540

IR	IRDa-Infrarotschnittstelle für Diagnosezwecke
CF	CompactFlash für die komfortable Speicherung der Anwenderprogramme und Firmware
MMI	Bedien- und Visualisierungsgeräte an der Bediengeräteschnittstelle RS 422
USB	USB-Schnittstelle als zusätzliche serielle Programmiergeräte-Schnittstelle
XRIO	Ankopplung von mehreren RIO-Busknoten für die flexible Realisierung von digitalen und analogen E/A, Zählern und Positonierung von NC-Achsen.
CANopen	CANopen Feldbus-Anschluß
EtherNet	RJ-45-Anschluß für Vernetzung und Programmierung über TCP/IP
PRG	Diagnose- oder Programmiergeäte-Schnittstelle RS 232
DRV	Funktionelle Erweiterung durch Erweiterungsmodule, hier die Anbindung von digitalen Servoantrieben über eine SML(Speed Loop Module)-Schnittstelle

2 Installation

2.1 XCx

Die XCx sind in geerdeten geschlossenen Gehäusen aus Metall (z.B. Schaltkasten, Schaltschrank) auf einer metallischen Trägerplatte elektrisch gut leitend zu installieren.

Zum Schutz der Geräte vor Entladung statischer Elektrizität muss sich das Bedienpersonal vor dem Öffnen von Schaltkästen oder Schaltschränken elektrostatisch entladen.

2.1.1 Montagelage

Die Montagelage der XCx 500 und 540 ist beliebig.

Die XCx 300 muss, wegen der direkt angesetzten RIO-Module, senkrecht montiert werden.

2.1.2 Montageabstände

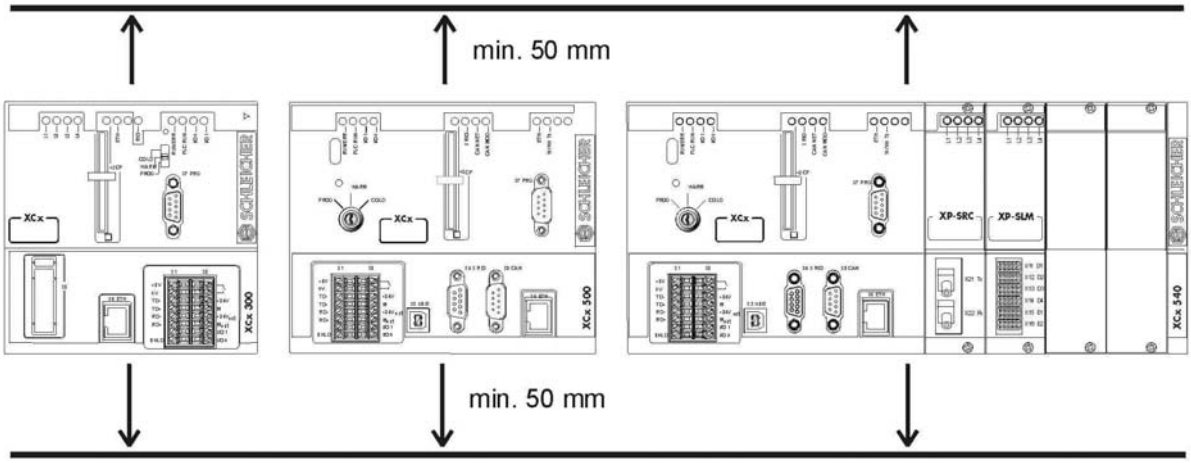


Abbildung 4: Montageabstände

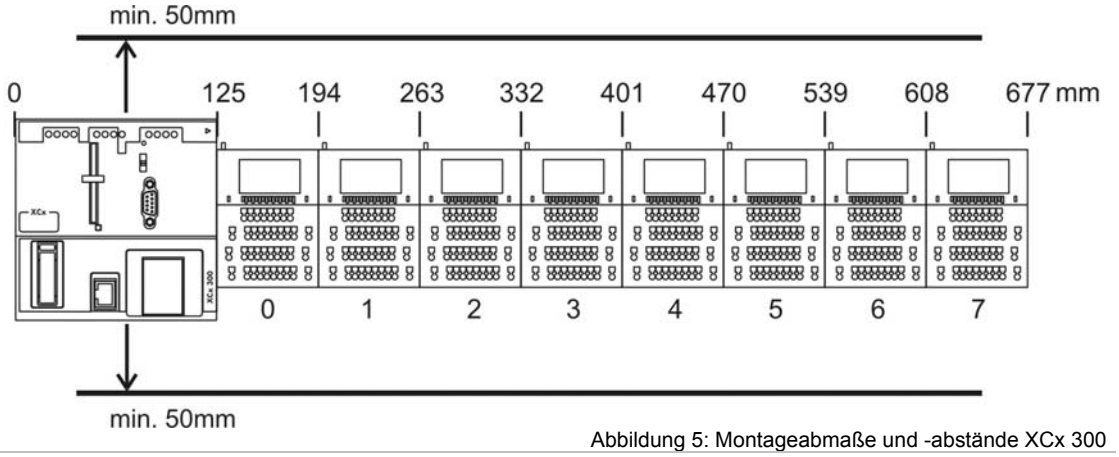


Abbildung 5: Montageabmaße und -abstände XCx 300

Der Maximalausbau mit 8 RIO-Modulen an der XCx 300 soll eingehalten werden. Ein weiterer Ausbau wird nicht empfohlen.

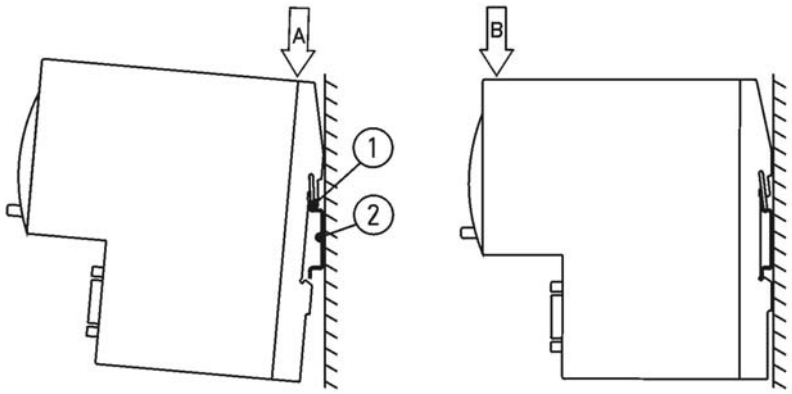
2.1.3 Montage der XCx 500 und 540

Hutschienemontage

- Es muss eine Hutschiene Typ TS 35 mm / 7,5 nach DIN EN 50022 verwendet werden.
- Die Hutschiene muss elektrisch gut leitend auf einer elektrisch leitenden, verwindungssteifen Trägerplatte montiert werden. Dazu sind Maschinenschrauben M5 mit Zahnscheiben im Abstand von 150 mm zu verwenden.

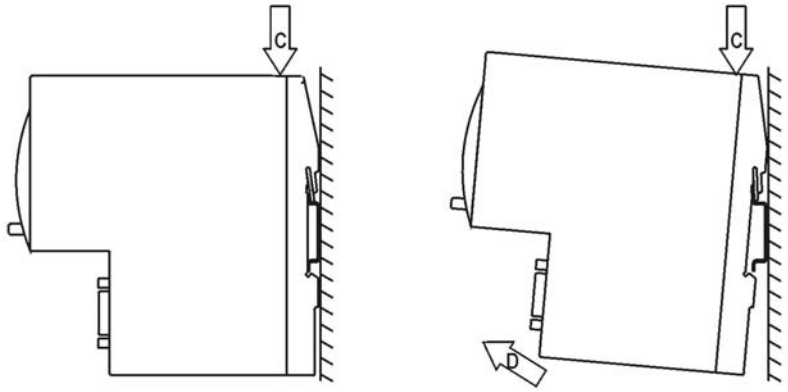
Die Hutschienenmontage ist für normale mechanische Belastung ausreichend.
Bei hoher mechanischer Belastung muss die XCx 500 und XCx 540 durch Aufschrauben befestigt werden.

Montage



A Gerät mit der Führung (1) auf die Hutschiene (2) aufsetzen.
B drücken, bis das Gerät einrastet.

Demontage



C Gerät nach unten drücken.

D Im gedrückten Zustand abheben.

Montage durch Aufschrauben

Die XCx 300 ist nicht für die Befestigung durch Aufschrauben vorgesehen.

Die Steuerung muss elektrisch gut leitend auf einer elektrisch leitenden Trägerplatte montiert werden.

Dazu sind 4 Maschinenschrauben M4 mit Zahnscheiben zu verwenden.

Die Befestigungslöscher für die Schraubmontage befinden sich in der Grundplatte des Gerätes an der rechten und linken Seite. Sie sind von außen frei zugänglich.

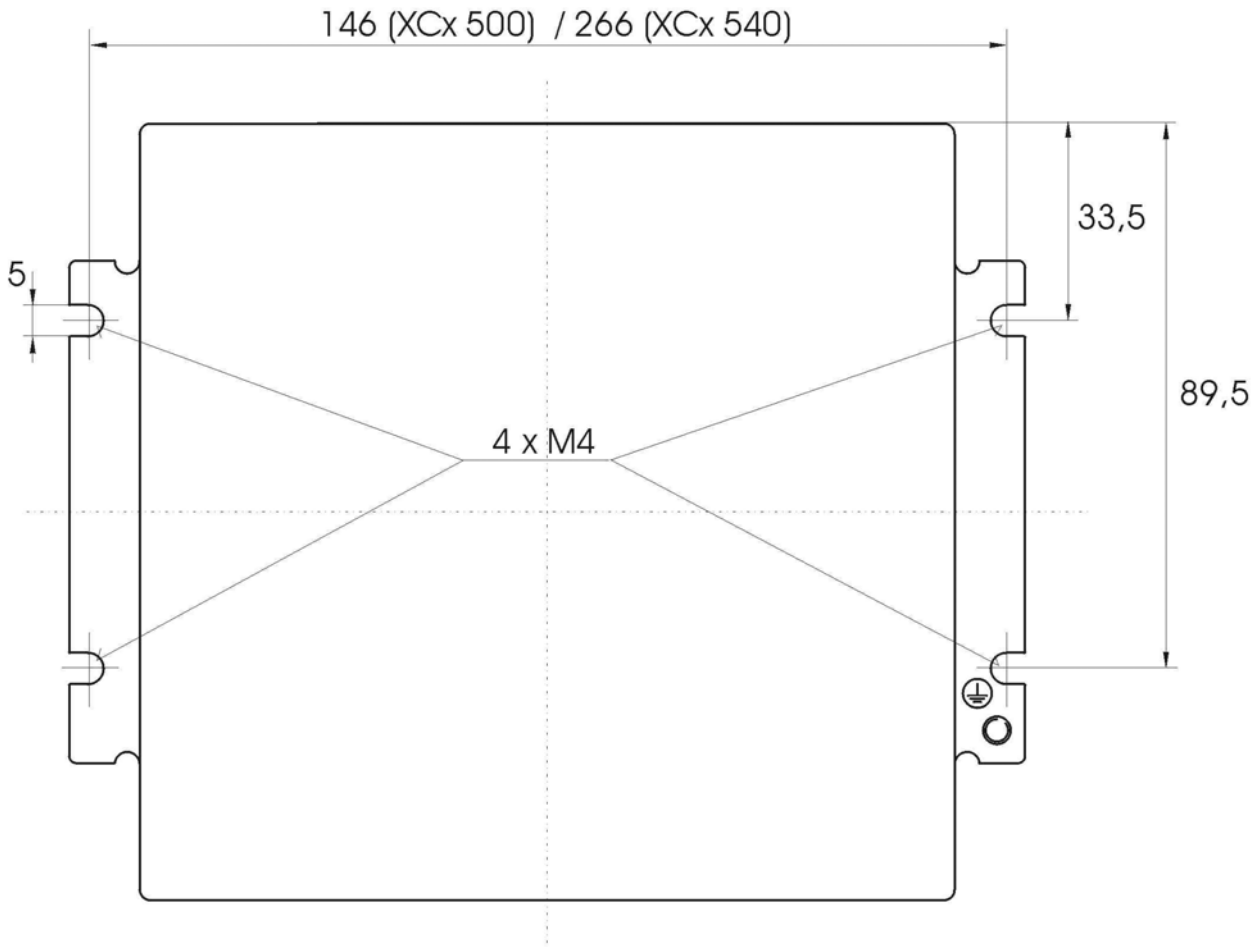


Abbildung 6: Abmaße der Befestigungslöcher für Schraubmontage der XCx 500/540

2.1.4 Montage der XCx 300 auf der Hutschiene

Bei der XCx 300 müssen die Steuerung und die RIO-Module auf der selben Hutschiene montiert werden.

Besonders wichtig ist die verwindungssteife Befestigung der Hutschiene, um die Kontaktsicherheit der Module untereinander zu gewährleisten.

- Es muss eine Hutschiene Typ TS 35 mm / 7,5 nach DIN EN 50022 verwendet werden.
- Die Hutschiene muss elektrisch gut leitend auf einer elektrisch leitenden, verwindungssteifen Trägerplatte befestigt werden.
- Zur Befestigung sind Maschinenschrauben M5 mit Zahnscheiben im Abstand von 150 mm zu verwenden.
- Vor der Betätigung der Federkraft-Klemmen der Module müssen die orangen Kontaktschieber, auf der Oberseite der Module, geöffnet sein.
- Vor dem Schließen der Kontaktschieber müssen die Module untereinander ausgerichtet werden.

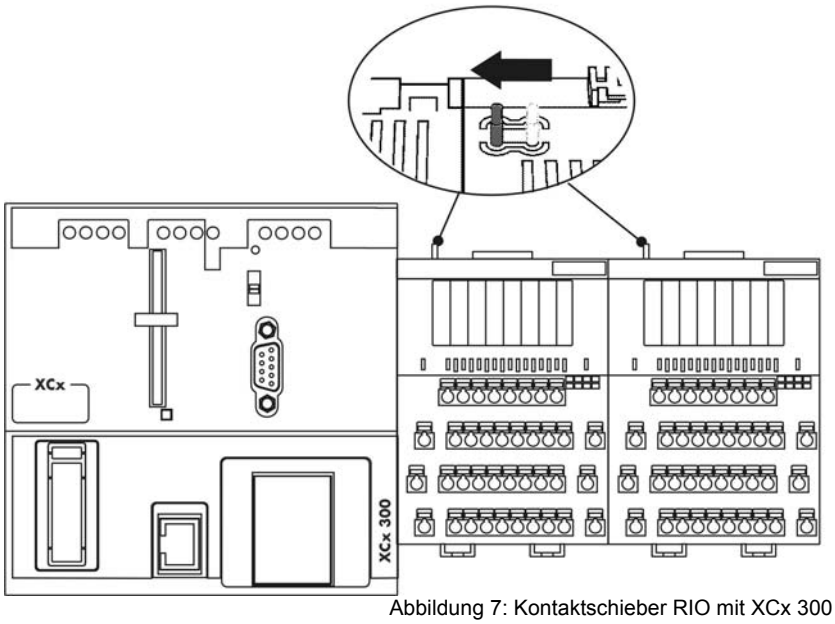


Abbildung 7: Kontaktschieber RIO mit XCx 300

- Steuerung und Module müssen mit Endklammern (im Bild unten schwarz dargestellt) gegen Verrutschen gesichert sein.

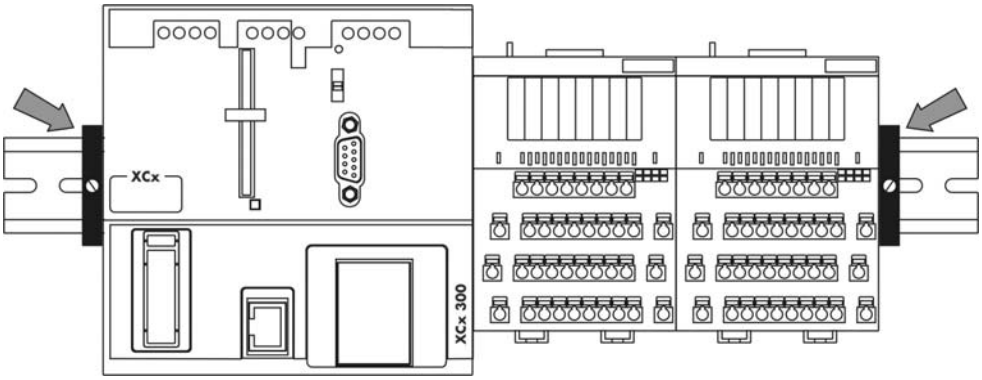
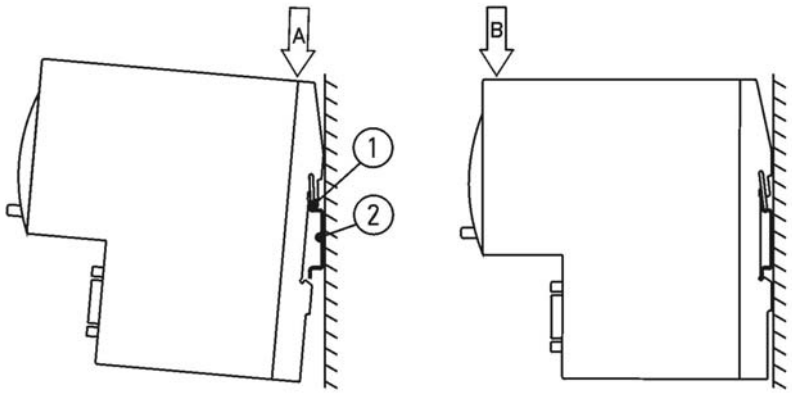


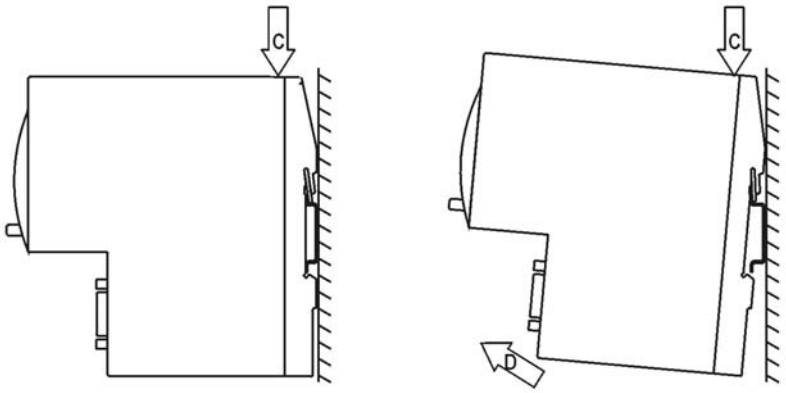
Abbildung 8: Endklammern an der XCx 300

Montage XCx 300



A Gerät mit der Führung (1) auf die Hutschiene (2) aufsetzen.
B Drücken, bis das Gerät einrastet.

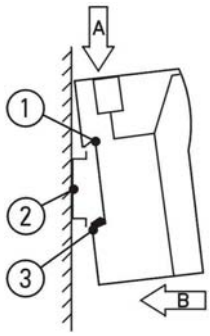
Demontage XCx 300



C Gerät nach unten drücken.

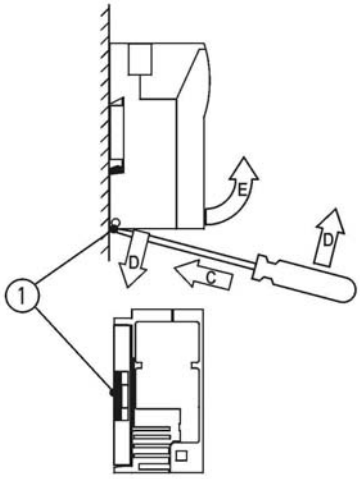
D Im gedrückten Zustand abheben.

Montage RIO-Module



A Gerät leicht geneigt mit der Führung (1) auf die Hutschiene (2) aufsetzen.
B An die Hutschiene (2) drücken, bis der Riegel (3) einrastet.

Demontage RIO-Module



Den orangen Kontaktschieber auf der Moduloberseite öffnen (nach rechts schieben).
C Schraubendreher in den Riegel (1) stecken.
D Riegel mit dem Schraubendreher nach unten hebeln. Der Riegel verbleibt in der geöffneten Position.
E Gerät ankippen und abnehmen. Danach den Riegel (1) wieder zurückschieben.

2.1.5 Anschluss der Spannungsversorgung

XCx 500 und XCx 540

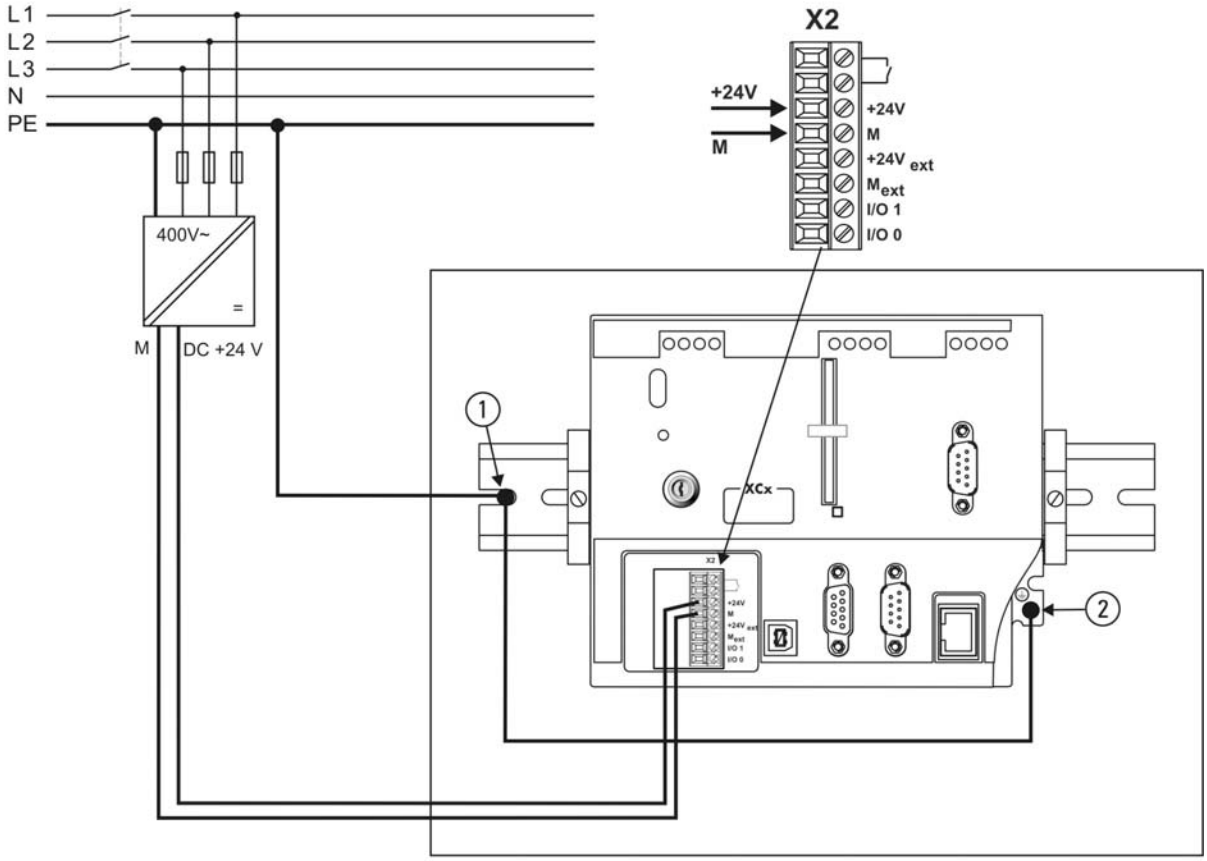


Abbildung 9: Anschluss der Spannungsversorgung XCx 500 und XCx 540

Wird keine Schirmschiene verwendet, kann die metallische Trägerplatte an einem zentralen Punkt (1) mit Masse (PE) verbunden werden.

Auf der Gehäuserückwand kann bei Bedarf der Erdungspunkt (2) mit einer Maschinenschraube M4 mit Zahnscheibe angeschlossen werden.

XCx 300

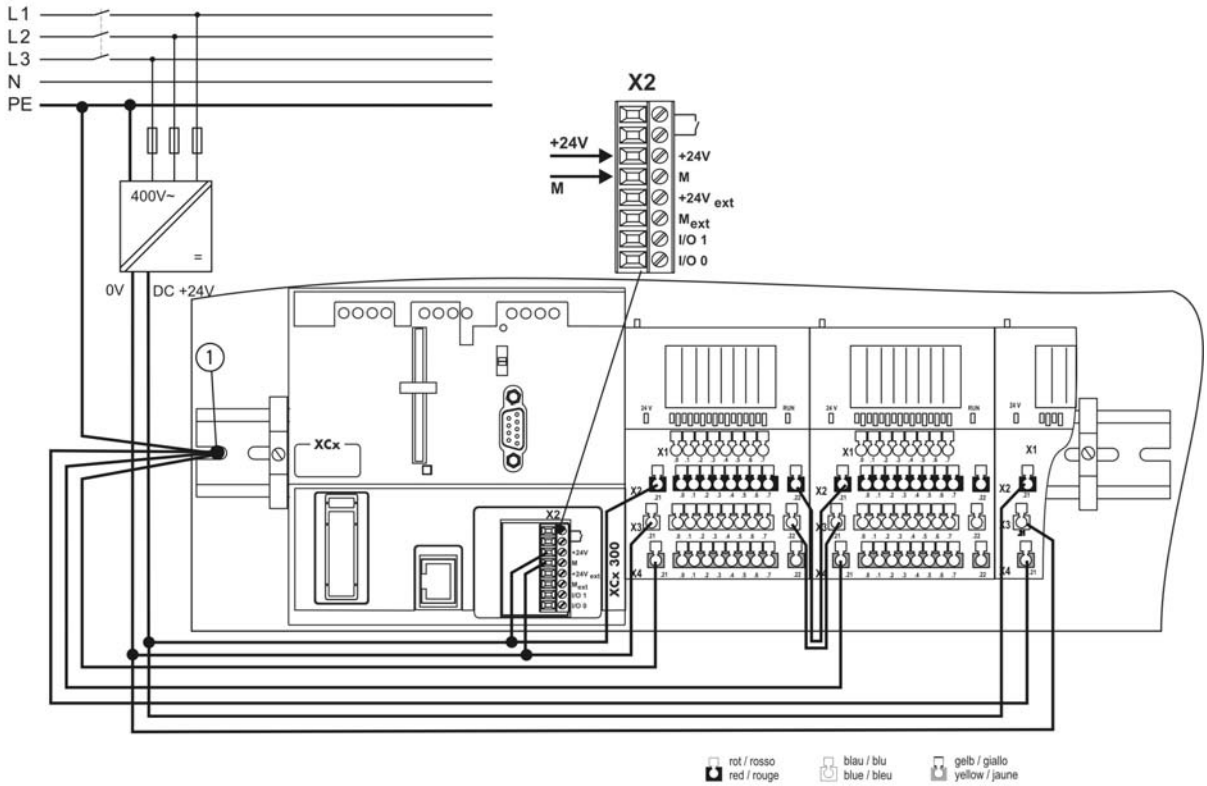


Abbildung 10: Anschluss der Spannungsversorgung XCx 300

Wird keine Schirmschiene verwendet, kann die metallische Trägerplatte an einem zentralen Punkt (1) mit Masse (PE) verbunden werden.

Der Maximalausbau mit 8 RIO-Modulen an der XCx 300 soll eingehalten werden. Ein weiterer Ausbau wird nicht empfohlen.

Hinweise zur Erdung, Klemmen und Verdrahtung der RIO-Module siehe auch Kapitel XRIO.

2.2 XRIO

Mit XRIO wird die direkte I/O-Ebene der XCx 500 und XCx 540 realisiert. Dazu werden die Erweiterungsmodule des modularen Systems RIO (Remote **Input Output)** verwendet.

Die RIO-Module sind in geerdeten geschlossenen Gehäusen aus Metall (z.B. Schaltkasten, Schaltschrank) auf einer metallischen Trägerplatte elektrisch gut leitend zu installieren.

Zum Schutz der Module vor Entladung statischer Elektrizität muss sich das Bedienpersonal vor dem Öffnen von Schaltkästen oder Schaltschränken elektrostatisch entladen.

2.2.1 Montagelage

Die senkrechte Montagelage muss eingehalten werden.

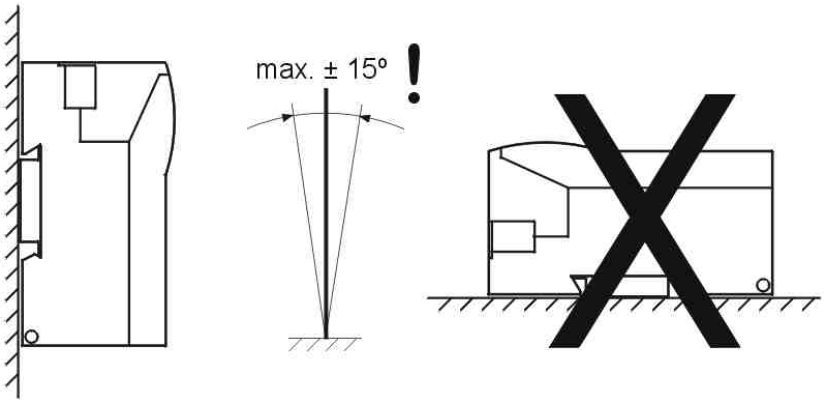


Abbildung 11: Montagelage XRIO

2.2.2 Montageabmaße und -abstände

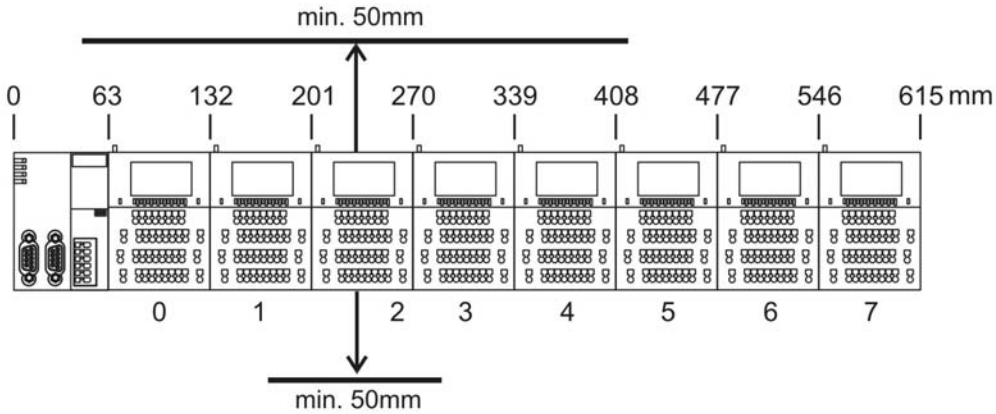


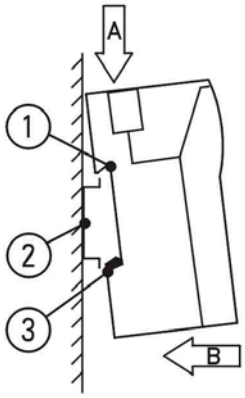
Abbildung 12: Montageabmaße und -abstände XRIO mit Buskoppler

2.2.3 Hutschienenmontage

Hutschiene Type TS 35mm/7,5 nach DIN EN 50022 verwenden.

Die Hutschiene muss elektrisch gut leitend auf der Trägerplatte montiert werden.

Montage



A Gerät leicht geneigt mit der Führung (1) auf die Hutschiene (2) aufsetzen.
B An die Hutschiene (2) drücken, bis der Riegel (3) einrastet.

Abbildung 13: Montage XRIO

Die Module müssen direkt nebeneinander montiert werden und gegen Verrutschen mit einer Endklammer gesichert werden.

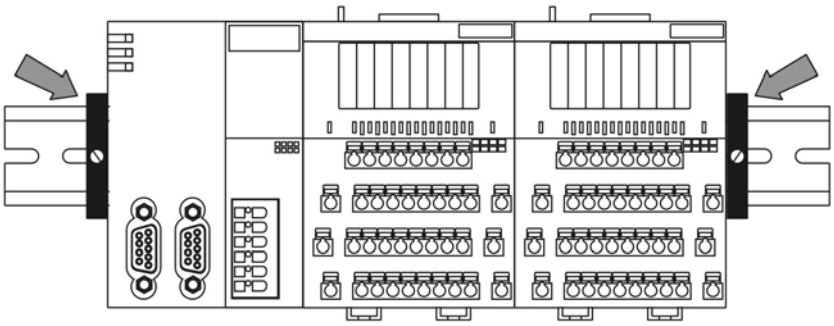
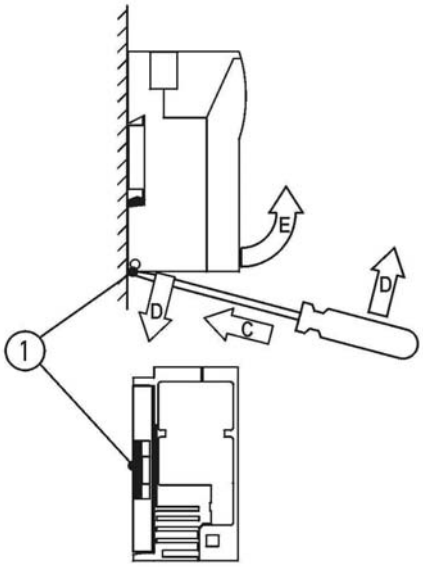


Abbildung 14: Endklammern XRIO

Demontage



Den orangen Kontaktschieber auf der Moduloberseite öffnen (nach rechts schieben).
C Schraubendreher in den Riegel (1) stecken.
D Riegel mit dem Schraubendreher nach unten hebeln. Der Riegel verbleibt in der geöffneten Position.
E Gerät ankippen und abnehmen. Danach den Riegel (1) wieder zurückschieben.

Abbildung 15: Demontage XRIO

2.2.4 Verbindung der Module untereinander

Die orangen Kontaktschieber auf der Moduloberseite verbinden die Kommunikationsübertragung zwischen den Modulen und dem Buskoppler.

Die Kontaktschieber müssen vor der Betätigung der Federkraftklemmen geöffnet werden, um die mechanische Beanspruchung der Kontaktstellen zu verringern.

Vor dem Schließen der Kontaktschieber sind die Module zueinander auszurichten.

Den Kontaktschieber beim Schließen nicht mit Gewalt betätigen.

Die Kontaktschieber müssen vor der Inbetriebnahme geschlossen werden.

Die Kontaktschieber dürfen während des Betriebes nicht geöffnet werden.

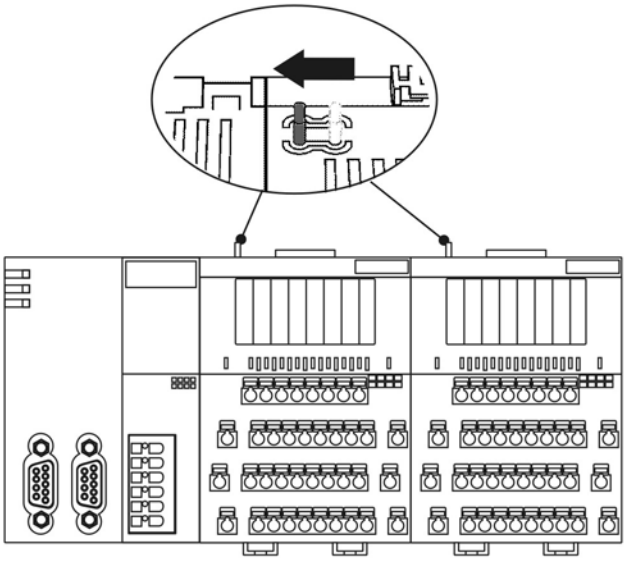


Abbildung 16: Kontaktschieber RIO mit XRIO-Buskoppler

2.2.5 Federkraftklemmen des XRIO-Buskopplers

Am Koppler werden zum Anschluss der Spannungsversorgung Federkraftklemmen eingesetzt.

Zum Betätigen der Federkraftklemme wird ein Schraubendreher mit einer Klinge 0,6 x 3,5 nach DIN 5264 B benötigt. Der Schaftdurchmesser darf die Schneidenbreite von 3,5 mm nicht überschreiten.

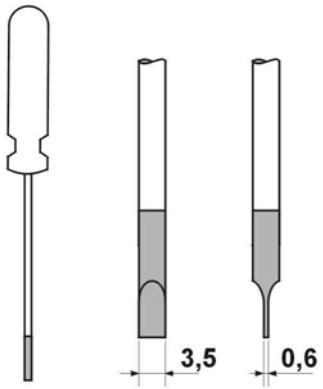
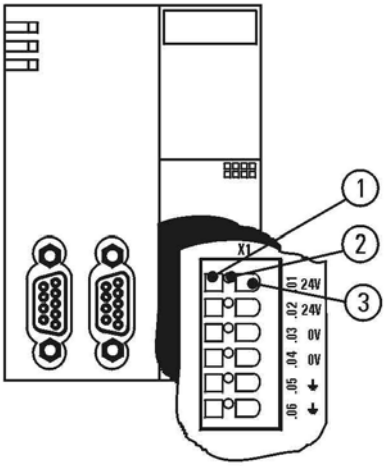


Abbildung 17: Schneidenbreite Schraubendreher



1 Betätigungsöffnung
2 Messpunkt
3 Klemmraum

Abbildung 18: Federkraftklemmen des Kopplers

Öffnen der Klemme

Schraubendreher senkrecht in die Betätigungsöffnung (1) drücken.

Draht in den Klemmraum (3) einführen.

Schließen der Klemme

Schraubendreher aus der Betätigungsöffnung (1) entfernen.

Messen an der Klemme

Jede Klemme besitzt einen Messpunkt (2), der mit einer üblichen 2mm-Messspitze zugänglich ist.

2.2.6 Federkraftklemmen der RIO-Erweiterungsmodule

Lieferzustand: Klemmen geöffnet

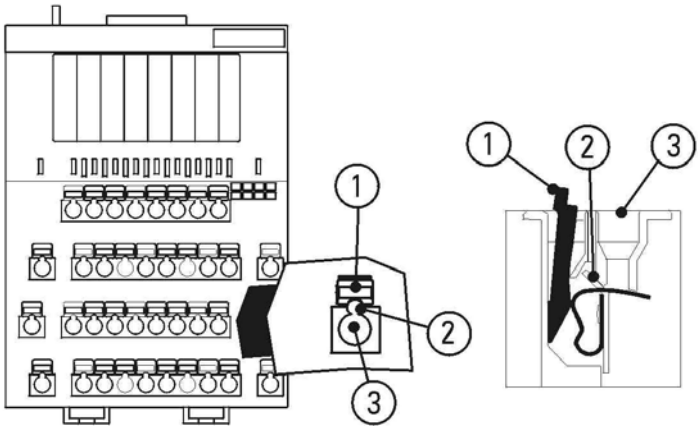
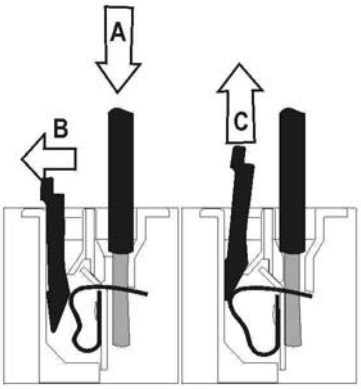


Abbildung 19: Federkraftklemmen der Erweiterungsmodule

Die Klemmen sind mit einem Klemmkeil (1) vorgespannt, der Klemmraum (3) ist geöffnet. Jede Klemme besitzt einen Messpunkt (2), der mit einer üblichen 2 mm Messspitze zugänglich ist.

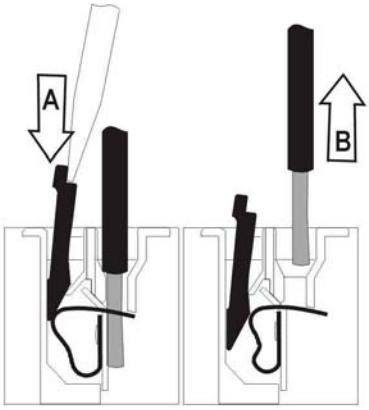
Schließen der Klemme



A Draht in den Klemmraum einführen.
Klemmkeil in Richtung **B** drücken.
Durch die Spannung der Feder wird der Klemmkeil nach oben **C** gedrückt, er verbleibt in der Klemme.

Öffnen der Klemme

Vor dem Öffnen der Klemmen muss der Kontaktschieber des Moduls geöffnet sein, um die mechanische Beanspruchung der Kontakstellen zu verringern.

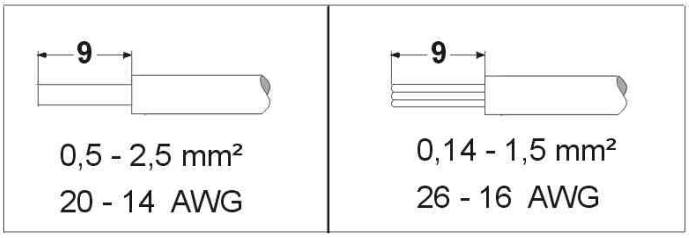


Klemmkeil mit Schraubendreher in Richtung **A** schieben. Der Klemmkeil hebelt die Federkraftklemme auf und verbleibt in dieser Stellung. Kabel in Richtung **B** entnehmen.

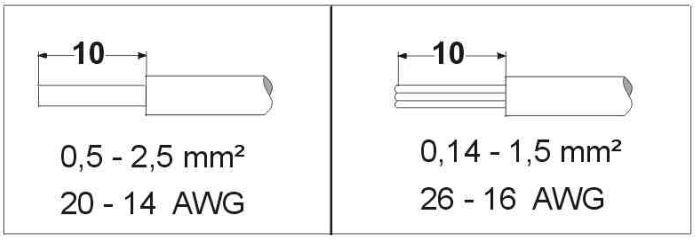
Die Federkraftklemme kann auch ohne Klemmkeil geöffnet werden. Dazu kann an Stelle des Klemmkeiles ein Schraubendreher verwendet werden.

2.2.7 Anschlussquerschnitte und Abisolierlänge RIO

XRIO Buskoppler



RIO Erweiterungsmodule



2.2.8 Anschluss der Spannungsversorgung für XRIO

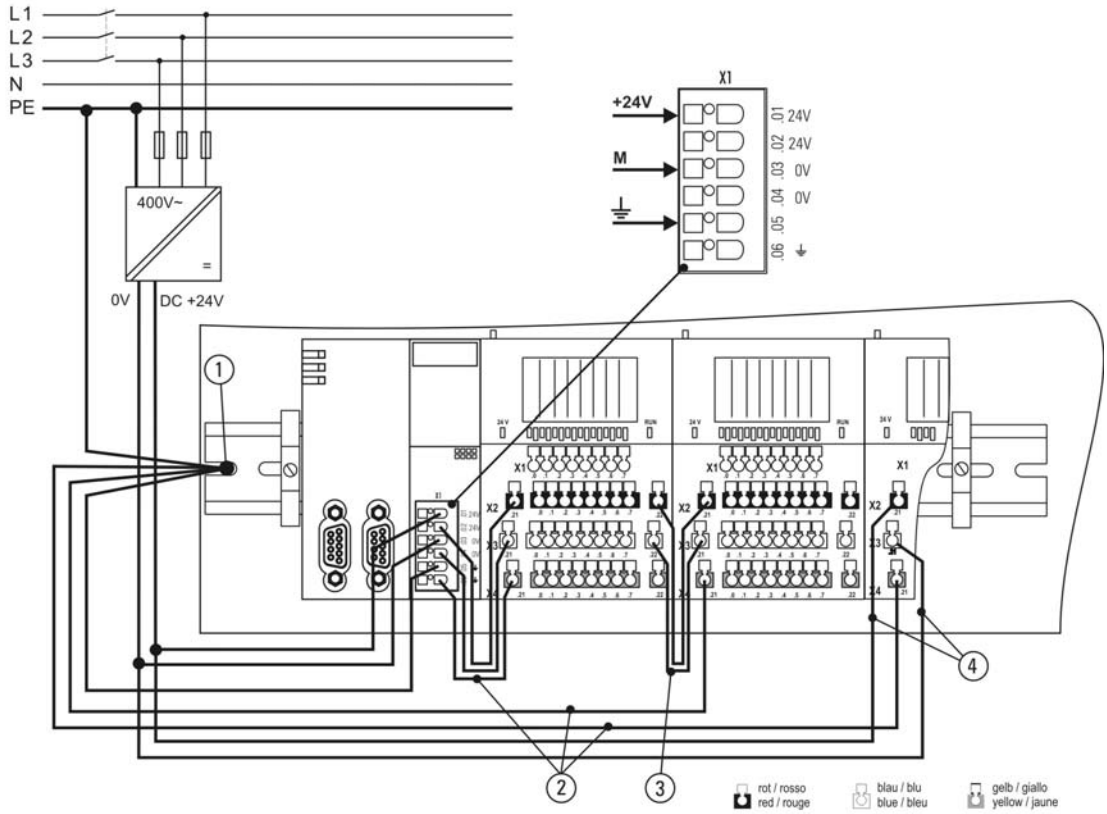


Abbildung 20: Anschluss der Spannungsversorgung XRIO an der XCx 500 und XCx 540

Das Anschlussbild ist für einen Buskoppler mit Erweiterungsmodulen des Typs RIO 8 I/O gültig. Die Erklärungen zu den Anschlussbildern sind in den folgenden Abschnitten zu finden.

Masseverbindung der Hutschiene (1)

Die zur Aufnahme der Module vorgesehene Hutschiene muss großflächig und gut leitend auf der Trägerplatte montiert werden.

Wird keine Schirmschiene verwendet kann die metallische Trägerplatte an einem zentralen Punkt (1) mit Masse (PE) verbunden werden.

Masseverbindung des Kopplers und der Erweiterungsmodule (2)

Die Koppler sowie das 8-fach-Erweiterungsmodul (8I/O) besitzen eine mit dem Erdungssymbol gekennzeichnete Anschlußklemme. Diese Klemme ist über eine möglichst kurze Leitung (2,5 mm2) mit Masse (oder mit PE-Potential) zu verbinden, um die Störunempfindlichkeit zu erhöhen.

Optimale EMV wird erreicht, wenn das erste Erweiterungsmodul rechts neben dem Koppler über die Potential-Weiterleitungsklemme des Kopplers versorgt werden.
Alle anderen Module können auch einzeln versorgt werden.

Zum Ableiten von EMV-Störungen dient die im Klemmfuß der Module integrierte Kontaktfeder. Diese Feder stellt die Verbindung des Schirmpotentials der Leiterplatte zur Hutschiene her. Eine Montage ohne oder mit defekter Kontaktfeder ist nicht zulässig.

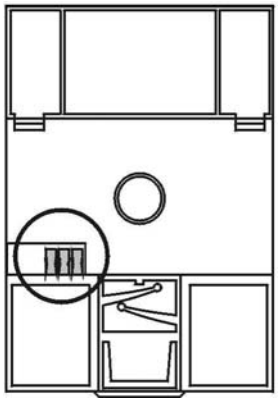


Abbildung 21: Kontaktfedern im Klemmfuß

Weiterschleifen der Versorgungsspannung (3)

Zur Herstellung einer optimalen Verdrahtung können die Versorgungsspannungen von Modul zu Modul weitergeschleift werden. Bei Modulen mit digitalen Ausgängen muss die Strombelastung der Weiterleitungsklemmen beachtet werden. Bei Überschreitung des Maximalstromes muss eine Zwischeneinspeisung vorgesehen werden (5).

Zwischeneinspeisung der Versorgungsspannumg (4)

Beim Weiterschleifen der Versorgungsspannung von Modul zu Modul über die Weiterleitungsklemmen muss die Strombelastung der Klemmen beachtet werden.

Es ist unbedingt darauf zu achten, dass die Belastung einer Klemme **Imax = 8A nicht überschreitet.**

Es sind Zwischeneinspeisungen vorzunehmen, wenn der Maximalstrom überschritten werden kann.

Versorgungsspannung für Module mit Kombikanälen

Bei Modulen mit digitalen Kombikanälen ist darauf zu achten, dass das Anlegen von 24 V an einen Kombikanal ohne Einspeisung der Versorgungsspannung unzulässig ist.
Es kommt sonst über die Ausgangsschaltung des Kombikanales zur Rückspeisung in den Versorgungsspannungsanschluß des Moduls, in dessen Folge eine Fehlfunktion oder Zerstörung der Ausgangsschaltung auftreten kann.

Es ist bei Not-Aus nicht zulässig nur die Spannungsversorgung der Module mit Kombikanälen abzuschalten. Es muss die Spannungsversorgung der Module gleichzeitig mit der Spannungsversorgung der Sensoren und Aktoren abgeschaltet werden.
Sinngemäß betreffen die oben gemachten Aussagen auch digitale Ausgangskanäle, wenn sie in fehlerhafter Weise mit 24 V beschaltet werden.

Leitungsführung

Die Verdrahtung der Module soll immer senkrecht nach unten verlegt werden, um das Ausklappen der Module zu ermöglichen.

2.2.9 Buskabel XRIO

Als Buskabel sind grundsätzlich geschirmte Kabel zu verwenden. Die maximale Kabellänge beträgt 10 m zwischen zwei Anschlusspunkten.

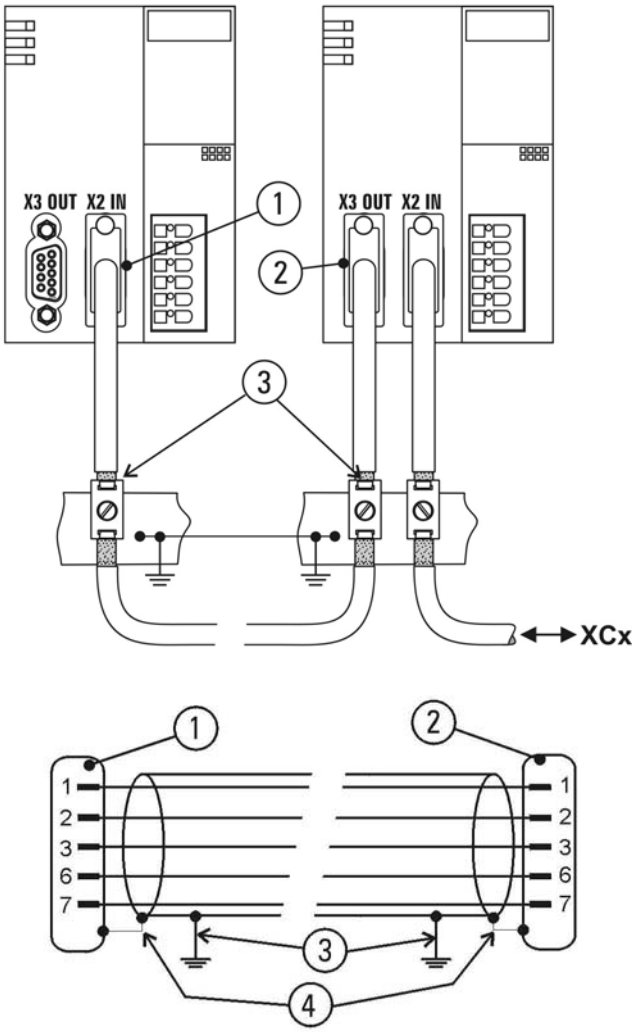


Abbildung 22: Buskabel XRIO

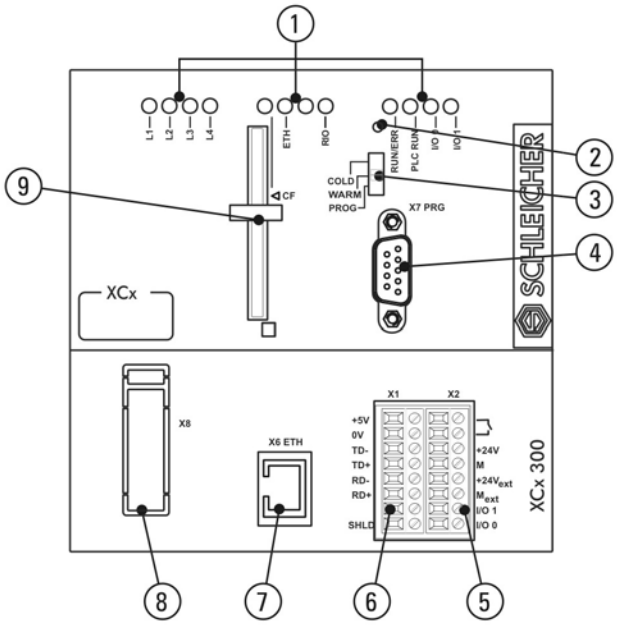
1 Steckverbinder D-Sub, 9-polig, Stifte
2 Steckverbinder D-Sub, 9-polig, Stifte
3 Schirmerdung kurz vor dem Gerät
4 Beidseitige Schirmauflage auf den Steckverbindergehäusen

Es ist auch möglich ein handelsübliches 1:1 Kabel zu verwenden, bei dem alle Anschlüsse verdrahtet sind.

3 Anzeigen, Anschlüsse und Bedienelemente

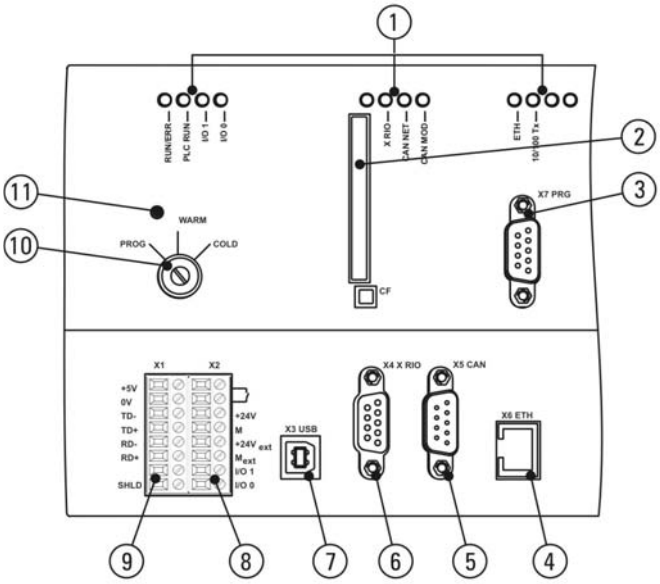
3.1 Übersicht

XCx 300



1	LED-Anzeigen
2	RESET-Taster
3	Betriebsartenschalter
4	X7 PRG RS232-Schnittstelle für Programmiergeräte
5	X2 Anschluss für Versorgungsspannung, direkte E/A und Betriebs-Kontakt
6	X1 RS422-Schnittstelle für den Anschluss von Bedien- und Anzeigegeräten
7	X6 ETH RJ 45 EtherNet-Anschluss
8	Steckplatz für einen Erweiterungsmodul
9	Compact Flash (CF)

XCx 500 / XCx 540



1	LED-Anzeigen
2	Compact Flash (CF)
3	X7 PRG RS232-Schnittstelle für Programmiergeräte
4	X6 ETH RJ 45 EtherNet-Anschluss
5	X5 CANopen Feldbusanschluss
6	X4 X RIO direkte Ankopplung von RIO-Modulen
7	X3 USB- für zusätzliche Programmiergeräte
8	X2 Anschluss für Versorgungsspannung, direkte E/A und BUSY-Kontakt
9	X1 RS422-Schnittstelle für den Anschluss von Bedien- und Anzeigegeräten
10	Betriebsartenschalter
11	RESET-Taster

3.2 LED-Anzeigen

XCx 300

LED-Bezeichnung	Farbe	Zustand	Bedeutung
L1 / L2 / L3 / L4			frei (für optionalen Erweiterungsmodul reserviert)
CF			Compact Flash
	grün	ein	Speicherzugriff
	rot	ein	Zugriffsfehler
ETH			EtherNet Netzwerk
	grün	blinkend	Netzwerkzugriffe
	rot	ein	Netzwerkfehler
RIO			RIO Direktanschluss
	grün	ein	Operational
	grün	blinkend	Pre-Operational
	rot	ein	Busfehler
	gelb	ein	Frame-Fehler
RUN/ERROR			CPU-Status
		aus	Betriebsspannung fehlt
	grün	ein	Betriebsspannung ok, kein Fehler
	rot	ein	CPU läuft nicht (Watchdog)
		blinkend	CPU hat fatalen Fehler festgestellt
PLC RUN			SPS-Status
		aus	SPS Stop
	gelb	ein	SPS läuft
		blinkend	SPS läuft, aber Ausgänge sind abgeschaltet (Betriebsbereit-Relais abgefallen)
I/O 0 **I/O 1**			Kombi I/O
	gelb	aus	Eingang / Ausgang nicht gesetzt
		ein	Eingang / Ausgang gesetzt

XCx 500 / XCx 540

LED-Bezeichnung	Farbe	Zustand	Bedeutung
RUN/ERROR			CPU-Status
		aus	Betriebsspannung fehlt
	grün	ein	Betriebsspannung ok, kein Fehler
	rot	ein	CPU läuft nicht (Watchdog)
		blinkend	CPU hat fatalen Fehler festgestellt
PLC RUN			SPS-Status
		aus	SPS Stop
	gelb	ein	SPS läuft
		blinkend	SPS läuft, aber Ausgänge sind abgeschaltet (Betriebsbereit-Relais abgefallen)
I/O 1 **I/O 2**			Kombi I/O
	gelb	aus	Eingang / Ausgang nicht gesetzt
		ein	Eingang / Ausgang gesetzt
CF			Compact Flash
	grün	ein	Speicherzugriff
	rot	ein	Zugriffsfehler
XRIO			RIO Direktanschluss
	grün	ein	Operational
	grün	blinkend	Pre-Operational
	rot	ein	Busfehler
	gelb	ein	Frame-Fehler
CAN NET			CAN Netzwerkwerkstatus
		aus	CAN State Prepared
	grün	ein	CAN State Operational
		blinkend	CAN State Pre-Operational
	rot	ein	Bus Off
		blinkend	CAN-Fehler
CAN MOD			CAN Modulstatus
	grün	ein	CAN-Stack initialisiert
		blinkend	ungültige CAN-Konfiguration
	rot	ein	Steuereinheit nicht bereit oder schwerer Fehler
		blinkend	Fehler in der Steuerung
ETH			EtherNet Netzwerk
	grün	blinkend	Netzwerkzugriffe
	rot	ein	Netzwerkfehler
10/100 Tx			EtherNet Übertragungsrate
	gelb	ein	Übertragungsrate 100 Mbit/s
		aus	Übertragungsrate 10 Mbit/s

3.3 Compact Flash

Auf der CF Card ist das Betriebssystem der Steuerung, wichtige Konfigurationsdateien sowie das SPS- Programm (Bootprojekt und SPS- Source) und die für den Betrieb der CNC notwendigen Dateien abgespeichert. CF Zugriffe werden durch die CF LED angezeigt. Die CF Card kann bei eingeschalteter Steuerung gezogen oder gesteckt werden (hot plugabel), es müssen allerdings folgende Festlegungen beachtet werden:

Die CF Card darf nur gezogen werden wenn:

- kein Zugriff erfolgt (CF LED muss aus sein)
- die Steuerung im Betriebszustand STOP steht

Die CF Card darf nur gesteckt werden wenn:

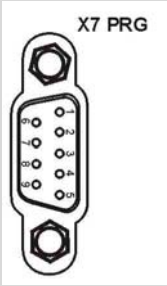
- die Steuerung im Betriebszustand STOP steht

Datenverluste oder die Auslösung des Watch Dog können bei Nichtbeachtung erfolgen.

Wird RESET ausgelöst oder die Steuerung ausgeschaltet, während ein Zugriff auf die CF Card erfolgt (CF LED leuchtet grün), tritt Datenverlust auf der CF Card auf.

3.4 X7 PRG RS232-Schnittstelle für Programmiergeräte

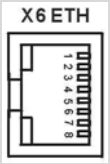
X7 D-Sub, 9-polig, Stecker



Pin	Bezeichnung		Erläuterung
1	(DCD)	(Data Carrier detect)	nicht belegt
2	RD	Receive Data	Eingang
3	SD	Send Data	Ausgang
4	DTR	Data Terminal ready	Brücke nach Pin 6
5	GND	Logic Ground	Nicht für Schirm
6	DSR	Data set ready	Brücke nach Pin 4
7	RTS	Request to send	Brücke nach Pin 8
8	CTS	Clear to send	Brücke nach Pin 7
9	(Ri)	(Ring Indicator)	nicht belegt

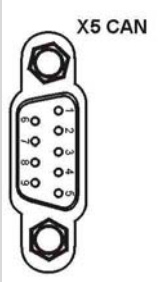
3.5 X6 ETH RJ 45 EtherNet-Anschluss

X6 RJ 45

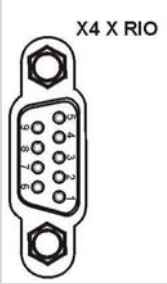


Pin	Bezeichnung	Erläuterung
1	TX+	Sendedaten plus
2	TX-	Sendedaten minus
3	RX+	Empfangsdaten plus
4	nc	nicht angeschlossen
5	nc	nicht angeschlossen
6	RX-	Empfangsdaten minus
7	nc	nicht angeschlossen
8	nc	nicht angeschlossen

3.6 X5 CANopen Feldbusanschluss

X5 D-Sub, 9-polig, Stecker			
	Pin	Bezeichnung	Erläuterung
	1	nc	nicht angeschlossen
	2	CAN_L	
	3	V-	Ground
	4	nc	nicht angeschlossen
	5	Drain	Schirmanschluß optional
	6	V-	Ground
	7	CAN_H	
	8	nc	nicht angeschlossen
	9	V+	Stromversorgung

3.7 X4 X RIO Anschluss

X4 D-Sub, 9-polig, Buchse			
	Pin	Bezeichnung	
	1	**RD+**	Empfangsdaten plus
	2	**TD+**	Sendedaten plus
	3	**0 V**	Versorgungsspannung 0 V
	4		nicht angeschlossen
	5	**+5 V**	Versorgungsspannung
	6	**RD-**	Empfangsdaten minus
	7	**TD-**	Sendedaten minus
	8		nicht angeschlossen
	9		nicht angeschlossen

3.8 X3 USB-(Universal Serial Bus) Anschluss

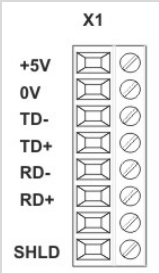
In Vorbereitung

Steckverbinder Buchse Typ B entsprechend der Universal Serial Bus Specification.

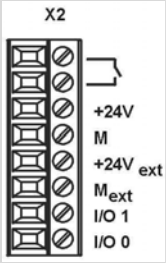
Der USB-Steckverbinder in Richtung Host (Hub) ist immer vom Typ A, Richtung Gerät vom Typ B. Damit wird eine Vertauschung oder der Aufbau einer Ringverbindung verhindert.

X3 USB-Anschluss Buchse Typ B			
	Pin	Bezeichnung	Erläuterung
	1	VCC (VBus)	Stromversorgung über den Bus (busbetrieben)
	2	- Data	Daten -
	3	+ Data	Daten +
	4	Ground (Gnd)	Ground der Stromversorgung

3.9 X1 RS422-Schnittstelle für den Anschluss von Bedien- und Anzeigegeräten

X1 Schraubblockklemme 8-polig		
	Bezeichnung	Erläuterung
	**+5 V**	Versorgungsspannung +5 V
	0 V	Versorgungsspannung 0 V
	TD-	Sendedaten minus
	TD+	Sendedaten plus
	RD-	Empfangsdaten minus
	RD+	Empfangsdaten plus
		nicht angeschlossen
	SHLD	Schirm

3.10 X2 Anschluss für Versorgungsspannung, direkte E/A und BUSY-Kontakt

X2 Schraubblockklemme 8-polig		
	Bezeichnung	Erläuterung
		Betriebs-Kontakt (geschlossen wenn die SPS läuft)
	+24 V	Versorgungsspannung DC +24 V
	M	Versorgungsspannungs-Masse
	+24 V ext	Versorgungsspannung DC +24 V für die direkten Ein-/Ausgänge
	Mext	Versorgungsspannungs-Masse für die direkten Ein-/Ausgänge
	I/O 1	direkter Ein-/Ausgang* 1
	I/O 2	direkter Ein-/Ausgang* 2
		*Der Anschluss kann als Ein- oder Ausgang genutzt werden. Auch bei der Verwendung als Eingang muss die Versorgungsspannung für die Ein-/Ausgänge angeschlossen sein.

Die beiden direkten digitalen Ein-/Ausgänge können im SPS-Programm verarbeitet werden. Wegen ihrer kurzen Reaktionszeiten, unabhängig von der Belastung der Steuereinheit, sind sie für zeitkritische Anwendungen einsetzbar.

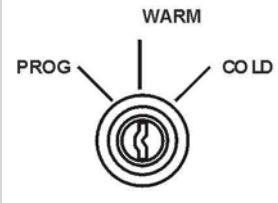
Kennwerte siehe Technische Daten.

In den mit MULTIPROG mitgelieferten Templates für die XCx-Steuereinheiten sind die Variablen bereits angelegt.
Arbeitsblatt Global_Variables, Gruppe XFIO_Variables

Variable	Klemme	
x2_I00	I/O 0	Eingang
x2_O00	I/O 0	Ausgang
x2_I01	I/O 1	Eingang
x2_O01	I/O 1	Ausgang

3.11 Betriebsartenschalter

Der Betriebsartenschalter ist als Schlüsselschalter ausgeführt, er verfügt über drei Stellungen. Mit diesen drei Stellungen wird das Hochlaufverhalten der Steuerung bestimmt. Über den Schlüsselschalter soll eine unberechtigte Veränderung des Hochlaufverhaltens verhindert werden.

Betriebsartenschalter		
	Stellung	Erläuterung
	**PROG**	SPS-Stop, Betriebsart Programmierung Nur in dieser Stellung: • kann vom Programmiersystem aus ein neues SPS-Programm oder ein Bootprojekt in die Steuerung übertragen werden • ist der RESET-Taster wirksam
	WARM	Warmstart der SPS (Defaultstellung)
	COLD	Kaltstart der SPS (Reinitialisierung der Retainvariablen)

Die Stellung des Betriebsartenschalters kann im SPS-Programm abgefragt werden.

In den mit MULTIPROG mitgelieferten Templates für die XCx-Steuereinheiten ist die Variable bereits angelegt.
Arbeitsblatt Global_Variables, Gruppe XFIO_Variables

Variable	Wert	Stellung
x2_SB00	0	PROG
	1	WARM
	2	COLD

3.12 RESET-Taster

Der RESET-Taster löst in der Betriebsart PROG (siehe Betriebsartenschalter) einen Hardware-Reset aus, gleichbedeutend mit Power-Off.

Wird RESET ausgelöst, während ein Zugriff auf die CF Card (CF LED leuchtet grün) erfolgt, tritt Datenverlust auf der CF Card auf.

Der RESET-Taster kann vom SPS-Programm aus abgefragt werden.

In den mit MULTIPROG mitgelieferten Templates für die XCx-Steuereinheiten ist die Variable bereits angelegt.
Arbeitsblatt Global_Variables, Gruppe XFIO_Variables

Variable	Wert	Stellung
x2_SB01	0	nicht betätigt
	1	betätigt

4 Schnelleinführung

Die in diesem Kapitel beschriebene Inbetriebnahme der XCx kann ohne tiefgreifendes Wissen durchgeführt werden. Um schnelle Erfolge zu erreichen ist es notwendig die Inbetriebnahmeschritte genau zu befolgen und die Rahmenbedingungen (wie z.B. die I/O-Konfiguration) genau einzuhalten.

Wichtige Themen, die hier nur angerissen werden, werden in den folgenden Kapiteln tiefgreifender behandelt.

Die Inbetriebnahme kann immer wieder neu begonnen werden, wenn mit einer Grundinitialisierung (siehe Seite 79) die Geräte-Defaultwerte d.h. der Auslieferungszustand wieder hergestellt wird.

Wichtig sind auch die allgemeinen Hinweise zur Inbetriebnahme, im gleichnamigen Kapitel (siehe Seite 78), wenn Probleme auftreten.

4.1 Inbetriebnahme der Kommunikation

4.1.1 Kommunikationsschema

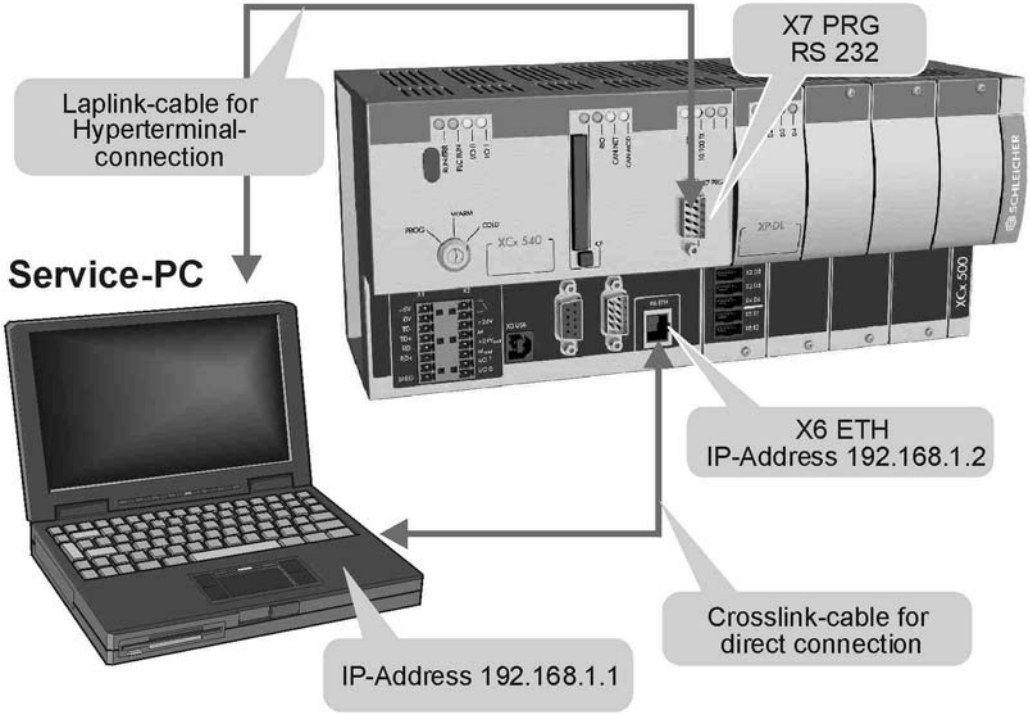


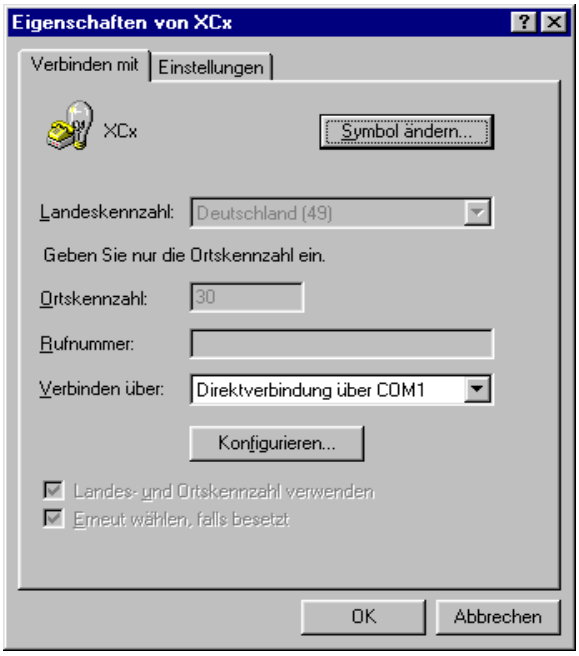
Abbildung 23: Anschluss des Service-PC an der XCx

4.1.2 Inbetriebnahme der seriellen Verbindung über die RS232-Schnittstelle

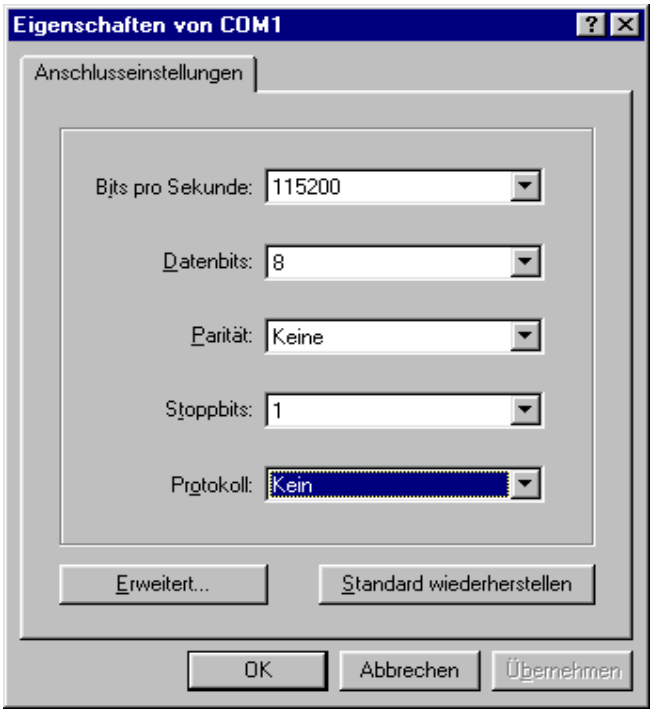
Die Verbindung über die serielle Schnittstelle ist bei der ersten Inbetriebnahme der XCx sehr hilfreich. Sie ist leicht in Betrieb zu nehmen und die ersten Reaktionen der XCx können beurteilt werden.

- Die Kabelverbindung zwischen dem Service-PC COM1 oder COM2 und der XCx X7 PRG herstellen.
- Auf dem PC unter *Start/Programme/Zubehör/Hyperterminal* das Programm *Hypertrm.exe* starten, einen Namen eingeben z.B. XCx und ein Symbol aussuchen.

- In *Eigenschaften von ...* unter *Verbinden über* Direktverbindung über COM1 auswählen und dann auf den Button *Konfigurieren* klicken.



- In *Eigenschaften von COM1* diese Parameter einstellen.



- Die XCx einschalten oder RESET auslösen.
  Während des Steuerungsanlaufes wird im Hyperterminal-Dialogfenster das Bootprotokoll ausgegeben.

4.2 Inbetriebnahme der Ethernet-Schnittstelle

Es wird empfohlen die Ethenet-Verbindung für die Programmierung mit MULTIPROG zu nutzen, um die höchstmögliche Performance bei der Konfiguration, Programmierung und Inbetriebnahme auszuschöpfen. Voraussetzung dafür ist ein PC mit Netzwerkkarte für Ethernet und ein Cross-Link-Kabel für die Direktverbindung zur XCx. Für die korrekte Konfiguration von MULTIPROG wird die IP-Adresse der XCx benötigt.

4.2.1 Auslesen der aktuellen IP-Adresse aus der XCx

Standardmäßig werden vor der Auslieferung der XCx folgende Werte eingetragen:
IP-Adresse: 192.168.1.2
Subnet Mask: 255.255.255.00
Username: target
Passwort: target

Um das zu überprüfen kann im Dialogfenster von *Hyperterminal* der Befehl *version* eingegeben werden.

Darauf hin wird die Versionsinformation ausgegeben.

```
-> version
VxWorks .........
Kernel: .........
Made on .........
Boot line:
ata=........
e=192.168.1.2:ffffff00 u=target pw=target tn=X
Cx .........
value = .......
->
```

In der Zeile e= steht die IP-Adresse (192.168.1.2) mit Subnet Mask (fffff00 == 255.255.255.00), Username (u=target) und Passwort (pw=target)

Notieren Sie sich die Angaben, sie werden bei der Konfiguration von MULTIPROG benötigt.

4.2.2 Einstellen der IP-Adresse des Service-PC

1. Die IP-Adresse des Service-PC wird eingestellt über *Start/Einstellungen/Systemsteuerung/Netzwerk/Protokolle*.
2. Dann TCP/IP auswählen und *Eigenschaften* anklicken.
   IP-Adresse eingeben: **192.168.1.1**
   Subnet Mask eingeben: **255.255.255.0**
3. Ein Neustart des PCs ist erforderlich.
4. Auslesen der IP-Adresse zum Überprüfen des Service-PCs

   Windows® 9x *Start/Ausführen/winipcfg*
   Im Bild Netzkarteninfo ist anstelle von *PPP-Adapter* der <Name Ider Netzwerkkarte> anzuwählen.

   Windows® NT *Start/Programme/MS-DOS Eingabeaufforderung ipconfig* eingeben und starten.

   Unter IP-Adresse der Netzwerkkarte sollte jetzt **192.168.1.1** stehen.
   Unter Subnet Mask sollte jetzt **255.255.255.0** stehen.
   Mit O.K. quittieren bzw. Fenster schliessen.

4.2.3 Ethernet-Verbindung herstellen und prüfen

1. Den Schlüsselschalter der XCx auf PROG stellen und den Ethernet-Anschluss des PC mit der Ethernet-Schnittstelle X6 ETH der XCx über ein Cross-Link-Kabel verbinden.
2. Auf dem PC *Start/Programme/MS-DOS-Eingabeaufforderung* wählen.
3. Eingabe: *doskey* (Doskey installiert wird ausgegeben. Als Eingabehilfe für einen erneuten Test verwenden).
4. Eingabe :ping 192.168.1.2
   (diese IP-Adresse ist bei Auslieferung der XCx voreingestellt).
5. Ist alles O.K. wird ausgegeben:
   ping wird ausgeführt...
   Antwort von 192.168.1.2.....
   Antwort...
   Antwort ..
   Antwort ..

Wird die XCx nicht erkannt, wird eine Fehlermeldung ausgegeben. Es sind die IP-Adresse und die Kabelanschlüsse zu überprüfen.

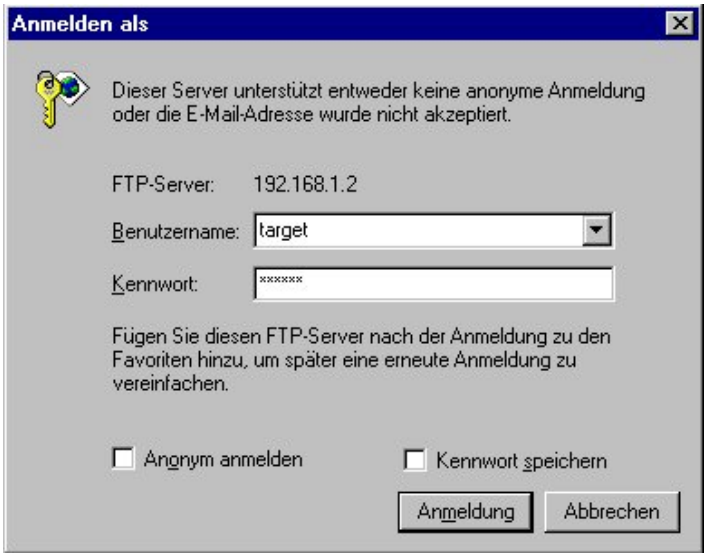
4.3 Ändern der IP-Adresse der XCx

Das Ändern der IP-Adresse ist nicht notwendig, wenn die XCx wie vorher beschrieben über ein Crosslink-Kabel direkt am PC betrieben wird. Andernfalls kann wie folgt vorgegangen werden

- Zum Ändern ein Programm für den FTP(File Transfer Protokoll)-Zugriff (z.B.Windows® Internet Explorer 6) auf dem PC starten.

Für den Zugriff per FTP muss die TCP/IP Verbindung zur XCx aktiv sein. (Siehe Überprüfen der Verbindung mit Ping, siehe oben)

- In der Adresszeile *ftp://192.168.1.2* eingeben und starten.
- Zur Anmeldung den Benutzernamen und das Kennwort der XCx eingeben:
  Benutzername = target
  Kennwort = target



- Es wird der Inhalt des Compact Flash im Windows® Internet Explorer angezeigt.
- Aus dem Verzeichnis \OS die Datei *BootLine.ini* auf die Festplatte kopieren.
- Die Datei *BootLine.ini* editieren.
  Die gewünschte IP-Adresse und g.g.f. den Benutzernamen und das Passwort ändern:

```
InetOnEthernet  =192.168.1.1
InetMask        =255.255.255.0
UserName        =target
Password        =target
```

- Die Datei speichern und zurück auf den Compact Flash der XCx schreiben.
  Wenn der Schreibvorgang fehlschlägt muss das Nur-Lesen-Attribut der Datei *BootLine.ini* aufgehoben werden. Dazu mit Hyperterminal über die RS232-Schnittstelle den Befehl *attrib "bootline.ini","-R"* eingeben und ausführen.
- Den Betriebsartenschalter der XCx auf PROG und stellen die XCx aus- und einschalten oder RESET auslösen.

4.4 Installation der Programmiersoftware MULTIPROG, OPC-Server, AddOns

4.4.1 Installation von MULTIPROG

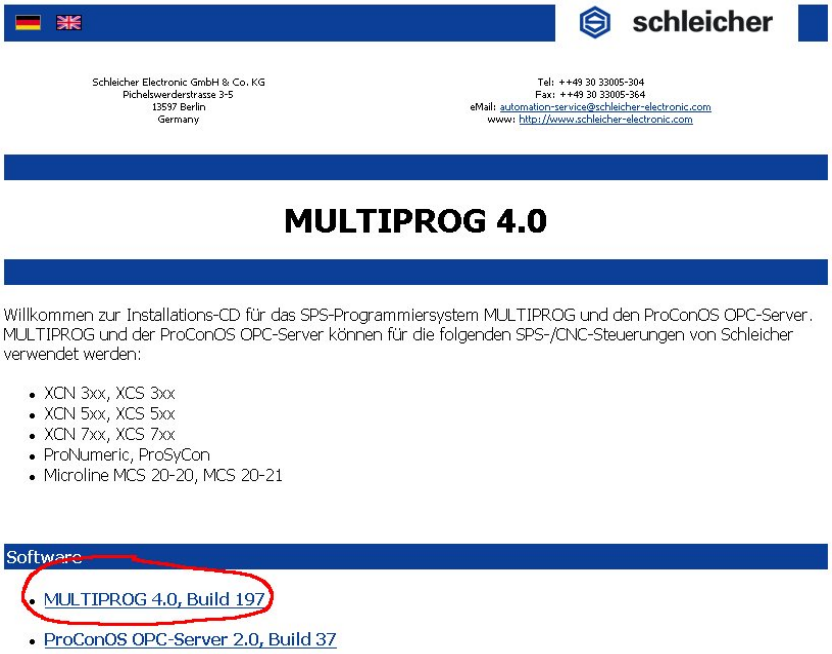
Die gesamte Programmiersoftware besteht aus den Software-Komponenten MULTIPROG, dem OPC-Server (**O**LE for **P**rocess **C**ontrol), AddOns für MULTIPROG und dem Schleicher-Dialog.

Alle Software-Komponenten müssen einzeln, in dieser Reihenfolge nacheinander installiert werden.

Folgende Software-Pakete können geliefert werden:

Name	Inhalt	Artikel-Nr.
MULTIPROG Einzelplatzlizenz	CD 1: • Programmiersoftware MULTIPROG CD 2: • Sevice Pack	R4.320.0640.0 (320 386 28)
MULTIPROG Update	CD 1: • Programmiersoftware MULTIPROG CD 2: • Sevice Pack	R4.320.0650.0 (320 386 29)
MULTIPROG Generallizenz	CD 1: • Programmiersoftware MULTIPROG CD 2: • Sevice Pack	R4.320.0660.0 (320 386 30)
Service Pack	CD: • Steuerungssoftware für alle Schleicher-Steuerungen • OPC-Server • AddOns • Schleicher-Dialog • Weitere Hilfsmittel wie Dokumentation und Service-Informationen	R4.320.0590.0 (320 385 65)

- Die CD MULTIPROG in das Laufwerk des PC einlegen.
  Es wird mit der AutoRun-Funktion der CD der Internet Explorer gestartet.



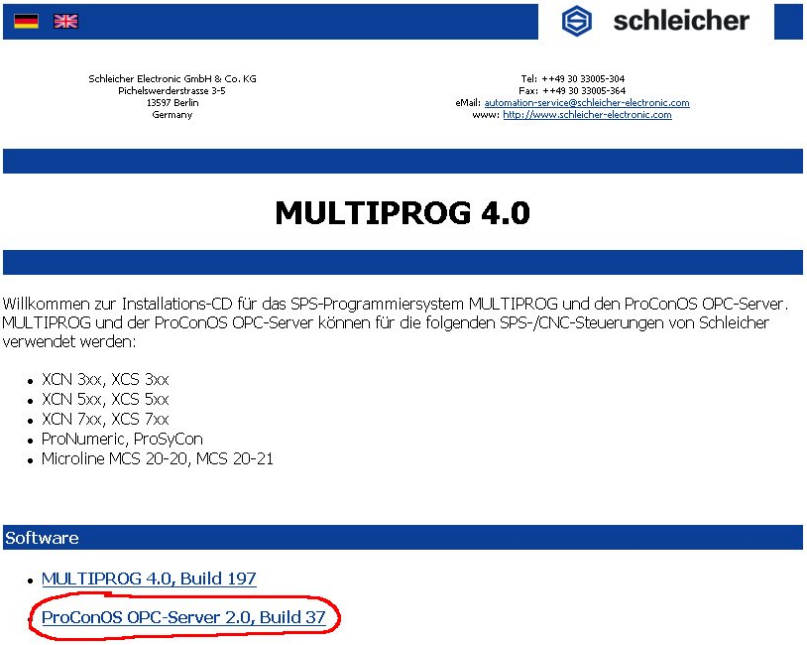
- Bitte die "Hinweise zur Installation" beachten.
  Dann *MULTIPROG* (hier Version 4.0) auswählen und die Installation starten.

Ist eine Version von MULTIPROG 1.2 bereits installiert, darf die installierte Version nicht überschrieben werden, wenn weiter mit den alten Projekten gearbeitet werden soll.
MULTIPROG muss dann in einem neuen Pfad installiert werden.

Alle anderen Installations-Einstellungen können unverändert bleiben.

Am Ende der Installation erfolgt eine Aufforderung zum Neustart des Computers. Falls Sie anschließend auch den ProConOS OPC-Server installieren wollen, ist ein Neustart des Computers jetzt noch nicht erforderlich.

- *Zur Installation des OPC-Servers im Internet Explorer den ProConoS OPC-Server auswählen und die Installation starten.*

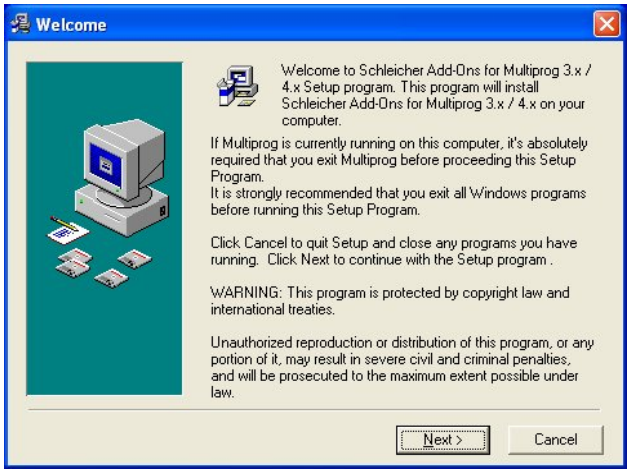


Der OPC-Server soll im MULTIPROG-Pfad installiert werden.
Alle anderen Installations-Einstellungen können unverändert bleiben.
Ein Neustart des PCs nach der Installation ist erforderlich.

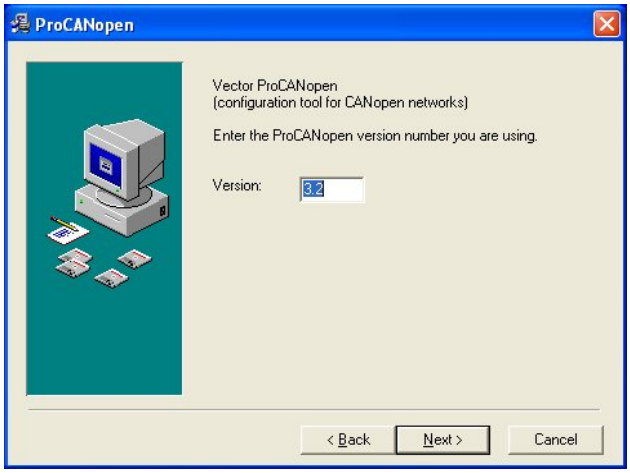
Als nächster Schritt müssen die AddOn's für MULTIPROG installiert werden.

- Dazu die CD *Service Pack* einlegen.
  Es wird mit der AutoRun-Funktion der CD der Internet Explorer gestartet.

- Dann unter der Rubrik für die vorhandene Steuerung (hier XCx 300) *AddOns für MULTIPROG* auswählen und die Installation starten.



- Im Lauf der Installation muss die aktuelle ProCANopen Version (hier 3.2) eingetragen werden. ProCANopen wird später zur Inbetriebnahme des CANopen Netzes benötigt.



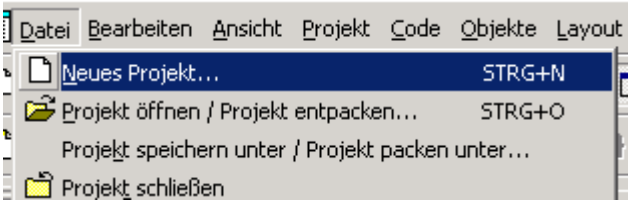
Ein Neustart des PCs nach der Installation ist nicht erforderlich.

Anschließend wird die Bedienoberfläche Schleicher-Dialog installiert.

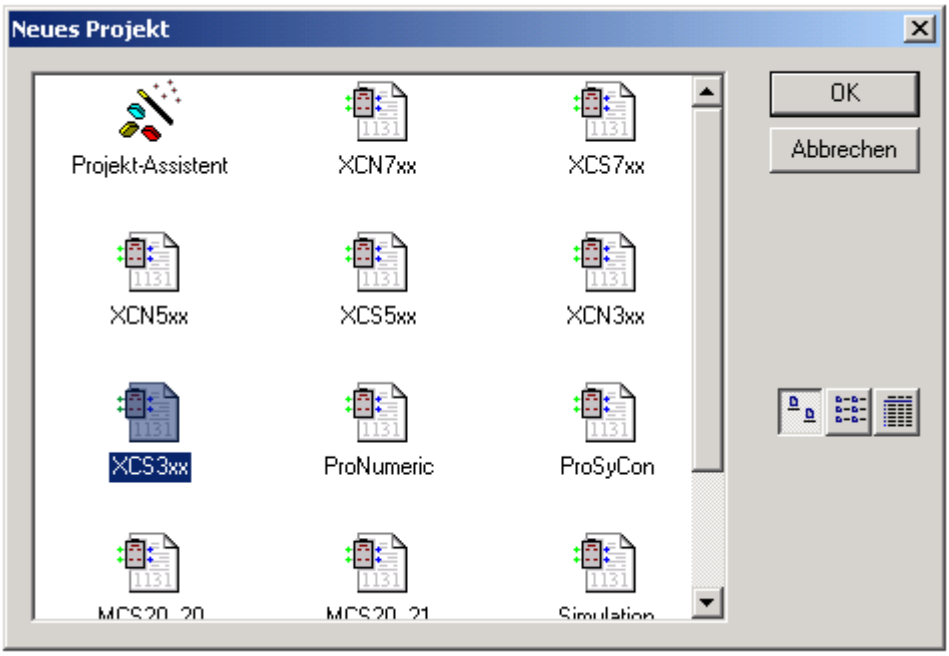
- Dazu im Internet Explorer *Dialog* auswählen und starten.

4.4.2 MULTIPROG starten und ein neues Projekt öffnen

- MULTIPROG starten, *Datei/Neues Projekt* wählen.

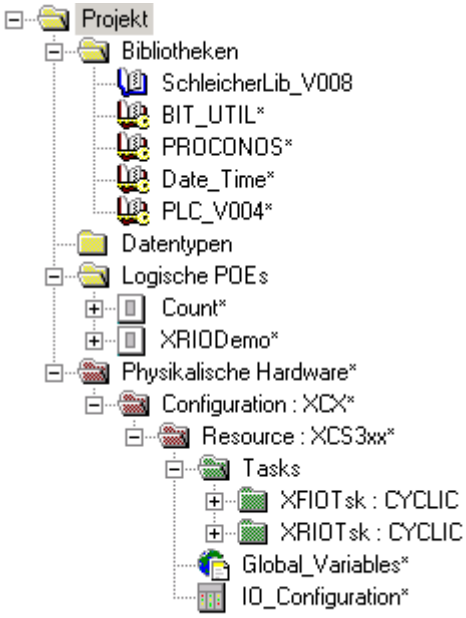


- Ein Projekt für den vorhandenen Steuerungstyp (hier XCS3xx) auswählen, mit *OK* ausführen.

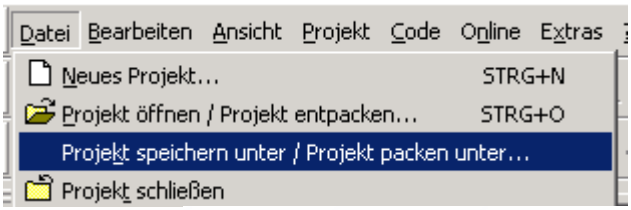


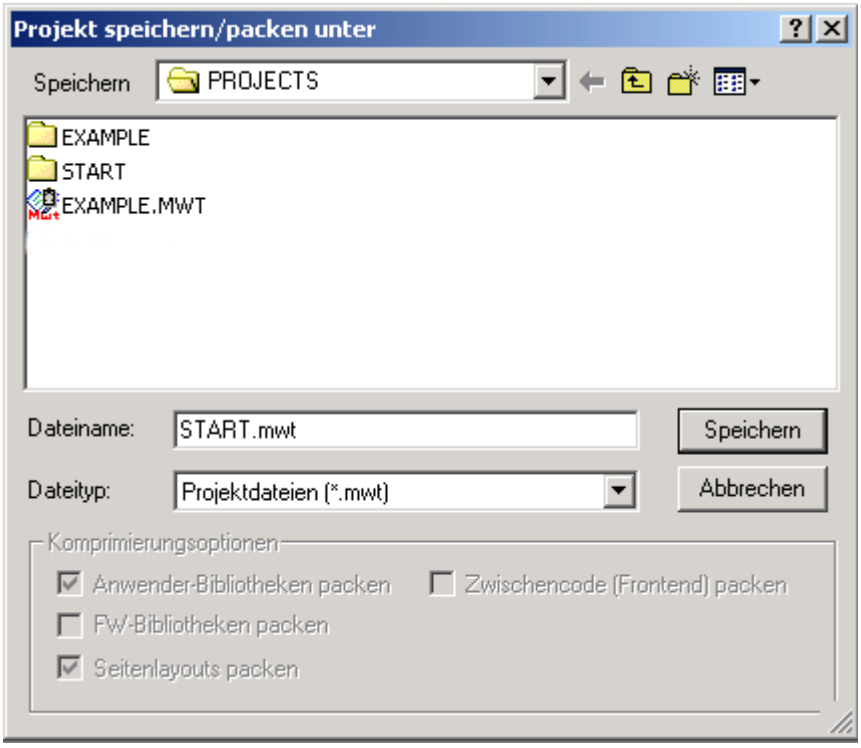
- Die Fragen nach der Konvertierung der Bibliotheken immer mit *Ja* beantworten.

- Ist das Projekt erfolgreich geöffnet, wird im Projekt-Fenster der Projektbaum dargestellt. Es sind bereits Logische POEs enthalte, die voll funktionsfähig sind und für eine einfache Schnellinbetriebnahme ausreichen.



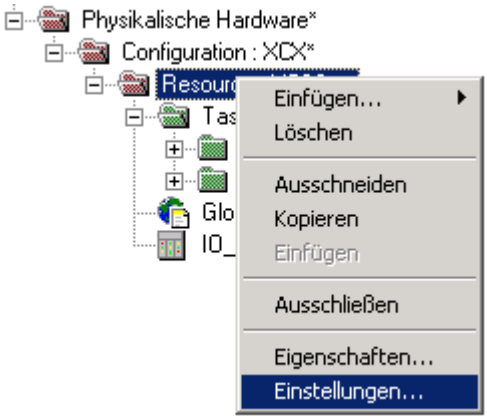
- Unter *Datei/Projekt speichern unter* das Projekt mit einem neuen Namen (hier START) speichern.



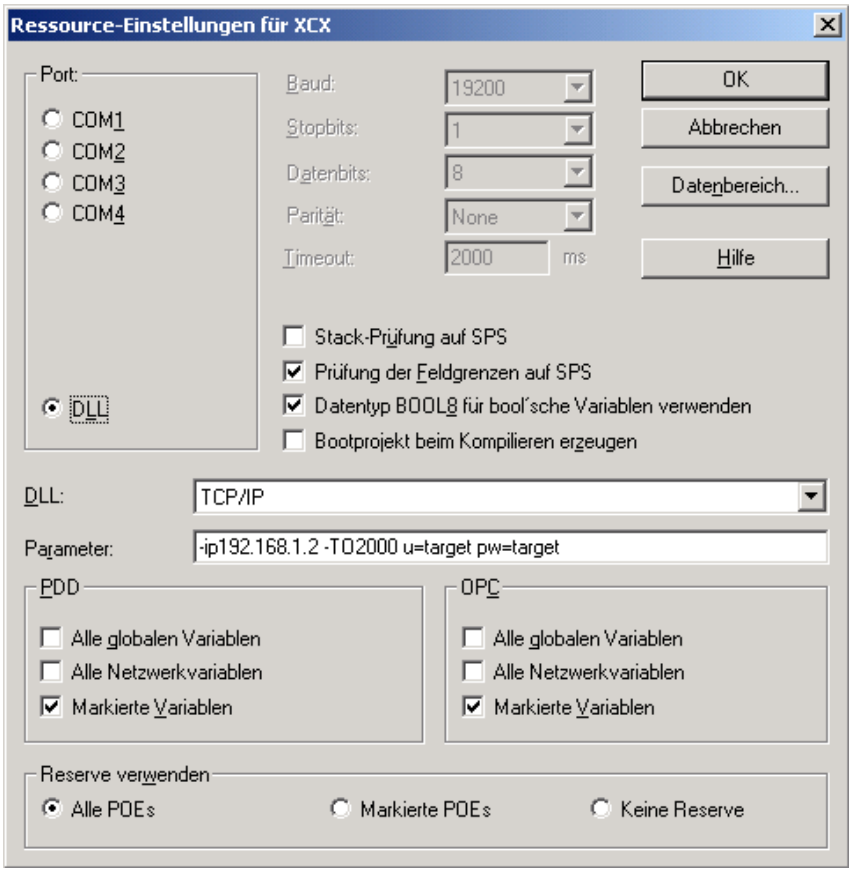


4.4.3 Ethernet-Verbindung in MULTIPROG einstellen

- Im Projektbaum *Ressource : XCS3xx* mit der rechten Maustaste anklicken, um das Kontextmenü zu öffnen.



- Dann *Einstellungen* wählen und unter *Parameter* die IP-Adresse der XCx eintragen.
  Die IP-Adresse wird wie im Abschnitt *Auslesen der aktuellen IP-Adresse aus der XCx* beschrieben ermittelt.



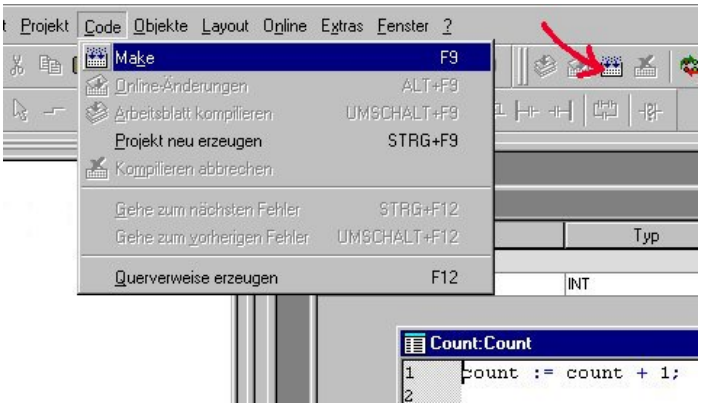
Bedeutung der Parameter:

-ip192.168.1.2 = IP-Adresse der XCx
-TO2000 = Time Out 2000 ms
u=target = Username
pw=target = Passwort

Alle anderen Einstellungen bleiben erhalten.

4.4.4 Ein Projekt Kompilieren und zur XCx senden

- Das Projekt Kompilieren, dazu *Code/Make* wählen.
  (alternativ <F9> oder Button *Make* siehe Pfeil)



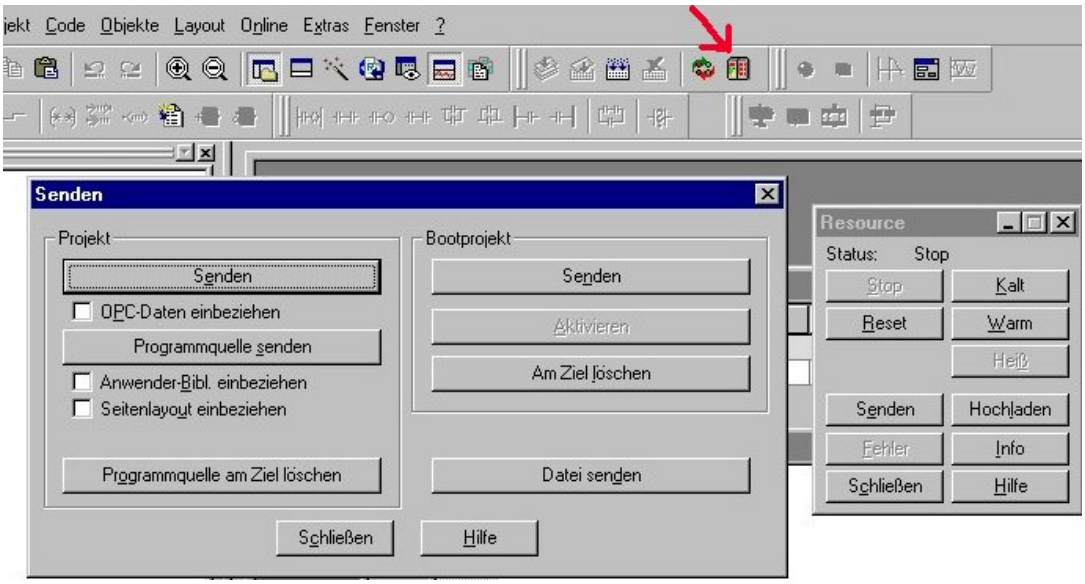
- Im Meldungsfenster unten wird der Kompiler-Lauf protokolliert. Hier werden auch Fehlermeldungen, Warnungen und weitere Infos angezeigt.
  Ein Anwählen der Auswahlbox Fehler, Warnungen usw.. zeigt die Meldung detaillierter an.
  Werden Fehler angezeigt, kann mit Doppelklick auf der Fehlerzeile direkt in die Zeile des SPS-Programms, die den Fehler verursachte verzweigt werden.
- Das Projekt über die Ethernet-Verbindung übertragen.

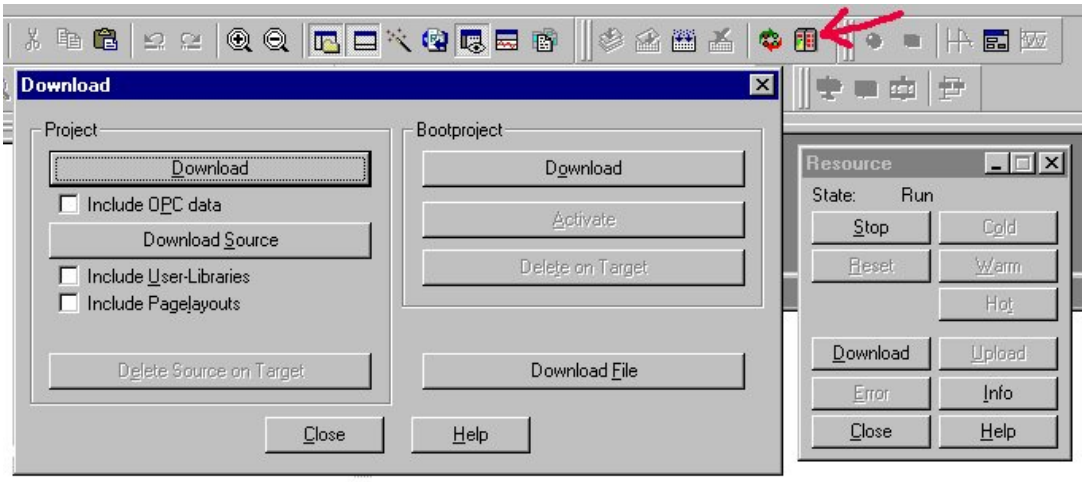
  1. Auf den Button *Projekt-Kontrolldialog* klicken (siehe Pfeil).

  2. Durch einen Klick auf den Button *Senden* im Kontrolldialog XCx das Dialogfenster Senden öffnen.

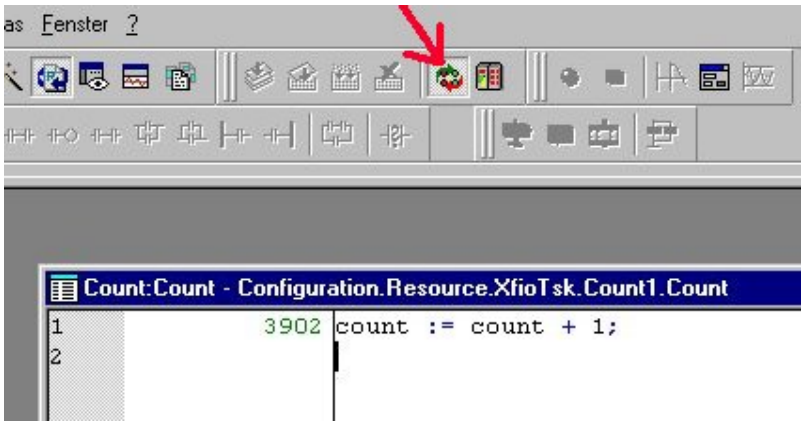
  3. Hier in der Rubrik Projekt wieder *Senden* wählen, das auf der XCx vorhandene Projekt überschreiben.

  4. Mit dem Button *Kalt* (Kaltstart) im Kontrolldialog wird das Programm auf der XCx gestartet. Die LED PLC RUN leuchtet dauernd gelb.





- Mit dem Button Debug ein/aus (siehe Pfeil) kann auf dem Arbeitsblatt der Inhalt der Variablen online angezeigt werden.



4.4.5 Einfügen der Koppelspeicher-Datentypen und -Variablen in das Projekt

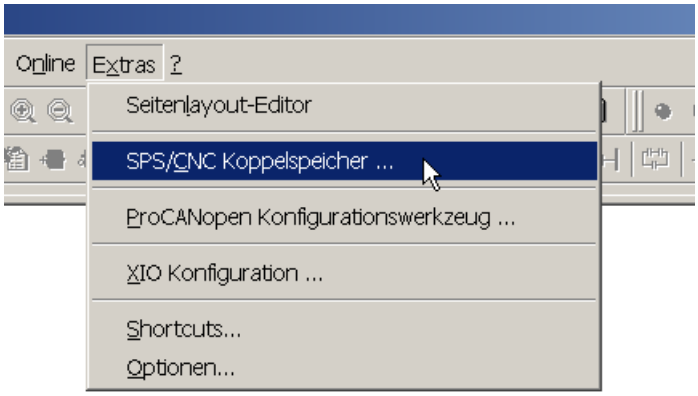
Um den vollen Zugriff auf alle vordefinierten Variablen zu erlangen ist es an dieser Stelle der Inbetriebnahme angebracht die Koppelspeicherstruktur in das Projekt einzufügen

Der Koppelspeicher besitzt eine umfangreiche Datenstruktur, die zur Kommunikation zwischen SPS, CNC und Visualisierungssystemen verwendet wird.
Außerdem kann über den Koppelspeicher auf Datenbereiche wie z.B. Version Number, Error Page und Log Book zugegriffen werden.

Eine genaue Beschreibung sämtlicher Variablen des Koppelspeichers ist als Online-Hilfe der Software "Schleicher Dialog" verfügbar.

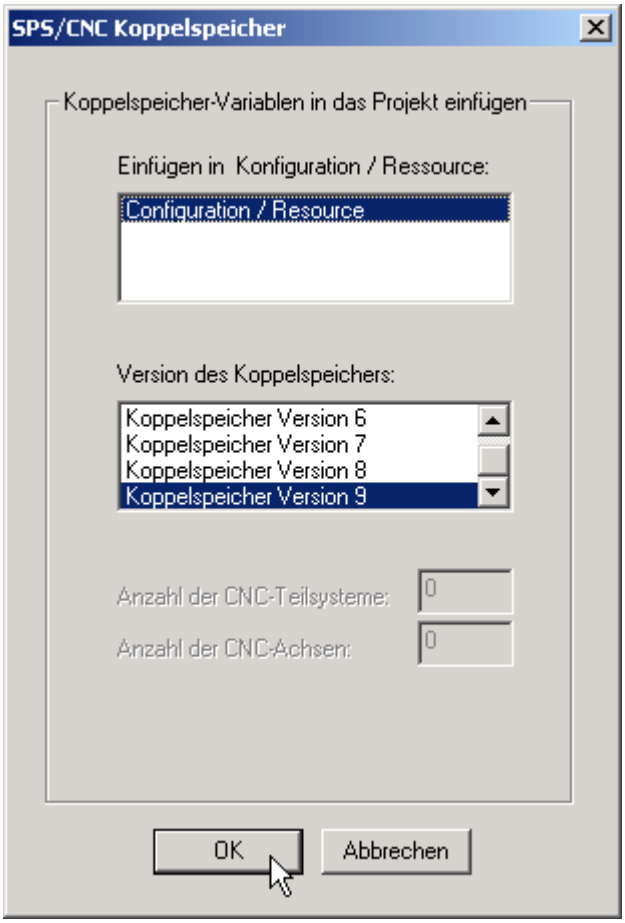
Um auf Variablen des Koppelspeichers zugreifen zu können, müssen die entsprechenden Datentypen und Variablen in das SPS-Projekt eingefügt werden. Diese sind in den Projekt-Templates von MULTIPROG noch nicht enthalten. Sie müssen vom Anwender selbst mit Hilfe des Koppelspeicher-AddOns für MULTIPROG in das SPS-Projekt eingefügt werden. Damit soll sichergestellt werden, dass der Anwender mit der für die Steuerungssoftware passenden Koppelspeicher-Variante arbeitet.

- Das Einfügen der Koppelspeicher-Datentypen und -Variablen in ein SPS-Projekt erfolgt über das Menü *Extras/SPS/CNC-Koppelspeicher*.



Es öffnet sich ein Dialogfenster mit folgenden Auswahl- bzw. Eingabemöglichkeiten.

- Einfügen in Konfiguration / Ressource: Auswahl der Ressource des SPS-Projektes, in die die Koppelspeicher-Variablen eingefügt werden sollen
- Version des Koppelspeichers: Auswahl der Koppelspeicher-Version
- Anzahl der CNC-Teilsysteme / CNC-Achsen: Eingabe der Anzahl der Teilsysteme und Achsen für CNC-Steuerungen (für reine SPS-Steuerungen sind diese beiden Eingabefelder deaktiviert)
- OK-Button: Beim Verlassen des Dialogfensters mit OK werden die Koppelspeicher-Datentypen und -Variablen in das SPS-Projekt eingefügt.
- Abbrechen-Button: Beim Verlassen des Dialogfensters mit Abbrechen wird das SPS-Projekt nicht verändert.



Nach dem Einfügen der Koppelspeicher-Datentypen und -Variablen muss das SPS-Projekt neu übersetzt und zur Steuerung übertragen werden.

Zugriff auf den Koppelspeicher

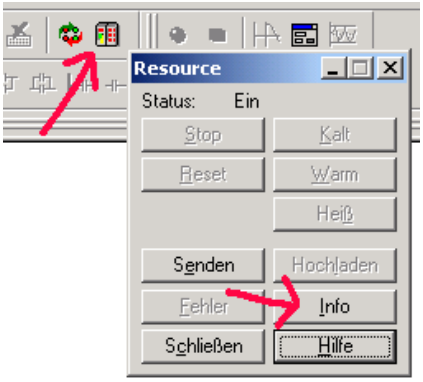
Das SPS-Programm hat Zugriff auf den gesamten Koppelspeicher über die globale Variable `plcMem` (bei SPS-Steuerungen) bzw. `cncMem` (bei CNC-Steuerungen). Mit der Punktschreibweise kann auf die einzelnen Komponenten des Koppelspeichers zugegriffen werden. Zum Beispiel kann das SPS-Programm die Versionsnummer der Betriebssoftware der Steuerung folgendermaßen auslesen: `cncMem.plcSect.lOSVersion`.

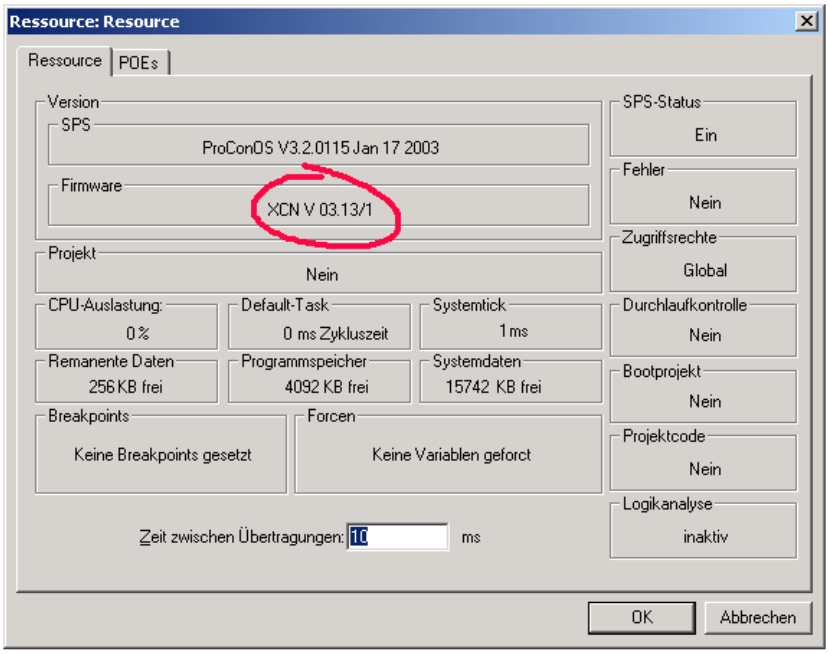
Visualisierungssysteme haben Zugriff auf den Koppelspeicher über die OPC-Schnittstelle. Die Versionsnummer der Betriebssoftware kann beispielsweise aus der OPC-Variablen `cmpS_lOSVersion` ausgelesen werden.

Hinweise zur Auswahl der Koppelspeicher-Version

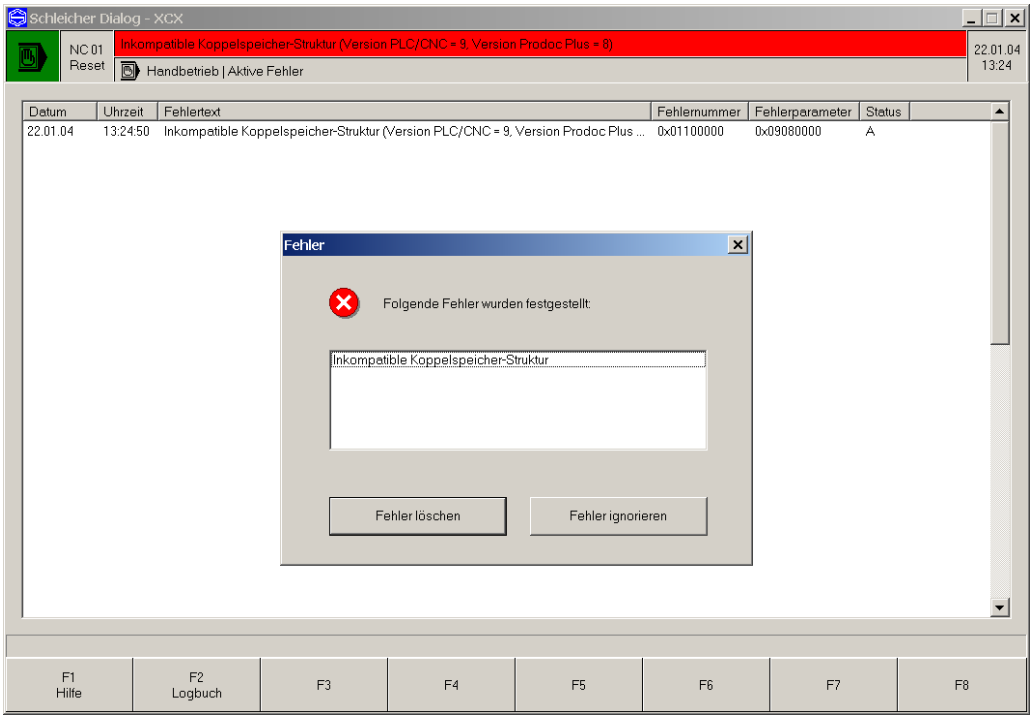
Die Koppelspeicher-Struktur wird von Schleicher von Zeit zu Zeit aktualisiert bzw. erweitert. Zur Unterscheidung der einzelnen Varianten dient eine Versionsnummer. Die Versionsnummer wird bei größeren Änderungen an der Koppelspeicher-Struktur erhöht, bei der eine Änderung von Adressen von Variablen erforderlich ist. Koppelspeicher-Varianten mit unterschiedlicher Versionsnummer sind daher inkompatibel. Nur bei übereinstimmender Versionsnummer ist eine Kompatibilität gewährleistet.

Grundsätzlich sollte immer die aktuellste Koppelspeicher-Version verwendet werden. Wenn es sich jedoch um eine Steuerung mit einer älteren Betriebssoftware handelt, muss eine hierzu passende ältere Koppelspeicher-Version verwendet werden. Die Dokumente XCx300Rev.doc, XCx5x0Rev.doc, XCx700Rev.doc erteilten Auskunft darüber, welche Koppelspeicher-Version mit welcher Betriebs-software-Version verwendet werden kann. Die Dokumente sind auf der Schleicher Homepage *http://www.schleicher-electronic.com* unter Betriebsanleitungen XCx verfügbar. Die Version der Betriebssoftware der Steuerung kann im Info-Dialogfenster zu der entsprechenden Ressource im SPS-Projekt angezeigt werden.





Falls die Betriebssoftware der Steuerung und die verwendete Koppelspeicher-Version nicht übereinstimmen, wird beim Start der SPS eine Fehlermeldung in den Fehlerspeicher eingetragen. Diese Fehlermeldung wird Schleicher Dialog folgendermaßen angezeigt.



Falls eine solche Fehlermeldung auftritt, muss das SPS-Projekt korrigiert werden. Im diesem Beispiel arbeitet die Betriebssoftware der Steuerung mit der Koppelspeicher-Version 9. Das SPS-Projekt enthält aber noch die Datentypen und Variablen für die Koppelspeicher-Version 8. Das SPS-Projekt muss korrigiert werden, indem die Datentypen und Variablen für die Koppelspeicher-Version 9 eingefügt werden. Anschließend muss das SPS-Projekt neu übersetzt und zur Steuerung übertragen werden.

Weitere Hintergrundinformationen zum Koppelspeicher

Beim Einfügen des Koppelspeichers in ein SPS-Projekt werden folgende Elemente hinzugefügt bzw. aktualisiert:

- Das Datentyp-Arbeitsblatt *SharedMemory_Types*; hier ist die Datenstruktur des Koppelspeichers deklariert.
  Das Datentyp-Arbeitsblatt wird ab der Koppelspeicher-Version 8 eingefügt. Bei früheren Versionen wird kein Datentyp-Arbeitsblatt eingefügt, da in dem Fall die Deklaration der Koppelspeicher-Struktur in der zugehörigen Anwender-Bibliothek *SchleicherLib_Vxxx* enthalten ist (Koppelspeicher-Version 7 in *SchleicherLib_V007*, Koppelspeicher-Version 6 in *SchleicherLib_V006* usw.).
- Die globalen Variablen `plcMem` (für SPS-Steuerungen) bzw. `cncMem` (für CNC-Steuerungen). Siehe Tabelle mit den globalen Variablen, Gruppe *SharedMemory_Variables*.
  Diese Variablen repräsentieren den gesamten nichtremaneten (non retain) Bereich des Koppelspeichers. Über die Punktschreibweise kann das SPS-Programm wie oben erläutert auf die einzelnen Komponenten (Variablen) des Koppelspeichers zugreifen.
  Ab der Koppelspeicher-Version 8 existiert für CNC-Steuerungen neben dem nichtremanenten auch ein remanenter (retain) Bereich des Koppelspeichers. Die Werte der Variablen dieses Bereiches bleiben im Gegensatz zum nichtremanenten Bereich auch nach dem Ausschalten der Steuerung erhalten. Hierfür wird zusätzlich die globale Variable `cncRMem` eingefügt.

Die globalen Variablen `cmpS...`, `cmeS...`, `cmcS...`, `cmsS...`, `cmaS...`. Siehe Tabelle mit den globalen Variablen, Gruppen *PLC_Common*, *CNC_Common*, *CNC_System_x* (x steht für die Nummer des CNC-Teilsystems, für jedes Teilsystem existiert eine solche Gruppe mit systemspezifischen Variablen) und *CNC_Axis_y* (y steht für die Nummer der CNC-Achse, für jede Achse existiert eine solche Gruppe mit achsspezifischen Variablen).
Diese Variablen sind für den Zugriff von Visualisierungssystemen oder ähnlichen Programmen auf den Koppelspeicher vorgesehen. Sie stellen den gesamten Koppelspeicher über die OPC-Schnittstelle zur Verfügung.
Diese Variablen stellen den Koppelspeicher in unstrukturierter Form zur Verfügung. Es werden nur einfache Datentypen (BOOL, DINT, REAL, STRING) und Felder von einfachen Datentypen verwendet. Diese Vorgehensweise ist erforderlich, da strukturierte Daten (wie sie die Variablen `plcMem` und `cncMem` beinhalten) nicht über die OPC-Schnittstelle übertragen werden können.

4.5 Zugriff auf die I/O Ebene mit XRIO

4.5.1 Anschlussprinzip XRIO

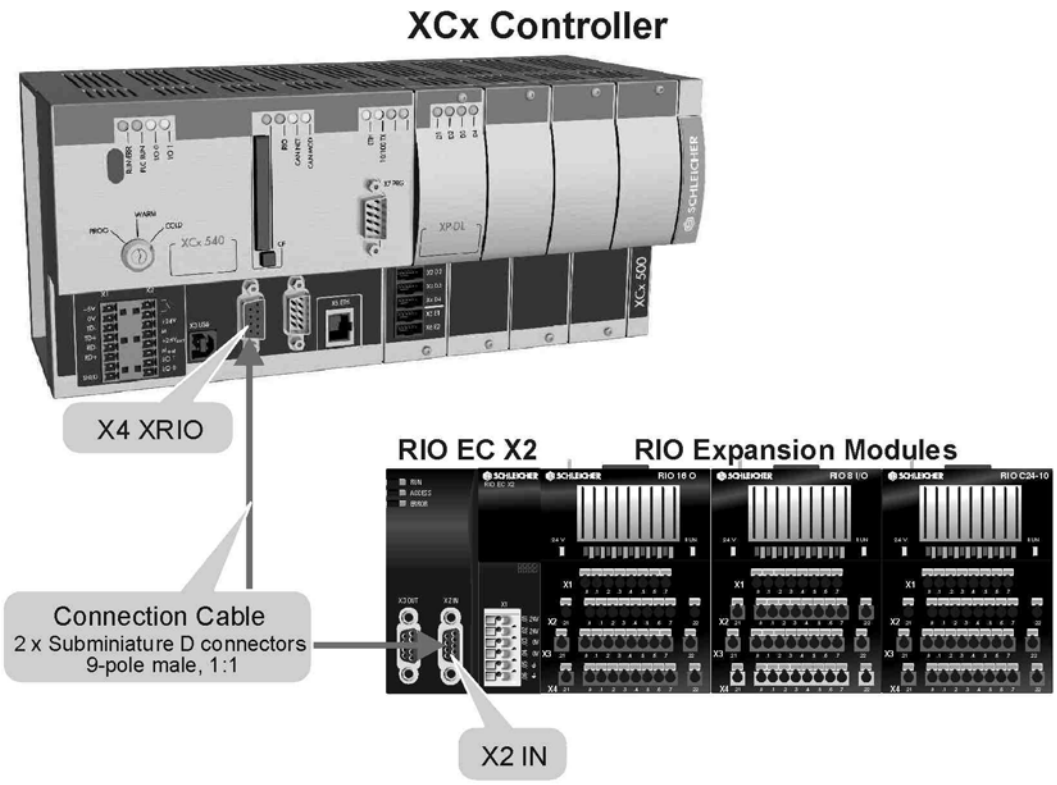
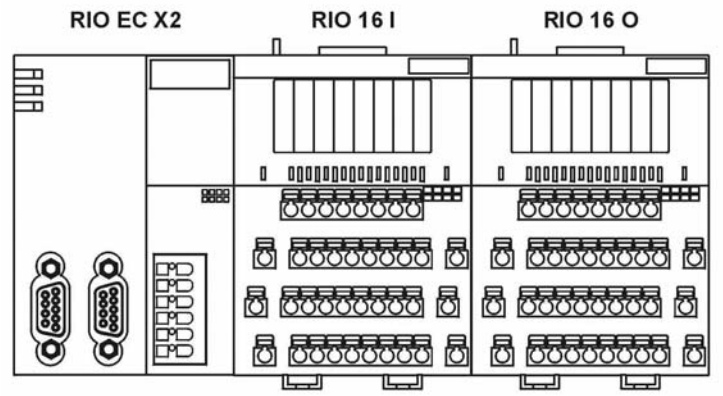


Abbildung 24: XRIO-Anschluss an der XCx

4.5.2 Beispiel einer XRIO-Konfiguration

Das Konfigurationsbeispiel wird für die folgenden Abschnitte benutzt.

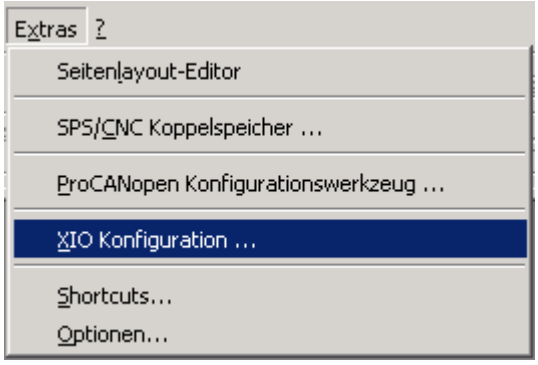


Modul	RIO 16I	RIO 16O
Eingänge nicht entprellt	IB0, IB1 IX0.0...IX0.7, IX1.0...IX1.7	-
Eingänge entprellt	IB2, IB3 IX2.0...IX2.7, IX3.0...IX3.7	-
Ausgänge	-	QB0, QB1 QX0.0...QX0.7, QX1.0...QX1.7

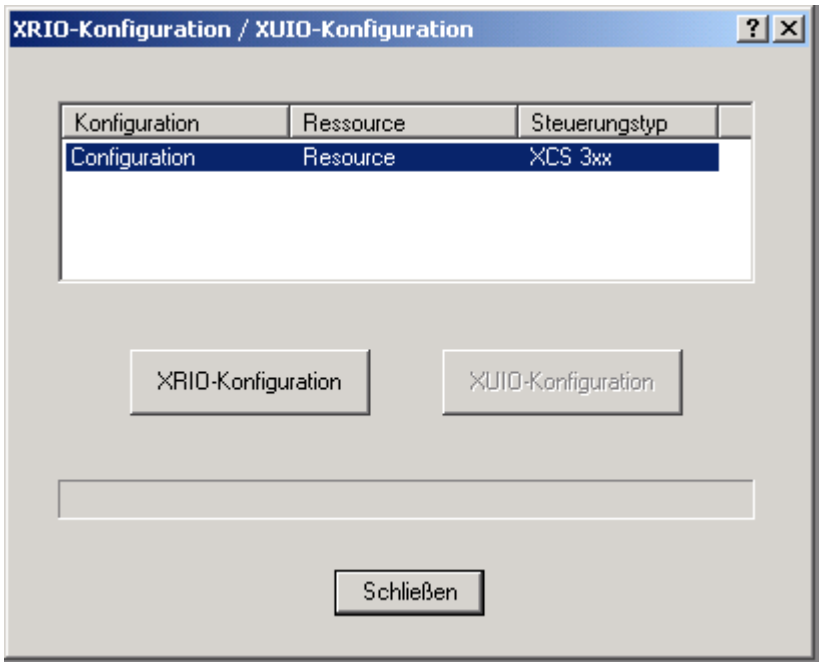
4.5.3 Einlesen der XRIO-Konfiguration

Mit der Installation der *AddOns* (siehe MULTIPROG-Installation) wird ein XRIO-Konfigurator zur Verfügung gestellt, der die Eintragung des XRIO-Treibers und die Variablendeklaration für XRIO vollständig automatisiert.

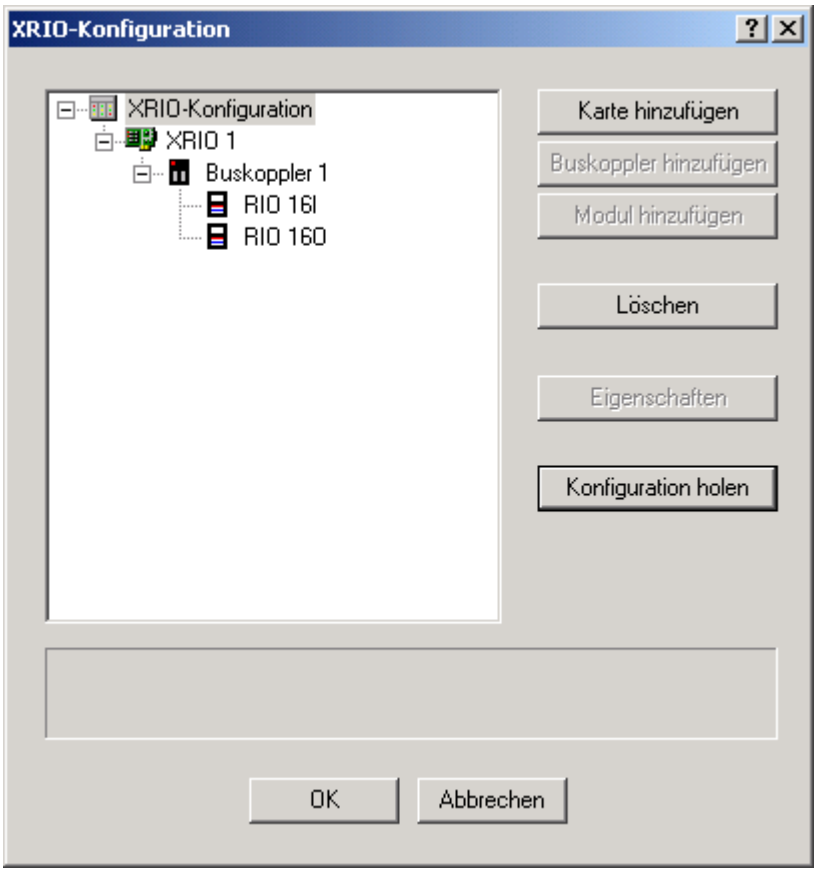
- Über das Menü *Extras/XIO Konfiguration* kann der Konfigurator gestartet werden.



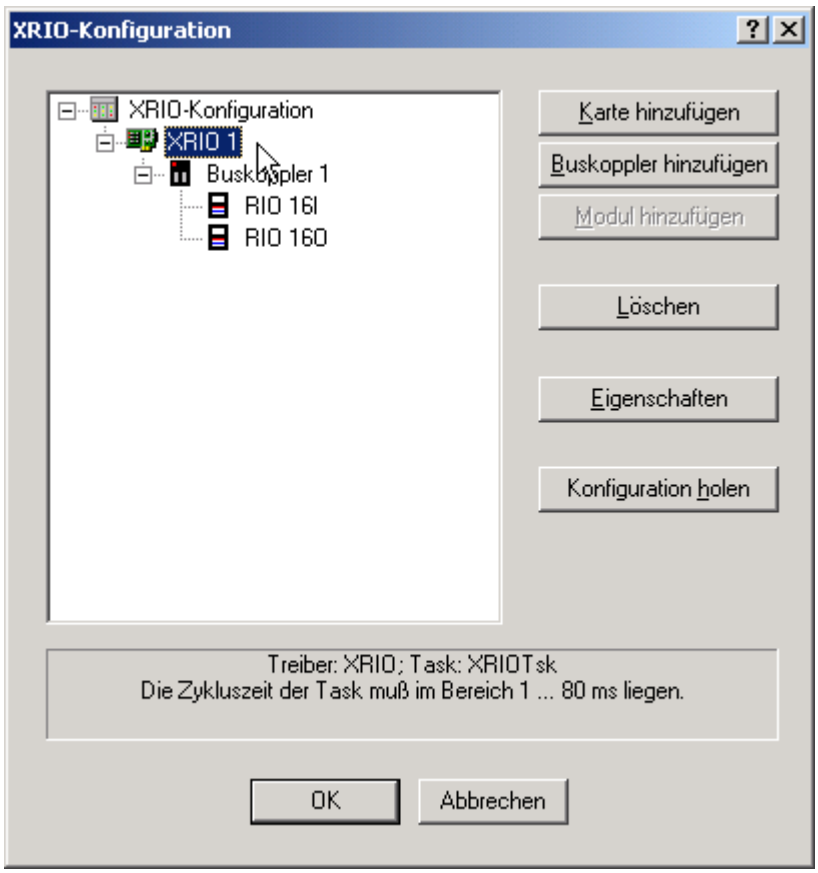
- Es kann der Steuerungstyp gewählt werden (hier nur XCS 3xx) und mit den Buttons die Art der Konfiguration (hier nur XRIO mit dem Button *XRIO-Konfiguration*).



- Die aktuelle XRIO-Konfiguration kann mit dem Button *Konfiguration holen* eingelesen werden.
  Schlägt die FTP-Verbindung fehl muss die im Abschnitt "Ethernet-Verbindung in MULTIPROG einstellen " eingestellte IP-Adresse überprüft werden. Es ist auf die korrekte Schreibweise der Eintragung zu achten.



- Anschließend müssen die Eigenschaften des RIO-Knotens eingestellt werden.
  Dazu auf den RIO-Knoten klicken und den damit aktivierten Button *Eigenschaften* betätigen.



- Dann kann der Treiber (hier XRIO) und die Task ausgewählt werden (hier XRIOTsk), in der der Treiber laufen soll.
  Der DUMMYIO-Treiber kann für die Inbetriebnahme und Programmierung der XCx ohne real vorhandener XRIO-Verbindung gewählt werden.

- Die Einstellungen durch klicken auf OK bestätigen, es wird der Treiber parametriert und die Variablendeklartion wird eingetragen.

Treten Fehler bei der Kommunikation über die Ethernet-Verbindung zur XCx auf, muss die korrekte parametrierung der Schnittstelle, wie im Abschnitt Ethernet-Verbindung in MULTIPROG einstellen beschrieben, überprüft werden.

- Im Projektbaum von MULTIPROG wurde unter Global_Variables die Variablendeklaration für die oben aufgeführte Konfiguration eingetragen.

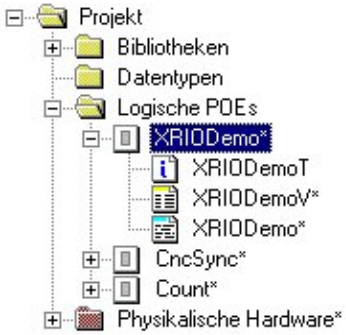
Global_Variables:Configuration.Resource

	Name	Typ	Adresse	Beschreibung	Verwendung	Anfangsw...	Reman...	PDD	OPC
	⊞ **Global_Variables**								
	⊞ **XFIO_Variables**								
	⊟ **XRIO_Variables**								
	xrio1IB0	BYTE	%IB 0	XRIO card 1, module 1, bc 0, RIO 16I, DIGITAL	VAR_GLOBAL		☐	☐	☐
	xrio1IB1	BYTE	%IB 1	XRIO card 1, module 1, bc 0, RIO 16I, DIGITAL	VAR_GLOBAL		☐	☐	☐
	xrio1IB2d	BYTE	%IB 2	XRIO card 1, module 1, bc 0, RIO 16I, DIGITAL	VAR_GLOBAL		☐	☐	☐
	xrio1IB3d	BYTE	%IB 3	XRIO card 1, module 1, bc 0, RIO 16I, DIGITAL	VAR_GLOBAL		☐	☐	☐
	xrio1QB0	BYTE	%QB 0	XRIO card 1, module 2, bc 0, RIO 16O, DIGITAL	VAR_GLOBAL		☐	☐	☐
	xrio1QB1	BYTE	%QB 1	XRIO card 1, module 2, bc 0, RIO 16O, DIGITAL	VAR_GLOBAL		☐	☐	☐
	⊞ **Network_Variables**								
	⊞ **PLC_Common**								

Für das Eingangsmodul RIO 16 I werden zuerst die nicht entprellten Eingänge (xrio1IB0 und xrio1IB1), gefolgt von den entprellten Eingängen (xrioIB2d und xrioIB3d) deklariert.

4.5.4 Der erste I/O-Zugriff durch das SPS-Programm

Für die Demonstration des I/O-Zgriffes sollen die ersten nicht entprellten 8 Eingänge des RIO 16 I auf die ersten 8 Ausgänge des RIO 16 O gelegt werden.
Eine POE mit dem Namen *XRIODemo* ist bereits angelegt und eine Instanz der POE ist in der Task *XRIOTsk* eingefügt.



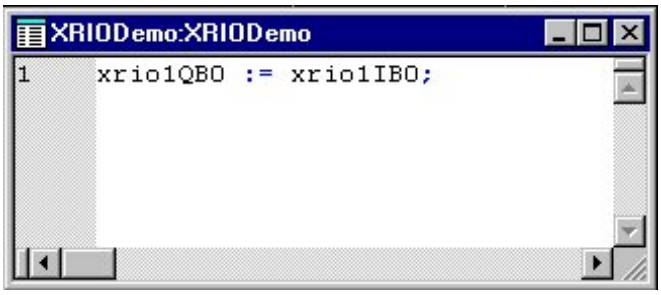
Die Arbeitsblätter der POE sind leer und sollen die lokale I/O-Variablendeklaration und das SPS-Programm aufnehmen.

- Zuerst müssen die I/O-Variablen deklariert werden. Dazu auf *XRIODemoV* klicken und die folgende Eintragungen vornehmen. Wichtig ist dass die Verwendung mit VAR_EXTERNAL angegeben wird, da die I/O-Variablen als globale Variablen vordefiniert wurden. Ganz einfach können die Einträge auch mit Kopieren/Einfügen (<Strg+C>,<Strg+V>) aus dem Arbeitsblatt *Global_Variables* auf dieses Arbeitsblatt übertragen werden.

XRIODemoV:XRIODemo

Name	Typ	Verwendung	Beschreibung
Default			
xrio1IB0	BYTE	VAR_EXTER...	XRIO card 1, module 1, bc 0, RIO 16I, DIGITAL
xrio1QB0	BYTE	VAR_EXTER...	XRIO card 1, module 2, bc 0, RIO 16O, DIGITAL

- Als nächster Schritt kann das SPS-Programm auf dem Arbeitsblatt XRIODemo erstellt werden.



Nach dem Kompilieren und Übertragen des Projektes kann die XCx gestartet werden.

Nach dem Start leuchten auf dem XRIO-Kloppler die grünen LED RUN und ACCESS. Die Eingänge 0 bis 7 auf dem RIO 16 I können mit DC 24 V angesteuert werden. Der entsprechende Ausgang auf dem RIO 16 O wird daraufhin gesetzt.

4.5.5 Diagnose

Mit dem Funktionsbaustein XRIO_STATE aus der Firmware Bibliothek PLC_Vxxx kann eine Diagnose durchgeführt werden. Er muss zyklisch in einer Task aufgerufen werden.

Beispiel zum Aufruf des Funktionsbausteines (Programm in ST):

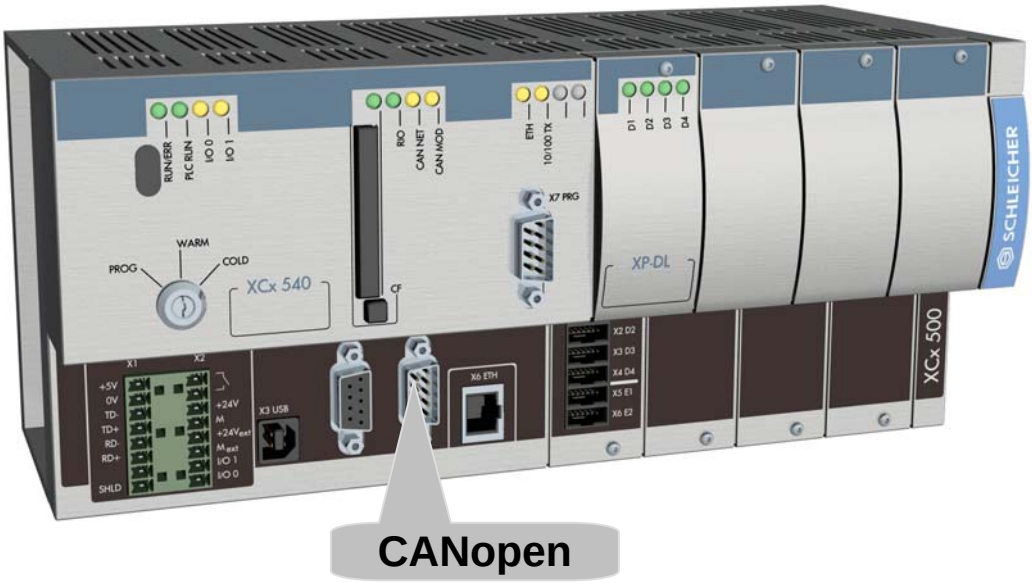
```
XRIO_STATE_1(ENABLE:=enable,CARD:=card,STATE:=state
,MODULES:=modules);
state   := XRIO_STATE_1.STATE;
modules := XRIO_STATE_1.MODULES;
error   := XRIO_STATE_1.ERROR;
enable  := FALSE;

Variables:
card        USINT
enable      BOOL
error       WORD
modules     XRIO_ModInfoType
state       XRIO_StateType


TYPE definition:
  XRIO_ModuleInfo  : STRUCT
    id             : UDINT;
    EBytes         : UINT;
    ABytes         : UINT;
    modClass       : UINT;
    general1       : UINT;
END_STRUCT (* XRIO_ModInfoType *);

XRIO_ModInfoType   : ARRAY [1..16] OF
XRIO_ModuleInfo;
  XRIO_StateType   : STRUCT
    nofModules     : INT;
    activeErrBits  : USINT;
    staticErrBits  : USINT;
    overload       : UDINT;
    _24VFail       : UDINT;
    stateErrCount  : UDINT;
END_STRUCT (* XRIO_StateType *);
```

4.6 CANopen für dezentrale I/O

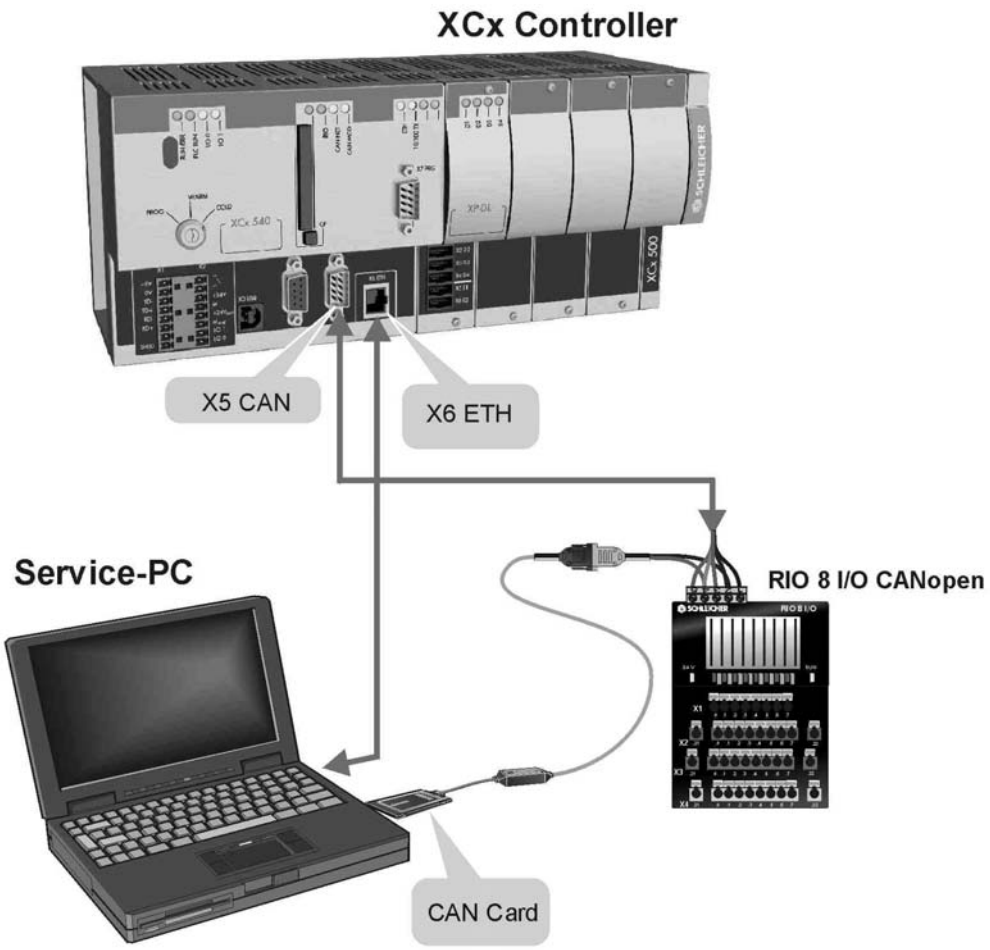


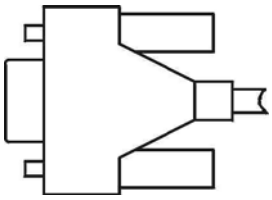
4.6.1 Spezifikationen

- CANopen arbeitet mit zwei Typen von Telegrammen:
  SDO (service data objects) sind Telegramme, die vom Empfänger **bestätigt** werden müssen,
  PDO (process data objects) sind Telegramme, die vom Empfänger **nicht bestätigt** werden müssen.
- Während der Netzwerk-Konfiguration werden die PDO`s für den Datenaustausch definiert und bekommen eine sogenannte COB ID. Der Empfänger einer Nachricht kennt zu jeder Zeit welches Telegramm für diesen Knoten bestimmt ist.
- Einige Komponenten unterstützen nur sogenanntes Default Mapping und arbeiten mit festen COD ID´s, die in der CANopen Definition festgeschrieben sind.
- Die Standardkommunikation für PDO ist **COS** (Change Of State): Eine PDO wird nur gesendet, wenn sich die Information innerhalb der PDO ändert.

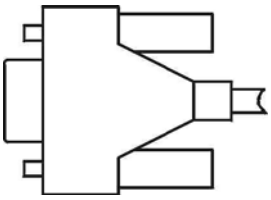
4.6.2 Anschlussprinzip und Verkabelung

Der beschriebene Aufbau ist eine Minimalkonfiguration, die für die weitere Inbetriebnahme als Beispiel dient.

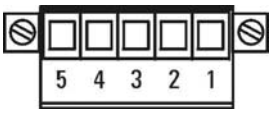




D-Sub, 9-polig, Buchse zur CAN Card im PC



D-Sub, 9-polig, Buchse zur XCx X5



Open Style Connector am RIO 8 I/O CANopen

Pin	Pin	Pin	
	3	1	0 V
2	2	2*	CAN_L
	5	3	Drain
7	7	4*	CAN_H
	9	5	DC +24 V

* Zwischen Pin 2 und 4 am RIO 8 I/O CANopen muss ein Abschlusswiderstand von 120 Ohm geschaltet werden.

4.6.3 Einstellungen am I/O-Modul RIO 8 I/O CANopen

Am Kompaktmodul RIO 8 I/O CANopen die Knotennummer 2 und die Datenübertragungsrate 125 kBaud einstellen.
Dazu den DIP-Schalter auf der Moduloberseite einstellen:

	Knotennummer							Daten-übertragungsrate		
Schalter	1	2	3	4	5	6	7	8	9	10
Stellung	OFF	ON	OFF	OFF	OFF	OFF	OFF	ON	ON	OFF

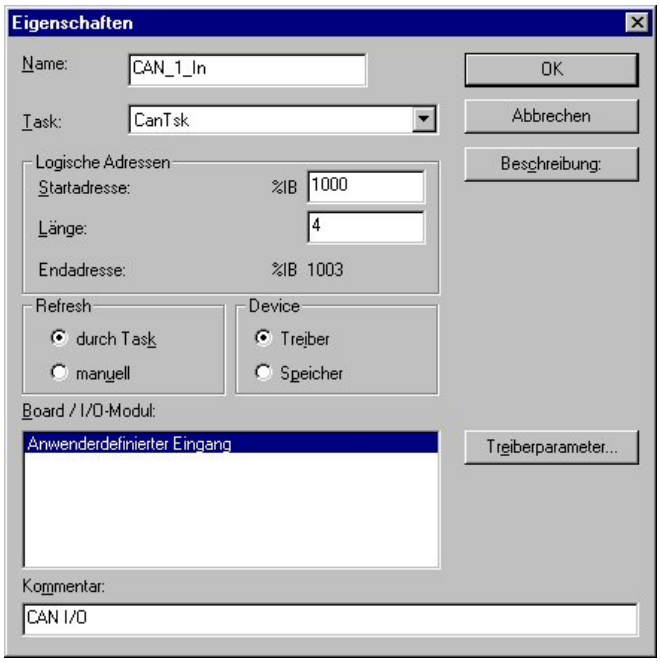
4.6.4 Deklaration des I/O-Treibers für CANopen

Ort und Prinzip der Deklaration des I/O Treibers für das CANopen-Netzwerk sind die selben wie für XRIO.

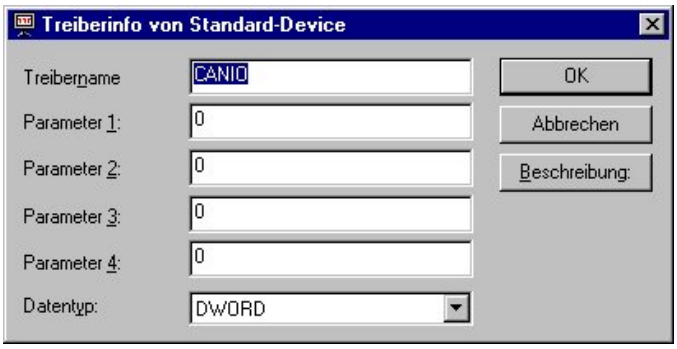
- Am Ende des Projektbaums befindet sich der Container IO_Configuration.

Dort sind bereits die I/O-Konfigurationen CAN_1_In und CAN_1_Out erstellt. Für dieses Beispiel müssen folgende Parameter eingetragen werden:

- Als *Startadresse* werden die logischen Adressen IB1000 für CAN_1_In und QB1000 für CAN_1_Out eingetragen.
- Die *Task* (mit welcher der I/O Adressraum synchronisiert wird) muss *CanTsk* sein.
- Im Paramter *Länge* werden so viele I/O Bytes deklariert, wie im CANopen Netzwerk ausgetauscht werden sollen (hier 4, da minimal Doppelwortabstände angewendet werden).



- In den *Treiberparametern* muss der Treibername CANIO eingestellt sein, der Datentyp ist DWORD



4.6.5 Deklaration von Netzwerkvariablen in MULTIPROG

- Im Fenster des Projektbaums unter *Global_Variables* sind im Ordner Network_Variables die benötigten Variablen vordefiniert.

Global_Variables:Configuration.Resource

Name	Typ	Verwendung	Beschreibung	Adresse	Anfangsw...	Reman...	PD
⊞ Global_Variables							
⊞ I/O_Variables							
⊟ Network_Variables							
IB1000	USINT	VAR_GLOBAL		%IB 1000		☐	☐
IB1001	USINT	VAR_GLOBAL		%IB 1001		☐	☐
IB1002	USINT	VAR_GLOBAL		%IB 1002		☐	☐
IB1003	USINT	VAR_GLOBAL		%IB 1003		☐	☐
IW1000	UINT	VAR_GLOBAL		%IW 1000		☐	☐
IW1002	UINT	VAR_GLOBAL		%IW 1002		☐	☐
ID1000	UDINT	VAR_GLOBAL		%ID 1000		☐	☐
QB1000	USINT	VAR_GLOBAL		%QB 1000		☐	☐
QB1001	USINT	VAR_GLOBAL		%QB 1001		☐	☐
QB1002	USINT	VAR_GLOBAL		%QB 1002		☐	☐
QB1003	USINT	VAR_GLOBAL		%QB 1003		☐	☐
QW1000	UINT	VAR_GLOBAL		%QW 1000		☐	☐
QW1002	UINT	VAR_GLOBAL		%QW 1002		☐	☐
QD1000	UDINT	VAR_GLOBAL		%QD 1000		☐	☐
⊞ PLC_Common							

- I/O-Bits werden im Arbeitsblatt „I_O_Variables“ mit den Adressen IX1000.0.. und QX1000.0 .. deklariert. (Im Beispiel wird QX1000.7 benutzt, um das Ergebnis am RIO 8 I/O sichtbar zu machen.)

Global_Variables:Configuration.Resource

Name	Typ	Verwendung	Beschreibung	Adresse	An
⊞ Global_Variables					
⊟ I/O_Variables					
CAN_1_IX0	BOOL	VAR_GLOBAL	CANopen 1. Netzwerk Eingang 0	%IX 1000.0	
CAN_1_QX7	BOOL	VAR_GLOBAL	CANopen 1. Netzwerk Ausgang 7	%QX 1000.7	
XRIO_1_IX0	BOOL	VAR_GLOBAL	XRIO Modul 1 (RIO 16 I) Eingang 0	%IX 0.0	
XRIO_1_IX1	BOOL	VAR_GLOBAL	XRIO Modul 1 (RIO 16 I) Eingang 1	%IX 0.1	
XRIO_2_QX0	BOOL	VAR_GLOBAL	XRIO Modul 2 (RIO 16 O) Ausgang 0	%QX 0.0	

4.6.6 Konfiguration des CANopen-Netzwerkes mit ProCANopen

- Die ProCANopen Software ist erforderlich, um ein Netzwerk zu konfigurieren.
  Zusätzlich benötigen Sie eine CAN-Feldbuskarte im Service PC z.B. „CANcardY“.
- Die Eigenschaften und Fähigkeiten der Komponenten sind in einer „eds“ Datei (electronic data sheet) deklariert. Die Eds-Datei muss in das Unterverzeichnis von ProCANopen mit dem Namen Eds kopiert werden.
- ProCANopen bildet die (abbildbaren) Objekte der Knoten ab. Z.B. Ausgangs Bytes (repräsentieren Eingangs Bits vom RIO) der RIO Module werden mit Eingangs Bytes der XCx verknüpft.
- Darüber hinaus werden zusätzliche Informationen für den Feldbus konfiguriert:
  Welcher Knoten ist der „NMT manager“ ?
  Welcher Knoten ist der „Configuration manager“ ?
  „Guarding“ und „guarding time“.
  „Sync time“ und „sync window length“.
- Nach der Netzwerkkonfiguration kann diese im Netzwerk gespeichert werden.
  Im Netzwerk speichern bedeuted, der Knoten - welcher als „Configuration manager“ (meistens XCx) ausgewählt ist – bekommt via CANopen die Information, wie das Netzwerk zu konfigurieren ist. Der „Configuration manager“ speichert die Information (z.B.XCx auf der flash disk) und nach dem Einschalten konfiguriert die XCx das Netzwerk.
- Nachdem das Netzwerk konfiguriert ist, kann der „NMT manager“ das CAN Netzwerk starten. Netzwerk starten bedeutet, Status „operational“ und Datenaustausch von PDOs (process data objects).

4.6.7 Installation von ProCANopen

- Um ProCANopen zu installieren, folgen Sie bitte der Dokumentation die mit der Software und der CAN-Karte geliefert wurde.
- Sie müssen die Treiber und die ProCANopen Software in zwei Schritten installieren.
- Kopieren sie die aktuellen EDS-Dateien für XCS und XCN von der CD in den Pfad .\ProCANopen\EDS\...
- Wenn die Dialogsprache Englisch gewünscht ist: Editieren Sie in der Datei \....\proCANopen\exe\vector.ini die Zeile language=0049 in =001.

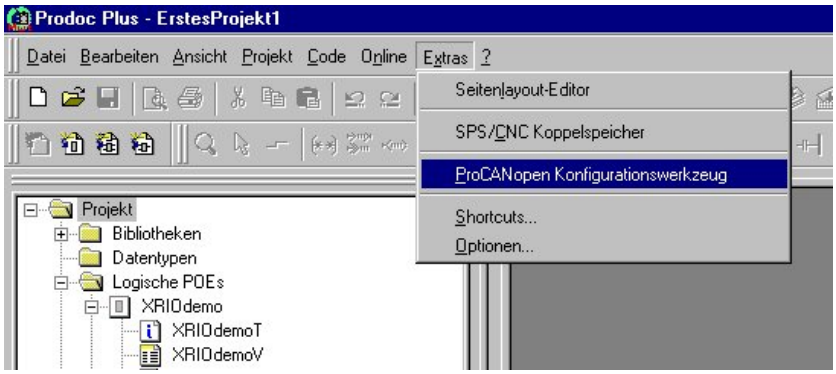
- Hinweise für Anwender, welche bereits ProCANopen Version 2.1 und MULTIPROG 1.2 für die Schleicher Steuerungen MicroLine und ProNumeric/ProSycon installiert haben:
- Sie benötigen nur die Update Version von ProCANopen.
- Überschreiben Sie nicht die bereits installierte Version !
- Installieren Sie ProCANopen V3.2 in einem neuen Pfad auf der Hard Disk z.B. \ProCANopen3.
- Wenn Sie eine CANCardX im Einsatz haben, ist es manchmal erforderlich, ein Update der Firmware und der Options auf der Karte vorzunehmen: Bitte notieren Sie die Seriennummer der Karte und setzen Sie sich mit Ihrem lokalen Händler in Verbindung.
- In Abhängigkeit von Ihrem PC Betriebssystem müssen Sie verschiedene Treiber installieren. Einige Neue Treiber sind nicht kompatibel mit der älteren ProCANopen Version 2.1 ! Dies bedeutet, ProCANopen Version 2.1 mit dem neuen Treiber V3.x funktioniert nicht mehr online mit dem Feldbus !
- Mit ProCANopen V2.1 projektierte Konfigurationsdateien können mit ProCANopen V3.2 weiter benutzt werden

---

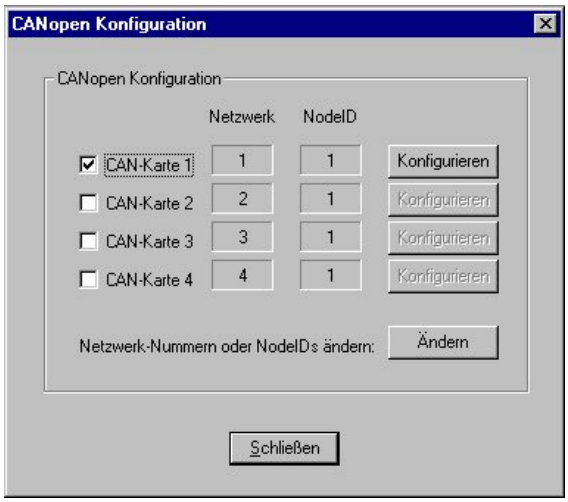
4.6.8 Einbindung von ProCANopen in MULTIPROG

Die Installation der AddOn's bereitet MULTIPROG so vor, dass ProCANopen direkt von MULTIPROG aus gestartet werden kann.

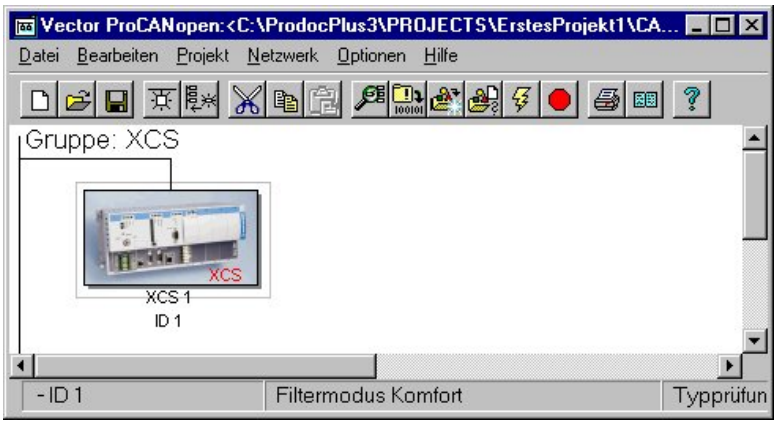
- Im Menü *Extras/ProCANopen Konfigurationswerkzeug* wählen.



- Dann kann die CAN-Karte der XCx angewählt werden. Im Beispiel wird nur die eine standartmäßig vorhandene Karte benutzt. Als Knotennummer (NodeID) kann 1 beibehalten werden.

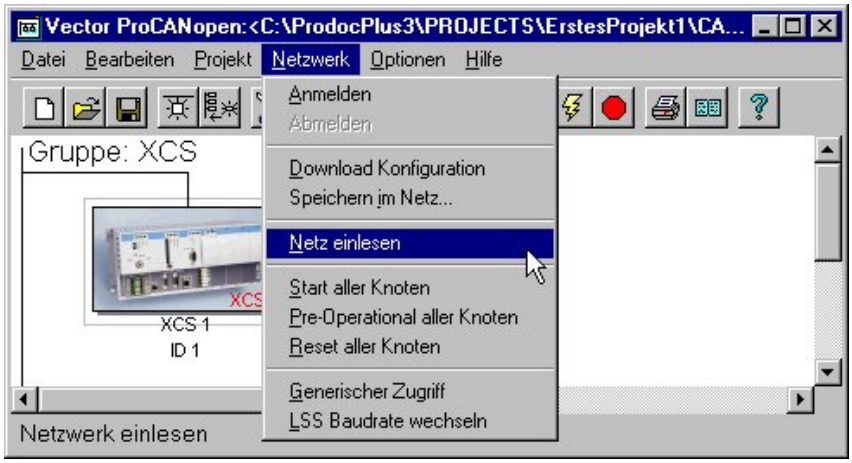


- Auf den Button *Konfigurieren* klicken. ProCANopen startet direkt mit dem richtigen CANopen Projekt.

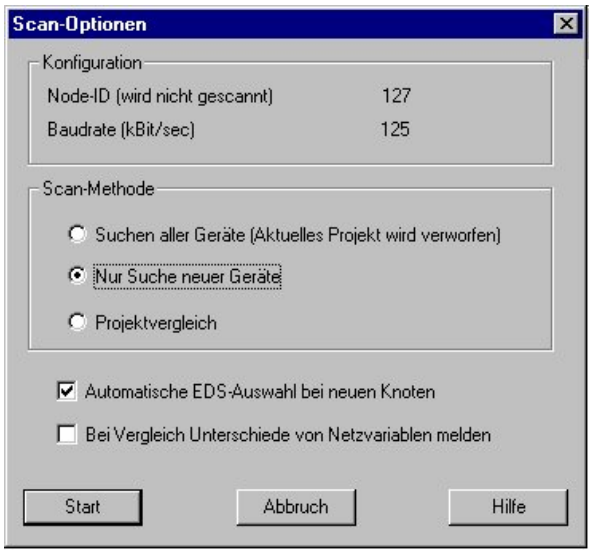


4.6.9 Erste Verbindungen mit ProCANopen

Zuerst muss das Netzwerk eingelesen werden.

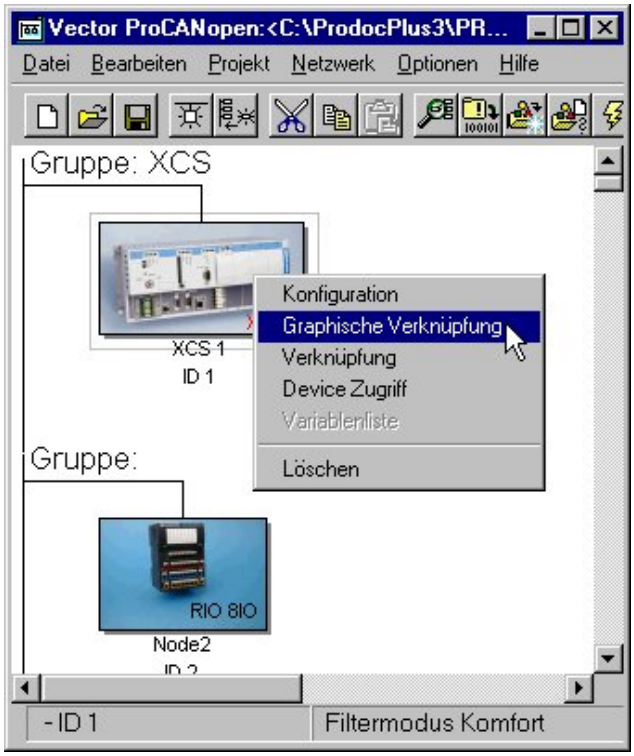


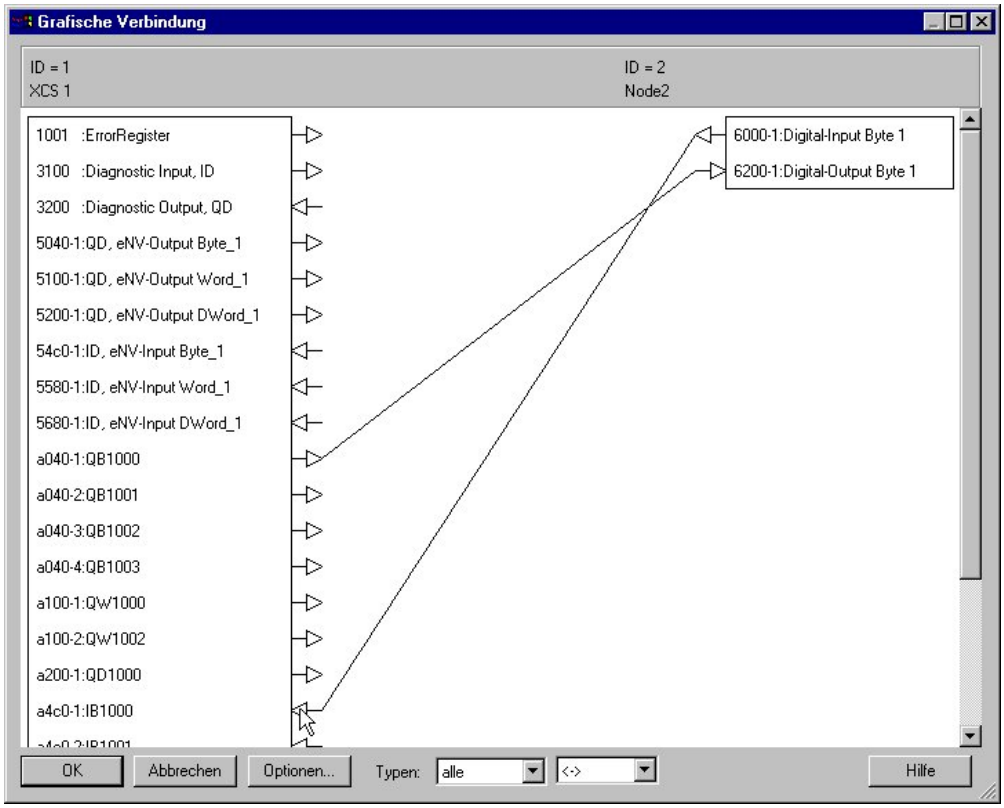
Da das Netzwerk mit Knoten 1 XCx bereits vorkonfiguriert ist muss das Einlesen mit der Scan-Option *Nur Suche neuer Geräte* ausgeführt werden.



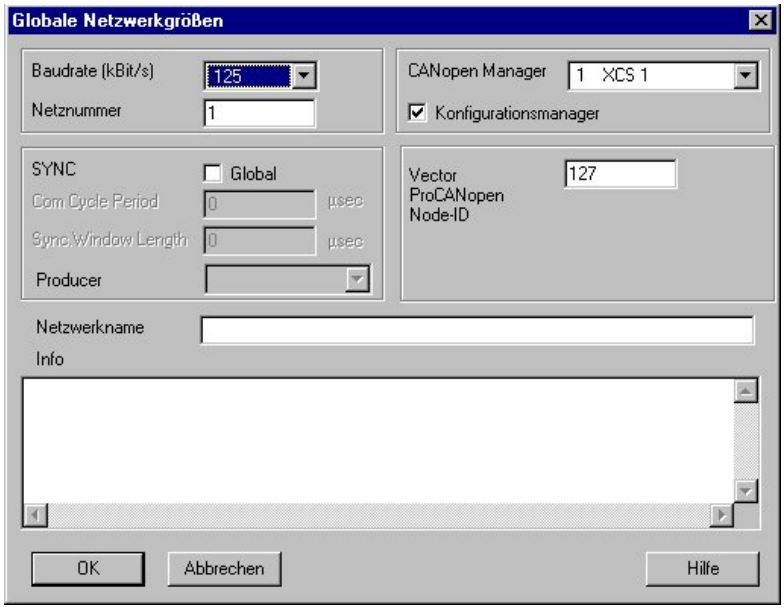
Dann können die Verknüpfungen der Netzwerk-Knoten projektiert werden.

- Mit der rechten Maustaste auf die XCx klicken, im Kontextmenue des Knotens *Graphische Verknüpfung* auswählen und dann auf den Knoten klicken, mit dem die Verbindung hergestellt werden soll (im Beispiel Knoten 2 RIO 8 I/O).

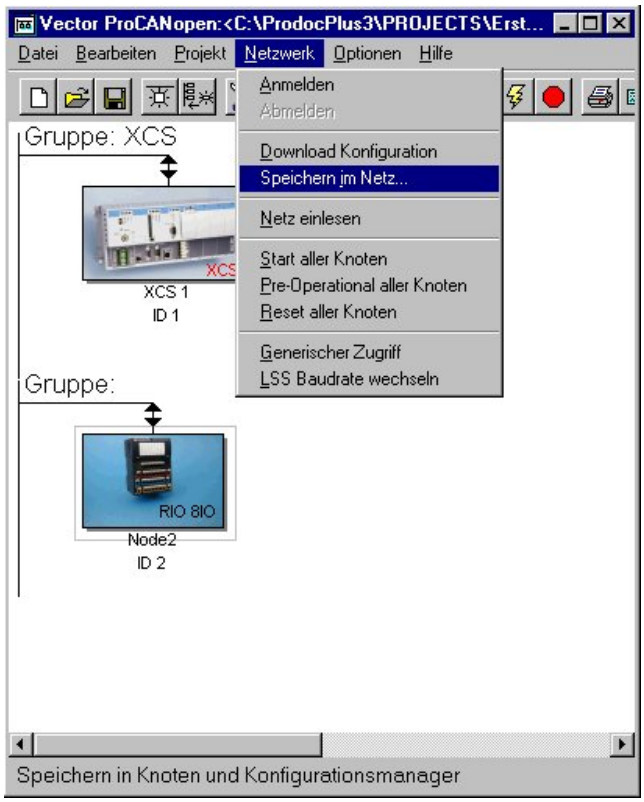




- In *Projekt/Globale Konfiguration* den Canopen manager Node 1 XCx als Konfigurationsmanager auswählen.

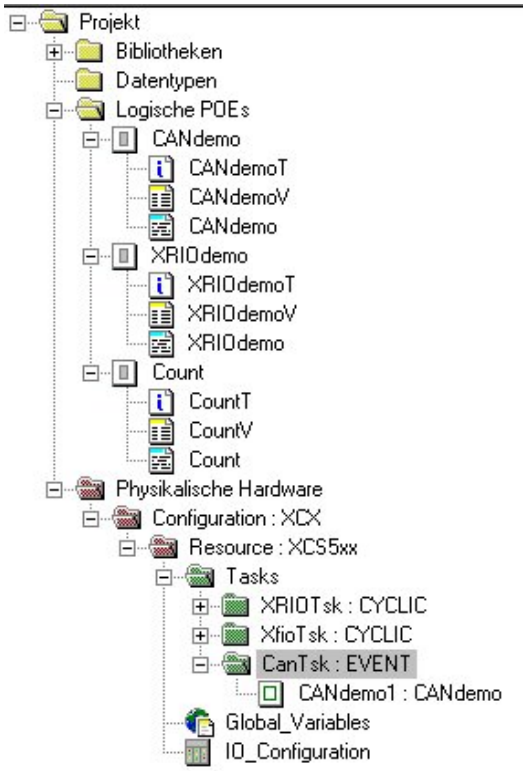


- Mit *Speichern im Netzwerk* wird die CAN Konfiguration in den Konfigurationsmanager gespeichert.
  Die XCx wird die Daten in den Compact Flash speichern und nach dem Einschalten wird das Netzwerk gebootet.

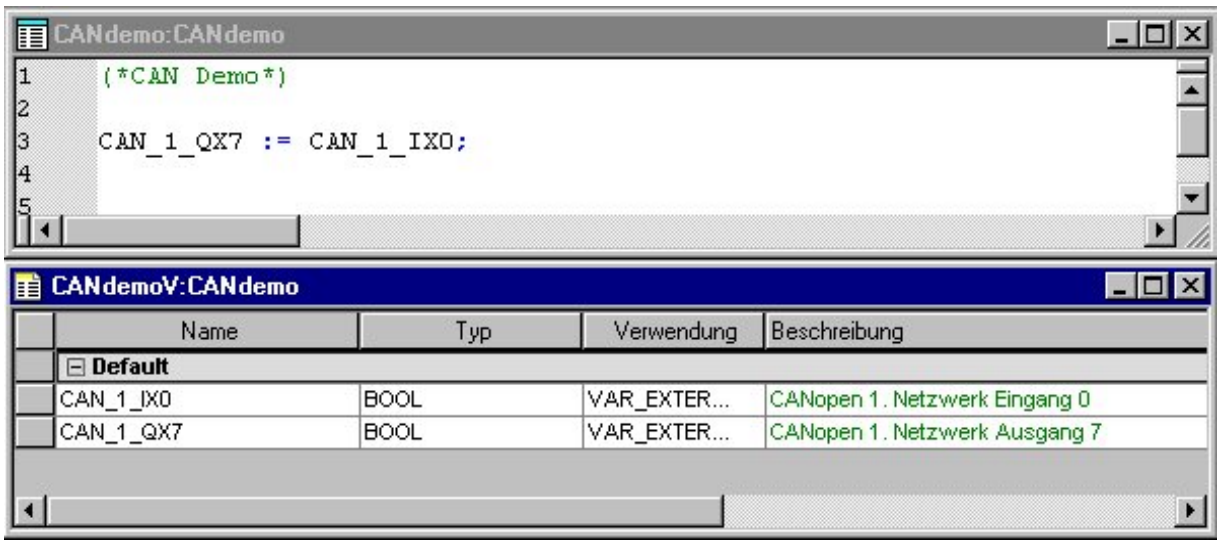


Weitere Hinweise zur CAN Konfiguration sehe Betriebsanleitung Inbetriebnahmehinweise für Feldbussysteme Artikel-Nr. 322 152 48.

Zum Test der Netzwerkverbindung muss eine neue POE (hier CANdemo) erstellt und in der CanTsk instanziiert werden.

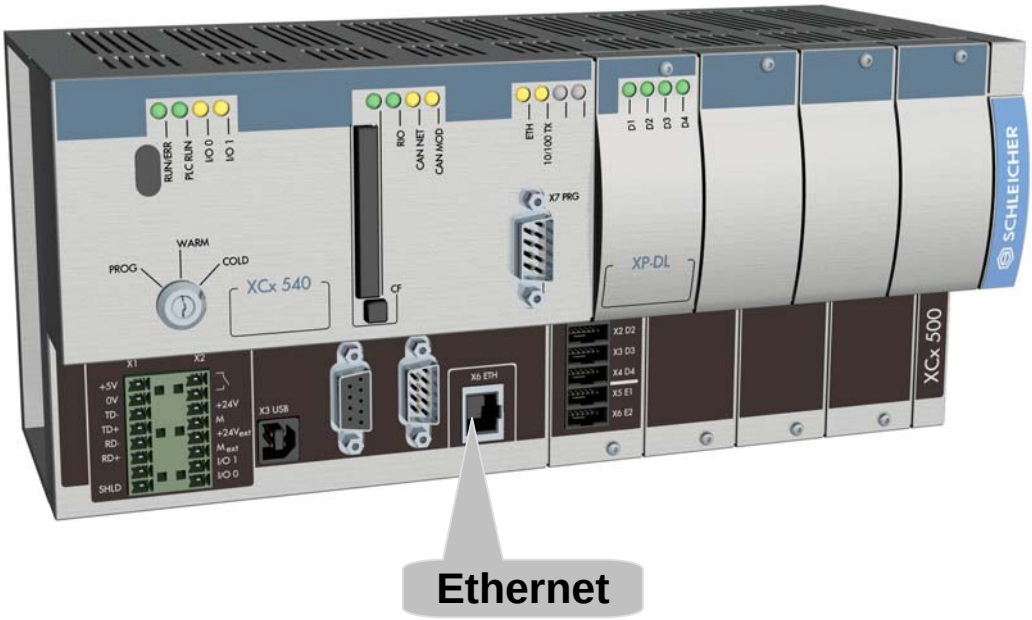


Die POE CANdemo mit der dazugehörigen Variablen-Arbeitsblatt:



Wird an den Eingang 0 des RIO 8 I/O CANopen 24 V angelegt, wird der Ausgang 7 auf 1 gesetzt.

4.7 Die Web-Server Funktionen der XCx



4.7.1 Allgemeine Funktionen und Konzept

- Der Hauptvorteil der Web-Server Technologie ist, dass alle Daten für Maschine und Visualisierung an einer Stelle abgelegt werden: Compact Flash der XCx.
- Der Web-Server ist im Betriebssystem der XCx implementiert.
- Applicationen (Web Pages) werden in einen gesonderten Bereiche auf den Compact Flash übertragen (mit FTP oder direktem Kopieren auf die Disk). Von dort liest der Web-Server die Daten und sendet sie zum Browser.
- Der Browser ist der „thin-client“ für die Datenvisualisierung. Die Applikation (HTML,JavaScript,Java) wird von der Steuerung geladen und läuft innerhalb einer Shell ab.
- Andere Visualisierungspatches müssen auf jedem Bediengerät installiert werden (Webfactory etc.). Dies nennt man ein „fat-client“ Konzept.

4.7.2 Schleicher spezifisches Applet

Normalerweise ist die Web Technologie ein Herunterladen in einer Richtung zum Browser und die Web Seite selbst ist dynamisch (Animations gifs oder Flash-files). Ein zyklischer Parameterrefresh ist nicht möglich !

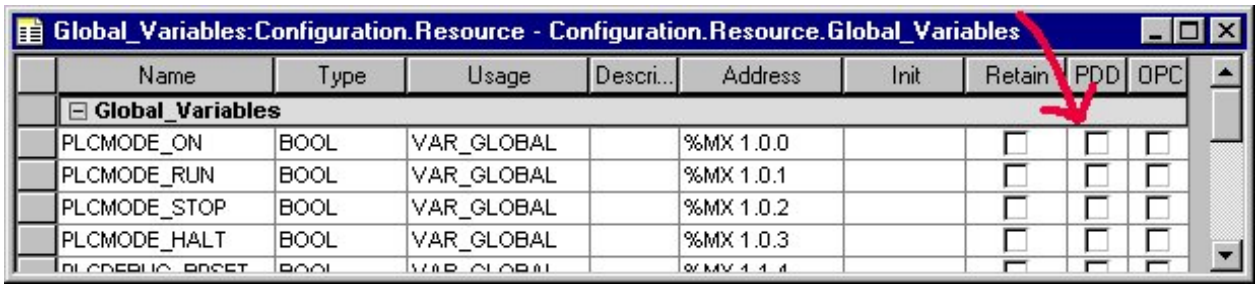
Schleicher liefert ein spezielles Java Applet, das einen Datenaustausch bidirektional zwischen Browser und Steuerung ermöglicht.

Dieses Applet unterstützt Funktionen, die von der HTML/ Java script language aufgerufen werden können.

Diese Funktionen ermöglichen der Applikation, einzelne oder mehrere Variablenwerte der SPS zu schreiben.

4.7.3 Deklaration von Variablen zur Visualisierung

Die Variablen die visualisiert werden sollen sind in MULTIPROG mit der Checkbox PDD zu markieren (PDD = Process Data Directory).



Diese Variablen werden in der XCx in einer internen Liste aktualisiert. Der Web Server kann die Variablen dieser Liste lesen und schreiben.

4.7.4 Applikationsbeispiel

Die XCx wird mit einer Standard Browser Applikation ausgeliefert.

Diese Applikation ermöglicht PDD markierte Variablen zu lesen und zu schreiben. Zusätzlich wird ein Statusüberblick gegeben.

4.7.5 Browser / Komponenten

- Alle standard PC's mit Ethernet sind möglich.
- MS Explorer V 5 oder höher oder Netscape Navigator V 6 oder höher sind erforderlich.
- Einige Terminals mit Windows®-CE sind möglich, wenn der Browser die Anforderungen wie Java Script 1.5, Java 2 , HTTP1.1 erfüllt.

4.8 Allgemeine Hinweise zur Inbetriebnahme

4.8.1 Steuerungsanlauf mit leerer Puffer-Batterie (batterieloser Betrieb)

Die XCx- Steuerungen verwenden einen batterie-gepufferten SRAM zum nullspannungssicheren Speichern von remanenten Daten. Hierzu zählen SPS Retain-Variablen, CNC-Daten (cncRMem) aber auch Einträge des ErrorLogBooks.

Falls die Steuerung über einen längeren Zeitraum nicht eingeschaltet wurde (> 3 Monate) ist u. U. die Pufferung nicht mehr wirksam und der SRAM- Inhalt ungültig; die SPS geht nicht mehr automatisch auf RUN.

Der Anwender kann in diesem Fall über den Dialog oder im Terminal (VxWorks-Prompt: Anweisung SetClock und Validate SRAM) den SRAM erneut „validieren“.

Um die Validierung bei bestimmten Applikationen zu vermeiden, ist ab SW V06.09/1 der sog. batterielose Betrieb möglich.

Hierzu muss lediglich im Run-Script „/ata0/OS/Run.vxs“ der Eintrag ***pc_main(0,"/ata0")*** durch ***pc_main(4,"/ata0")*** ersetzt werden:

```
cd "/ata0"
CheckBootIni "OS/bootline.ini",0
pc_main(4,"/ata0")
```

Der Eintrag bewirkt ein Löschen bzw. die automatische Reinitialisierung und Validierung des SRAMs. [1]

Die SPS startet im Anschluss automatisch (Schalterstellung COLD, WARM).

[1] Falls die Uhrzeit ebenfalls ungültig ist, wird diese auf den 1.1.2006, 0:00 Uhr gesetzt.

4.8.2 Grundinitialisierung

Die Grundinitialisierung dient zum Rücksetzen der Steuerung auf die Geräte-Defaultwerte, d.h. den Auslieferungszustand.
Kundendaten auf der Compact Flash Card werden nicht beeinflusst.

Durchführen der Grundinitialisierung

1. Betriebsartenschalter in die Stellung PROG stellen.
2. Reset-Taster ständig gedrückt halten.
3. Betriebsartenschalter in die Stellung COLD stellen.
   Warten bis die LEDs RUN/ERR und PLC/RUN gleichzeitig gelb blinken.
4. Reset-Taster loslassen, die RUN/ERR LED blinkt rot, die PLC/RUN LED blinkt gelb.
5. Reset-Taster innerhalb von 8 s kurz betätigen, die Grundinitialisierung beginnt.
   Wird der Reset-Taster nicht innerhalb von 8 s betätigt, läuft die Steuerung wie nach einem normalen RESET an.

Nach der Grundinitialisierung wird die Steuerung im "Abgesicherten Mode" gestartet.
Im "Abgesicherten Mode" laufen weder SPS noch CNC, der Zugriff auf die Steuerung über FTP ist aber möglich.

Die Grundinitialisierung setzt die IP-Adresse der Steuerung auf die Standard-IP-Adresse 192.168.1.2

Der "Abgesicherte Mode" kann nur durch einen Neustart der Steuerung (Aus- und Einschalten der Versorgungsspannung oder RESET) verlassen werden.

Ausgeführte Vorgänge bei der Grundinitialisierung

- Rücksetzen der BootLine (im internen Flash) auf die Werte der Default- BootLine
  (z.B. auf die Standard IP- Adresse 192.168.1.2)
- Hochfahren im „Abgesicherten Mode“, d.h. ohne Ausführung von *run.vxs*.
- Ausführen des sog. Init Runscripts (*init.vxs*)
  Hier können weitere Kommandos abgesetzt werden, wie z.B. ClearSram(0) zum Löschen des SRAM- Bereiches (Retain- und Super-Retain-Variablen, remanente CNC-Daten und Error Logbook).
  Das Setup kopiert neben *run.vxs* auch die Files Init.vxs auf die Compact Flash Card.
  Backup.vxs kann auch aus der VxWorks Kommando Shell aufgerufen werden. Es erzeugt ein Image von *ata0* auf *ata1*. Dabei wird *Image.vxs* in *Init.vxs* umbenannt. Bei einer Grundinitialisierung wird dann *ata1* auf *ata0* kopiert.

5 Bedienung der XCx mit dem Schleicher-Dialog

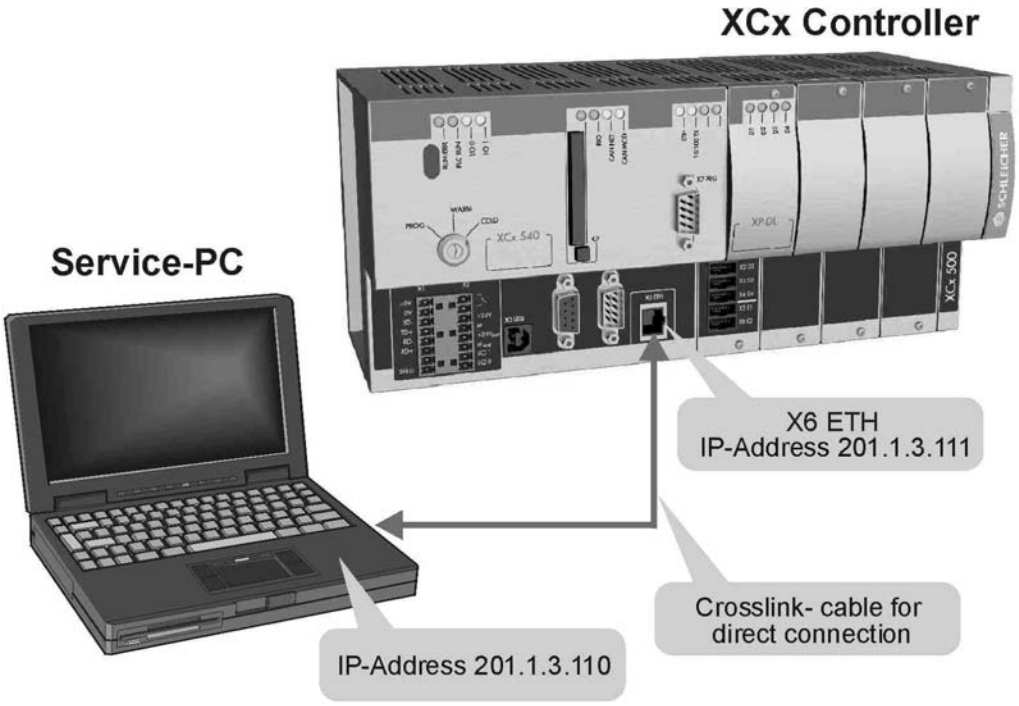
5.1 Übersicht

Der Schleicher-Dialog stellt Bildschirmseiten zur Inbetriebnahme und Bedienung der XCS und XCN zur Verfügung.

Je nach verwendeter Steuerung wird zwischen Schleicher-Dialog für XCS und XCN unterschieden. Die Erkennung des Steuerungstyps führt das Programm automatisch aus.

Voraussetzung ist ein auf dem PC installierter OPC-Server.

Die Ethernet-Verbindung des PC mit der XCx muss in Betrieb sein.



5.2 Installation auf dem PC und Vorbereitung des OPC-Servers

5.2.1 Installation des OPC-Servers

OPC-Server 1.x

Aufruf des Setup-Programms
PcosOpcr.exe

Nach der Installation des OPC-Servers 1.x müssen die Schleicher AddOns für MULTIPROG installiert werden (Setup-Programm *MwtAddOns.exe*). Auch wenn das Setup-Programm für die Schleicher AddOns für MULTIPROG bereits früher ausgeführt wurde, muss es nach der Installation des OPC-Servers erneut ausgeführt werden.

OPC-Server 2.0

Aufruf des Setup-Programms
ProConOS OPC-Server 2.0Desktop_xx.exe

5.2.2 Installation des Schleicher-Dialoges

Zur Installation muss das Installationsprogramm des Schleicher-Dialoges *Setup.exe* gestartet werden.

Die Programmdateien für den Schleicher-Dialog werden standardmäßig im Verzeichnis *D:\Schleicher\Dialog* installiert. Es kann jedoch auch ein anderes Verzeichnis gewählt werden.

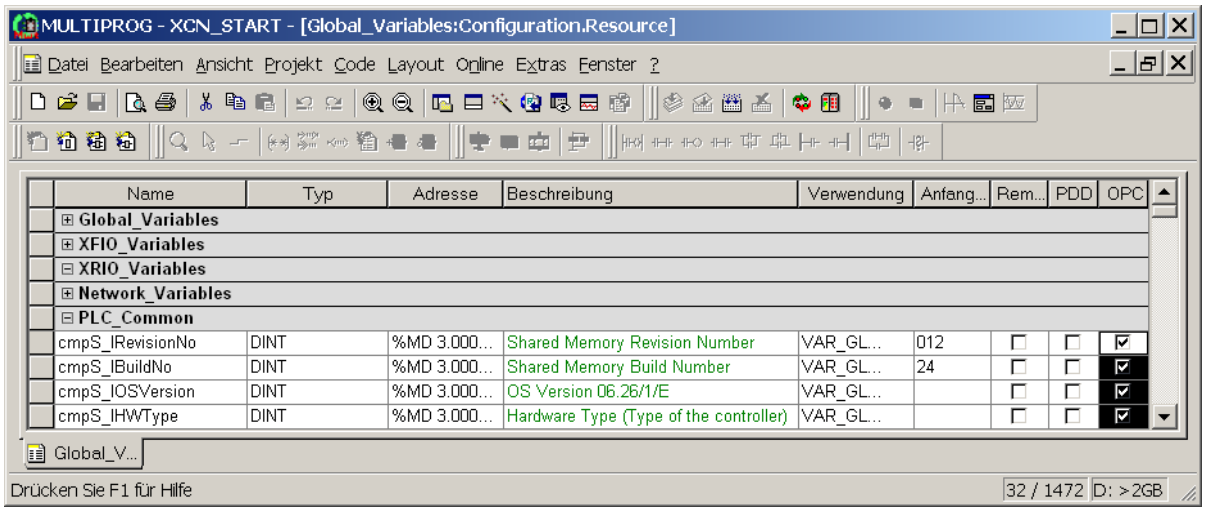
Während der Installation muss vom Benutzer festgelegt werden, welche Version des OPC-Servers (1.x oder 2.0) verwendet werden soll.

5.2.3 Deklaration der OPC-Variablen im SPS-Projekt mit MULTIPROG

Alle Variablen, die der OPC-Server zur Verfügung stellen soll, müssen zuvor in der Variablen-Tabelle des zugehörigen MULTIPROG Projektes als OPC-Variablen gekennzeichnet werden, indem die CheckBox *OPC* für diese Variablen aktiviert wird.

Der Anwender muss die globalen Variablen und / oder die lokalen Variablen einer POE als OPC-Variablen deklarieren.

Die Variablen des Koppelspeichers sind standardmäßig als OPC-Variable deklariert.



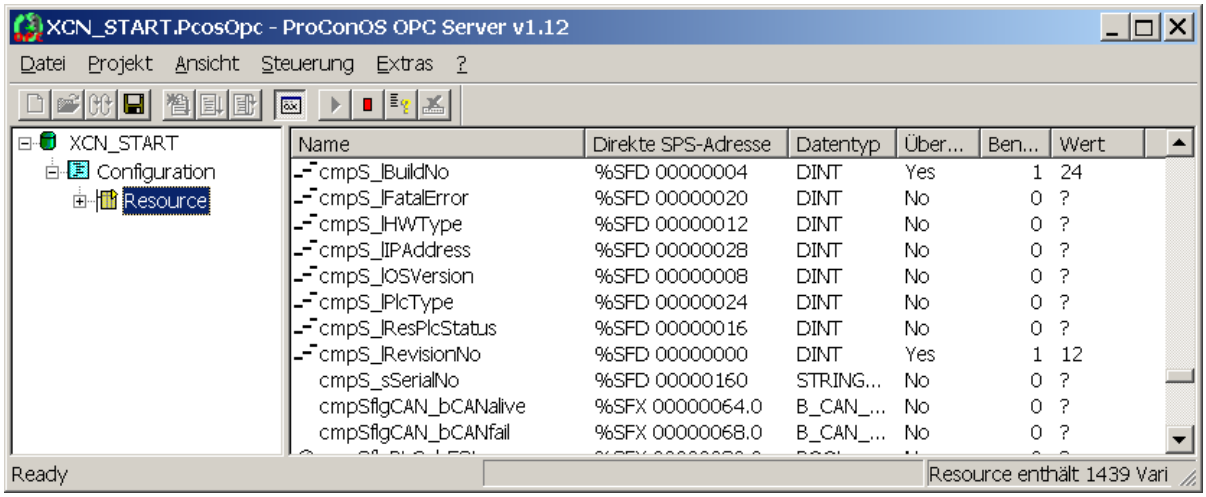
- Übersetzen des SPS-Projektes.
- Übertragen des Projektes mit MULTIPROG an die Steuerung.
- Starten der Steuerung.

5.2.4 Konfiguration und Test des OPC-Servers

OPC-Server 1.x

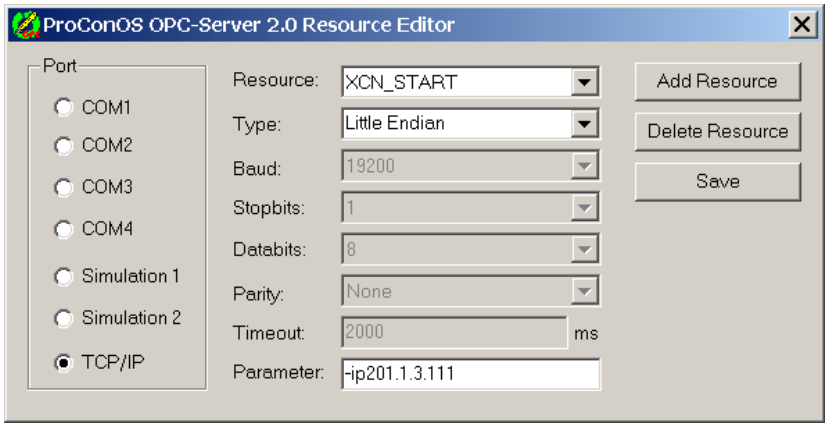
Ein neues OPC-Projekt erzeugen und ein übersetztes SPS-Projekt einfügen:

- ProConOS OPC-Server starten
- MULTIPROG Projekt in den OPC-Server einfügen (Menü *Projekt*, Menüpunkt *Projekt einfügen*, gewünschtes MULTIPROG Projekt auswählen, z.B. XCN_start.mwt)
- OPC-Server Projekt speichern (Menü *Datei*, Menüpunkt *OPC-Projekt speichern unter*, den vorgeschlagenen Projektnamen übernehmen)
- Zur Kontrolle, ob die Verbindung zwischen OPC-Server und Steuerung ordnungsgemäß funktioniert, kann der Wert einer oder mehrerer Variablen im Fenster des OPC-Servers angezeigt werden (dazu in der Baumstruktur im linken Teil des Fensters den gewünschten Knoten anwählen und einen Doppelklick auf einen Variablennamen im rechten Teil des Fensters ausführen, daraufhin wird der Wert der Variablen angezeigt).



OPC-Server 2.0

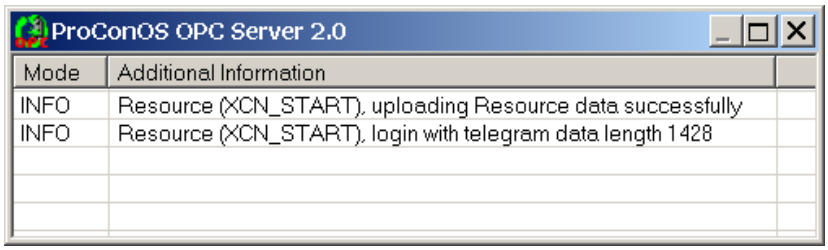
- **OPC-Ressource-Editor** starten.
  (Über *Start/Programme/KW-Software\ProConOS OPC-Server 2.0 Desktop/OPC Resource Editor* das Programm starten.)
  Den Namen der Ressource und die IP-Adresse der Steuerung eingeben. Dazu muss eine neue Ressource angelegt werden (Button *Add Resource*).
  Die neuen Einstellungen abspeichern (Button *Save*) und den OPC-Ressource-Editor beenden (Button *X*).



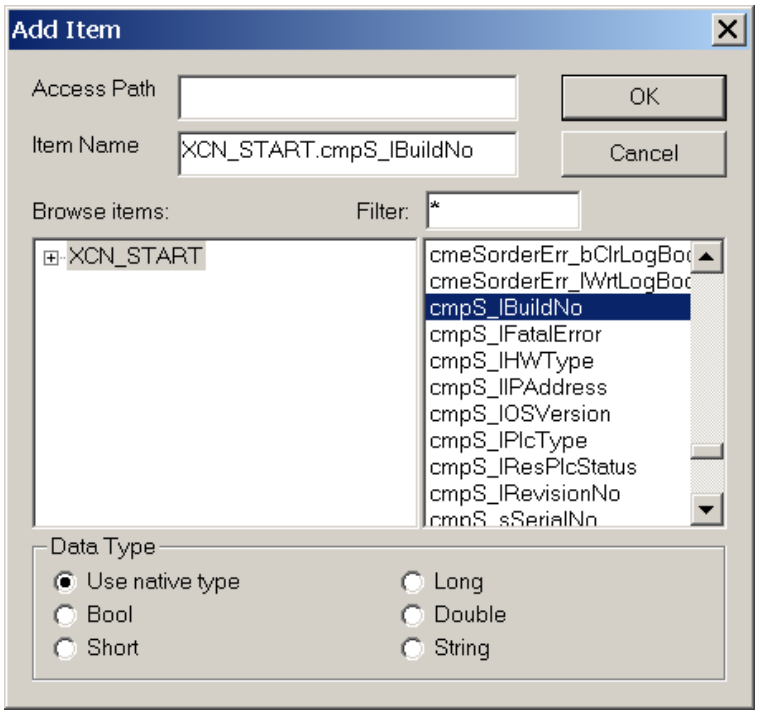
- **ProConOS OPC-Server** starten.
  (Über *Start/Programme/KW-Software\ProConOS OPC-Server 2.0 Desktop/ProConOS OPC-Server 2.0 Desktop* das Programm starten.)
  Es erscheint ein kleines Icon im Windows-System-Tray.

- Mit der rechten Maustaste auf dieses Icon klicken und *Server Status* ... auswählen.

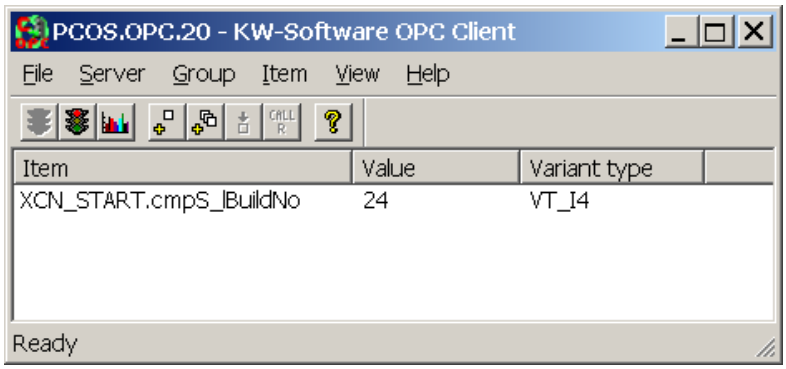
- Die Verbindung zur Steuerung wurde ordnungsgemäß erstellt, wenn folgende Meldungen im Statusfenster des OPC-Servers erscheinen.



- **OPC-Test-Client** starten.
  (Über *Start/Programme/KW-Software\ProConOS OPC-Server 2.0 Desktop/Tools/OPC Test-Client* das Programm starten.)
  Mit dem Befehl *Connect*... im Menü *Server* eine Verbindung zum OPC-Server herstellen.
  Mit dem Befehl *Add item* ... im Menü *Group* eine Variable auswählen.



- Nach Bestätigung mit OK wird der Wert der ausgewählten Variablen im Hauptfenster des OPC-Test-Clients angezeigt.



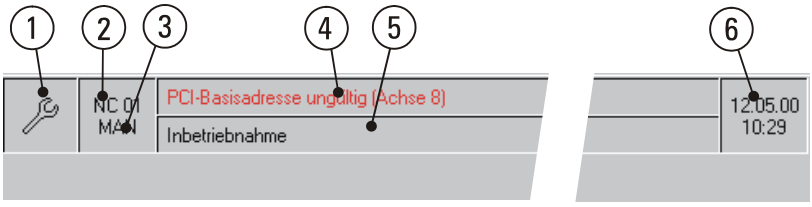
- **Schleicher-Dialog** starten.
  Nachdem nun alle Einstellungen vorgenommen worden sind, kann der Schleicher-Dialog gestartet werden.

5.3 Allgemeiner Aufbau der Bedienoberfläche

Aufbau der Fenster

1	Status- und Meldebereich
2	Arbeitsbereich Der Arbeitsbereich ist der Bereich des Bildschirms, eines Fensters oder einer Dialogbox, in dem Einstellungen vorgenommen werden können, bzw. Informationen dargestellt werden.
3	Hinweisbereich
4	Softkeybereich Im Softkeybereich werden die Funktionstasten beschriftet.

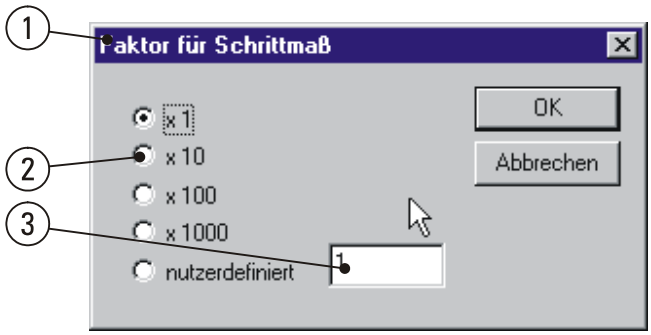
Status- und Meldebereich



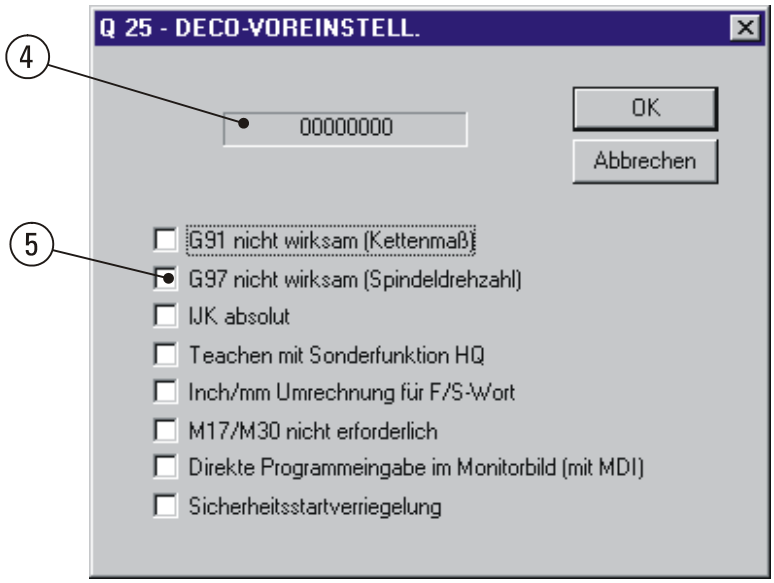
1	Aktuelle Bedienbereiche auswählen
2	Angewähltes NC-Teilsystem
3	Aktuelle NC-Betriebsart
4	Fehlermeldungen
5	Aktuelle Position im Steuerungsmenü
6	Datum und Uhrzeit

Aufbau der Dialogboxen

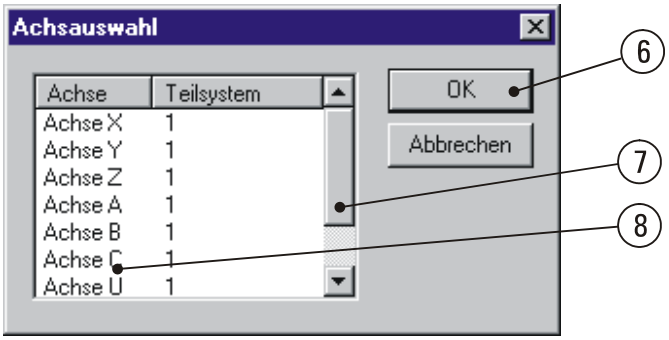
Der Aufbau und die Bedienung der Dialogboxen entspricht den Konventionen von Windows®.



1	Titelleiste Die Titelleiste enthält Informationen, die den Benutzer über den Inhalt des Fensters und evtl. über seine aktuelle Tätigkeit informiert.
2	Auswahlliste (nur eine Option auswählbar)
3	Eingabefeld Ein Eingabefeld ist ein Element, das für Dateneingaben des Benutzers vorgesehen ist. Die darin enthaltenen Werte können gelöscht, verändert oder auch unverändert belassen werden, wenn das Feld fokussiert ist.



4	Ausgabefeld Das Ausgabefeld dient der Darstellung von Informationen an den Benutzer. Die darin enthaltenen Daten können weder verändert noch gelöscht werden.
5	Auswahlliste (mehrere Optionen auswählbar)



6	Aktionstaste
7	Verschiebebalken Ist der Fensterinhalt größer als der angezeigte Bereich, wird dies durch ein Verschiebebalken gekennzeichnet. Der Schieberegler auf dem Verschiebebalken zeigt die aktuelle Position des sichtbaren Bereichs in Relation zum Gesamtbereich an. Ein Anzeigeausschnitt kann mit Hilfe des Verschiebebalkens gewählt werden.
8	Auswahlliste (nur eine Option auswählbar)

Die Zustände der Bildschirmelemente

fokussiert	Reserve G53 selbsthaltend Änderung interner Q-Pa	Ein Bildschirmelement ist fokussiert, wenn es durch den Fokusrahmen gekennzeichnet ist. Nur ein fokussiertes Element kann bearbeitet (Eingabefeld) oder selektiert werden. Innerhalb einer Gruppe kann der Focus mit den Cursortasten bewegt werden.
selektiert	Reserve G53 selbsthaltend Änderung interner Q-Pa	Ist ein Bildschirmelement fokussiert kann es selektiert (angewählt) werden. Ein selektiertes Feld wird als solches kenntlich dargestellt (z.B. Marke, Farbhinterlegung).
gesperrt		Ein Bildschirmelement gilt als gesperrt, wenn es nicht fokussiert werden kann. Dieser Zustand wird grafisch kenntlich angezeigt (Kontrastminderung).

5.4 Schleicher-Dialog der XCS

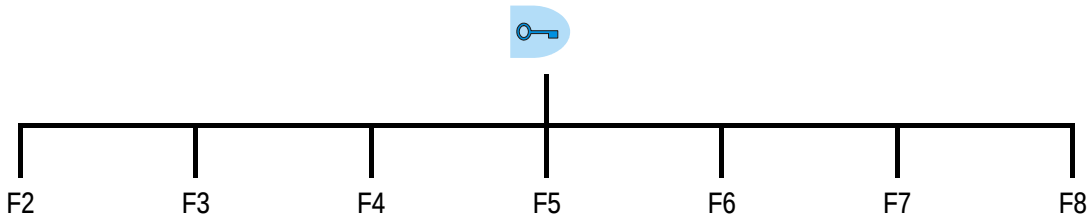
Dieses Kapitel ist nur für XCS gültig.

Das Steuerungsmenü und die Bedienbereiche der XCS

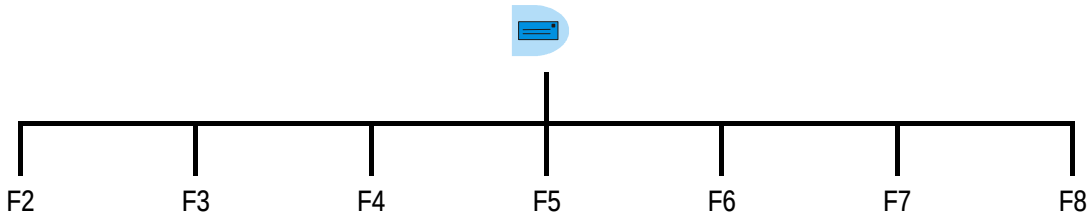
Die oberste Ebene des Steuerungsmenüs besteht aus Bedienbereichen. Die Bedienbereiche orientieren sich an den für die Bedienung wichtigen Tätigkeiten.

Der Softkey F1 wird immer zum Aufruf der Hilfeseiten verwendet. Die Hilfeseiten enthalten weitergehende Informationen zum Inhalt der tieferliegenden Bedienebenen.

Zugangsberechtigung <Ctrl+F4>



Fremdsoftware starten <Ctrl+F5>



Start Applikation 1	Start Applikation 2	Start Applikation 3	Start Applikation 4	Start Applikation 5	Start Applikation 6

Inbetriebnahme <Ctrl+F6>
(Softkeyebene 1)

F2	F3	F4	F5 Boot-Ein-stellungen	F6 OPC-Variablen	F7 Info	F8 Weiter >>
			Editieren	Wert ändern		
			CAN-Ein-stellungen	Editieren		
				Neu		
				Löschen		

Inbetriebnahme <Ctrl+F6>
(Softkeyebene 2)

F2 Programmein stellungen	F3	F4	F5 Uhr stellen	F6 Verbinden	F7 Info	F8 << Zurück
Eigen-schaften						
Ressource hinzufügen						
Ressource löschen						
Ressource nach oben						
Ressource nach unten						

5.5 Schleicher-Dialog der XCN

Dieses Kapitel ist nur für XCN gültig.

Das Steuerungsmenü und die Bedienbereiche der XCN

Die oberste Ebene des Steuerungsmenüs besteht aus Bedienbereichen. Die Bedienbereiche orientieren sich an den für die Maschine wichtigen Tätigkeiten.

Der Softkey F1 wird immer zum Aufruf der Hilfeseiten verwendet. Die Hilfeseiten enthalten weitergehende Informationen zum Inhalt der tieferliegenden Bedienebenen.

Handbetrieb <Ctrl+F1>

F2 Referen-zieren	F3 Verfahren	F4	F5	F6	F7 Teilsystem	F8
Achsen	Jog					
	Schrittmaß					
	Zielwert-vorgabe					
	Handrad					
	Achse					
	Eilgang					
	Nullsetzen					
	Override					

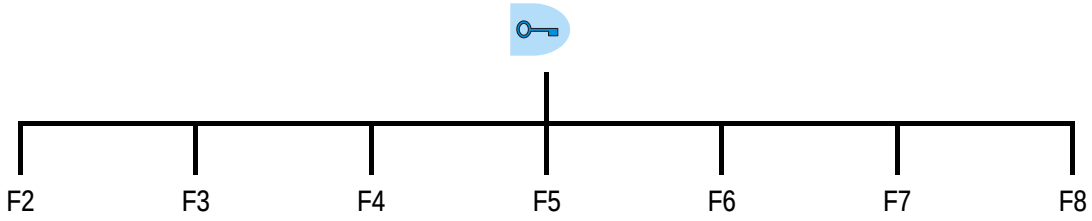
Automatik <Ctrl+F2>

F2 Programm	F3	F4	F5	F6	F7 Teilsystem	F8
Aktivieren						
MDI						
Einzelsatz						
Blocksatz						
Satzfolge						
Eilgang						
Override						

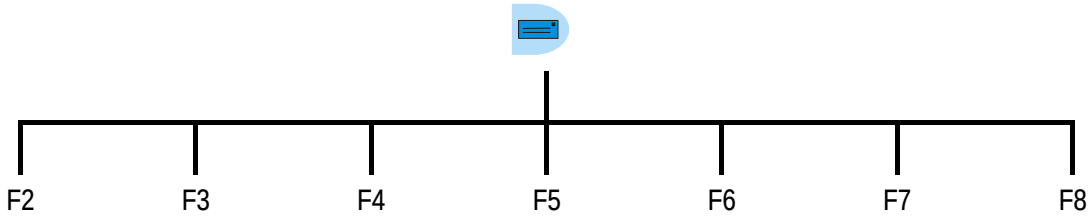
Programmieren <Ctrl+F3>

F2 CNC-Programme	F3 R-Parameter	F4 Werkzeug-daten	F5 Nullpunkt-ver-schiebungen	F6 Koordinaten-systeme	F7	F8
Programm editieren	Wert ändern	Wert ändern	Wert ändern	Wert ändern		
Neues Programm	Editieren					
Programm aktivieren	Neu					
Programm kopieren	Löschen					
Programm löschen						
Programm schützen						
Neues Projekt						
Projekt aktivieren						
Projekt kopieren						
Projekt löschen						
Projekt schützen						
Ansicht						
Editieren						
Neu						
Übertragen						
Aktualisieren						
Löschen						
Verzeichnis						

Zugangsberechtigung <Ctrl+F4>



Fremdsoftware starten <Ctrl+F5>



Start Applikation 1	Start Applikation 2	Start Applikation 3	Start Applikation 4	Start Applikation 5	Start Applikation 6

Inbetriebnahme <Ctrl+F6>
(Softkeyebene 1)

F2 Grundein-stellungen	F3 CNC-System	F4 Antriebs-konfiguration	F5 Boot-Ein-stellungen	F6 OPC-Variablen	F7 Info	F8 Weiter >>
Editieren	Editieren	Editieren	Editieren	Wert ändern		
	Anzeige-modus	Antriebs-parameter	CAN-Ein-stellungen	Editieren		
	Achs-zuordnung	DriveTop		Neu		
				Löschen		

Inbetriebnahme <Ctrl+F6>
(Softkeyebene 2)

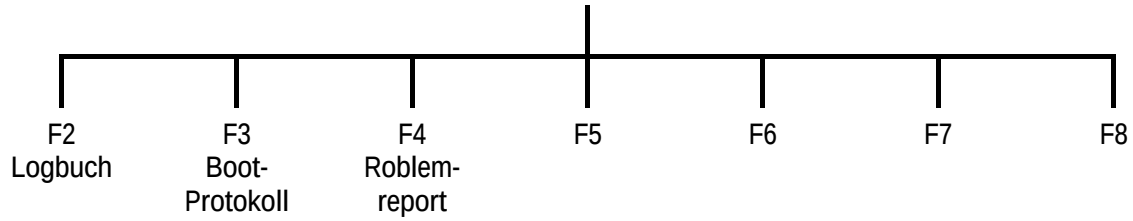
F2 Programmein stellungen	F3 SPS/CNC-Optionen	F4 Protokol-lierung	F5 Uhr stellen	F6 Verbinden	F7 Info	F8 << Zurück
Eigen-schaften	Optionen freigeben	Übernehmen				
Ressource hinzufügen		System-parameter hinzufügen				
Ressource löschen		System-parameter löschen				
Ressource nach oben		Antriebs-parameter hinzufügen				
Ressource nach unten		Antriebs-parameter löschen				

5.6 Aufruf Activ-Error-Buffer und Log-Book

Die Fehlermeldungen im Active-Error-Buffer und Log-Book sind auf jeder Bedienebene über die Tastenkombination <Ctrl+?> aufrufbar.

Die Einträge werden im Kapitel Fehlermeldungen detailiert erklärt.

Fehler <Ctrl+?>



Öffnen						
Speichern						
Drucken						
Aktualisieren						
Löschen						

6 XRIO

In diesem Kapitel befinden sich tifergehende Hinweise zu XRIO. Die Inbetriebnahme von XRIO wird im Kapitel Schnelleinstieg beschrieben. Angaben zur mechanischen und elektrischen Installation befinden sich im Kapitel Installation dieser Betriebsanleitung.

Mit XRIO wird die direkte I/O-Ebene der XCx realisiert. Dazu werden die Buskoppler RIO EC X2 und die Erweiterungsmodule des modularen Systems RIO (**R**emote **I**nput **O**utput) verwendet (siehe auch Abschnitt Erweiterungsmodule aus dem System RIO).

XRIO ist ein Schleicher-spezifischer Bus, der eine effiziente und schnelle Datenübertragung ermöglicht. XRIO ist elektrisch wie Interbus-S aufgebaut. Es ist eine Punkt-zu-Punkt-Verbindung mit je einer Datenleitung pro Übertragungsrichtung, entsprechend RS422 mit 500 kBaud Datenübertragungsrate.

Der Buskoppler besitzt je eine Schnittstelle für ankommende und abgehende Datenleitungen.

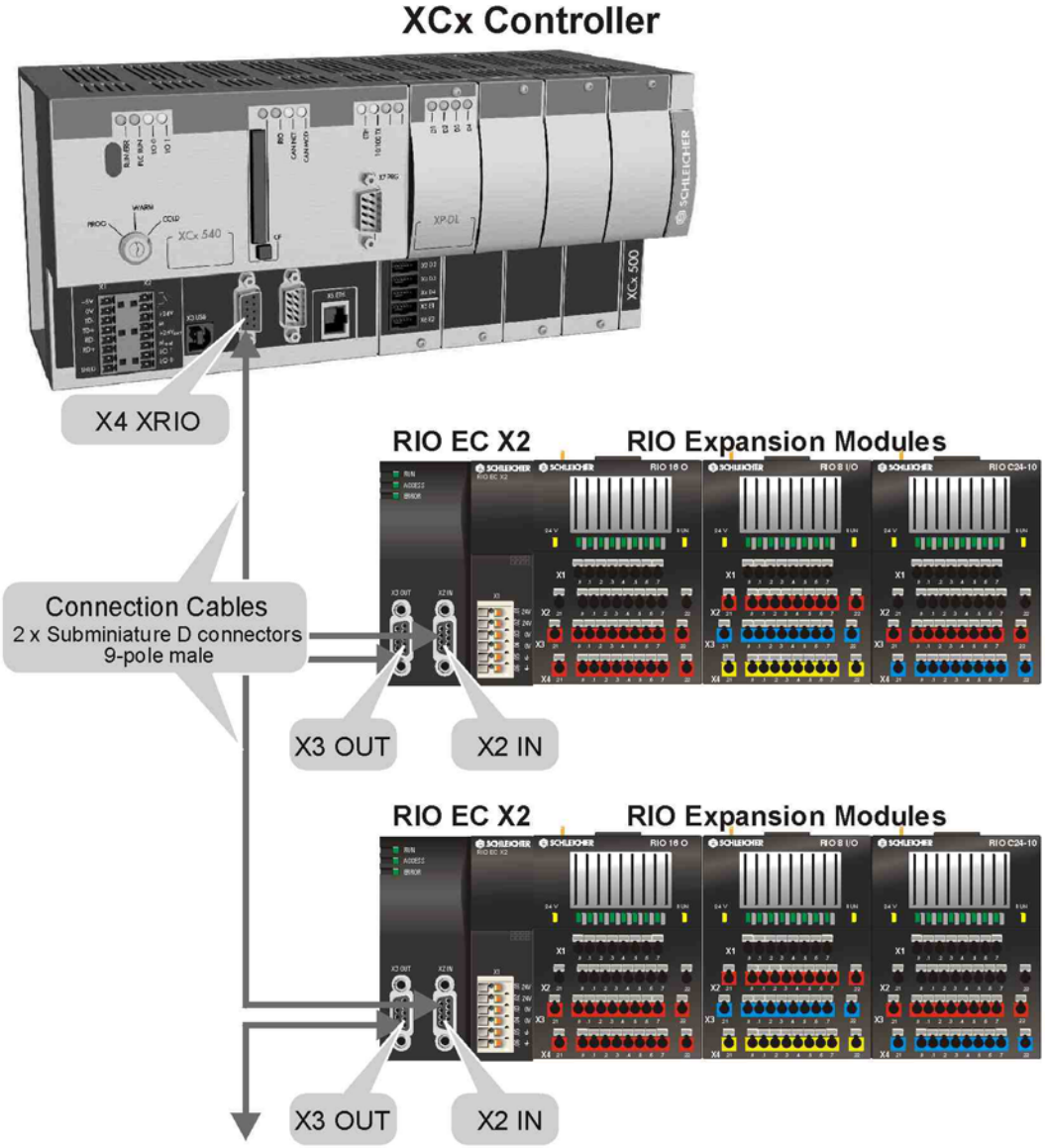


Abbildung 25: SLM-Modul Frontansicht

- Der Buskoppler RIO EC X2 wird mit einem 9poligen D-Sub Kabel mit X2 IN direkt an X4 der XCx angeschlossen.
- Die maximale Buskabellänge beträgt 10 m Zwischen 2 Anschlusspunkten. Es können preiswerte abgeschirmte Buskabel mit handelsüblichen Steckverbindern verwendet werden.
- Weitere Buskoppler werden am Anschluss X3 OUT angeschlossen. Es können insgesamt bis zu 4 Buskoppler RIO EC X2 angeschlossen werden.
- Die LED RUN, ACCES und ERROR des Buskopplers zeigen den Zugriff durch das Betriebssystem der XCx auf den Buskoppler. D.h. wird der Buskoppler ohne XCx betrieben oder ist die SPS in Stop oder in der Anlaufphase leuchtet RUN rot.
  Im normalen Betrieb mit der XCx leuchten RUN und ACCES grün
- Am Buskoppler werden die I/O Module wie bei allen anderen Schleicher Buskopplern rechts angereiht (siehe auch Abschnitt Erweiterungsmodule aus dem System RIO).

Weiter Angaben siehe Betriebsanleitung RIO Buskoppler Artikel Nr.: R4.322.1840.0 (322 156 98).

Die Betriebsanleitungen können kostenlos von unseren Internetseiten *www.schleicher-electronic.com* geladen werden.

6.1 XRIO-Regeln

- Die maximal Anzahl der zu übertragenen Bytes beträgt 127.
- Jedes Buskoppelmodul überträgt zusätzlich 3 oder 4 Header-Bytes.
- Die minimale Task-Zykluszeit wird im wesentlichen durch die Übertragungszeit bestimmt.
- Die maximale Task-Zykluszeit beträgt 80 ms.
  Wird die Maximalzeit überschritten, spricht die Watch-Dog-Funktion der einzelnen RIO-Module an, die LED XRIO der XCx leuchtet rot.
- Der XRIO- Treiber darf je „XRIO- Verbindung“ nur genau einmal aufgerufen werden, d.h. eine Aufteilung des I/O- Bereichs einer XRIO- Verbindung auf verschiedene Tasks ist nicht erlaubt und führt zu Fehlfunktionen.

Erst im vierten Zyklus nach SPS-Start, enthält der Abbildspeicher der Eingänge gültige Daten.

6.2 Erkennung der XRIO-Konfiguration durch das Betriebssystem der XCx

Das Betriebssystem der XCx erkennt automatisch die XRIO-Konfiguration. Bei Differenzen der aktuellen Konfiguration zur vorher abgespeicherten Konfiguration wird der automatische SPS-Start nicht ausgeführt.
Die XRIO-Konfiguration wird auf dem Compact Flash abgespeichert:
beim ersten Einschalten der XCx in der Konfigurationsdatei *\ata0\OS\PLC\sysconf.txt* und in der Systeminformation *\ata0\OS\logfiles\sysinfo.txt*
bei jedem weiteren Einschalten (sowie bei Reset über den Projekt-Kontrolldialog von MULTIPROG und beim Anlegen eines neuen Projektes in MULTIPROG) nur in der Systeminformation *\ata0\OS\logfiles\sysinfo.txt.*(siehe dazu Die Datei sysinfo.txt).

Die Information über die aktuelle XRIO-Konfiguration aus der Datei *\ata0\OS\logfiles\sysinfo.txt* ist wichtig für die Deklaration des I/O-Treibers im SPS-Projekt.

- Wird beim Einschalten der XCx ein Unterschied zwischen der Konfigurationsdatei und der aktuellen XRIO-Konfiguration festgestellt wird der automatische SPS-Start der XCx (Schlüsselschalter der XCx steht auf WARM oder COLD) nicht ausgeführt.
  Der manuelle Start über den Projekt-Kontrolldialog von MULTIPROG ist aber möglich.

Der manuelle Start der XCx mit unkorrekter XRIO-Konfiguration kann zu unvorhersehbaren Zuständen der I/O-Peripherie führen. Es müssen Vorkehrungen zum Schutz von Menschen und Maschinen vor Beschädigungen getroffen werden.

- Bei Bedarf kann die Konfigurationsdatei der XCx aktualisiert werden:
  Dazu auf die Koppelspeicherzelle *cmpSwrdPlcRw_lWrtSysConf* den Wert -1 eintragen. (Nach erfolgreicher Aktualisierung steht dort der Wert 100.)
  Alternativ dazu kann die Konfigurationsdatei gelöscht werden.

  Danach die XCx aus-/einschalten oder den Schlüsselschalter auf die Position PROG stellen und RESET an der XCx auslösen.

6.2.1 Die Datei sysinfo.txt Abschnitt XRIO

Die hier vorgestellte Datei *sysinfo.txt* zeigt eine fiktive XRIO-Konfiguration, die nur der besseren Demonstration so erstellt wurde. Die Datei wurde hier mit Kommentaren versehen, um den Informationsgehalt zu erhöhen.

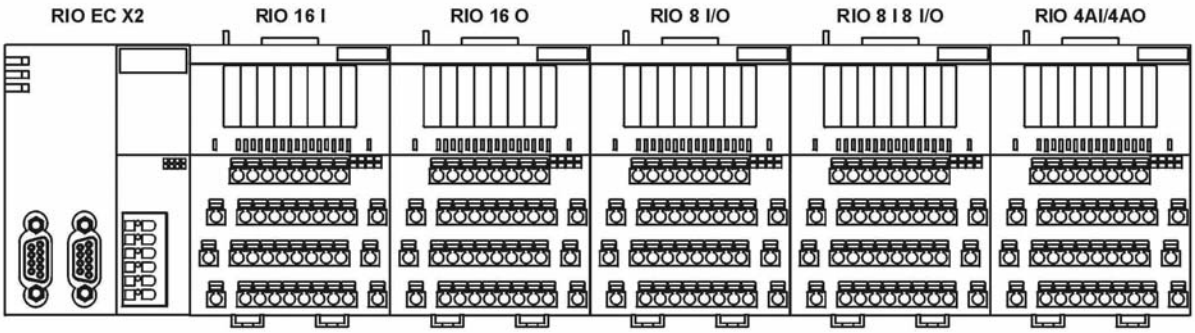


Abbildung 26: Buskoppler RIO EC X2 mit 5 RIO-Erweiterungsmodulen

```
[XRIO]
Cards =1                ; eine XRIO Verbindung
Mod01 =5                ; Anzahl der Module
#IB01 =18               ; Länge des Eingans-Adressbereiches in Byte
%IBS01=0                ; Startadresse
%IBE01=17               ; Endadresse
#QB01 =14               ; Länge des Ausgangsadressbereiches in Byte
%QBS01=0                ; Startadresse
%QBE01=13               ; Endadresse

[XRIO_C01M01]
Name=RIO 16I            ; Modulname
ID  =6                  ; Modul-ID
Type=DIGITAL            ; Modul-Klasse
#IB =4                  ; 4 Eingangsbyte: 2 unentprellt, 2 entprellt
#QB =0                  ; 0 Ausgangsbytes
%IB =0                  ; erstes Eingangsbyte des Moduls, Byteadresse 0
                        ; %IB 0 nicht entprellt
                        ; %IB 1 nicht entprellt
                        ; %IB 2 entprellt
                        ; %IB 3 entprellt

[XRIO_C01M02]
Name=RIO 16O            ; Modulname
ID  =4                  ; Modul- ID
Type=DIGITAL            ; Modul- Klasse
#IB =0                  ; 0 Eingangsbytes
#QB =2                  ; 2 Ausgangsbytes
%QB =0                  ; erstes Ausgangsbyte des Moduls, Byteadresse 0
                        ; %QB 0
                        ; %QB 1
[XRIO_C01M03]
Name=RIO 8I/O           ; Modulname
ID  =2                  ; Modul- ID
Type=DIGITAL            ; Modul- Klasse
#IB =2                  ; 2 Eingangsbytes
#QB =2                  ; 2 Ausgangsbytes
%IB =4                  ; erstes Eingangsbyte des Moduls, Byteadresse 4
                        ; %IB 4 nicht entprellt
                        ; %IB 5 entprellt
%QB =2                  ; erstes Ausgangsbyte des Moduls, Byteadresse 2
                        ; %QB 2 (dummy)
```

```
                        ; %QB 3

[XRIO_C01M04]
Name=RIO 8I 8I/O        ; Modulname
ID  =8                  ; Modul- ID
Type=DIGITAL            ; Modul- Klasse
#IB =4                  ; 2 Eingangsbytes
#QB =2                  ; 2 Eingangsbytes
%IB =6                  ; erstes Eingangsbyte des Moduls, Byteadresse 6
                        ; %IB 6 nicht entprellt
                        ; %IB 7 nicht entprellt
                        ; %IB 8 entprellt
                        ; %IB 9 entprellt
%QB =4                  ; erstes Ausgangsbyte des Moduls, Byteadresse 4
                        ; %QB 4 (dummy!)
                        ; %QB 5

[XRIO_C01M05]
Name=RIO 4AI/4AO 20mA   ; Modulname
ID  =14                 ; Modul- ID
Type=ANALOG             ; Modul- Klasse
#IW =4                  ; 4 Eingangsworte
#QW =4                  ; 4 Ausgangsworte
%IW =10                 ; erstes Eingangswort des Moduls, Adr. 10
                        ; %IW 10
                        ; %IW 12
                        ; %IW 14
                        ; %IW 16
%QW =6                  ; %QW 6
                        ; %QW 8
                        ; %QW 10
                        ; %QW 12
```

6.3 Adressierung der I/O-Module

- Bei der Adressierung der Eingänge liegen die entprellten und die unentprellten Eingänge immer direkt nacheinander
  Die Entprellung der Eingänge auf 2 ms Eingangssignalverzögerung wird per Software im Buskoppler realisiert. (Die Eingänge auf den I/O Modulen sind zusätzlich mit 01,ms entprellt).
- Bei der Adressierung der Ausgänge belegen auch Module mit nur 1 Byte Ausgänge immer 1 Wort.
  Damit wird für folgende Module eine gerade Anfangsadresse sichergestellt.

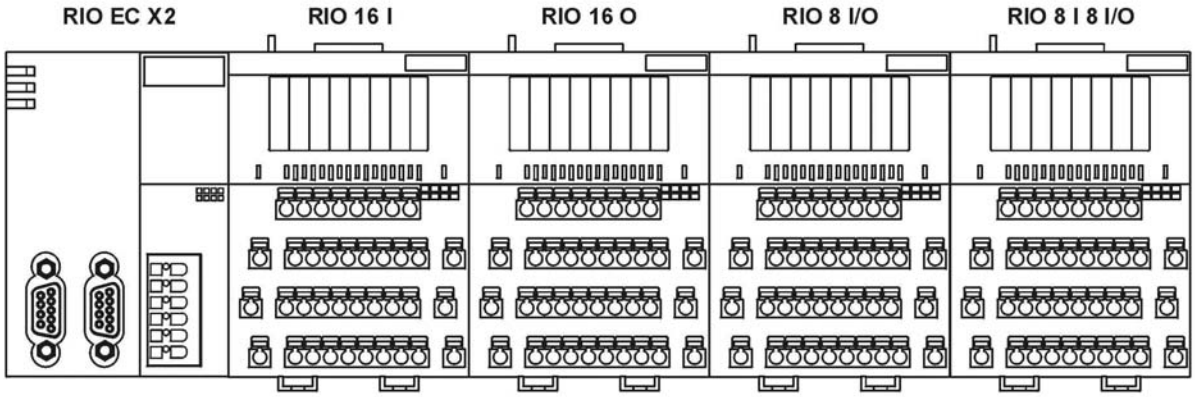


Abbildung 27: Buskoppler RIO EC X2 mit 4 RIO-Erweiterungsmodulen

Modul	RIO 16I	RIO 8I/8I/O	RIO 8I/O	RIO 16O
Eingänge nicht entprellt	IB0, IB1	IB4, IB5	IB8	-
Eingänge entprellt	IB2, IB3	IB6, IB7	IB9	-
Ausgänge	-	QB1(+QB0)	QB3(+QB2)	QB4, QB5

6.4 XRIO Flags im Koppelspeicher Gültigkeit der Prozessdaten

Mit Hilfe von System-Flags auf dem Koppelspeicher

plcMem.plcSect.flgXRIO.bXRio[] *ARRAY[1..4] OF BOOL*
oder
cmpSflgXRIO_bXRio[] *ARRAY[1..4] OF BOOL*

kann die Gültigkeit der XRIO Prozessdaten für die entsprechende XRIO-Verbindung (1..4) im SPS-Programm abgefragt werden:

TRUE = Eingangsabbild gültig (SPS-RUN, keine Übertragungsfehler)
FALSE = Eingangsabbild ungültig (SPS-STOP oder keine Betriebsbereitschaft bzw. Übertragungsfehler)

6.5 Der XRIO-Treiber

Der I/O-Treiber XRIO übernimmt die Ansteuerung der über Steckverbinder X4 XRIO angeschlossenen RIO-Module, stellt entsprechende Ein- und Ausgangsabbilder bereit und übernimmt deren Refresh.

Die Übertragungszeiten auf dem XRIO-Bus werden hauptsächlich durch die Anzahl der zu übertragenden Bytes bestimmt. Als Richtwert dient eine Transmit- Time von 22µs je Byte.

Berechnungsvorschrift:
Zeit = 22µs (Anzahl I/O- Bytes + 3 Steuer- Bytes)

Achtung!

Der XRIO-Treiber darf je „XRIO-Verbindung“ nur genau einmal aufgerufen werden, d.h. eine Aufteilung des I/O-Bereichs einer XRIO-Verbindung auf verschiedene Tasks ist nicht erlaubt und führt zu Fehlfunktionen.

Der XRIO-Treiber muss in der Task aufgerufen werden in der die I/O Verarbeitet werden, damit konsistente Daten über die Task-Laufzeit verfügbar sind. (Zu Beginn der Task wird das Eingangsabbild gelesen, am Ende der Task wird das Ausgangsabbild geschrieben.)

Das Taskintervall kann 1 bis 80 ms betragen. Bei Taskintervall > 80 ms wird der Watchdog der RIO-Module ausgelöst. Die LED XRIO der XCx leuchtet rot.

Die nachfolgende Tabelle gibt eine Übersicht über die gegenwärtig durch den XRIO-Treiber unterstützten User-Parameter:

XRIO User Parameter:		
DRIVER_PARAMETER	Code	Inhalt
DRIVER_PAR1	0 = default	• Prozessdaten (Ausgänge) werden bei SPS- STOP automatisch auf Null gesetzt, • Es wird ein I/O- Refresh je Task- Zyklus durchgeführt.
DRIVER_PAR1	Bit 0	• nicht benutzt
DRIVER_PAR1	Bit 1	• nicht benutzt
DRIVER_PAR1	Bit 2	• ignoriere Abweichungen zwischen der konfigurierten und der realen Anzahl I/O- Bytes
DRIVER_PAR1	Bit 3	• zusätzlicher Prozessdaten- Refresh (im IN- Treiber Teil zu konfigurieren) **Achtung:** Diese Funktion ist z.Z. nicht freigegeben!
DRIVER_PAR1	Bit 6	• 2 Stop-Bits wenn gesetzt (im IN- Treiber Teil zu konfigurieren)
DRIVER_PAR1	Bit 7	• erzwinge XRIO Normalmodus (500 kBit/s) (im IN- Treiber Teil zu konfigurieren)
DRIVER_PAR2	Bit 0	• nicht benutzt
DRIVER_PAR2	Bit 1:	• verhindert Task- Wechsel während des Prozessdaten-Updates
DRIVER_PAR2	Bit 2	• nicht benutzt

6.6 Berechnung der "Estimated Transmission Time"

Die Übertragungszeiten auf dem XRIO- Bus werden durch die Anzahl der zu übertragenden Bytes bestimmt. Als Richtwert dient eine Transmit- Time von 22µs je Byte bei einer Übertragungsrate 500 kByte/s (kbaud).

4 Mbaud und 1 stop bit:
((∑ I/O-Bytes je Modul + 4) x 2,75 µs/Byte + (no. of ECs)x 4 µs/EC + (no. of IO modules) x 1µs/module + 10 µs

4 Mbaud und 2 stop bits
((∑ I/O-Bytes je Modul + 4) x 3,00 µs/Byte + (no. of ECs)x 4 µs/EC + (no. of IO modules) x 1µs/module + 10 µs

500 kbaud und 1 stop bit
((∑ I/O-Bytes je Modul + 3) x 22 µs/Byte

500 kbaud und 2 stop bits
((∑ I/O-Bytes je Modul + 3) x 24 µs/Byte

Beispiel (ECX2, 6 dig.IO, 2 P24, 1 A10):

4 Mbaud und 1stop bit

(61 + 4) x 2,75 µs + 4 x 4 µs + 9 x 1µs + 10 µs = 0,214 ms --> gemessen 0,20 ms

4 Mbaud und 2 stop bits

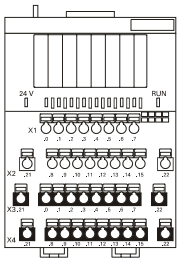
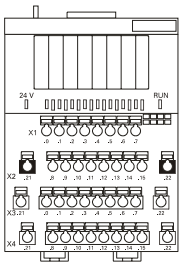
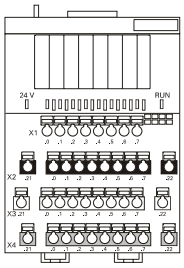
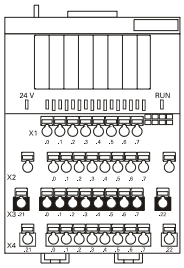
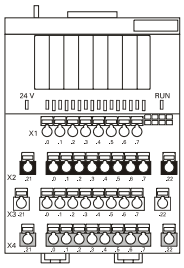
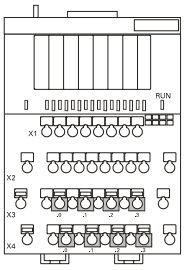
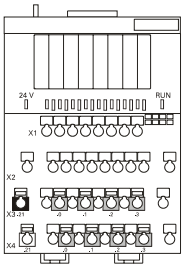
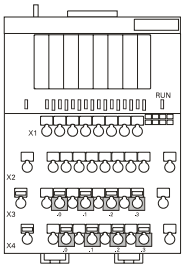
(61 + 4) x 3,00 µs + 4 x 4 µs + 9 x 1µs + 10 µs = 0,230 ms --> gemessen 0,21 ms

500 kbaud und 1stop bit
(61 + 3) x 22,00 µs = 1,410ms --> gemessen 1,39 ms

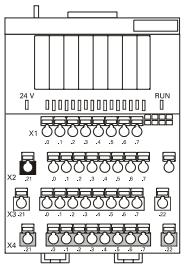
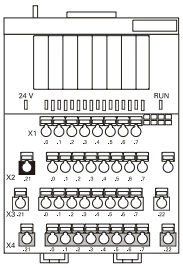
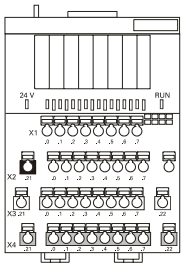
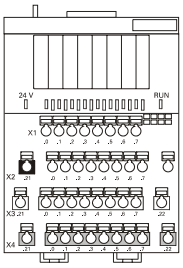
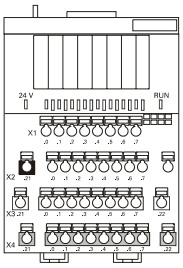
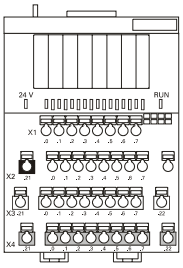
500 kbaud und 2 stop bits
(61 + 3) x 24,00 µs = 1,540 ms --> gemessen 1,51ms

6.7 Erweiterungsmodule aus dem System RIO

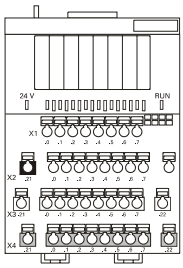
6.7.1 Übersicht

Digitalmodule			
	RIO 16 I 16 Eingänge DC 24 V Zweileiter-Anschlußtechnik		RIO 16 O 16 Ausgänge 1A Zweileiter-Anschlußtechnik
	RIO 8 I/O 8 Kombi-I/O Alle Kombi I/O als Eingänge DC 24 V oder Ausgänge 1A einzeln nutzbar. Vierleiter-Anschlußtechnik		RIO 8 I 8 I/O 8 Eingänge DC 24 V 8 Kombi-I/O Alle Kombi I/O als Eingänge DC 24 V oder Ausgänge 1A einzeln nutzbar. Zweileiter-Anschlußtechnik
	RIO 8 O 2A 8 Ausgänge 2 A Vierleiter-Anschlußtechnik		RIO 4 I 230 VAC 4 Eingänge AC 230 V
	RIO 4 O R 4 Ausgänge Relais		RIO 4 I 120 VAC 4 Eingänge AC 120 V

Analogmodule

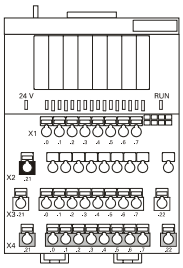
Spannung ±10 V		Strom 20mA	
	RIO 4AI ±10 V 4 Analogeingänge Auflösung 12 Bit		RIO 4AI 20mA 4 Analogeingänge Auflösung 12 Bit
	RIO 4AI/4AO ±10 V 4 Analogeingänge 4 Analogausgänge Auflösung 12 Bit		RIO 4AI/4AO 20mA 4 Analogeingänge 4 Analogausgänge Auflösung 12 Bit
		Strom 4...20mA	
			RIO 4AI 4-20mA 4 Analogeingänge Auflösung 12 Bit
			RIO 4AI/4AO 4-20mA 4 Analogeingänge 4 Analogausgänge Auflösung 12 Bit

Temperaturmodule



RIO T10-10

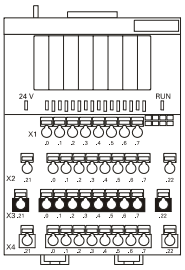
4 Eingänge für Temperaturmessung mit Pt100/Pt1000



RIO T20-10

4 Eingänge für Temperaturmessung mit Thermoelementen

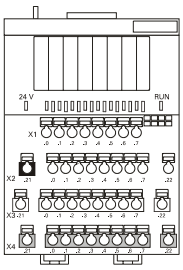
Zählermodul



RIO C24-10

4 Zähler 16 Bit oder
2 Zähler 32 Bit

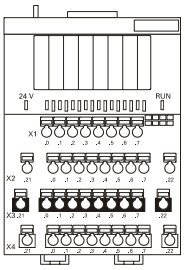
Achsinterfacemodul



RIO A10-10

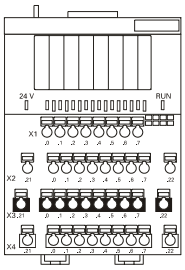
Interface für eine Achse

I Positioniermodule



RIO P05-10

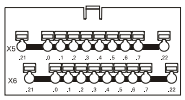
Positionierung von zwei Achsen
(5 V Zählereingänge)



RIO P24-10

Positionierung von zwei Achsen
(24 V Zählereingänge)

Potentialverteiler (Klemmenerweiterung)



RIO KE 16

2 Verteiler mit je
10 Klemmstellen

Nur für Module mit Aufnahmelaschen geeignet.

Weiter Angaben siehe Betriebsanleitung RIO Erweiterungsmodule Artikel Nr.: R4.322.1720.0 (322 154 14).
Aktuelle Bestellangaben und Artikelnummern siehe unsere Internetseiten *www.schleicher-electronic.com*

6.7.2 Tabellen der Modultypen und Modulklassen

Modultypen

Module- ID	HID	Name	Module Type	Input Bytes	Output Bytes	I/O- Bytes effektive
0x0001	0x02	RIO 8I/O	DIGITAL	1+1	2	3
0x0002	0x06	RIO 16I	DIGITAL	2+2	0	3
0x0003	0x04	RIO 16O	DIGITAL	0	2	3
0x0004	0x08	RIO 8I 8I/O	DIGITAL	2+2	2	3
0x0005	0x0A	RIO 4AI/4AO ±10V	ANALOG	8	8	9
0x0006	0x0C	RIO 4AI ±10V	ANALOG	8	0	9
0x0007	0x0E	RIO 4AI/4AO 20mA	ANALOG	8	8	9
0x0008	0x10	RIO 4AI 20mA	ANALOG	8	0	9
0x0009	0x12	RIO RNO	DIGITAL	2+2	0	3
0x000A	0x42	RIO C24-10	COUNTER	10+2	10	11
0x000B	0x42	RIO C24-10	COUNTER	10+2	10	11
0x000C	0x62	RIO P24-10	COUNTER	10+2	10	11
0x000D	0x62	RIO P24-10	COUNTER	10+2	10	11
0x000E	0x16	RIO T10-10	ANALOG	8	0	9
0x000F	0x1E	RIO 4AI 0..10V	ANALOG	8	0	9
0x0010	0x18	RIO 4AI/4AO 4..20	ANALOG	8	8	9
0x0011	0x1A	RIO 4AI 4..20mA	ANALOG	8	0	9
0x0012	0x1C	RIO 4AI/4AO 0..10	ANALOG	8	8	9
0x0013	0x24	RIO 4OR	DIGITAL	0	2	3
0x0014	0x20	RIO T20-10	ANALOG	8	0	9
0x0015	0x00	RIO P05-10	COUNTER	10+2	10	11
0x0016	0x00	RIO P05-10	COUNTER	10+2	10	11
0x0017	0x44	RIO 8O 2Amp.	DIGITAL	0	2	3
0x0018	0x26	RIO 4I 115VAC	DIGITAL	2	0	3
0x0019	0x46	RIO 4I 230VAC	DIGITAL	2	0	3
0x001A	0x00	RIO 8I TTL	DIGITAL	2	0	3
0x001B	0x00	RIO 8O neg	DIGITAL	0	2	3
0x001C	0x70	RIO A 10-10	ANALOG	8+4	8	9
0x001C	0x70	ECX2	ANALOG	2	2	3

Input Bytes sind die Summe von entprellten und nicht entprellten Eingangs-Bytes.

I/O-Bytes effektive sind die Summe aller zuschiebenden resp. zu übertragenden Bytes.

Modulklassen

Modul- Typ	Code
ANALOG	0
DIGITAL	1
COUNTER	2
NOMOD	3

6.8 Erweiterungsmodul XRIO

Das Modul kann für die Erweiterung der XCx 300 und XCx 540 eingesetzt werden. Es besitzt folgende Eigenschaften:

- Eine XRIO-Busschnittstelle
- entsprechend RS422
- 500 kBaud Datenübertragungsrate

6.8.1 Anzeigen und Anschlüsse

Anzeigen			
LED	Farbe		
L1			nicht benutzt
L2	grün	ein	Operational
	grün	blinkend	Preoperational
	rot	ein	Busfehler
	gelb	ein	Frame-Fehler
L3			nicht benutzt
L4			nicht benutzt
Anschluss X31			
Pin	Bezeichnung		
1	RD+		Empfangsdaten plus
2	TD+		Sendedaten plus
3	0 V		Versorgungsspannung 0 V
4	nc		nicht angeschlossen
5	+5 V		Versorgungsspannung
6	RD-		Empfangsdaten minus
7	TD-		Sendedaten minus
8	nc		nicht angeschlossen
9	nc		nicht angeschlossen

6.8.2 Technische Daten

Grunddaten	
Schnittstelle	XRIO
Datenübertragungsrate	500 kBaud
Anschlusstechnik	D-Sub, 9-polig, Buchse

7 Feldbus CANopen

7.1 Grundlagen

Allgemeines

CANopen basiert auf dem CAN Application Layer für industrielle Anwendungen CAL. Das CANopen- Kommunikationsprofil CiA DS-301 spezifiziert die Mechanismen zur Konfiguration und Kommunikation zwischen Geräten in Echtzeitumgebungen.
CANopen benutzt die Datenübertragungschicht nach ISO 11898 und CAN 2.0 A+B:

- Beschreibung der Gerätedetails über ein EDS (Electronic Data Sheet).
- Objektorientierte Kommunikation mit PDOs und SDOs.
- Übertragung von Echtzeitdaten mit ´purem´ CAN als PDO (Process Data Object). PDOs können von allen Teilnehmern ereignisgesteuert oder synchronisiert gesendet werden.
- Komplexe Daten oder niederpriore Dienste werden über SDOs (Service Data Objects) übertragen bzw. abgewickelt.
- CANopen Configuration-Manager (oder -Master) übernehmen z.B. das Netzwerkmanagement bei Netzanlauf, sind aber nicht zur Kommunikation der Slaves untereinander notwendig.
- Bis zu theoretisch 127 Teilnehmer sind an einem Bus möglich. Praktisch wird die Anzahl der Bus-Teilnehmer durch die jeweilige Bus-Topologie – insbesondere aber durch die Art der eingesetzten CAN-Transceiver Bausteine - begrenzt.
  (bei Schleicher CAN-Geräten auf z.Z. 64)

CANopen mit der XCx

Die SPS der XCx kann auf Netzvariablen sowie Parametrier- und Diagnosefunktion über den CANIO-Treiber zugreifen. Die Installation des Treibers und die Inbetriebnahme des Netzes wird im Kapitel Inbetriebnahme der Betriebsanleitung XCx beschrieben.

Daneben ist es möglich auf direkte Netzvariablen zuzugreifen, siehe dazu Zugriff auf direkte Netzvariablen.

Die Parametrier- und Diagnosefunktionen betreffen die Hard- und Software der CANopen-Schnittstelle der XCx, die hier als CAN-Prozessor umschrieben werden. Siehe dazu Zugriff auf Parametrier- und Diagnosefunktionen sowie Die Parametrier- und Diagnosefunktionen.

7.1.1 Process- Data- Objects (PDO)

Als in PDOs "mappfähige“ Objekte ("mappable objects") stehen der Steuereinheit Byte-, Word,- Doppelword-Netzvariablen zur Verfügung. Netzvariablen werden im SPS- Programm über ein Prozeßdatenabbild mit Hilfe des CANIO- Treibers angesprochen.
Außerdem werden sogenannte direkte Netzvariablen bereitgestellt, auf die ohne speziellen Treiber und ohne Prozeßabbild (!) zugegriffen werden kann.
Zur Projektierung der Kommunikationsbeziehungen im CANopen-Netzwerk, ist der Einsatz eines CANopen Konfigurators, wie z.B. ProCANopen, zu empfehlen. Aus den mitgelieferten EDS- Dateien können hierzu alle Informationen zu Anzahl, Objekt- Index, Datentyp usw. entnommen werden.
Die XCx stellt als Gerät gemäß dem CiA DSP 302 (Framework for Programmable CANopen Devices) bzw. DSP 405 (Device Profile for IEC 1131 Programmable Devices) kein Default- Mapping für PDOs zur Verfügung.

Eine Aufstellung der mappfähigen Objekte ist unter Parametrier- und Diagnosedaten und Netzvariablen zu finden.

7.1.2 Service-Data-Objects (SDO)

Für das Lesen/Schreiben von bis zu 128 Byte konsistenten Nutzdaten stehen spezielle "SDO- Objekte" zur Verfügung.

Zum SPS- seitigen Zugriff auf diese Objekte, müssen vorher im SPS-Code entsprechende Arrays deklariert sein.

Objektindex	Richtung	Adressbereich	Azahl Bytes
0x4100	Rx	je nach IO Konfiguration	128
0x4110	Rx		128
0x4120	Rx		32
0x4130	Rx		64
0x4200	Tx		128
0x4210	Tx		128
0x4220	Tx		32
0x4230	Tx		64

Rx bedeutet von extern per "Write Request" beschreibbar
Tx bedeutet von extern per "Read Request" lesbar

7.1.3 Nodeguarding

Über das Nodeguarding kann ein Guarding- Master den Ausfall eines Slaves erkennen. Dazu sendet er zyklisch Nachrichten auf den Guarding- Identifier (100Eh) des Slaves. Dieser antwortet mit einer Guarding- Nachricht, die u.a. ein Toggle- Bit enthält.

Wird eine Schleicher- SPS als Guarding- Master projektiert, können etwaige Guarding- Fehler im SPS- Programm über einen Funktionsbaustein ausgewertet werden.

7.1.4 Lifeguarding

Während das Nodeguarding vom Guarding- Master durchgeführt wird, um den Ausfall eines Teilnehmers zu erkennen, benutzt der überwachte Teilnehmer diese Guarding- Telegramme, um seinerseits den Ausfall des Masters zu erkennen. Diese Überwachungsfunktion eines Teilnehmers wird Lifeguarding genannt.

Eine Kabelbrucherkennung und damit eine Zwangsabschaltung der Ausgänge kann bei CANopen nur bei aktiviertem Node- und Lifeguarding erfolgen !

Zur Aktivierung des Lifeguardings muß der NMT-Manager die Objekte Guard- Time (100Ch) und Life-Time- Factor (100Dh) beschreiben.

Falls die Überwachungszeit

*Life-Time = Life-Time-Factor * Guard-Time [ms]*

abläuft, ohne daß ein Guarding-Telegramm beim Teilnehmer eintrifft wird ein Guarding-Fehler festgestellt:

- Ein RIO CANopen Buskoppler oder Kompaktmodul schaltet die LED NET rot blinkend und die Ausgänge ab.
- Eine Schleicher- SPS mit CANopen meldet den Guarding- Fehler mittels Funktionsbaustein an das SPS- Programm.

Ist eines der beiden o.g. Objekte gleich 0, wird kein Lifeguarding und damit auch keine Kabelbrucherkennung durchgeführt !

7.2 CANopen spezifische SPS- Adressen

7.2.1 Parametrier- und Diagnosedaten

Diagnosedaten		
CANopen Objekt Index	SPS Adresse	Inhalt
0x3100 Diagnostic Input	je nach IO Konfiguration	Eingangsdaten
0x3200 Diagnostic Output	je nach IO Konfiguration	Ausgangsdaten

7.2.2 Netzvariablen

Netzvariablen		
CANopen Objekt Index	SPS Adresse	Inhalt
0x54c0 IB Input Byte 0x5580 IW Input Word 0x5680 ID Input Dword	je nach IO Konfiguration	direkte NVs Eingangsdaten
0x5040 QB Output Byte 0x5100 QW Output Word 0x5200 QD Output DWord	je nach IO Konfiguration	direkte NVs Ausgangsdaten
0xa4c0 IB Input Byte 0xa580 IW Input Word 0xa680 ID Input Dword	je nach IO Konfiguration	allgemeine NVs Eingangsdaten
0xa040 QB Output Byte 0xa100 QW Output Word 0xa200 QD Output DWord	je nach IO Konfiguration	allgemeine NVs Ausgangsdaten

Alle Netzvariablen können auch per "Service- Daten- Object" (SDO) angesprochen werden.

Das Mapping auf Prozess- Daten- Objekte (PDO) kann ebenfalls per SDO erfolgen. Zur Vereinfachung, Fehlervermeidung und Diagnose ist jedoch dringend der Einsatz des CANopen Konfigurators ProCANopen zu empfehlen.

Alle Netzvariablen- Ausgänge (QW) werden bei SPS- Stop auf 0 gesetzt.

Netzvariablen werden mit Hilfe des CANIO- Treibers wie ein Prozeßabbild eingelesen und ausgegeben.

Direkte Netzvariablen werden ohne Prozeßabbild eingelesen und ausgegeben.

7.2.3 Zugriff auf Netzvariablen und I/O-Konfiguration

Netzvariablen sowie die Funktionscodes und Parameter der Parametrier- und Diagnosefunktion werden mit Hilfe der I/O-Konfiguration auf SPS-Adressen abgebildet und in der Variablen-Deklaration mit symbolischen Namen versehen.
Über Driver-Parameter ist eine nutzerspezifische Anpassung des CANIO- Treibers möglich:

CANIO User Parameter:		
DRIVER_PARAMETER	Code	Inhalt
DRIVER_PAR1	0 (default)	• Zugriff auf Diagnosedaten deaktiviert • Prozeßdaten (Ausgänge) werden bei SPS-STOP automatisch auf Null gesetzt
DRIVER_PAR1	Bit 0 = 1	• aktiviere Zugriff auf Diagnosedaten
DRIVER_PAR1	Bit 1 = 1	• Prozeßdaten (Ausgänge) werden bei SPS-STOP nicht automatisch auf Null gesetzt
DRIVER_PAR2	0 (default)	• Update des Prozeßabbildes ohne Benutzung von Semaphoren • gewährleistet 32- Bit Datenkonsistenz, keine Konsistenz über das komplette Abbild ! • schnell mit geringen Overhead • für reine I/O- Zugriffe geeignet
DRIVER_PAR2	Bit 0 = 1	• Update des Prozeßabbildes unter Benutzung von Semaphoren • Datenkonsistenz über den gesamten Netzvariablen- Bereich • langsam und Overhead behaftet, mit Wartezeit für die Freigabe der Semaphoren • verwendet den Timeout- Wert aus DRIVER_PAR3 • z.B. für überlagerte Protokolle
DRIVER_PAR2	Bit 1 = 1	• verhindert Taskwechsel während des Prozeßdaten- Updates
DRIVER_PAR2	Bit 2 = 1	• Double- Buffer Mode, Update des Prozeßabbildes unter Benutzung von Semaphoren • Datenkonsistenz über den gesamten Netzvariablen- Bereich • schnell aber mit Overhead behaftet, ohne Wartezeit für die Freigabe der Semaphoren • z.B. für den Ausstauch von konsistenten Datenstrukturen
DRIVER_PAR3	Wert	• Timeout in µs (bei Verwendung von Semaphoren) • default max. 500 µs

Beispiel

```
(* CAN1> - Don't remove this label*)
PROGRAM netin1 : INPUT
(
   VAR_ADR     := 1000, (* CAN card / network 1 *)
   END_VAR_ADR := 1255,
   DEVICE      := DRIVER,
   DRIVER_NAME := 'CANIO'
 (* use defaults *)
);
PROGRAM netout1 : OUTPUT
(
   VAR_ADR     := 1000,
   END_VAR_ADR := 1255,
   DEVICE      := DRIVER,
   DRIVER_NAME := 'CANIO'
 (* use defaults *)
);
(* diagnostic interface *)
PROGRAM netin1d : INPUT
(
   VAR_ADR     := 1256, (* CAN card / network 1 *)
   END_VAR_ADR := 1259,
   DEVICE      := DRIVER,
   DRIVER_NAME := 'CANIO',
   DRIVER_PAR1 := 1
);
PROGRAM netout1d : OUTPUT
(
   VAR_ADR     := 1256, (* CAN card / network 1 *)
   END_VAR_ADR := 1259,
   DEVICE      := DRIVER,
   DRIVER_NAME := 'CANIO',
   DRIVER_PAR1 := 1
);
(* <CAN1 - Don't remove this label *)
```

Deklaration im SPS- Programm:

```
VAR_GLOBAL
mNVInput1        AT %IW 1000 : INT;
mNVOutput1       AT %QW 1000 : INT;
mDiag1Input1     AT %IW 1256 : UINT;
mDiag1Input2     AT %IW 1258 : UINT;
mDiag1Output1    AT %QW 1256 : UINT;
mDiag1Output2    AT %QW 1258 : UINT;
VAR_END
```

Zu beachten ist ferner, daß DRIVER_PAR1=1 gesetzt werden muß, damit der CANIO- Treiber auf die Parametrier- und Diagnosefunktionen zugreift. Hierbei werden Abbilder für die Speicherzellen der Parametrier- und Diagnosefunktionen angelegt.

Wird der Double-Buffer Mode an einer Stelle der I/O- Konfiguration aktiviert, so gilt diese Einstellung für alle Treiber-Zugriffe.

7.2.4 Zugriff auf direkte Netzvariablen

Direkte Netzvariablen werden im SPS- Programm wie folgt deklariert:

```
VAR_GLOBAL
(* CAN card / network 1 *)
mNV1Input  AT %MW 3.1010000 : UDINT;
mNV1Output AT %MW 3.1010512 : UDINT;
(* CAN card / network 2 *)
mNV2Input  AT %MW 3.1020000 : UDINT;
mNV2Output AT %MW 3.1020512 : UDINT;
VAR_END
```

Analog kann auch direkt auf die entsprechenden Speicherzellen der Parametrier- und Diagnosefunktionen zugegriffen werden, d.h. ohne Zuhilfenahme des CANIO-Treibers.

```
VAR_GLOBAL
mDiag1Input1  AT %MW 3.1019992 : UINT;
mDiag1Input2  AT %MW 3.1019994 : UINT;
mDiag1Output1 AT %MW 3.1019996 : UINT;
mDiag1Output2 AT %MW 3.1019998 : UINT;
VAR_END
```

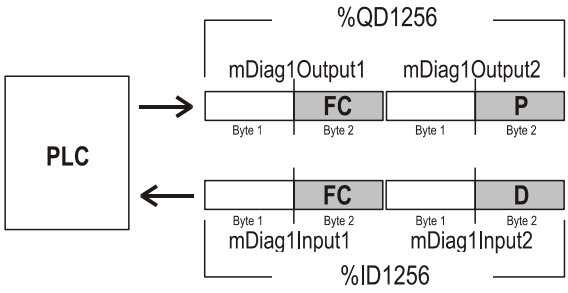
Hierbei werden keine Abbilder für die Speicherzellen der Parametrier- und Diagnosefunktionen angelegt.

7.2.5 Zugriff auf Parametrier- und Diagnosefunktionen

Die SPS kann im CAN-Prozessor Parametrier- und Diagnosefunktionen auslösen.

Die SPS fordert eine Funktion an, indem in der Variablen mDiag1Output1 der gewünschte Funktionscode eingetragen wird. Soll eine Funktion mit Parameter übergeben werden, muß zuerst in mDiag1Output2 der Parameter und danach in mDiag1Output1 der Funktionscode eingetragen werden.

Nach der Abarbeitung der Funktion wird der Funktionscode immer im mDiag1Input1 angezeigt, im mDiag1Input2 können dann die Diagnosedaten gelesen werden.



FC = **F**unktions**C**ode
P = **P**arameter
D = **D**aten

Möglich ist auch die direkte Zuweisung in 32 Bit- Variablen (z.B. %QD1256 und %ID1256).

7.3 SDO Funktionsbausteine

Unterstützte Funktionen

SDO- Write
SDO- Read
Read Error- and Emergency- Entries

Detailliertere Beschreibungen sind im CiA Draft-Standard 301 bzw. den jeweiligen Profilen (z.B. CiA DSP 405, 401 usw.) enthalten.

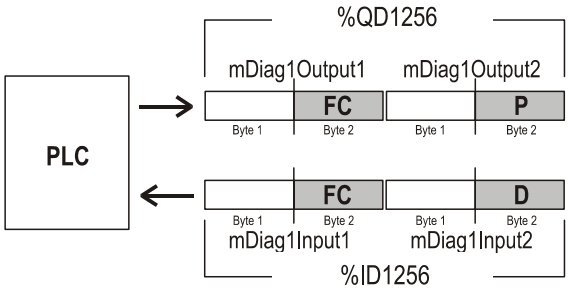
7.4 Die Parametrier- und Diagnosefunktionen

Die SPS kann im CAN-Prozessor* Parametrier- und Diagnosefunktionen durch Übertragung eines Codes auslösen.

Funktion	
0	keine Funktion
1,17	Knotennummer (NodeID) auslesen und einstellen, Wertebereich: 1..127 Reboot erforderlich!
2	Baudrate auslesen und einstellen, Wertebereich: 1..8 Reboot erforderlich!
3 ... 5	Reserviert
6	CAN Statuscode auslesen
7	Fehlerstatus auslesen
8	Firmware- Version auslesen
9..15	Reserviert
16	effektive Laufzeit des CAN Stacks (in 1/100 ms) auslesen
17..19	Reserviert
20	Hochlaufverzögerung auslesen und einstellen, Wertebereich: 1..60 s
21...101	Reserviert
102	Zykluszeit des CAN- Stacks auslesen und einstellen, Wertebereich: 1..10 ms Reboot erforderlich!
103...254	Reserviert
255	Rücksetzen

Die SPS fordert eine Funktion an, indem in der Variablen mDiag1Output1 der gewünschte Funktionscode eingetragen wird. Soll eine Funktion mit Parameter übergeben werden, muß zuerst in mDiag1Output2 der Parameter und danach in mDiag1Output1 der Funktionscode eingetragen werden.

Nach der Abarbeitung der Funktion wird der Funktionscode immer im mDiag1Input1 angezeigt, im mDiag1Input2 können dann die Diagnosedaten gelesen werden.



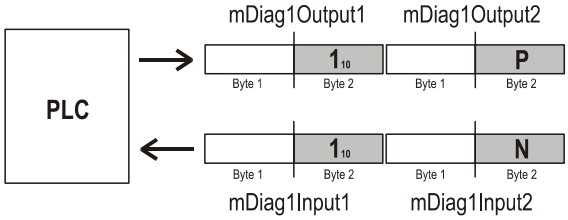
FC = **F**unktions**C**ode
P = **P**arameter
D = **D**aten

Möglich ist auch die direkte Zuweisung in 32 Bit- Variablen (z.B. %QD1256 und %ID1256).

*Als CAN-Prozessor wird hier die Hard- und Software der CANopen-Schnittstelle umschrieben.

7.4.1 Funktion 1 CANopen Knotennummer einstellen und auslesen

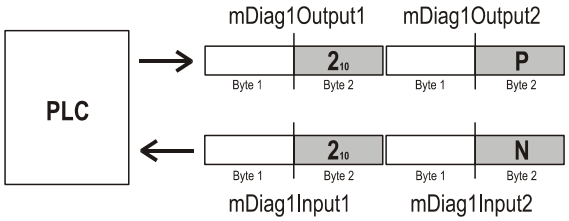
P Parameter	
0	Knotennumer auslesen
1 . . 127	Knotennummer 1 einstellen Knotennummer 127 einstellen



N aktuell eingestellte Knotennummer	
1 . . 27	Knotennummer 1 Knotennummer 127

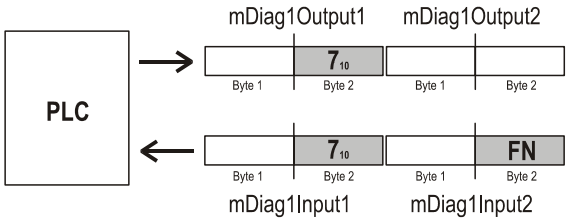
7.4.2 Funktion 2 Datenübertragungsrate auslesen und einstellen

P Parameter	
0	Auslesen der aktuell eingestellten Datenübertragungsrate
1	10 kBaud einstellen
2	20 kBaud einstellen
3	50 kBaud einstellen
4	125 kBaud einstellen
5	250 kBaud einstellen
6	500 kBaud einstellen
7	800 kBaud einstellen
8	1000 kBaud einstellen



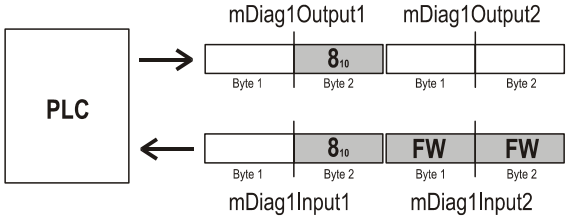
N aktuell eingestellte Datenübertragungsrate in kBaud	
1	10
2	20
3	50
4	125
5	250
6	500
7	800
8	1000

7.4.3 Funktion 7 CANopen- Fehlernummer auslesen



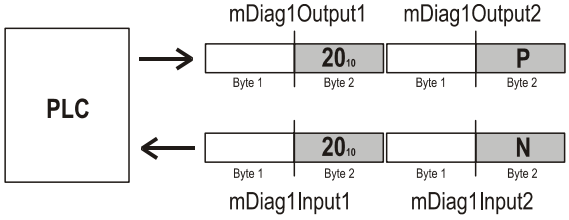
FN Fehlernummer siehe dazu Fehlermeldungen Seite 123

7.4.4 Funktion 8 CANopen- Firmware-Version auslesen



FW Firmware-Version
Die Firmware-Version wird hexadezimal-kodiert abgebildet.

7.4.5 Funktion 20 CANopen- Hochlaufverzögerung einstellen



P Hochlaufverzögerung in Sekunden (Vorgabewert)
N Hochlaufverzögerung in Sekunden (Istwert)

Die CAN-Teilnehmer benötigen nach dem Einschalten der Spannung verschieden viel Zeit um für CAN-Telegramme bereit zu stehen. Der NMT-Manager darf erst Telegramme senden, wenn alle Teilnehmer am Bus bereit sind.
Die Einstellung einer Hochlaufverzögerung ist daher nur erforderlich, wenn die XCx als NMT-Manager projektiert wird.

7.5 Fehlermeldungen

Fehlermeldungen des CAN-Prozessors werden im „Activ- Errorbuffer“ bzw. im „Log- Book“ eingetragen. Diese Eintragungen können mit dem Inbetriebnahmetool NC-Dialog angezeigt werden.

Die Fehlernummer kann mit der Diagnosefunktion 7 durch die SPS ausgelesen werden.

Eine Erklärung der Fehlermeldungen ist im Kapitel "Fehlermeldungen" der XCx-Betriebsanleitung enthalten.

7.6 Die CANopen-spezifische Firmware-Bibliothek in MULTIPROG

Zu den Bibliotheken (ausser SchleicherLib) ist eine Online-Hilfe vorhanden, die die aktuellen Inhalte tiefgreifend darstellt. Die Online Hilfe ist über das Kontextmenü der jeweiligen Bibliothek erreichbar. Das Kontextmenü wird aktiv wenn mit der rechten Maustaste auf das Icon der Bibliothek geklickt wird.

Hinweis zu den Variablendeklarationen der Beispielprogramme von Funktionsbausteinen

Die Beispielprogramme in den Hilfen zu den Funktionsbausteinen enthalten Variablendeklarationen nach IEC 61131-3 mit den Schlüsselwörtern VAR und END_VAR. Sollen die Beispielprogramme mit MULTIPROG 3 angewendet werden, müssen die Variablendeklarationen in Tabellenform, auf dem Variablen-Arbeitsblatt der benutzten POE, von Hand eingetragen werden.

7.6.1 Inhalt der Bibliothek CANopen_V001

Funktionsbaustein	Kurzbeschreibung
CO_NET_SDO_WRITE	sendet ein Service Data Object (SDO)
CO_NET_SDO_READ	empfängt ein Service Data Object (SDO)
CO_NET_GET_LOCAL_NODE_ID	liefert die eigene Node- ID zurück
CO_NET_GET_STATE	liefert den aktuellen CANopen- Status
CO_NET_GET_KERNEL_STATUS	liefert den sog. erweiterten CANopen- Kernelstatus
CO_NET_NMT	setzt den Status eines oder aller Geräte im CANopen- Netzwerks
CO_NET_RECV_EMY_DEV	liest etwaige Emergency- Nachrichten von einem bestimmten Netzwerk- Knoten
CO_NET_RECV_EMY	liest etwaige Emergency- Nachrichten von einem beliebigen Netzwerk- Knoten
CO_NET_RECV_ERR_DEV	liest etwaige Error- Nachrichten von einem bestimmten Netzwerk- Knoten
CO_NET_RECV_ERR	liest etwaige Error- Nachrichten von einem beliebigen Netzwerk- Knoten
CO_NET_SENDL2	sendet beliebige CAN Layer 2- Nachrichten
CO_NET_PING	führt ein Ping auf einen bestimmten Netzwerk- Knoten aus
CO_NET_RESTART_CAN	startet die CANopen Kommunikation neu (z. B. nach "bus- off")
CO_NET_RESTART_ALL	startet den kompletten CANopen- Stack neu
CO_NET_SHUTDOWN	stoppt den CANopen- Stack
CO_NET_CAN_SYNC	ermöglicht die Synchronisation zwischen SPS- Task und den CANopen- Stack

7.7 Erweiterungsmodul XF-CAN

Das Modul kann für die Erweiterung der XCx 300 und XCx 540 eingesetzt werden. Es besitzt folgende Eigenschaften:

- eine CANopen Feldbusschnittstelle
- entsprechend CAN Application Layer für industrielle Anwendungen CAL, Datenübertragungschicht nach ISO 11898 und CAN 2.0 A+B, Kommunikationsprofil CiA DS-301

7.7.1 Anzeigen und Anschlüsse

Anzeigen			
LED	Farbe		
L1	nicht benutzt		
L2	nicht benutzt		
L3			CAN Netzwerkwerkstatus
		aus	CAN State Prepared
	grün	ein	CAN State Operational
		blinkend	CAN State Pre-Operational
	rot	ein	Bus Off
		blinkend	CAN-Fehler
L4			CAN Modulstatus
	grün	ein	CAN-Stack initialisiert
		blinkend	ungültige CAN-Konfiguration
	rot	ein	Steuereinheit nicht bereit oder schwerer Fehler
		blinkend	Fehler in der Steuerung
Anschluss X41			
Pin	Bezeichnung		
1	nc		nicht angeschlossen
2	CAN_L		
3	V-		Ground
4	nc		nicht angeschlossen
5	Drain		Schirmanschluß optional
6	V-		Ground
7	CAN_H		
8	nc		nicht angeschlossen
9	V+		Stromversorgung

7.7.2 Technische Daten

Grunddaten	
Schnittstelle	CANopen
Anschlusstechnik	D-Sub, 9-polig, Buchse

8 Das Multi-Task-System

8.1 Übersicht

Basis ist ein Echtzeit-Betriebssystem, das durch Taskprioritäten gesteuert wird. Drei Prioritätsstufen für Tasks werden verwendet:

- Prioritätsstufe für Überwachungstasks (supervisor task level)
- Prioritätsstufe für Anwender-Tasks und Default-Task (user task level)
- Prioritätsstufe für Betriebssystem-Tasks (system task level)

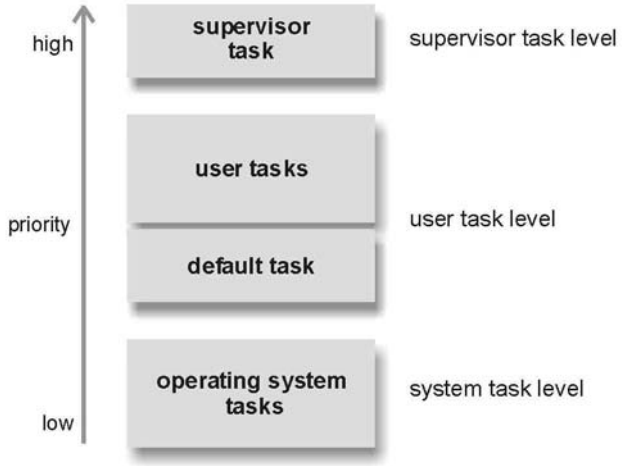


Abbildung 28: Tasksystem

Im Betriebssystem gibt es eine besonders geschützte Prioritätsstufe für die Überwachungstask. Die Überwachungstask ist eine Betriebssystem-Task, die in der höchsten Prioritätsstufe abgearbeitet wird. Die Überwachungstask ermittelt Fehler, wie z.B. eine Division durch Null oder die Überschreitung der Ausführungszeit einer Task und aktiviert die entsprechende Betriebssystem-Task.

Auf der Anwender- und Default-Task Stufe laufen alle Tasks, die vom Anwender eingefügt werden.
In diesem Bereich laufen auch einige wichtige Firmware-Tasks, die beim Parametrieren der Anwender-Tasks berücksichtigt werden müssen. Siehe dazu Abschnitt Task-Prioritäten.

Auf der Prioritätsstufe für Betriebssystem-Tasks laufen Tasks vom Anwender unbeeinflußt ab zB. Kommunikationstask, Debugtask, Speicherverwaltungstask und Systemkontrolltask.

8.2 Anwender-Tasks

Anwender-Tasks sind alle Tasks, die durch den Anwendungsprogrammierer eingefügt werden.
Die Default-Task gehört ebenfalls zur Prioritätsstufe für Anwender-Tasks. Sie ist die Anwender-Task mit der niedrigsten Priorität. Die Default-Task wird abgearbeitet, wenn zum entsprechenden Zeitpunkt keine Anwender-Task aktiv ist.

Eine falsche oder zumindest ungeeignete Wahl der Anwender-Task-Einstellungen hinsichtlich Typ, Priorität oder Interrupt- Mode usw. - insbesondere in Kombination mit langen Programmlaufzeiten - kann zu Steuerungsfehlfunktionen führen, da essentielle Betriebssystem-Tasks verdrängt werden.
Siehe dazu Abschnitt Task-Prioritäten.

Es können verschiedene Anwender-Tasktypen verwendet werden.

8.2.1 Zyklische Tasks

Zyklische Tasks führen die ihnen zugewiesenen Programme innerhalb eines definierten Zeitintervalls mit einer vom Anwender vorgegebenen Priorität aus.

In MULTIPROG können den einzelnen Tasks Prioritäten zwischen 0 und 31 zugeordnet werden. 0 steht für die höchste, 31 für die niedrigste Priorität. Die Task mit der höchsten Priorität wird als erste aufgerufen. Die User-Task Prioritäten werden auf die Prioritätsstufen des Echtzeitbetriebssystems abgebildet. (siehe Abschnitt Task-Prioritäten).

Wenn die Watchdog-Zeit einer zyklischen Task höher ist als die eingestellte Intervallzeit und die Ausführung der Task nicht beendet ist, bevor die eingestellte Intervallzeit erreicht wird, werden ein oder mehrere Ausführungs-Zyklen übersprungen.

8.2.2 Ereignis-Tasks

Ereignis-Tasks oder auch Event-Tasks werden vom Betriebssystem gestartet, wenn bestimmte Ereignisse auftreten.

Gegenwärtig sind folgende Ereignisse definiert.

Interne Bezeichnung	Ereignis-Nummer	Bemerkungen
Interrupts		
PLC_EVENT_XFIO_I0	0	XFIO Interrupt (Input 0, XCx 3/5)
PLC_EVENT_XUIO_0	0	U– Bus Interrupt 0 (XCx7, UBE32 0,1I Eingang 0)
PLC_EVENT_XFIO_I1	1	XFIO Interrupt (Input 1, XCx 3/5)
PLC_EVENT_XUIO_1	1	U– Bus Interrupt 1 (XCx7, UBE32 0,1I Eingang 1)
PLC_EVENT_XUIO_2	2	U– Bus Interrupt 2 (XCx7, UBE32 0,1I Eingang 2)
PLC_EVENT_XUIO_3	3	U– Bus Interrupt 3 (XCx7, UBE32 0,1I Eingang 3)
Synchronisation		
PLC_EVENT_POS	4	Lageregler- Task (nur XCN)
PLC_EVENT_CAN	5	CANopen- Task
PLC_EVENT_IPO	6	CNC IPO- Task (nur XCN)
PLC_EVENT_DECO	7	CNC DECO- Task (nur XCN)
PLC_EVENT_MCSIO	8	MCS / XCS20 IO-Treiber Synchronisation (microLine, XCx micro)
	9	Reserviert
PLC_EVENT_XFIO_I10	10	Messinterrupt aktiv 0
PLC_EVENT_XFIO_I11	11	Messinterrupt aktiv 1
	12	Reserviert
	13	Reserviert
	14	Reserviert
PLC_EVENT_AC_FAIL	15	AC Fail (ProNumeric)

Die Ereignisnummer wird in der Taskeinstellung von MULTIPROG verwendet, um das Ereignis zu spezifizieren, das die Ereignis-Task startet.

Die vorgegebene Priorität wird, außer bei gesetzter Bypass- Option vom System, berücksichtigt. (Bypass hebt den normalen Taskwechsel auf, so dass die zugewiesenen Programme sofort ausgeführt werden, wenn das Ereignis eintritt.)

Es werden bis zu 16 Ereignisse in eine Warteschlange gesetzt. Diese Ereignisse gehen daher nicht verloren und werden später ausgeführt. Dies gilt auch im Falle eines Auftretens neuer Ereignisse vor der Ausführung der zugewiesenen Ereignis-Task.

8.2.3 System-Tasks

System-Tasks bzw. Systemprogramme (SPG's) werden automatisch vom Betriebssystem gestartet, wenn im Zusammenhang mit dem Betriebssystem ein Ereignis auftritt.
Verschiedene SPG's sind verfügbar, wie in der folgenden Tabelle dargestellt:

Nr.	Name	Ereignis	Aktionen
SPG 0	WARM_START	wird bei einem Warmstart ausgeführt	• remanente Daten werden nicht initialisiert • nicht-gepufferte Daten werden initialisiert • die Open-Funktion des I/O-Treibers wird ausgeführt • Anwender-Tasks werden aktiviert • SPS wechselt in den Zustand 'Betrieb'
SPG 1	COLD_START	wird bei einem Kaltstart ausgeführt	• alle Daten werden initialisiert • die Open-Funktion des I/O-Treibers wird ausgeführt • Anwender-Tasks werden aktiviert • SPS wechselt in den Zustand 'Betrieb'
SPG 2	TO_STOP	wird ausgeführt, wenn die Programmausführung gestoppt wird	• Anwender-Tasks werden deaktiviert • alle Ausgänge werden aktualisiert • die Close-Funktion des I/O-Treibers wird ausgeführt • SPS wechselt in den Zustand 'STOP'
SPG 10	WATCHDOG	wird ausgeführt, wenn die Ausführung einer Task nicht innerhalb ihrer Watchdogzeit beendet ist	• Anwender-Tasks werden deaktiviert • alle Ausgänge werden aktualisiert • die Close-Funktion des I/O-Treibers wird ausgeführt • SPS wechselt in den Zustand 'STOP'
SPG 11	ZERODIV	wird ausgeführt, wenn während der Programmausführung eine Division durch Null aufgetreten ist	• Anwender-Tasks werden deaktiviert • alle Ausgänge werden aktualisiert • die Close-Funktion des I/O-Treibers wird ausgeführt • SPS wechselt in den Zustand 'STOP'
SPG 12	STACKOVER	wird ausgeführt, wenn ein Stacküberlauf aufgetreten ist. Wird nur ausgeführt, wenn das Kontrollkästchen 'Stack-Prüfung' im Dialog 'Ressource ... einrichten' in MULTIPROG aktiviert wurde.	• Anwender-Tasks werden deaktiviert • alle Ausgänge werden aktualisiert • die Close-Funktion des I/O-Treibers wird ausgeführt • SPS wechselt in den Zustand 'STOP'
SPG 13	BADCAL	wird ausgeführt, wenn eine herstellerspezifische POE aufgerufen wird, die nicht existiert	• Anwender-Tasks werden deaktiviert • alle Ausgänge werden aktualisiert • die Close-Funktion des I/O-Treibers wird ausgeführt • SPS wechselt in den Zustand 'STOP'
SPG 14	IOERROR	wird ausgeführt, wenn ein Fehler im I/O-Treiber auftritt, während der Prozeß abläuft	• SPS setzt Abarbeitung fort
SPG 16	MATHERR	wird ausgeführt, wenn ein Gleitkommafehler in einer arithmetischen Funktion auftritt	• Anwender-Tasks werden deaktiviert • alle Ausgänge werden aktualisiert • die Close-Funktion des I/O-Treibers wird ausgeführt

Nr.	Name	Ereignis	Aktionen
			• SPS wechselt in den Zustand 'STOP'
SPG 17	CPU_OVERLOAD	wird ausgeführt, wenn eine CPU-Überlastung auftritt	• Anwender-Tasks werden deaktiviert • alle Ausgänge werden aktualisiert • die Close-Funktion des I/O-Treibers wird ausgeführt • SPS wechselt in den Zustand 'STOP'
SPG 18	INITIODRV_ERR	wird ausgeführt, wenn beim Initialisieren des I/O-Treibers während eines Kalt- oder Warmstarts ein Fehler auftritt	• SPS wird nicht gestartet
SPG 19	BOUNDS_ERR	wird ausgeführt, wenn die Grenzen eines Felds oder einer Struktur überschritten wurden. Wird nur ausgeführt, wenn das Kontrollkästchen 'Index-Prüfung' oder das Kontrollkästchen 'Feldbegrenzungs-Prüfung' im Dialog 'Ressource ... einrichten' in MULTIPROG aktiviert wurde.	• Anwender-Tasks werden deaktiviert • alle Ausgänge werden aktualisiert • die Close-Funktion des I/O-Treibers wird ausgeführt • SPS wechselt in den Zustand 'STOP'
SPG 20	BUS_ERR	wird ausgeführt, wenn Variablen mit einem Datentyp ≥ 2 Bytes und ungeraden Adressen verwendet wurden oder wenn in MULTIPROG ein interner Fehler aufgetreten ist. Nur bei Motorola-Plattformen.	• Anwender-Tasks werden deaktiviert • alle Ausgänge werden aktualisiert • die Close-Funktion des I/O-Treibers wird ausgeführt • SPS wechselt in den Zustand 'STOP'
SPG 21	STRING_ERR	wird ausgeführt, wenn ein Fehler bei einer Zeichenfolge-Operation auftritt, z.B. wenn eine Zeichenfolge durch eine andere ersetzt werden sollte, aber nicht gefunden wurde.	• Das Verhalten einer Zeichenfolge-Ausnahme hat sich geändert! In der Standardeinstellung wird nach dem Auftreten einer Zeichenfolge-Ausnahme das SPG 21 aufgerufen. Zusätzlich wird ein Eintrag in den Fehlerkatalog vorgenommen, der die Modul- und Zeilennummer enthält. Die SPS bleibt im 'RUN'-Status.

System-Tasks werden nicht vom Watchdog überwacht.

8.2.4 Default- Task

Die Default-Task läuft mit der niedrigstmöglichen Anwender-Task-Priorität als sog. Hintergrund-Task und ist nicht zeitüberwacht. Sie wird dann aktiviert, wenn alle höherprioren Anwender-Tasks abgearbeitet wurden. Die Default-Task ist dabei so konfiguriert, dass sie einen Teil der zur Verfügung stehenden Restzeit beansprucht (max. 80 %) jedoch höchstens alle 20ms ausgeführt wird.
In jeder Ressource ist nur eine Default-Task erlaubt.

Alle Treiber der I/O-Konfiguration, die nicht explizit einer Anwender-Task zugeordnet wurden, führen zum automatischen Anlegen der Default-Task und werden im Kontext der Default-Task ausgeführt.

8.3 Anwender-Task-Information

Für jede Anwender-Task werden Informationen auf System-Variablen abgebildet. Die unten abgebildeten Typdefinitionen der Systemvariablen sind in der Bibliothek SchleicherLib im Abschnitt PLC_Types zu finden.

```
TYPE
 TaskNameType : ARRAY [1..10] OF BYTE;
END_TYPE

TYPE
 TaskInfoType0 : STRUCT
  MaxTask   : INT;      (* 00: *)                    max. mögliche Taskanzahl
  CurTask   : INT;      (* 02: *)                    aktuelle Taskanzahl
 END_STRUCT (* TaskInfoType0 *);
END_TYPE


TYPE
 TaskInfoType1 : STRUCT
  TaskName   : TaskNameType; (* 04: *)               Taskname
  TaskPrio   : INT;      (* 14: *)                   Taskpriorität
  TaskMode   : INT;      (* 16: *)                   Taskmode
  TaskPeriod  : INT;      (* 18: [ms] *)             Taskperiode in ms
  TaskStack  : INT;      (* 20: *)                   Größe des benutzten Task-Stacks
  MainPoe   : INT;      (* 22: assigned PLC program *) zugeordnetes SPS-Programm
  TaskWatchDog : INT;      (* 24: [ms] *)            Watch-Dog-Zeit in ms
  reserve0   : DINT;     (* 26: *)
  MaxStack   : INT;      (* 30: max. used stack *)   Größe des möglichen Task-Stacks
  CurDuration : INT;      (* 32: [ticks] *)          aktuelle Taskdauer einschließlich
                                                      bevorrechtigte Aufrufe
  MinDuration : INT;      (* 34: [ticks] *)          minimale Taskdauer
  MaxDuration : INT;      (* 36: [ticks] *)          maximale Taskdauer
  AveDuration : INT;      (* 38: [ticks] *)          mittlere Taskdauer
  CurDelay   : INT;      (* 40: [ticks] *)           aktuelle Taskverzögerung
  MinDelay   : INT;      (* 42: [ticks] *)           minimale Taskverzögerung
  MaxDelay   : INT;      (* 44: [ticks] *)           maximale Taskverzögerung
  AveDelay   : INT;      (* 46: [ticks] *)           mittlere Taskverzögerung
 END_STRUCT (* TaskInfoType1 *);
END_TYPE
```

Die Variablen werden mit den Typen *TaskInfoType0* und *TaskInfoType1* deklariert.

Global_Variables:Configuration.Resource - Configuration.Resource.Global_Variables

Name	Typ	Verwendung	Beschreibung	Adresse	Anfangsw...	Reman...
TaskInfo0	TaskInfoType0	VAR_GLOBAL		%MD 1.1000		☐
TaskInfo1	TaskInfoType1	VAR_GLOBAL		%MD 1.1004		☐
TaskInfo2	TaskInfoType1	VAR_GLOBAL		%MD 1.1068		☐
TaskInfo3	TaskInfoType1	VAR_GLOBAL		%MD 1.1132		☐
TaskInfo4	TaskInfoType1	VAR_GLOBAL		%MD 1.1196		☐
TaskInfo5	TaskInfoType1	VAR_GLOBAL		%MD 1.1260		☐

Die folgende Anwendertask-Information wird mit einem Offset von 64 ab 1004 deklariert (1004 + 64 = 1068 usw.).

Die Reihenfolge der Tasks wird durch den Rang der Task im Projektbaum *Physikalische Hardware/Configuration/Resource/Tasks* festgelegt.

8.4 Task-Prioritäten

Die Tabelle gibt eine Übersicht über die empfohlenen Task-Prioritäten bzw. deren Einordnung hinsichtlich wichtiger reservierter Firmware-Tasks (tfwLAGE, tfwCANhigh, tfwIPO).

MULTIPROG-Priorität	RTOS*- Priorität (defaultl)	RTOS* Task-Name	Verwendung
0	30	beliebig	z.B. Anwender-Task (Ereignis 0)
1	31	beliebig	z.B. Anwender-Task (Ereignis 1)
2	32	beliebig	z.B. Anwender-Task (Ereignis 4)
3	33	tfwLAGE	reserviert für Lageregel- Task (nur XCN)
4	34	beliebig	z.B. Anwender-Task (Ereignis 4, 5)
5	35	tfwCANhigh	reserviert für CAN- Stack Task (Option CAN_HIGH_PRIO = 1)
6	36	beliebig	z.B. Anwender-Task (Ereignis 5, 6)
7	37	tfwIPO	reserviert für IPO- Task (nur XCN)
8	38	beliebig	z.B. Anwender-Task (Ereignis 5)
9	39	tfwCANhigh	reserviert für CAN- Stack Task (Option CAN_HIGH_PRIO = 0)
10	40	beliebig	z.B. Anwender-Task (Ereignis 5)
11..15	41..45	beliebig	z.B. zyklische Anwender-Tasks
16..31	46	beliebig	z.B. sonstige, zyklische Anwender-Tasks
Default	127	default	Hintergrund-Task

*Real Time Operating System

Eine falsche oder zumindest ungeeignete Wahl der Anwender-Task-Einstellungen hinsichtlich Typ, Priorität oder Interrupt-Mode usw. - insbesondere in Kombination mit langen Programmlaufzeiten - kann zu Steuerungsfehlfunktionen führen, da essentielle Firmware-Tasks (tfwLAGE, tfwCANhigh, tfwIPO) verdrängt werden.

Das System unterstützt insgesamt 18 Anwender-Tasks (Prioritätsstufen 0..16 und die Default-Task).
Tasks mit Prioritätswerten >= 16 werden mit Priorität 16 ausgeführt.

8.5 Tasks und Watchdogs

Es gibt zu jeder anwenderdefinierten Task einen eigenen einstellbaren Watchdog.
Der Watchdog überprüft, ob die Taskausführung am Ende des Watchdog-Zeitintervalls beendet ist. Wenn die Taskausführung nach dieser Zeit nicht beendet wird, wird die System-Task SPG 10 'WATCHDOG' ausgeführt und die SPS geht in den 'STOP'-Zustand über, wenn keine weiteren Aktionen programmiert wurden. Zusätzlich wird ein Eintrag in den Fehlerkatalog vorgenommen. Das Watchdog-Zeitintervall beginnt, wenn die Task bereit für die Ausführung ist. Das Watchdog-Zeitintervall wird im Dialog 'Task ... einrichten' in MULTIPROG festgelegt.

Wenn die Ausführungsdauer der Task sowie die Watchdog-Zeit annähernd denselben Wert haben und eine hohe CPU-Auslastung vorliegt, ist es möglich, daß während der Umsetzung einiger Online-Bedienschritte die Watchdog-Zeit überschritten wird.

Ein Grund für dieses Verhalten kann sein, daß während des Debuggens im Online-Modus der Adreßstatus mit Durchlaufkontrolle ausgewählt wurde.

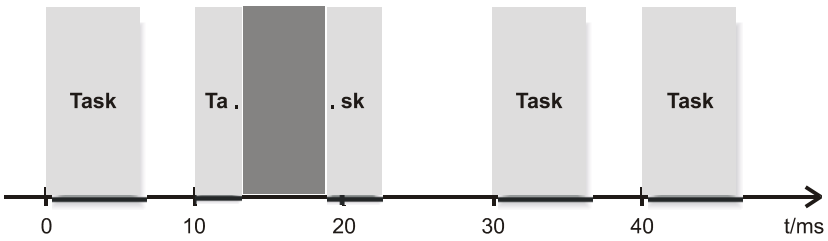


Abbildung 29: Task und Watchdog

In diesem Beispiel ist die Watchdog-Zeit der angezeigten Task auf 10 ms eingestellt. In der Abbildung überschreitet die Task ihre Watchdog-Zeit nach 20 ms. Wenn die Watchdog-Zeit der Task auf 20 ms eingestellt ist, wird sie beim nächsten Mal nach 30 ms ausgeführt. In diesem Fall wird die Ausführung einer Task nach 20 ms übersprungen.

8.6 Tasks einfügen und Programme zuweisen

Um eine Task einzufügen müssen in MULTIPROG folgende Schritte ausgeführt werden:

- Im Projektbaum, unter der Ressource für die jeweilige Steuerung, mit der rechten Maustaste auf den Ordner Tasks klicken, um das Kontextmenü zu öffnen.

- Den Menüpunkt Einfügen/Task wählen, es erscheint der Dialog Einfügen.
- Den Namen für die Task eingeben.
- Im Listenfeld Task-Typ den gewünschten Task-Typ einstellen. Es kann zwischen einer Default-Task, zyklischen Task, der Ereignis- oder System-Task gewählt werden.
  Hinweis: Wenn der Tasktyp 'DEFAULT' nicht in dem Listenfeld aufgeführt ist, besitzt die Ressource bereits eine Default-Task.
- Den Dialog. mit OK bestätigen.
  Es erscheint der Dialog Task-Einstellungen für ... Abhängig von der zuvor ausgewählten Task enthält der Dialog verschiedene Text- und Listenfelder.
- Für die jeweilige Task müssen folgende Parameter eingegeben werden:

zyklische Task	Zeitintervall
Ereignis-Task	Ereignisnummer (Nummer des Interrupt)
System-Task	Nummer eines Systemprogrammes

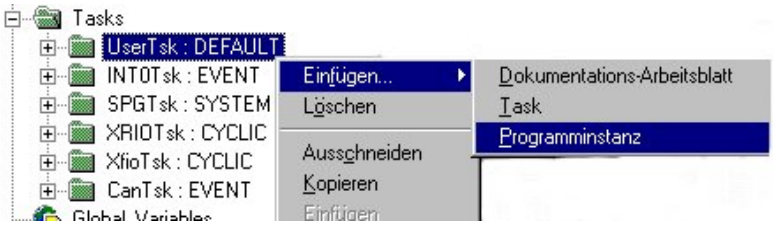
Bei der Vergabe der Priorität müssen unbedingt die Ausführungen im Abschnitt Task-Prioritäten beachtet werden.

Programme müssen Tasks zugewiesen werden um sie auszuführen. Zuweisen eines Programmes zu einer Task bedeutet, daß eine Instanz des Programmes ausgeführt wird, wenn die Task aktiviert wird. Von einem Programm können verschiedene Instanzen verschiedenen Tasks zugeordnet werden.

Einer Task können mehrere Programme zugewiesen werden. In diesem Fall wird das erste Programm im Taskverzeichnis als erstes ausgeführt. Danach wird das Programm darunter ausgeführt usw.

Um Programme zuzuweisen müssen in MULTIPROG folgende Bedienschritte durchgeführt werden:

- Im Projektbaum mit der rechten Maustaste auf das Symbol der Task klicken in die das Programm eingefügt werden soll.



- Im Kontextmenü Einfügen/Programminstanz wählen.
- Einen Instanznamen für das Programm in das Feld Programminstanz eingeben.
- Im Listenfeld Programmtyp das gewünschte Programm einstellen.
- Den Dialog mit OK bestätigen.
  Das Programmsymbol wird in den Projektbaum eingefügt.

9 Die SPS

- Betriebssytem: ProConOS
- Programmierung: MULTIPROG nach IEC 61131-3
- Kommunikation mit der CNC über Koppelspeicher

9.1 Programmierung

Die Programmierung der XCx erfolgt mit der Programmiersoftware MULTIPROG nach IEC 61131-3 auf einem PC.

Die Programmiersoftware besteht aus der Software MULTIPROG und den AddOns für MULTIPROG von Schleicher.

Das Programmiersystem mit Programmieranleitung ist als Zubehör zu beziehen. Siehe dazu Kapitel "Zubehör und Ersatzteile".

Die SPS wird mit dem fertig kofigurierten Projekt ausgeliefert, auf dessen Grundlage die Programmierung der SPS begonnen werden kann (siehe Kapitel Inbetriebnahme).

9.2 SPS-Betriebszustände und Startverhalten

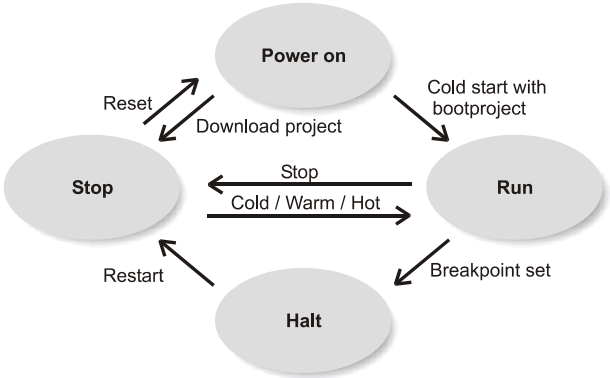
9.2.1 Betriebszustände

Betriebszustand	Beschreibung
EIN	Stromzufuhr ist eingeschaltet es ist kein Programm geladen
STOP	Programm ist geladen Anwendertasks sind inaktiv Eingänge des Prozessabbildspeichers werden nicht aktualisiert Ausgangssignale werden nicht an die Ein- und Ausgänge übermittelt
BETRIEB	Programmausführung ist aktiviert Anwendertasks sind aktiv Eingänge des Prozessabbildspeichers werden gemäß der I/O-Konfiguration aktualisiert Ausgänge des Prozessabbildspeichers werden gemäß der I/O-Konfiguration und der Programmausführung aktualisiert
HALT	Programmausführung wird an einem Haltepunkt angehalten Anwendertasks sind inaktiv Eingänge des Prozessabbildspeichers werden nicht aktualisiert Ausgänge des Prozessabbildspeichers werden nicht aktualisiert

Der aktuelle Zustand der SPS wird in MULTIPROG im Projekt-Kontrolldialog in der Zeile Status: angezeigt.
Wenn hinter dem aktuellen Zustand im Kontrolldialog 'Debug' angezeigt wird, bedeutet das, dass Haltepunkte gesetzt oder Variablen geforct wurden.

9.2.2 Wechseln der Betriebszuständen mit MULTIPROG

Über die graphische Benutzeroberfläche von MULTIPROG kann gesteuert werden, wann die Programmausführung auf der SPS gestartet und gestoppt wird. Die Schaltflächen für Wechsel, die im aktuellen Betriebszustand nicht möglich sind, sind im Projekt-Kontrolldialog abgeblendet.



Starten der Programmausführung

Zustandswechsel von → nach	Schaltfläche im Kontrolldialog	Beschreibung, was passiert
Stop → Betrieb	Cold	• es erfolgt ein Kaltstart • alle Daten werden initialisiert • SPG 1 wird aufgerufen • alle Anwendertasks werden aktiviert • die Programmausführung wird aktiviert
Stop → Betrieb	Warm	• es erfolgt ein Warmstart • nur nicht-gepufferte Daten werden initialisiert • SPG 0 wird aufgerufen • alle Anwendertasks werden aktiviert • die Programmausführung wird aktiviert
Stop → Betrieb	Hot	• es erfolgt ein Heißstart • es werden keine Daten initialisiert • alle Anwendertasks werden aktiviert • die Programmausführung wird aktiviert • nicht verfügbar, wenn Sie die Programmausführung zum ersten Mal nach dem Senden starten

Stoppen der Programmausführung

Zustandswechsel von → nach	Schaltfläche im Kontrolldialog	Beschreibung, was passiert
Betrieb → Stop	Stop	• alle Anwendertasks werden deaktiviert, wenn ihr Arbeitszyklus beendet ist • SPG 2 wird aufgerufen • die Ausgänge des Prozessabbildspeichers werden geschrieben • die Programmausführung wird gestoppt • die physikalischen Ausgänge werden auf Null oder Vorzugsabschaltlage gesetzt

Allgemeines Reset

Zustandswechsel von → nach	Schaltfläche im Kontrolldialog	Beschreibung, was passiert
Stop → Ein	Reset	• das Projekt wird gelöscht • es erfolgt ein allgemeines Reset

9.2.3 Startverhalten der SPS nach dem Einschalten der Versorgungsspannung

Das SPS-Startverhalten wird mit dem Betriebsartenschalter eingestellt.

Es können folgende Varianten ausgewählt werden:

- PROG SPS-Stop
- WARM SPS-Warmstart nach IEC 61131-3
- COLD SPS-Kaltstart nach IEC 61131-3

9.3 Systemvariablen

Systemvariablen informieren über den Systemzustand, wie z.B. über geforcte Variablen, Leistungsfähigkeit der CPU, etc. Diese Variablen haben feste Speicheradressen und können vom SPS-Programm verwendet werden, um die entsprechenden Informationen zu erhalten.

Alle Systemvariablen in der folgenden Tabelle sind bereits im Bereich Global_Variables des Arbeitsblattes Global_Variables deklariert.

Systemvariablen				
Name	Datentyp	Log. Adr. (Byte)	Log. Adr. (Bit)	Beschreibung
PLCMODE_ON	BOOL	0	0	TRUE := aktueller SPS-Zustand ist EIN
PLCMODE_RUN	BOOL	0	1	TRUE := aktueller SPS-Zustand ist BETRIEB
PLCMODE_STOP	BOOL	0	2	TRUE := aktueller SPS-Zustand ist STOP
PLCMODE_HALT	BOOL	0	3	TRUE := aktueller SPS-Zustand ist HALT
PLCDEBUG_BPSET	BOOL	1	4	TRUE := ein oder mehrere Haltepunkte sind gesetzt
PLCDEBUG_FORCE	BOOL	2	0	TRUE := eine oder mehrere Variablen sind geforct
PLCDEBUG_POWERFLOW	BOOL	2	3	TRUE := Durchlaufkontrolle ist aktiv
PLC_TICKS_PER_SEC	INT	44	-	Anzahl der Systemticks pro Sekunde, die von der SPS als Systemzeitbasis verwendet werden. Dieser Wert bestimmt die Zeitauflösung der SPS für Funktionsbausteine für Zeitverzögerung, wie TON, TOF oder TP und die kürzeste Zykluszeit für die DEFAULT-Task und zyklische Tasks.
PLC_SYS_TICK_CNT	DINT	52	-	Anzahl der gezählten SPS Systemticks

Zusätzlich zu diesen Systemvariablen sind weitere Variablen definiert, die Informationen zum System vorhalten.
Die Typdefinitionen der Variablen sind in der Bibliothek SchleicherLib im Abschnitt PLC_Types zu finden.

9.4 Bibliotheken und Funktionsbausteine in MULTIPROG

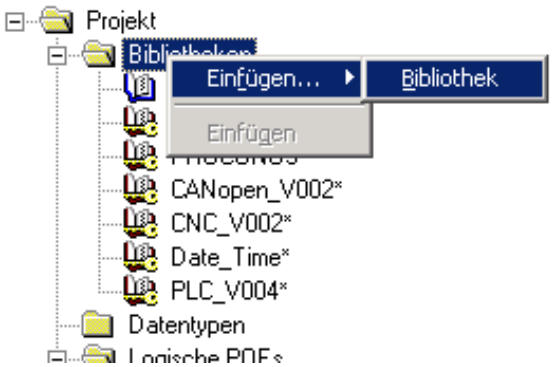
Funktionsbausteine sind in Bibliotheken zusammengefasst. Sie werden, je nach Steuerungstyp, beim Erstellen eines neuen MULTIPROG-Projektes automatisch eingebunden oder können bei Bedarf manuell eingebunden werden.

Bibliotheken	XCN 7xx	XCS 7xx	XCN 5xx	XCS 5xx	XCN 3xx	XCS 3xx	Pro-Numeric	Pro-SyCon	MCS 20-20	MCS 20-21	Simulation
PROCONOS	+	+	+	+	+	+	+	+	+	+	+
BIT_UTIL	+	+	+	+	+	+	+	+	+	+	+
CANopen_Vxxx	+	+	+	+	o	o	+	+	-	+	-
CFB_Vxxx	o	o	o	o	o	o	o	o	-	-	-
CNC_Vxxx	+	-	+	-	+	-	+	-	-	-	-
Date_Time	+	+	+	+	+	+	+	+	+	+	-
Microline	-	-	-	-	-	-	-	-	+	+	-
MMI	o	o	o	o	o	o	o	o	o	o	-
PLC_Vxxx	+	+	+	+	+	+	+	+	-	-	-
Profibus_Vxxx	o	o	o	o	o	o	o	o	-	-	-
Serial	o	o	o	o	o	o	o	o	o	o	-
SchleicherLib_Vxxx	+	+	+	+	+	+	+	+	-	-	-
XCx7_Vxxx	+	+	-	-	-	-	-	-	-	-	-

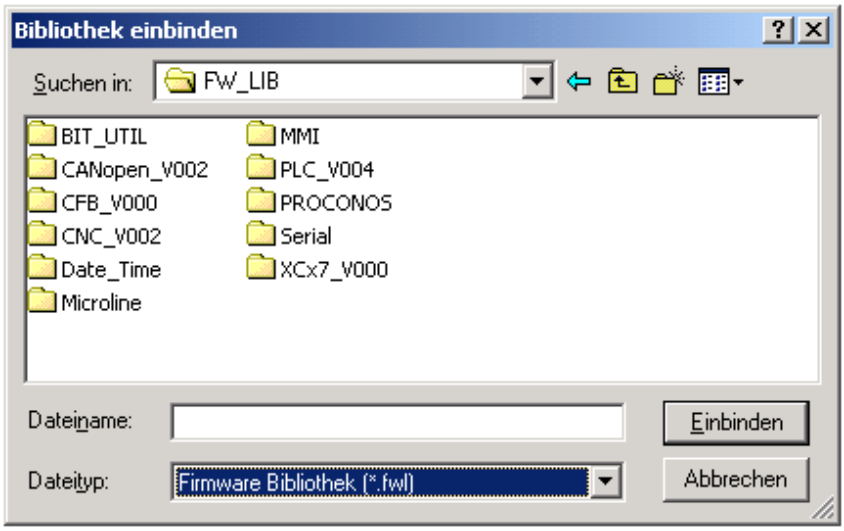
+ Werden beim Erstellen eines neuen Projektes automatisch eingebunden.
o Können je nach Bedarf manuell eingebunden werden.
- nicht möglich oder unnötig

Funktionsbausteine können folgendermaßen eingebunden werden:

- Mit der rechten Maustaste im Projektbaum von MULTIPROG das Kontextmenü Bibliotheken/Einfügen/Bibliothek öffnen.

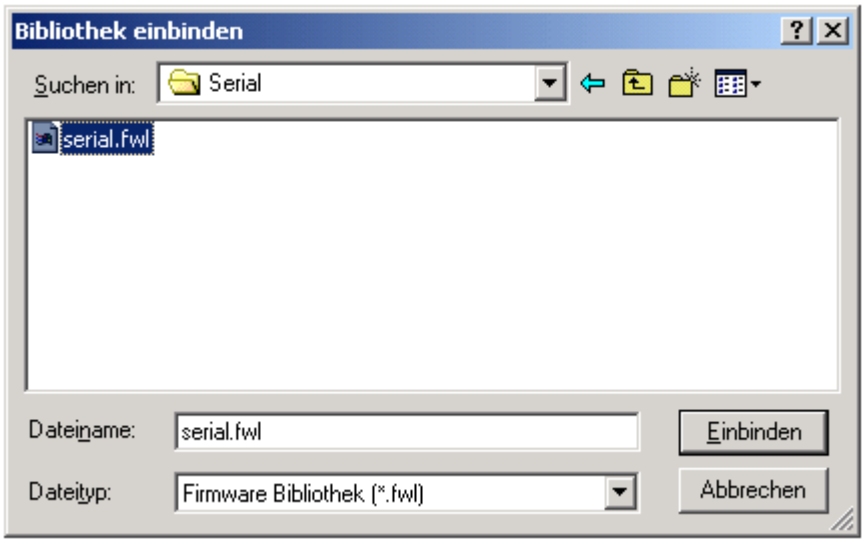


- Den Pfad .\KWSoft\MWT\PLC\FW_LIB und den Dateityp Firmware Bibliothek (*.fwl) wählen.



Jede Bibliothek ist in einem eigenen Pfad gespeichert.

Soll z.B. die Bibliothek *Serial* eingebunden werden, muss sie im gleichnamigen Pfad die Bibliothek ausgewählt werden.



Zu den Bibliotheken (außer SchleicherLib) ist eine Online-Hilfe vorhanden. Die Online Hilfe ist über das Kontextmenü der jeweiligen Bibliothek erreichbar. Das Kontextmenü wird aktiv wenn mit der rechten Maustaste auf das Icon der Bibliothek geklickt wird.

Hinweis zu den Variablendeklarationen der Beispielprogramme von Funktionsbausteinen

Die Beispielprogramme in den Hilfen zu den Funktionsbausteinen enthalten Variablendeklarationen nach IEC 61131-3 mit den Schlüsselwörtern VAR und END_VAR. Sollen die Beispielprogramme mit MULTIPROG angewendet werden, müssen die Variablendeklarationen in Tabellenform, auf dem Variablen-Arbeitsblatt der benutzten POE, von Hand eingetragen werden.

9.4.1 Bibliothek CANopen_Vxxx

Die Bibliothek enthält Funktionsbausteine für die Parametrierung und Diagnose des CANopen Netzwerkes.

Funktionsbaustein	Nr.	Kurzbeschreibung	Steuerungstypen
CO_NET_SDO_WRITE	150	sendet ein Service Data Object (SDO)	XCx ProNumeric ProSyCon MCS 20-21
CO_NET_SDO_READ	151	empfängt ein Service Data Object (SDO)	
CO_NET_GET_LOCAL_NODE_ID	152	liefert die eigene Node- ID zurück	
CO_NET_GET_STATE	153	liefert den aktuellen CANopen- Status	
CO_NET_GET_KERNEL_STATUS	154	liefert den sog. erweiterten CANopen-Kernelstatus	
CO_NET_NMT	155	setzt den Status eines oder aller Geräte im CANopen- Netzwerks	
CO_NET_RECV_EMY_DEV	156	liest etwaige Emergency- Nachrichten von einem bestimmten Netzwerk- Knoten	
CO_NET_RECV_EMY	157	liest etwaige Emergency- Nachrichten von einem beliebigen Netzwerk- Knoten	
CO_NET_RECV_ERR_DEV	160	liest etwaige Error- Nachrichten von einem bestimmten Netzwerk- Knoten	
CO_NET_RECV_ERR	161	liest etwaige Error- Nachrichten von einem beliebigen Netzwerk- Knoten	
CO_NET_SENDL2	162	sendet beliebige CAN Layer 2- Nachrichten	
CO_NET_PING	163	führt ein Ping auf einen bestimmten Netzwerk-Knoten aus	
CO_NET_RESTART_CAN	164	startet die CANopen Kommunikation neu (z. B. nach "bus- off")	
CO_NET_RESTART_ALL	165	startet den kompletten CANopen- Stack neu	
CO_NET_SHUTDOWN	166	stoppt den CANopen- Stack	
CO_NET_CAN_SYNC	170	ermöglicht die Synchronisation zwischen SPS-Task und den CANopen- Stack	

9.4.2 Bibliothek CFB_Vxxx

Die an IEC 61131-5 angelehnte Bibliothek CFB_Vxxx enthält Funktionsbausteine zur “peer-to-peer“ Kommunikation über TCP/IP.

Funktionsbaustein	Nr.	Kurzbeschreibung	Steuerungstypen
CONNECT_V	60	stellt eine "peer-to-peer" Verbindung zwischen zwei Teilnehmern her	XCx ProNumeric ProSyCon
USEND_V	61	sendet beliebige Daten	
URCV_V	62	empfängt beliebige Daten	

9.4.3 Bibliothek CNC_Vxxx

Die Bibliothek CNC_Vxxx enthält Funktionsbausteine für das Lesen und Schreiben von Systemdaten, SERCOS-, XRIO- und CAN-Antriebsparametern und PROFIBU-DP-Antriebsparametern.

Funktionsbaustein	Nr.	Kurzbeschreibung	Steuerungstypen
READ_Q_PARAM_*	200 bis 207	liest einen CNC-Systemdaten-Parameter	XCN ProNumeric
WRITE_Q_PARAM_*	208 bis 215	schreibt einen CNC-Systemdaten-Parameter	
SAVE_Q_PARAM_*	221	speichert die CNC-Systemdaten-Parameter auf der Festplatte	
SAVE_R_PARAM_*	220	speichert die CNC-Rechenparameter auf der Festplatte	
READ_SERC_PARAM	302	liest einen SERCOS-Parameter	
WRITE_SERC_PARAM	303	schreibt einen SERCOS-Parameter	
SET_SERC_PHASE	304	Umschaltung der SERCOS-Kommunikationsphase	
SET_SERC_COMMAND	308	ausführen eines SERCOS-Kommandos	
MC_ANALOG	300	XRIO Motion Control Baustein (mit Lageregler)	XCN
MC_ANALOG_1_AXIS	307	XRIO Motion Control Baustein für eine Achse (mit Lageregler)	
MC_CAN	301	CAN MotionControl Baustein	
MC_DP	309	PROFIBUS-DP Motion Control Baustein	
MC_DP_1_AXIS	310	PROFIBUS-DP Motion Control Baustein für eine Achse	

Hinweise:

Die Funktionsbausteine MC_ANALOG und MC_CAN sind von der Bibliothek PCL_Vxxx in die Bibliothek CNC_Vxxx übernommen worden (ab CNC_V003 / PLC_V005).

Die Funktionsbausteine MC_DP und MC_DP_1_AXIS gibt es ab der Version CNC_V004.

9.4.4 Bibliothek Date_Time

Die XCx verfügt über eine gepufferte Echtzeituhr mit Kalender (Berücksichtigung von Schaltjahren) und einer Auflösung von 1 Sekunde.

Datum und Uhrzeit können mit den Funktionsbausteinen aus der Bibliothek Date_Time gelesen und gesetzt werden.

Funktionsbaustein	Nr.	Kurzbeschreibung	Steuerungstypen
GET_TIME	130	Zeit lesen	XCx ProNumeric ProSyCon MCS xx-xx
GET_DATE	128	Datum lesen	
SET_TIME	131	Zeit setzen	
SET_DATE	129	Datum setzen	

9.4.5 Bibliothek MMI

Die Bibliothek MMI realisiert die Kommunikation mit einem Bediengerät der COP-Familie über die serielle Schnittstelle der Steuerung.

Funktionsbaustein	Nr.	Kurzbeschreibung	Steuerungstypen
PPF_COP_COMM	140	kommuniziert mit einem COP-Bediengerät (PNet-Protokoll)	XCx ProNumeric ProSyCon MCS xx-xx

9.4.6 Bibliothek PLC_Vxxx

Über den Umfang der Standard IEC- bzw. ProConOS-Funktionsbausteine hinaus, werden weitere, steuerungsspezifische Firmware-Funktionsbausteine in dieser Bibliothek bereitgestellt.

Funktionsbaustein	Nr.	Kurzbeschreibung	Steuerungstypen
PUT_ERROR	400	erzeugt eine nutzerdefinierte Fehlermeldung (bitte nicht mehr verwenden)	XCx ProNumeric ProSyCon
PUT_ERROR2	401	erzeugt eine nutzerdefinierte Fehlermeldung	
CLEAR_ERROR	402	löscht eine mit Lock-Flag abgesetzte Fehlermeldung	
READ_FILE	405	lesender Dateizugriff	
WRITE_FILE	406	schreibender Dateizugriff	
SEND_MAIL	410	sendet eine E-MAIL (SMTP Client)	
XFIO_CONFIG	420	XFIO Interrupt Konfiguration	XCx
XRIO_STATE	422	XRIO Statusinformationen	
GET_MTS	430	liefert den aktuellen Zeitwert in µs Ticks	XCx ProNumeric ProSyCon
OPEN_PROFILE	431	öffnet eine Datei im INI-Format	
NEW_PROFILE	432	legt eine neue Datei im INI-Format an	
FLUSH_PROFILE	433	schreibt aktualisierte Datei im INI-Format	
CLOSE_PROFILE	434	schließt eine Datei im INI-Format	
GET_PROFILE_STRING	435	liest einen String aus einer Datei im INI-Format	
GET_PROFILE_INT	436	liest einen Integer-Wert aus einer Datei im INI-Format	
GET_PROFILE_REAL	437	liest einen Real-Wert aus einer Datei im INI-Format	
WRITE_PROFILE_STRING	438	schreibt einen String in eine Datei im INI-Format	
WRITE_PROFILE_INT	439	schreibt einen Integer-Wert in eine Datei im INI-Format	
WRITE_PROFILE_REAL	440	schreibt einen Real-Wert in eine Datei im INI-Format	

Hinweise:

Die Funktionsbausteine MC_ANALOG und MC_CAN sind von der Bibliothek PCL_Vxxx in die Bibliothek CNC_Vxxx übernommen worden (ab CNC_V003 / PLC_V005).

Die Funktionsbausteine xxx_PROFILE_xxx gibt es ab der Version PLC_V006.

9.4.7 Bibliothek Profibus_Vxxx

Die Bibliothek Serial enthält Funktionsbausteine für die Kommunikation über die PROFIBUS-Karte.

Funktionsbaustein	Nr.	Kurzbeschreibung	Steuerungstypen
DP_NET_GET_STATE	190	liefert den Status der PROFIBUS-Karte	XCx ProNumeric ProSyCon
DP_NET_PUT_MSG	191	setzt eine Nachricht an das Message Interface der Hilscher Karte ab	
DP_NET_GET_MSG	192	holt eine Nachricht vom Message Interface der Hilscher Karte ab	

9.4.8 Bibliothek Serial

Die Bibliothek Serial enthält Funktionsbausteine für die serielle Kommunikation der Steuerungen.

Funktionsbaustein	Nr.	Kurzbeschreibung	Steuerungstypen
PORT_OPEN	135	öffnet eine serielle Schnittstelle	XCx ProNumeric ProSyCon MCS xx-xx
PORT_CLOSE	136	schließt eine serielle Schnittstelle	
PORT_READ	137	gibt Zeichen auf einer seriellen Schnittstelle aus	
PORT_WRITE	138	liest Zeichen von einer seriellen Schnittstelle	
PORT_STATE	139	liefert Statusinformation einer seriellen Schnittstelle	

9.4.9 Bibliothek SchleicherLib_Vxxx

Die Bibliothek SchleicherLib_Vxxx enthält Datentypdefinitionen der Firmware, die für MULTIPROG bereitgestellt werden. Funktionsbausteine sind in dieser Bibliothek nicht enthalten.

9.4.10 Bibliothek XCx7_Vxxx

Die Bibliothek XCx7_Vxxx enthält Funktionsbausteine die ausschließlich bei dem Steuerungstyp XCx 700 verwendet werden.

Funktionsbaustein	Nr.	Kurzbeschreibung	Steuerungstyp
UZB_VR	250	Funktionsbaustein für den Betrieb der UZB 2VR Module	XCx 700
UBA_ERR_CTRL	251	Fehlerhandling der UBA-Erweiterungsmodule	
READ_AXIS_PAGE	305	liest einen Parameter aus der einer Achse zugeordneten sogenannten Remotepage.	
WRITE_AXIS_PAGE	306	schreibt einen Parameter in die einer Achse zugeordneten sogenannten Remotepage	
READ_RP	252	lesender Zugriff auf System-U Remotepages	
WRITE_RP	253	schreibender Zugriff auf System-U Remotepages	
IBSM	254	InterBus-S Master (USK-DIM)	

Hinweis: Die Funktionsbausteine READ_RP, WRITE_RP und IBSM gibt es ab der Version XCx7_V001.

9.5 Das SPS-Betriebssystem ProConOS

9.5.1 Die Initialisierungsdatei ProConOS.INI

Mit Hilfe der Datei ProConOS.INI lassen sich erweiterte Einstellungen z.B. der Kommunikationstreiber, der Systemtasks und des CANopen-Stacks applikationsspezifisch anpassen.
Falls ProConOS.INI noch nicht existiert oder gelöscht wurde, wird die Datei initial mit Default-Werten beim Start der Steuerungssoftware angelegt.
Die Datei wird auf dem Compact Flash im Pfad */ata0/OS/PLC/ProConOS.INI* gespeichert.

Beschreibung der Section- und Key- Einträge:

```
[PLC]

; starte SPS User Tasks mit mit hoher Priorität
USR_HIGH_PRIO  = 1      ; yes = 1 (default), no = 0

; verwende den ProConOS Socket Kommunikations- Treiber
PC_SOCKET_DRV  = 1      ; yes = 1 (default), no = 0
; max. Anzahl der ProConOS- Clients
; bei gleichzeitigen Zugriff auf die Steuerung
PC_SOCKET_BLOG = 4      ; default

; Verwende den ProConOS Serial 0/1 Kommunikations- Treiber
PC_SERIAL0_DRV = 0      ; yes = 1, no = 0 (default)
PC_SERIAL0_BR  = 19200  ; Baudrate = 19200 (default)
PC_SERIAL1_DRV = 0      ; yes = 1, no = 0 (default)
PC_SERIAL1_BR  = 19200  ; Baudrate = 19200 (default)


[CNC]

; starte die CNC IPO Task mit hoher Priorität
IPO_HIGH_PRIO = 0 ; yes=1, no=0 (default)


[CAN]

; starte die CANopen Task mit hoher Priorität
CAN_HIGH_PRIO = 0             ; yes = 1, no=0 (default)

; Restart den CANopen Prozess nach PLC STOP (NMT master!)
RESTART_CAN  = 0  ; yes=1, no=0 (default)

; PLC STOP nach CAN heartbeat error
HBE_STOP_PLC = 1  ; yes = 1 (default), no = 0
; PLC STOP nach CAN Bus Off
CBO_STOP_PLC = 0  ; yes = 1 (default), no = 0


[PATHS]

; Ablagepfad für CNC Online Protokollierung
ONLCONFIGPATH = "/ata0/OS/ONL"
; Ablagepfad der NC- Dateien
NCFILESPATH   = "/ata0/OS/CNC"
```

10 Die CNC

10.1 Inbetriebnahme und Programmierung

Die Inbetriebnahme und Programmierung erfolgt mit der Software "Schleicher-Dialog". Siehe dazu das gleichnamige Kapitel in dieser Betriebsanleitung.
Der Schleicher-Dialog ist mit umfangreichen kontext-sensitiven- und Online-Hilfen ausgestattet.
Die Hilfedateien, wie z.B. die Datei Antriebsparameter, können bei Bedarf auch als PDF-Datei bezogen werden.

Der Befehlssatz der CNC-Programmierung ist Inhalt folgender Programmieranleitungen:

CNC-Programmierung XCx und ProNumeric deutsch	R4.322.2080.0 (322 381 61)
CNC Programming XCx and ProNumeric englisch	R4.322.2090.0 (322 381 62)

Sie können kostenlos vom Internet: *http:\\www.schleicher-electronic.com* als PDF-Dateien oder Windows-Hilfedateien geladen werden.

10.2 Der SPS / CNC Koppelspeicher

Für die Inbetriebnahme der CNC ist es unerlässlich die Koppelspeicherstruktur in das Projekt einzufügen.

Der Koppelspeicher besitzt eine umfangreiche Datenstruktur, die zur Kommunikation zwischen SPS, CNC und Visualisierungssystemen verwendet wird.
Außerdem kann über den Koppelspeicher auf Datenbereiche wie z.B. Version Number, Error Page und Log Book zugegriffen werden.

Im Abschnitt "Schnelleinführung" wird die Vorgehensweise beim Einfügen genau beschrieben.

Eine genaue Beschreibung sämtlicher Variablen des Koppelspeichers ist als Online-Hilfe der Software "Schleicher Dialog" verfügbar. Als PDF-Datei kann die Koppelspeicherbeschreibung von der Schleicher Homepage *http://www.schleicher-electronic.com* unter Betriebsanleitungen XCx geladen werden.

10.3 Wichtige Hinweise zum Multi-Task-System und CNC

Für den sicheren Betrieb der CNC ist es wichtig die Task-Hierarchien zu kennen und für die einzelnen Anwender-Tasks zu berücksichtigen.

Für die CNC gilt:

- Die Gesamtlaufzeit der Anwender-Tasks, die mit einer Priorität 0 bis 2 laufen, darf die Hälfte der Laufzeit der Lageregler-Task (Priorität 3) nicht überschreiten.
- Die mit der IPO-Ebene synchron laufende User-Task (Ereignis 6) muss eine Priorität zwischen 7 (IPO-Task) und 3 (Lageregler-Task) besitzen.
- Der CAN-Prozess wird vom System synchron zur IPO-Task aufgerufen, wenn die eingestellte CAN-Zykluszeit gleich dem IPO- oder gleich dem Lagereglertakt ist.
  Die an das Ereignis 5 gebundene User-Task läuft nach Abarbeitung des CAN-Prozesses.

11 SLM-Antriebsmodul XP-SLM

Das SLM-Antriebsmodul XP-SLM stellt ein serielles Hochgeschwindigkeits Interface mit SLM-Protokoll (Speed Loop Motor Protokoll) für den Datentransfer von und zu SLM-Antrieben und Stellern der Fa. Control Techniques Drives Ltd. zur Verfügung.
Über dieses Modul kann die XCN die Positionierung und Lageregelung von bis zu 4 unabhängigen NC-Achsen ausführen. Desweiteren stehen 2 Inkrementalgeber-Eingänge und 2 Interrupt-Messeingänge zur Verfügung.

Das Modul wird für die Erweiterung der XCx 540 eingesetzt. Es besitzt folgende Eigenschaften:

- bis zu 4 Antriebe
- RS 485 Schnittstelle
- SLM-Protokoll

Abbildung 30: SLM-Modul Frontansicht

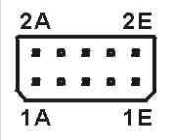
11.1 Anzeigen und Anschlüsse

Anzeigen		
LED	**Farbe**	
L1	rot / grün	Kommunikation Kanal 1 grün dauerhaft: Kommunikation ok rot dauerhaft: Kommunikationsfehler
L2	rot / grün	Kommunikation Kanal 2 grün dauerhaft: Kommunikation ok rot dauerhaft: Kommunikationsfehler
L3	rot / grün	Kommunikation Kanal 3 grün dauerhaft: Kommunikation ok rot dauerhaft: Kommunikationsfehler
L4	rot / grün	Kommunikation Kanal 4 grün dauerhaft: Kommunikation ok rot dauerhaft: Kommunikationsfehler

Anschlüsse		
X11 D1	Antriebe	Typ: har-link®, Buchse, 10-pol. Hersteller Fa. HARTING
X12 D2		
X13 D3		
X14 D4		
X15 E1	Geber / Interrupt	Belegung siehe unten
X16 E2		

11.2 Belegung der Buchsen am Modul (Draufsicht)

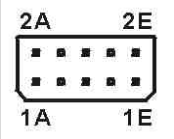
X11 D1 Antriebe



Pin	Bezeichnung	Bemerkung
1A	COM1+	SLM-Protokoll (RS485), Antrieb 1
2A	COM1-	SLM-Protokoll (RS485), Antrieb 1
1B	COM2+	SLM-Protokoll (RS485), Antrieb 2
2B	COM2-	SLM-Protokoll (RS485), Antrieb 2
1C	COM3+	SLM-Protokoll (RS485), Antrieb 3
2C	COM3-	SLM-Protokoll (RS485), Antrieb 3
1D	HWEN1	Reglerfreigabe (Hardware Enable)
2D	nc	not connected
1E	+24V-EXT	externe Spannungseinspeisung
2E	GND-EXT	externe Spannungseinspeisung

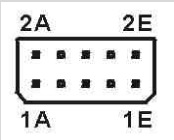
Neben den Anschlüssen COM1 stehen hier auch die Anschlüsse COM2 und COM3 für zwei weitere Antriebe zur Verfügung. Dies ist speziell für MultiAx-Antriebe der Fa. Control Techniques gedacht. Die Reglerfreigabe gilt für alle drei Achsen zusammen.

X12 D2 Antrieb



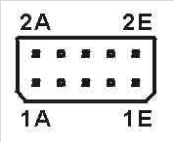
Pin	Bezeichnung	Bemerkung
1A	COM2+	SLM-Protokoll (RS485), Antrieb 2
2A	COM2-	SLM-Protokoll (RS485), Antrieb 2
1B	nc	not connected
2B	nc	not connected
1C	nc	not connected
2C	nc	not connected
1D	HWEN2	Hardware Enable (Reglerfreigabe)
2D	nc	not connected
1E	+24V-EXT	externe Spannungseinspeisung
2E	GND-EXT	externe Spannungseinspeisung

X13 D3 Antrieb



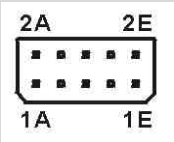
Pin	Bezeichnung	Bemerkung
1A	COM3+	SLM-Protokoll (RS485), Antrieb 3
2A	COM3-	SLM-Protokoll (RS485), Antrieb 3
1B	nc	not connected
2B	nc	not connected
1C	nc	not connected
2C	nc	not connected
1D	HWEN3	Hardware Enable (Reglerfreigabe)
2D	nc	not connected
1E	+24V-EXT	externe Spannungseinspeisung
2E	GND-EXT	externe Spannungseinspeisung

X14 D4 Antrieb



Pin	Bezeichnung	Bemerkung
1A	COM4+	SLM-Protokoll (RS485), Antrieb 4
2A	COM4-	SLM-Protokoll (RS485), Antrieb 4
1B	nc	not connected
2B	nc	not connected
1C	nc	not connected
2C	nc	not connected
1D	HWEN4	Hardware Enable (Reglerfreigabe)
2D	nc	not connected
1E	+24V-EXT	externe Spannungseinspeisung
2E	GND-EXT	externe Spannungseinspeisung

X15 E1 und X16 E2 Inkrementalgeber / Interrupt-Messeingänge



Pin	Bezeichnung	Bemerkung
1A	A	Geber-Eingang A
2A	/A	Geber-Eingang /A
1B	B	Geber-Eingang B
2B	/B	Geber-Eingang /B
1C	NULL	Geber-Eingang NULL
2C	/NULL	Geber-Eingang /NULL
1D	nc	not connected
2D	GND-EXT	Ground, extern
1E	INT+	Interrupt-Eingang+
2E	INT-	Interrupt-Eingang-

11.3 Technische Daten SLM Antriebsmodul XP-SLM

Grunddaten	
Artikel-Nr.	R4.503.0040.0 (503 383 32-M)
Anzahl der steuerbaren NC-Achsen	bis zu 4 Achsen

Elektrische Daten	
Leistungsaufnahme der internen Spannungsversorgung	typ. 2 W
Galvanische Trennung	ja, mittels Optokoppler
Versorgungsspannung extern	+24V-EXT, GND-EXT
Spannung	DC 24 V +/-20% max. 5% Restwelligkeit
Leistungsaufnahme	1,3 W
Serielle SLM-Schnittstelle	COM1+, COM1-; COM2+, COM2-; COM3+, COM3-; COM4+, COM4-;
Anzahl	4
Physik	RS485; 2,5 MBaud
Protokoll	durch ASIC CT2239-003 der Fa. Control Techniques
Hardware Enable (Reglerfreigabe)	HWEN1; HWEN2; HWEN3; HWEN4
Anzahl	4
Schaltpegel	H-Pegel ≥ +24V-EXT - 0,5 V L-Pegel ≤ 1 V
Ausgangsstrom	0,5 A
Signalverzögerung	<300 µs (Hardware)
Geber-Eingänge	A, /A, B, /B, NULL, /NULL
Anzahl	2
Physik	RS422
max. Eingangs-Frequenz	2 MHz
Interrupt-Eingänge	INT+, INT-
Anzahl	2
Schaltpegel	H-Pegel = +11 V bis +30 V L-Pegel = -30 V bis +5 V
Eingangsstrom	min. H-Pegel (+11 V) ≥ 2 mA max. L-Pegel (+5 V) ≤ 2 mA typ. (+24 V) 8 mA max. (+30 V) ≤ 15 mA
Signalverzögerung	<100 ns (Hardware)
Triggerung	Flanken-Triggerung

Zubehör	
XP-SLM-K1-3,5	Verbindungskabel 3,5m von XP-SLM-Modul nach Antrieb (10-adrig, einseitig Harting-Stecker) Artikel-Nr.: R4.506.0010.0 (506 383 85)

11.4 Schnittstellen

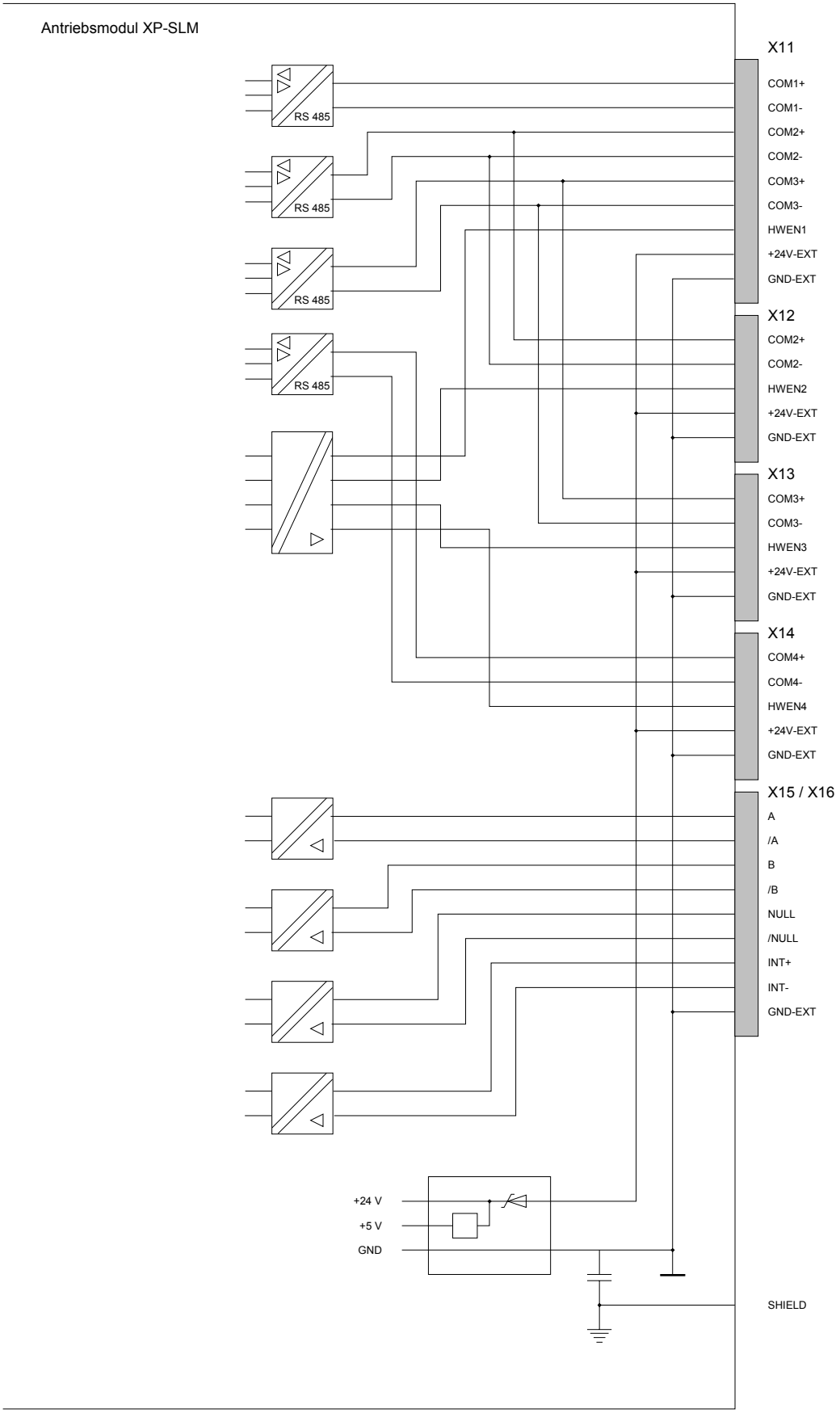


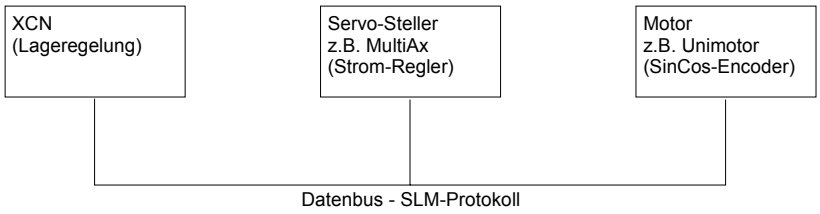
Abbildung 31: Schnittstellen XP-SLM

Externe Spannungsversorgung

Die externe Spannungsversorgung kann durch ein 24V-Netzteil oder durch den Antrieb (z.B. MultiAx der Fa. Control Techniques) eingespeist werden. Siehe dazu auch Kapitel 11.6 Verdrahtungs-Beispiel mit MultiAx.

Serielle SLM-Schnittstelle

Über eine SLM-Schnittstelle werden immer 3 Teilnehmer miteinander verbunden:
- Motion-Contoller (XCN mit XP-SLM)
- Umrichter (z.B. MultiAx der Fa. Control Techniques)
- Motor (z.B. Unimotor Fa. Control Techniques)



Hardware Enable (Reglerfreigabe)

Geber-Eingänge

An die Geber-Eingänge können optional zwei Handräder angeschlossen werden.
(Derzeitig sind die Eingänge nur hardwaremäßig vorhanden, eine SW-Schnittstelle ist in Vorbereitung.)

Interrupt-Eingänge

An die Interrupt-Eingänge können Messtaster angeschlossen werden.

Es gibt 2 Interrupt-Eingänge auf der Karte.
(Derzeitig wird softwareseitig nur der Interrupt-Eingang von Stecker X15 ausgewertet.)

Für Messfunktionen sind folgende $-Funktionen und Koppelspeicher-Variablen zu beachten:

$-Funktionen	
$53	Abbruch der Fahrbewegung mit Kompensation des Schleppabstandes durch Interruptsignal
$54	Restweg löschen durch Interruptsignal

Koppelspeicher		
cncMem.axSect[n].flgN2P.bRapAxStpAct	Rapid Axis Stop Active	$53 Messzyklus ist aktiv
cncMem.axSect[n].flgN2P.bRapMeasAct	Rapid Measurement Active	$54 Messzyklus ist aktiv
cncMem.axSect[n].flgN2P.bMeasValOk		Meßwert ist gültig
cncMem.axSect[n].wrdN2P.lCurMeasPos	Current Measurement Position	aktuelle Position des Antriebs zum Messzeitpunkt ($53 / $54)

11.5 Applikationsbeispiel

An eine XCN mit dem SLM Antriebsmodul XP-SLM sollen ein Steller MultiAx und zwei Motoren Unimotor der Fa. Control Techniques angeschlossen werden. Die Antriebe werden steuerungsseitig über die Stecker X1 und X2 angeschlossen.

Das Diagramm zeigt die prinzipielle Verdrahtung der Antriebe (ohne den Leistungsteil).

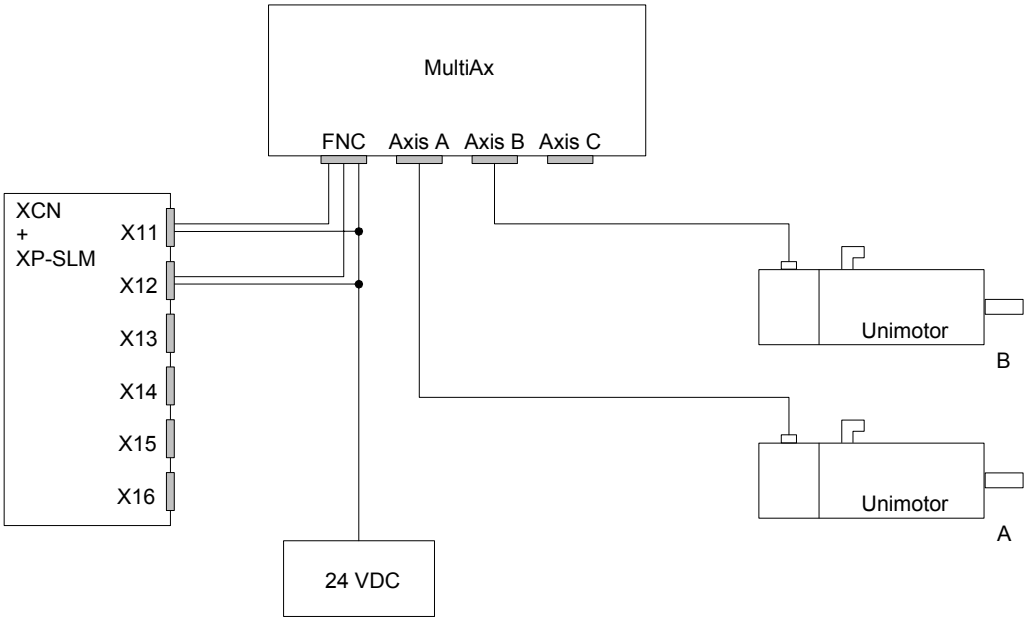


Abbildung 32: Applikationsbeispiel XP-SLM

11.6 Verdrahtungs-Beispiel mit MultiAx

SLM Antriebsmodul XP-SLM an MultiAx (15 pol. High Density D-Sub)

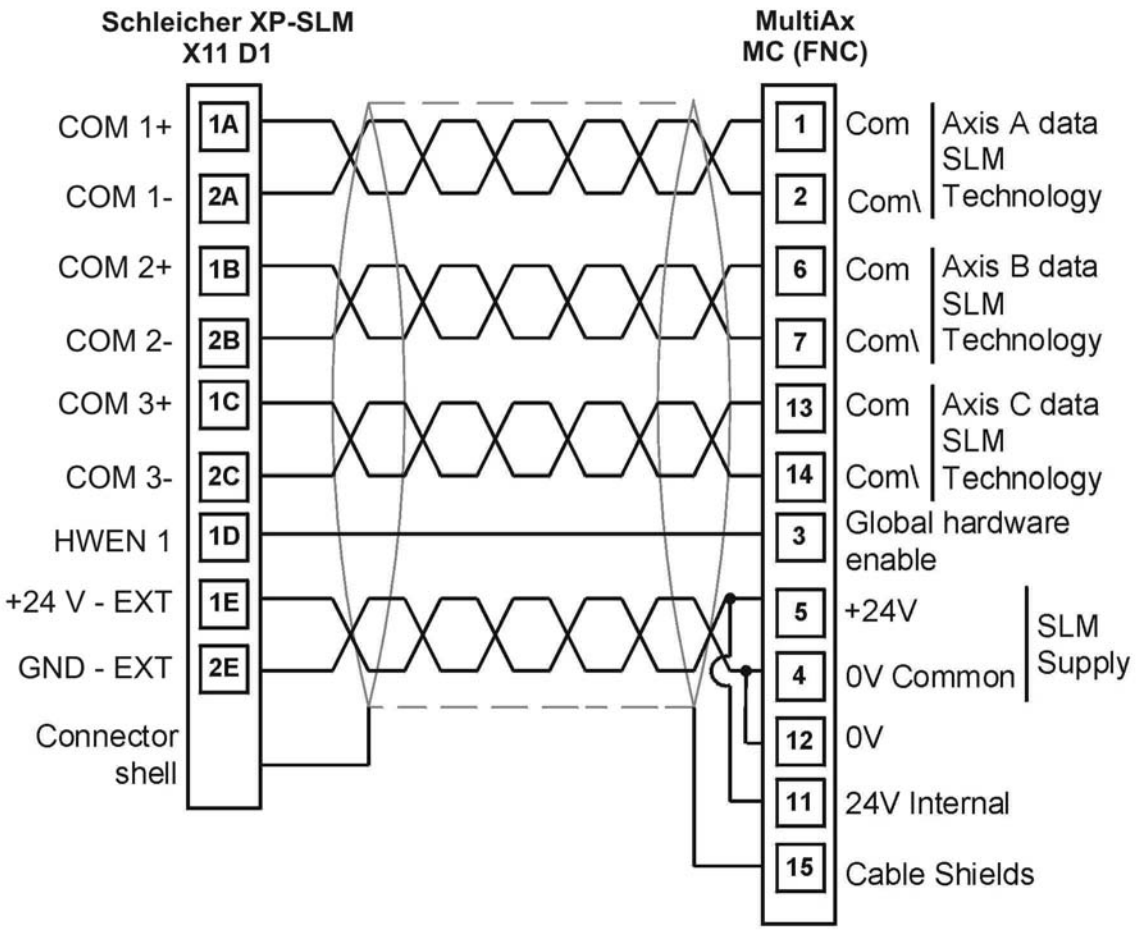


Abbildung 33: Verdrahtung XP-SLM an MultiAx

Diese Art der Verdrahtung wird nur empfohlen, wenn die Modifikation mit der elektronischen Sicherung in MultiAx implementiert ist.

Ohne diese Schutzmaßnahme ist diese Verdrahtung fehlerhaft und kann die Spannungsversorgung von MultiAx beschädigen.

Eine Verdrahtung ohne diese Modifikation läßt die Gewährleistung für den Antrieb hinfällig werden.

Hinweise zu MultiAx

Werden bei ausgeschalteter Spannungsversorgung (AC) des Antriebs Positions-Daten benötigt, so muß eine externe 24V Spannungsversorgung an MualtiAx (MC (FNC) Pin 5 und 4) angeschlossen werden.

Steckerbelegung des Anschlusse MC (FNC):

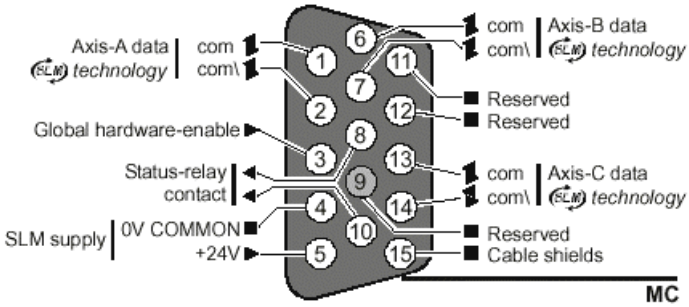


Abbildung 34: Steckerbelegung XP-SLM MC (FNC)

11.7 Verdrahtungs-Beispiel mit M'Ax

SLM Antriebsmodul XP-SLM an M'Ax (15 pol. High Density D-Sub)

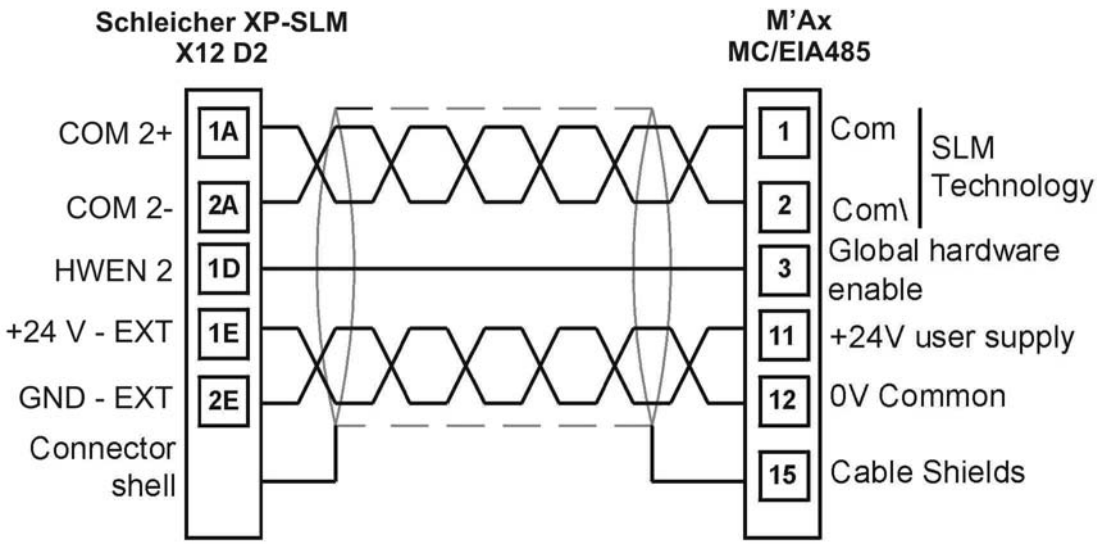


Abbildung 35: Verdrahtung XP-SLM an M'Ax

SLM Antriebsmodul XP-SLM an M'Ax (RJ45)

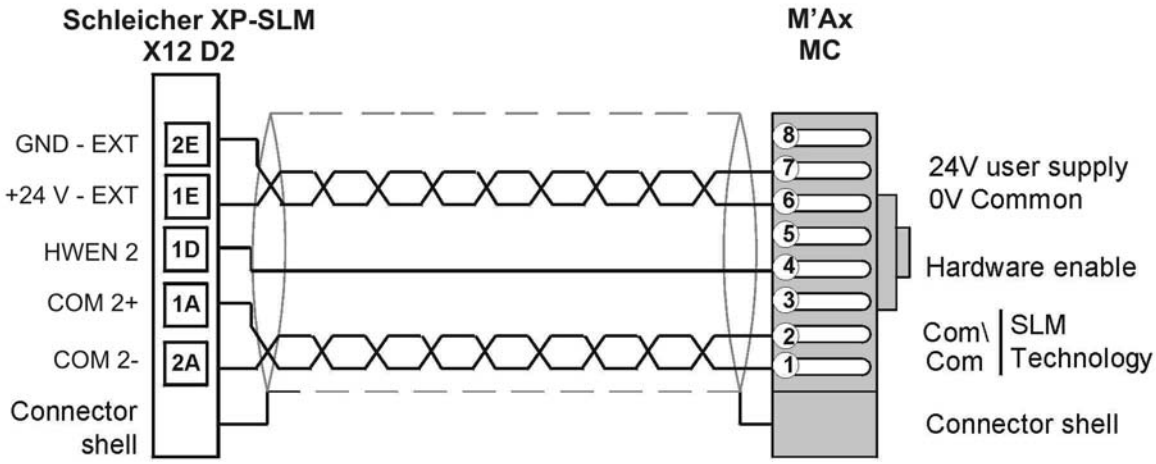


Abbildung 36: Verdrahtung XP-SLM an M'Ax RJ45

Hinweise zu M'ax

Werden bei ausgeschalteter Spannungsversorgung (AC) des Antriebs Positions-Daten benötigt, so muß eine externe 24 V Spannungsversorgung an M'Ax (MC/EIA485 Pin 5 und 4) angeschlossen werden.

Steckerbelegung des Anschlusses MC/EIA485:

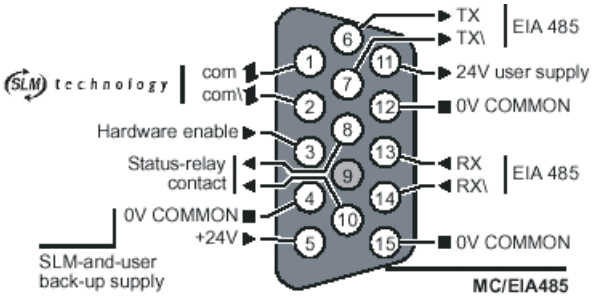


Abbildung 37: Steckerbelegung XP-SLM MC/EIA485

Steckerbelegung des Anschlusses MC/EIA485:

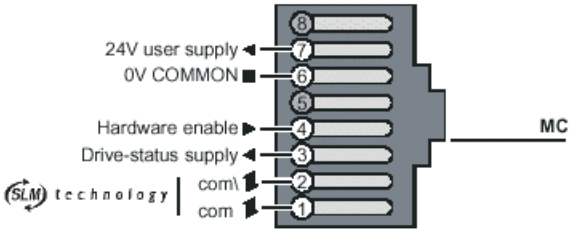


Abbildung 38: Steckerbelegung XP-SLM MC/EIA485

12 SERCOS-Modul XP-SRC

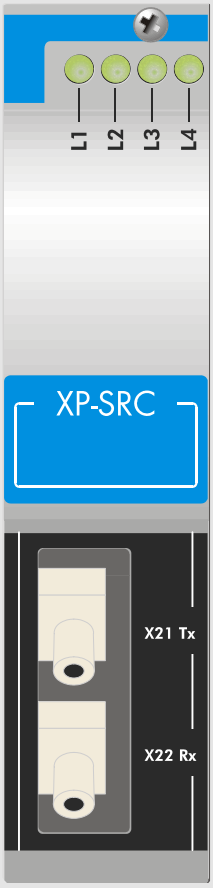
Das SERCOS-Antriebsmodul XP-SRC stellt einen SERCOS-Ring zur Verfügung, an dem bis zu 8 unabhängige SERCOS-Antriebe über einen Lichtwellenleiter betrieben werden können.

In der Steuerung können maximal zwei Antriebsmodule XP-SRC Verwendung finden.

- SERCOS-Master nach IEC1491
- 8 Achsen
- Lichtwellenleiteranschluß nach IEC 874-2

Abbildung 39: SERCOS-Modul Frontansicht

12.1 Anzeigen und Anschlüsse



Anzeigen			
	Bezeichnung	Farbe	
L1	SERCOS PHASE 1	grün/rot	SERCOS Hochlaufphase: grün an - Phase 1 durchlaufen rot an - Phase 1 nicht durchlaufen
L2	SERCOS PHASE 2	grün/rot	SERCOS Hochlaufphase: grün an - Phase 2 durchlaufen rot an - Phase 2 nicht durchlaufen
L3	SERCOS PHASE 3	grün/rot	SERCOS Hochlaufphase: grün an - Phase 3 durchlaufen rot an - Phase 3 nicht durchlaufen
L4	SERCOS PHASE 4	grün/rot	SERCOS Hochlaufphase: grün an - Phase 4 durchlaufen rot an - Phase 4 nicht durchlaufen
			alle LED's rot: Modul in Phase 0
			alle LED's aus: keine Achse zugeordnet
Anschlüsse			
X21	Tx	SERCOS-Ring Ausgang	F-SMA Schraubverbindung für Lichtwellenleiter
X22	Rx	SERCOS-Ring Eingang	

12.2 Technische Daten

Grunddaten	
Artikel-Nr.	R4.503.0020.0 (503 383 31-M)
Anzahl der steuerbaren NC-Achsen	bis zu 8 Achsen (1 SERCOS-Ring)

SERCOS-Schnittstelle	
Interface	SERCOS Master nach IEC 1491
Anzahl	1 SERCOS-Ring
Physik	Kunststoff-Lichtwellenleiter, 2 / 4 / 8 / 16 MBaud
Controller	SERCON 816
Zykluszeit	1 ms ... 30 ms

Anschlußtechnik	
SERCOS-Lichtwellenleiteranschluß IN / OUT	F-SMA Schraubverbindungen nach IEC 874-2

12.3 Applikationsbeispiel

An eine XCN mit dem SERCOS Antriebsmodul XP-SRC können bis zu 8 SERCOS-Antriebe angeschlossen werden. Dabei ist das Modul XP-SRC der Master und die Antriebe sind die Slaves. Die Verdrahtung der Lichtwellenleiter erfolgt ringförmig, d.h. es wird jeweils ein Ausgang mit einem Eingang verbunden.

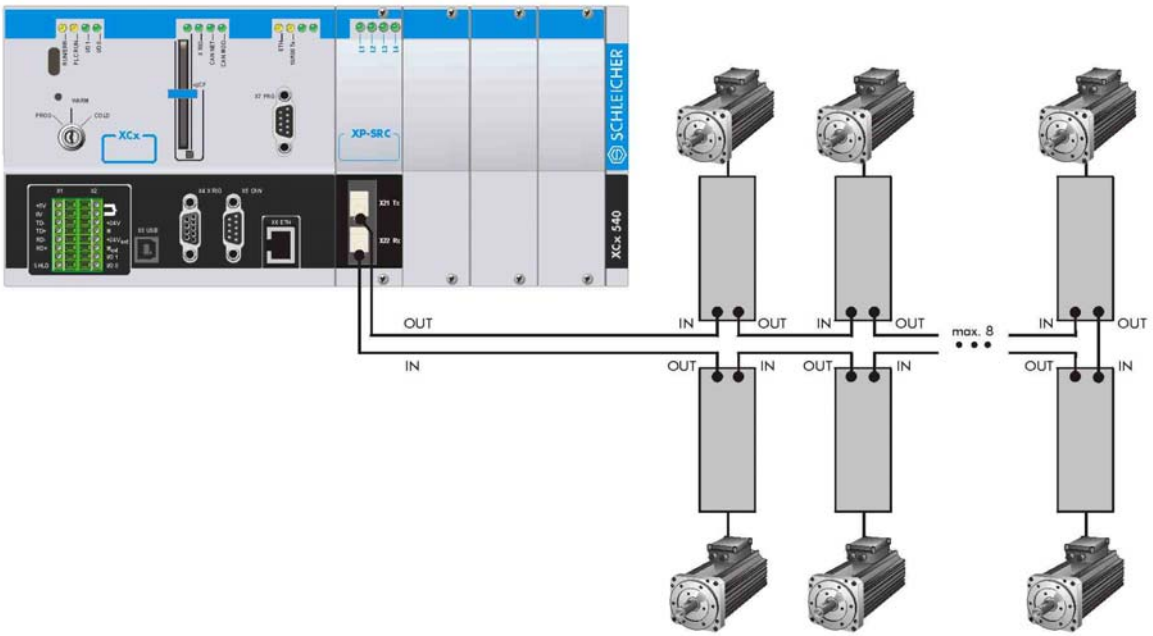


Abbildung 40: SERCOS-Modul Applikationsbeispiel

12.4 Kommunikationsaufbau über den SERCOS-Ring

Der Kommunikationsaufbau über den SERCOS-Ring erfolgt in 5 Phasen (Phase 0..4).

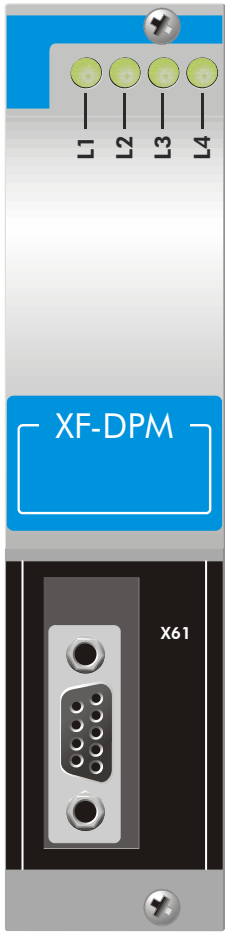
Der Kommunikationsaufbau wird immer dann neu gestartet, wenn

- die Steuerung neu eingeschaltet wird,
- der Taster Reset (an der XCN) betätigt wird,
- die Antriebsparameter geändert wurden oder
- ein Fehler (z.B. Kabelbruch) am SERCOS-Ring festgestellt wird.

Bis zum Erreichen der Kommunikationsphase 4 können mehrere Sekunden vergehen. Der Kommunikationsaufbau kann an den LED-Anzeigen verfolgt werden.

13 PROFIBUS-DP Module

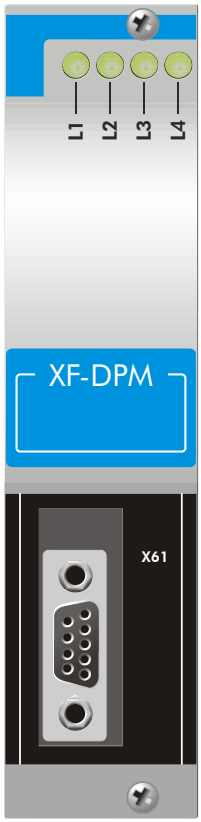
13.1 PROFIBUS-DP Master XF-DPM



- Eine PROFIBUS-DP Master Feldbusschnittstelle
- entsprechend IEC 61158 Typ 3
- bis 12 MBaud Datenübertragungsrate

Abbildung 41: PROFIBUS-DP Master Frontansicht

13.1.1 Anzeigen und Anschlüsse XF-DPM

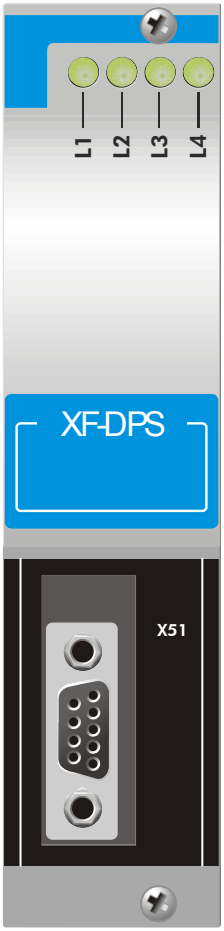


Anzeigen			
LED	Bez.	Farbe / Zustand	Beschreibung
L1	BF	rot	keine Busverbindung (bus fail)
		grün	Busverbindung aktiv
		gelb zyklisch blinkend	Busverbindung aktiv, jedoch kein Prozessdatenaustausch
L2			nicht benutzt
L3	SYS	gelb azyklisch blinkend	Hardware-Fehler
		grün zyklisch blinkend	bereit für Busverbindung, jedoch (noch) keine Busverbindung aktiv
		grün azyklisch blinkend	Fehler in der Busverbindung
		grün	Busverbindung aktiv
L4	COM	gelb	zyklischer Datenaustausch aktiv
		rot	Fehler in der Busverbindung
Anschluss X61			
Pin		Bezeichnung	
1		nc	nicht angeschlossen
2		nc	nicht angeschlossen
3		RxD/TxD-P	Receive/Transmit Data P
4		CNTR-P	Repeater Control Signal
5		DGND	Data Ground (0 V)
6		VP	Spannung Plus (+5 V)
7		nc	nicht angeschlossen
8		RxD/TxD-N	Receive/Transmit Data N
9		nc	nicht angeschlossen

13.1.2 Technische Daten XF-DPM

Grunddaten	
Schnittstelle	PROFIBUS-DP Master
Datenübertragungsrate	bis 12 MBaud
Anschlusstechnik	D-Sub, 9-polig, Buchse
Betriebsumgebungstemperatur	0°C bis 45°C

13.2 PROFIBUS-DP Slave XF-DPS



- Eine PROFIBUS-DP Slave Feldbusschnittstelle
- entsprechend IEC 61158 Typ 3
- bis 12 MBaud Datenübertragungsrate

Abbildung 42: PROFIBUS-DP Slave Frontansicht

13.2.1 Anzeigen und Anschlüsse XF-DPS

Anzeigen			
LED	Bez.	Farbe / Zustand	Beschreibung
L1	BF	rot	keine Busverbindung (bus fail), Bruch des Feldbuskabels oder Master betreibt den Bus nicht (mehr)
		grün	Busverbindung aktiv
		gelb zyklisch blinkend	Busverbindung aktiv, jedoch kein Prozessdatenaustausch
L2	DIA	gelb zyklisch blinkend	statische Diagnose aktiv
L3	SYS	gelb azyklisch blinkend	Hardware-Fehler
		grün zyklisch blinkend	bereit für Busverbindung, jedoch (noch) keine Busverbindung aktiv
		grün azyklisch blinkend	Fehler in der Busverbindung
		grün	Busverbindung aktiv
L4	COM	gelb	zyklischer Datenaustausch aktiv
Anschluss X51			
Pin		Bezeichnung	
1		nc	nicht angeschlossen
2		nc	nicht angeschlossen
3		RxD/TxD-P	Receive/Transmit Data P
4		CNTR-P	Repeater Control Signal
5		DGND	Data Ground (0 V)
6		VP	Spannung Plus (+5 V)
7		nc	nicht angeschlossen
8		RxD/TxD-N	Receive/Transmit Data N
9		nc	nicht angeschlossen

13.2.2 Technische Daten XF-DPS

Grunddaten	
Schnittstelle	PROFIBUS-DP Slave
Datenübertragungsrate	bis 12 MBaud
Anschlusstechnik	D-Sub, 9-polig, Buchse
Betriebsumgebungstemperatur	0°C bis 45°C

13.3 Die PROFIBUS-DP Konfigurationsdaten

Sektion DPxx in der Datei ProConOS.INI

```
; DP parameter Definitionen:
 [DP00]                  ; e.g. card# 0
; any description
DESCRIPTION     = Profibus DPS #0 (XCx5 series)
BUS_ADDR        = 2      ; 0..126
MASTER_FCONF    = 1      ; the master force slaves configuration
                         ; 1 = yes (default)/ 0 = no
```

Der ProConOS.ini- Eintrag MASTER_FCONF steuert das Verhalten hinsichtlich der DP Konfigurationsdaten.

0: die vom Master gesendeten Konfigurationsdaten werden mit den Einstellungen der DPSconf0x.ini verglichen

1: der Master überschreibt etwaige Konfigurationsdaten (gesendete Daten werden immer als gültig akzeptiert)

Hinweis:
Die aktuelle XDP- Konfiguration ist der Datei Sysinfo.txt (/ram0/OS/LOG/Sysinfo.txt) zu entnehmen.

Standard Module Konfiguration

DP Slave: IO Bereich: 0..72

Nr.	Offset	Anzahl	Byte /Word	Input / Output	Consistency
0	0	4	B	I	-
1	0	4	B	O	-
2	4	4	B	I	-
3	4	4	B	O	-
4	8	16	B	I	c
5	8	16	B	O	c
6	24	16	B	I	c
7	24	16	B	O	c
8	40	16	W	I	c
9	40	16	W	O	c

DP Slave-Konfiguration

Die Modul Struktur Definition der DP Slave-Konfiguration erfolgt nach Code:

modType	modLen
0 - input byte without consistency 1 - input word without consistency 2 - output byte without consistency 3 - output word without consistency 4 - input byte with consistency 5 - input word with consistency 6 - output byte with consistency 7 - output word with consistency 8 - dummy place	0 - 1 Byte/Word 1 - 2 Bytes/Words 2 - 3 Bytes/Words 3 - 4 Bytes/Words 4 - 8 Bytes/Words 5 - 12 Bytes/Words 6 - 16 Bytes/Words 2 - 20 Bytes/Words 8 - 32 Bytes/Words 9 - 64 Bytes/Words

Beispiel der DPSconf0x.ini (default)

```
[DP00]
NumberOfModules = 10

[mod00]
; 4 bytes input
modType = 0
modLen  = 3

[mod01]
; 4 bytes output
modType = 2
modLen  = 3

[mod02]
; 4 bytes input
modType = 0
modLen  = 3

[mod03]
; 4 bytes output
modType = 2
modLen  = 3

[mod04]
; 16 bytes input with consistence
modType = 4
modLen  = 6

[mod05]
; 16 bytes output with consistence
modType = 6
modLen  = 6

[mod06]
; 16 bytes input with consistence
modType = 4
modLen  = 6

[mod07]
; 16 bytes output with consistence
modType = 6
modLen  = 6

[mod08]
; 16 words input with consistence
```

```
modType = 5
modLen  = 6

[mod09]
; 16 words output with consistence
modType = 7
modLen  = 6
```

14 Analog-Achsen mit MC_ANALOG und RIO A10-10 - ein Beispiel

An einem Beispiel soll der Firmware-Funktionsbaustein MC_ANALOG und das Achsinterface RIO A10-10 vorgestellt und ihre Funktionalität beschrieben werden. Der Anwender soll mit geringstem Aufwand analoge Achsen projektieren können. Alle nachfolgende Angaben beziehen sich auf das Beispiel *XCN_McRioA10_10.zwt*, dass über den Technischen Service zur Verfügung gestellt werden kann.

Grundsätzlich werden nur sehr wenige Signale des Achsinterfaces zur Steuerung einer analogen Achse durch die CNC benötigt. Im Beispiel wird aber der Zugriff auf alle Signale ermöglicht um ein Einarbeiten in die Thematik zu erleichtern.

Weitere Informationen zu diesem Thema sind den folgenden Dokumentationen zu entnehmen:

- Online-Hilfe der Bibliothek PLC_Vxxx, Kapitel "Funktionsbaustein MC_ANALOG"
- Betriebsanleitung "RIO Erweiterungsmodule", Kapitel "Achsinterface RIO A10-10"

14.1 Beispiel / Voraussetzung

Steuerung:	XCN 500	
RIO:	RIO EC X2, RIO A10-10, RIO A10-10, RIO 16I, RIO 16O	
Achsen:	CNC-Achse 1, in Simulation	
	CNC-Achse 2, MC-ANALOG-Achse 1, 1. RIO A10-10 Modul	
	CNC-Achse 3, MC-ANALOG-Achse 2, 2. RIO A10-10 Modul	
Programm:	Datentyp:	McRioA10_10
	POE:	AnalogAxis, McRioA10_10, CncSync (Arbeitsblatt RioA10_10)
	Globale Variablen:	Gruppe AnalogAxis
	belegte Adressen:	AT %MW0 (zum Dekodieren und Kodieren von WORD-Daten - für den einfacheren Bit-Zugriff auf Wort-Variable)

14.2 Schritt 1: XRIO-Konfiguration

Das Modul RIO A10-10 kann an einem beliebigen Platz in der Modulreihe und in beliebiger Kombination angeschlossen werden.

XRIO ist an die Motion-Control-Task (*McSync*) zu binden (in ProdocPlus: Menü *Extras-XRIO/Konfiguration*). Der dabei auftretende Hinweis: "...muss an zyklische Task gebunden werden..." ist zu ignorieren. Die aktuellen XRIO Variablen liegen zum Vergleich in der Gruppe XRIO. Die neu eingelesenen Variablen werden in der Gruppe XRIO-Variables abgelegt.

14.3 Schritt 2: Achsinterface RIO A10-10

Minimal ist die Versorgungsspannung (X2.21/X3.21), der Geber, der analoge Sollwert (X2.7/X2.22) und die Betriebsbereitschaft des Servos (Ready X1.1) anzuschließen. Im Beispiel wurden zusätzlich die Endlagenschalter als Öffner (X1.2, X2.2) und bei der Y- Achse der Referenzpunktschalter als Schließer (X2.1) aufgelegt.

Der Endschalter Plus der Z-Achse wird im Beispiel zusätzlich als Referenzpunktschalter ausgewertet. Er ist am Modul nicht aufgelegt.

14.4 Schritt 3: Lageregel- und IPO Abtastrate

Der Datenaustausch mit dem Achsinterface RIO A10-10 ist abhängig von der Größe der XRIO-Konfiguration.

Der Antriebsparameter CYCLE_TIME sollte mindestens doppelt so groß wie die - unter *ata0\OS\PLC\Sysconf.txt*, Abschnitt [XRIO], Schlüsselwort ETT01 = 660 µs - für den Datenaustausch XRIO eingetragene Zeit sein (Default-Wert von CYCLE_TIME = 2000).

Die Abtastrate IPO (im Schleicher-Dialog unter *Inbetriebnahme/CNC-Grundeinstellungen*) muss ein ganzzahliges Vielfaches des Lageregeltaktes sein.

Bei z.B. 8 analogen Achsen ist die CYCLE_TIME = 6000, wenn keine weiteren XRIO-Module verwendet werden. Daraus ergibt sich eine Abtastrate IPO = 6000.

14.5 Schritt 4:Task-Struktur

Steuerungsintern läuft die Motion-Control-Task (Lageregel-Task) mit Priorität 3 und die IPO-Task mit Priorität 7. Die CAN-Task läuft in Abhängigkeit des Parameters CAN_HIGH_PRIO (*ProConOS.ini*) intern mit der Priorität 9 (CAN_HIGH_PRIO=0, default) bzw. mit der Priorität 3 (CAN_HIGH_PRIO=1).

Daraus ergibt sich für die SPS-Task:

Lageregler:	McSync, Ereignis 4, Priorität 2
IPO:	CncSync, Ereignis 6, Priorität 6
CAN (CAN_HIGH_PRIO=0):	CAN, Ereignis 5, Priorität 10
CAN (CAN_HIGH_PRIO=1):	CAN, Ereignis 5, Priorität 4
XRIO:	RIO, zyklisch, Priorität 11

14.6 Schritt 5: RIO A10-10 und MC_ANALOG

1. Initialisieren Sie in den Globalen Variablen in der Gruppe AnalogAxis den ersten Achsnamen (hier Y) mit der (Analog-Achs-) Nummer eins. Die Werte der folgenden Achsnamen werden jeweils um eins erhöht. Die Nummern dürfen keine Lücken aufweisen.

In der Variablen *NumberOfAnalogAxis* wird die Anzahl der analogen Achsen, die ausgewertet werden sollen, angegeben.

```
VAR_GLOBAL
  NumberOfAnalogAxis : INT := 2;
  Y                  : INT := 1;
  Z                  : INT := 2
END_VAR
```

2. Die Daten aller möglichen acht Achsen werden in einer Struktur (TypRioA10_10) abgebildet. Sie verbinden die "Roh"- Eingangsdaten des RIO A10-10 in der POE AnalogAxis mit dem Anwender-Funktionsbaustein: McRioA10_10:

```
(* Input RIO A10-10 to PLC
   ------------------------------------------------------------------------- *)
(* Analog Axis 1 *)
RioA10_10[Y].Rio2Plc.iw1_ActPosMSW := xrio1IW0; (* n   *)
RioA10_10[Y].Rio2Plc.iw2_ActPosLSW := xrio1IW2; (* n+2 *)
RioA10_10[Y].Rio2Plc.iw3_Input     := xrio1IW4; (* n+4 *)
RioA10_10[Y].Rio2Plc.iw4_Message   := xrio1IW8; (* n+8 *)

(* Analog Axis 2 *)
RioA10_10[Z].Rio2Plc.iw1_ActPosMSW := xrio1IW12;
RioA10_10[Z].Rio2Plc.iw2_ActPosLSW := xrio1IW14;
RioA10_10[Z].Rio2Plc.iw3_Input     := xrio1IW16;
RioA10_10[Z].Rio2Plc.iw4_Message   := xrio1IW20;
```

3. Schaltet die Freigabe der Achse: EnableOperation (im Beispiel mit TRUE deklariert):

```
(* Function MC_ANALOG
   ------------------------------------------------------------------------- *)
(* Enable Operation Axis
   Beispiel: allgemeine Freigabe einzelner Achsen *)
RioA10_10[Y].Status.EnableOperation := EnableOperation;
RioA10_10[Z].Status.EnableOperation := EnableOperation;
```

4. Aufruf des Funktionsbausteines McRioA10_10 (dieser Anwender-Funktionsbaustein ruft den Firmware- Funktionsbaustein MC_ANALOG auf):

```
(* Function *)
McRioA10_10(NumberOfAxis := NumberOfAnalogAxis);

SumError := McRioA10_10.SumError;
```

5. Verbinden Sie Ausgangsdaten mit dem RIO A10-10:

```
(* Output RIO PLC to A10-10
   ------------------------------------------------------------------------- *)
```

```
(* Analog Axis 1 *)
(* n+2 *) xrio1QW2 := RioA10_10[Y].Plc2Rio.qw2_AxFeed;
(* n+4 *) xrio1QW4 := RioA10_10[Y].Plc2Rio.qw3_Output;
(* n+6 *) xrio1QW6 := RioA10_10[Y].Plc2Rio.qw4_Config;

(* Analog Axis 2 *)
xrio1QW10 := RioA10_10[Z].Plc2Rio.qw2_AxFeed;
xrio1QW12 := RioA10_10[Z].Plc2Rio.qw3_Output;
xrio1QW14 := RioA10_10[Z].Plc2Rio.qw4_Config;

RETURN;
```

14.7 Datentyp TypRioA10_10

Alle Istwerte, Zustände, Meldungen, Sollwerte und Konfigurationen des RIO A10_10 und der dazugehörenden Status- und Control-Variablen (STATUS_WORD und CONTROL_WORD) werden in dieser Struktur als einzelne Variable für alle möglichen 8 Achsen abgebildet.

```
TypRioA10_10 -+                                      Allgemein
   [1..8]     +- CurPos                                  Geber- Istwert
              +- AxFeed                                  Drehzahlsollwert
              |
              +- In --+                               Istwerte
              |       +- Enable                          Reglerfreigabe
              |       +- IO                              freier Ein-/ Ausgang
              |       +- Ready                           Betriebsbereitschaft
              |       +- RefPCAM                         Referenzpunktnocken
              |       +- LimSwP                          Endlagenschalter Plus
              |       +- LimSwM                          Endlagenschalter Minus
              |       +- ZeroImpEnc                      Nullimpuls Geber
              |       +- Togelbit                        Rückmeldung
              |       |
              |       +- Message ---+                 Meldungen
              |                     +- ErrMeasCirc       Kabelbruch
              |                     +- ZeroImpEnc        Nullimpuls Geber
              |                     +- StopReady         Stopp BB fehlt
              |                     +- StopLimSwP        Stopp Endschalter Plus
              |                     +- StopLimSwM        Stopp Endschalter Minus
              |                     +- StopErrMeasCirc   Stopp Kabelbruch
              |
              +- Out -+                               Sollwerte
              |       +- Enable                          Reglerfreigabe
              |       +- IO                              freier Ein-/ Ausgang
              |       +- Toggelbit                       Rückmeldung
              |       |
              |       +- Config ----+                 Konfiguration Geber
              |                     +- AbsPosEnc         Absolutwertgeber
              |                     +- EnableResetPos    Freigabe Istwert loeschen
              |                     +- AbsEncBinaer      Abs.Geber-Code binaer
              |                     +- EncInverting      Geber invertiert
              |                     +- AbsEncData0       Abs.Geber Daten 0
              |                     +- AbsEncData1       Abs.Geber Daten 1
              |                     +- AbsEncClock0      Abs.Geber Takt 0
              |                     +- AbsEncClock1      Abs.Geber Takt 1
              |                     +- ResetPos          Istwert loeschen
              |                     +- ReadyOn           Test BB ein
              |                     +- LimSwPOn          Test
              |                     |                    Endlagenschalter Plus
              |                     +- LimSwMOn          Test
              |                     |                    Endlagenschalter Minus
              |                     +- ErrMeasCircOn     Test Kabelbruch
              |
              +- Status  -+                           STATUS_WORD
              |           +- ReadyToSwitchOn             Einschaltbereit
              |           +- SwitchedOn                  Eingeschaltet
              |           +- EnableOperation             Betrieb freigegeben
              |           +- Fault                       Fehler im Antrieb
              |           +- HomingState0                Statusbit 1 Referenz.
```

```
|           +- HomingState1                 Statusbit 2 Referenz.
|
+- Control -+                            CONTROL_WORD
|           +- SwitchOn                     Einschalten
|           +- DisableVoltage               Spannung sperren
|           +- QuickStop                    Schnellhalt
|           |                               (z.Zt. nicht aktiv)
|           +- OperationEnable              Betrieb freigegeben
|           +- HomingStart                  Start Referenzfahrt
|           +- FaultReset                   Reset Stoerung
|           +- Stop                         Halt (z.Zt. nicht aktiv)
|
+- Rio2Plc--+                            Eingangsdaten vom RIO Modul
|           +- iw1_ActPosMSW                Geber-Istwert MSW
|           +- iw2_ActPosLSW                Geber-Istwert LSW
|           +- iw3_Input                    Eingaenge, Meldungen
|           +- iw4_Message                  Meldungen, Ueberwachungen
|
+- Plc2Rio--+                            Ausgangsdaten zum RIO Modul
            +- qw2_AxFeed                   Drehzahlsollwert
            +- qw3_Output                   Ausgaenge, Konfiguration
            +- qw4_Config                   Konfiguration
```

14.8 Schritt 6: RIO A10-10 und CNC

In der POE CncSync Arbeitsblatt RioA10_10 werden die Endlagen- und Referenspunktschalter an die CNC-Koppelebene übergeben.

```
(* Achsinterface RIO A10-10 Axis 1
   ------------------------------------------------------------------------- *)
(* Beispiel 1:
   Endlagenschalter (Oeffner) & Referenzpunktschalter (Schliesser) *)
cncMem.axSect[2].flgP2N.bLimSwM  := RioA10_10[Y].In.LimSwM;
cncMem.axSect[2].flgP2N.bLimSwP  := RioA10_10[Y].In.LimSwP;
cncMem.axSect[2].flgP2N.bRefPCam := RioA10_10[Y].In.RefPCam;


(* Achsinterface RIO A10-10 Axis 1
   ------------------------------------------------------------------------- *)
(* Beispiel 2:
   Endlagenschalter (Oeffner) & Endlage-Plus ist Referenzpunktschalter *)
cncMem.axSect[3].flgP2N.bLimSwP  := cncMem.sysSect[3].flgN2P.bResNcRef
                                    OR RioA10_10[Z].In.LimSwP;
cncMem.axSect[3].flgP2N.bRefPCam := RioA10_10[Z].In.LimSwP;

RETURN;
```

Hier werden die Zuordnungen der Analog-Achsen (z.B. Y: 1. Analog-Achse zu 2. CNC-Achse) zu den CNC-Achsen getätigt.

Im Arbeitsblatt Freigaben sind der Vorschub- und die Reglerfreigabe auf TRUE zu setzen.

Die Endlagen- und Referenzpunktschalter müssen nicht über das RIO A10-10 eingelesen werden. Dann sind hier die anderen Eingänge zuzuweisen.

14.9 Anmerkungen

1. Vorsicht! Im Beispiel sind alle Freigaben für die Achsen gesetzt (Anfangswert in der Variabelendeklaration: *Enable). Sie müssen bewusst zurückgesetzt werden.
2. Das Achsinterface RIO A10-10 ist ein allgemein gültiges Modul. Es muss nicht zwingend mit der CNC und dem Funktionsbaustein MC_ANALOG verwendet werden. Konfigurationen wie die Überwachung der Endschalter sind für reine SPS- Lageregler als Hardware-Unterstützung gedacht. Das gleiche gilt für die entsprechenden Meldungen.
3. Die Fehler-Variable McRioA10_10.SumError wird gesetzt, wenn eine der aufgerufenen Achsen (*NumberOfAnalogAxis*) einen Kabelbruch (Messkreisfehler) erkennt oder der Funktionsbaustein MC_ANALOG mit ungültigen Parametern aufgerufen wird.
4. Die Impulse des Gebers werden durch das RIO A10-10 vervierfacht.
5. Der Maschinenparameter "Achse nicht referenzieren" ist z.Zt. noch im Schleicher-Dialog unter *Inbetriebnahme/CNC-Systemparameter* (Q.052, Bit 3) und unter *Inbetriebnahme/Antriebskonfiguration/Antriebsparameter* (NOT_HOMING_n) zu beschreiben.
6. Die "Maximalgeschwindigkeit" ist "zweimal" im Schleicher-Dialog (unter *Inbetriebnahme/CNC-Systemparameter* (Q.023) und unter *Inbetriebnahme/Antriebskonfiguration/Antriebsparameter* (MAX_VELOCITY_n)) einzutragen (siehe Online-Hilfe).
7. Zu beachten ist hierbei, dass die Parameter eine unterschiedliche Bedeutung haben. Q.023 gibt die maximale Achsgeschwindigkeit der Maschine an, während MAX_VELOCITY die maximal mögliche Geschwindigkeit des Antriebes angibt.
8. Das Nichterreichen des Genauhalt der Achse wird nicht als Fehler ausgegeben.
9. Die Achse fährt den ersten NC-Satz und wechselt nicht zum zweiten NC-Satz. Fragen Sie die Koppelspeicher-Variable `cncMem.axSect[n].flgN2P.bInPos` ab und parametrieren Sie das Genauhaltfenster in der Antriebskonfiguration entsprechend.

15 Fehlermeldungen

Die Fehlermeldungen werden im Active-Error-Buffer und im Error-Log-Book gespeichert, sie sind mit Fehlernummern und zusätzlichen Angaben gekennzeichnet.

Active-Error-Buffer und Error-Log-Book sind im Schleicher-Dialog auf jeder Bedienebene über die Tastenkombination <Ctrl+?> aufrufbar.

In der Online-Hilfe des Schleicher-Dialoges werden detaillierte Erläuterung der Fehlermeldungen zur Verfügung gestellt.

15.1 Aufbau der Fehlermeldungen

Fehlermeldungen bestehen aus einer hexadezimalen Fehler-Nr., einem Fehlertext und bis zu 3 optionalen Parametern, die in den Fehlertext eingeblendet werden können.

Fehler-Nr.(hexadezimal) / Par. 1 / Par. 2 / Par. 3. / Fehlertext

Fehler, denen kein Fehlertext zugeordnet wurde, werden in folgender Darstellung ausgegeben:

Fehler %ErrX%, Parameter %ParX% (Par1=%Par1%, Par2=%Par2%, Par3=%Par3%)

Folgende Zahlenwerte können mit dieser Nomenklatur in die Fehlermeldung eingeblendet werden:

%ErrX%	Fehlernummer (32 Bit, hexadezimal)
%ParX%	Parameterwert (32 Bit, hexadezimal), enthält die Parameter 1 bis 3: D31...D24 -- Parameter 1 D23...D16 -- Parameter 2 D15...D0 -- Parameter 3
%Par%	Parameterwert (32 Bit, dezimal), enthält die Parameter 1 bis 3: D31...D24 -- Parameter 1 D23...D16 -- Parameter 2 D15...D0 -- Parameter 3
%Par1%	Parameterwert 1 (8 Bit, dezimal, mit Vorzeichen)
%Par2%	Parameterwert 2 (8 Bit, dezimal, mit Vorzeichen)
%Par3%	Parameterwert 3 (16 Bit, dezimal, mit Vorzeichen)

Beispiel (Darstellung im Error-Log-Book):

Datum	Uhrzeit	Fehlertext	Fehler-nummer	Fehler-parameter	Status
04.03.03	08:04:06	Fehler 0x04100006, Parameter 0x00640001 (Par1=0, Par2=100, Par3=1)	0x04100006	0x00640001	O
04.03.03	08:04:04	Power-On	0x01100006	0x00000000	O
04.03.03	08:04:01	NC nicht betriebsbereit	0x02000000	0x00000000	A

Die Spalte Status im Error-Log-Book kann folgende Werte annehmen:

A	(Active) aktive Fehlermeldung im Active-Error-Buffer
L	(Lock) gelockte Fehlermeldung im Active-Error-Buffer
I	(In) Fehlermeldung aufgetreten und in den Active-Error-Buffer eingetragen
O	(Out) Fehlerursache beendet, aus dem Active-Error-Buffer ausgetragen

Aufbau der Fehlernummern

Die Fehlernummer ist eine 32 Bit Zahl, das High-Wort enthält die Gruppierung, das Low-Wort die fortlaufende Fehlernummer.

Gruppierung der Fehlernummern

0x0100nnnn	Laufzeitfehler SPS
0x0110nnnn	allgemeine SPS-Fehlermeldungen
0x0120nnnn	XRIO-Fehler
0x0200nnnn	allgemeine CNC-Fehlermeldungen
0x0210nnnn	teilsystemspezifische CNC-Fehler
0x0211nnnn	Decoderfehler
0x0212nnnn	Satzeinfügefehler
0x0213nnnn	Fehler der Schneidenradiuskompensation
0x0214nnnn	Fehler Interpolationsvorbereitung
0x0220nnnn	achsspezifische Fehler
0x0221nnnn	SERCOS Diagnosestatus (nur bei ProNumeric/PN-MIC)
0x0222nnnn	SERCOS Zustandsklasse 1 (nur bei ProNumeric/PN-MIC)
0x0223nnnn	herstellerspezifischer SERCOS-Fehler (nur bei ProNumeric/PN-MIC)
0x0224nnnn	herstellerspezifischer SERCOS-Fehlertext (nur bei ProNumeric/PN-MIC)
0x0230nnnn	antriebsspezifischer Fehler (nur bei XCN)
0x0300nnnn	CAN-spezifische Betriebssystemfehler
0x0301nnnn	CAN-Treiberfehler
0x0400nnnn bis 0x04FFnnnn	anwenderdefinierte Fehler, die durch den Funktionsbaustein PUT_ERROR in das Error-Log-Book eingetragen werden können

16 Technische Daten

Spannungsversorgung	
Versorgungsspannung	DC 24 V (19,2 V .. 30 V einschl. 5% Restwelligkeit)
Stromaufnahme	max. 4,5 A (incl. aller digitalen I/O)
Galvanische Trennung (zur internen Elektronik)	X1 (RS422) ja X2 (digitale E/A) ja X3 USB nein X4 XRIO ja X5 CAN ja X6 ETH (Ethernet) ja X7 PRG (RS232) nein

Serielle Schnittstellen	
RS 232	Programmier und Diagnoseschnittstelle X7 PRG
RS 422	Bediengeräteschnittstelle X1

Weitere Schnittstellen	
USB	Programmier- Diagnose und Bediengeräteschnittstelle X3
Ethernet	Programmier- Diagnose und Bediengeräteschnittstelle X6
CAN	CANopen Feldbusschnittstelle
XRIO	Proprietäre Feldbusschnittstelle zum Anschluß des RIO IO- Systems über spezielle Buskoppelmodule (RIO EC X2)

direkte Ein-/Ausgänge (auf dem Steckverbinder X2)	
Ausgänge	DC 24 V, max. 500 mA, General Purpose
Anzahl	2
Schaltpegel	H-Pegel ≥ +24V-EXT - 0,5 V L-Pegel ≤ 1 V
Signalverzögerung	<300 µs (Hardware)
Eingänge	
Anzahl	2
Schaltpegel	H-Pegel = +11 V bis +30 V L-Pegel = -30 V bis +5 V
Eingangsstrom	min. H-Pegel (+11 V) ≥ 2 mA max. L-Pegel (+5 V) ≤ 2 mA typ. (+24 V) 8 mA max. (+30 V) ≤ 15 mA
Signalverzögerung	<100 ns (Hardware)
Triggerung	Flanken-Triggerung

BUSY-Kontakt	DC 24 V, max. 2 A, General Purpose
Anzahl	1
Kontaktart	potentialfreier Relaiskontakt, Schließer

Sonstige Technische Daten	
Prozessor	XCx500: CPU Intel StrongARM SA1110, 32 Bit, 206 MHz XCx300: CPU Intel PXA210, 32 Bit, 200 MHz
Speicherausstattung	SDRAM 32 MB SRAM 1 MB FLASH (intern) 1 MB max. 16384 kB Datenspeicher, 4096 kB Programme SPS- Merker: 2048 kB nicht remanente Merker, 256 kB remanente Merker
Real-Time Clock	Batteriegepuffert mit Kalender und Schaltjahr, Auflösung: 1s
Pufferung	min. 3 Monate (Setzt eine volle Aufladung der Zelle voraus, was mit einer ununterbrochenen Betriebszeit der Steuereinheit von 4 h erreicht wird.)
Pufferelement	Vanadium-Pentoxid-Lithium-Zelle 3 V / 50 mAh + SuperCAP
Bearbeitungszeiten je 1000 AW	Bit 50 µs Byte / Word 30 µs

Klimatische Umgebungsbedingungen	
Betriebsumgebungstemperatur	0 ... +55°C (Kl. KV nach DIN 40040), freie Luftzirkulation Einschränkung auf 0 bis +45°C bei Verwendung der XF-DPS und XF-DPM Module.
Lagertemperatur	-25 ... +70°C (Kl. HS nach DIN 40040)
Relative Luftfeuchte	10 ... 95% (Kl. F nach DIN 40040), keine Betauung
Luftdruck im Betrieb	860 ... 1060 hPa

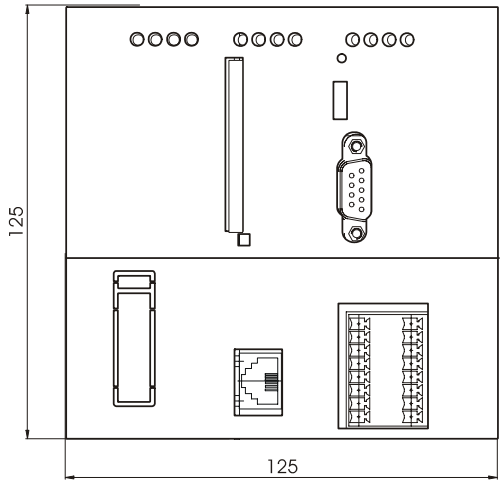
Elektrische Sicherheit	
Schutzart	IP 20 nach EN 60529
Luft-/Kriechstrecken	nach DIN EN 61131-2 zwischen Stromkreisen und Körper sowie zwischen galvanisch getrennten Stromkreisen, entsprechend Überspannungskategorie II, Verschmutzungsgrad 2
Prüfspannung	AC 350 V / 50 Hz für Geräte-Nennspannung DC 24 V

Elektromagnetische Verträglichkeit (EMV)	
Elektrostatische Entladung	nach EN 61000-4-2: 8 kV Luftentladung, 4 kV Kontaktentladung
Elektromagnetische Felder	nach EN 61000-4-3: Feldstärke 10 V/m, 80 ... 1000 MHz
Schnelle Transienten (Burst)	nach EN 61000-4-4: 2 kV auf DC-Versorgungsleitungen, 1 KV auf E/A-Signalleitungen
Stoßspannung (Surge)	nach EN 61000-4-5: 1 kV CM und 0,5 kV DM auf DC-Versorgungsleitungen, 0,5 kV CM und 0,5 kV DM auf DC- E/A- Signalleitungen
Störaussendung	nach EN 55011: Grenzwertklasse A, Gruppe 1

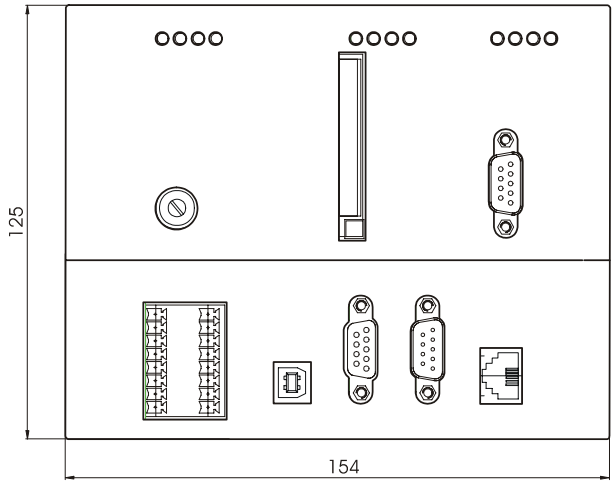
Mechanische Festigkeit	
Schwingen	nach DIN EN 60068-2-6 10 ... 57 Hz konstante Amplitude 0,075mm 57 ... 150 Hz konstante Beschleunigung 1 g
Schocken	nach DIN EN 60068-2-27, Sinus-Halbwelle 15g / 11ms
Freier Fall	nach DIN EN 60068-2-32: Fallhöhe 1m (mit Originalverpackung)

17 Abmessungen

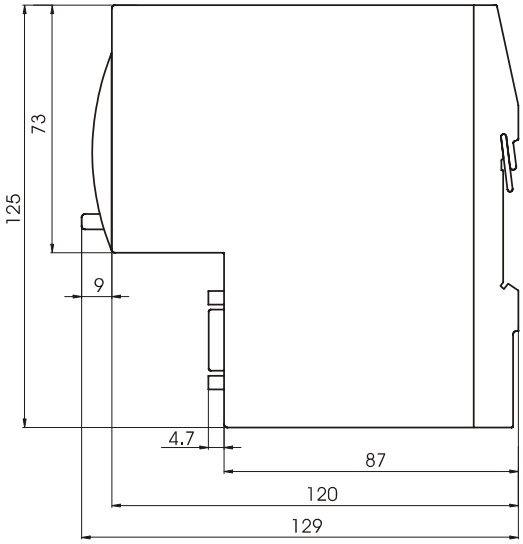
XCx 300



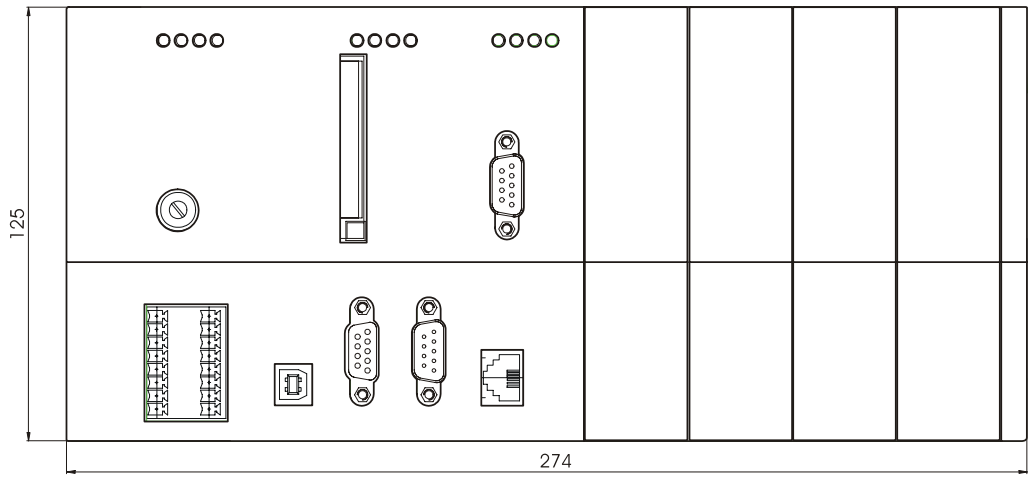
XCx 500



XCx 300 / 500 / 540



XCx 540



18 Anhang

18.1 Warenzeichenvermerke

- WINDOWS ist ein eingetragenes Warenzeichen der Microsoft Corporation.
- CANopen ist ein eingetragenes Warenzeichen von CAN in Automation e.V,
- ProCANopen ist ein eingetragenes Warenzeichen von Vector Informatik GmbH
- CANalyzer ist ein eingetragenes Warenzeichen von Vector Informatik GmbH
- VxWorks ist ein eingetragenes Warenzeichen der Wind River Systems Inc.

Alle anderen Warenzeichen oder Produktnamen sind eingetragene Warenzeichen der jeweiligen Firmen.

19 Sicherheitshinweise

Der im folgenden verwendete Begriff Automatisierungssysteme umfaßt Steuerungen, sowie deren Komponenten (Module), andere Teile (wie z.B. Baugruppenträger, Verbindungskabel), Bediengeräte und Software, die für die Programmierung, Inbetriebnahme und Betrieb der Steuerungen genutzt wird. Die vorliegende Betriebsanleitung kann nur einen Teil des Automatisierungssystems (z.B. Module) beschreiben.

Die technische Auslegung der SCHLEICHER Automatisierungssysteme basiert auf der Produktnorm EN 61131-2 (IEC 61131-2) für speicherprogrammierbare Steuerungen. Für die Systeme und Geräte gilt grundsätzlich die CE-Kennzeichnung nach der EMV-Richtlinie 89/336/EWG und sofern zutreffend auch nach der Niederspannungsrichtlinie 73/23/EWG.

Die Maschinenrichtlinie 89/392/EWG ist nicht wirksam, da die in der Richtlinie genannten Schutzziele auch von der Niederspannungs- und EMV-Richtlinie abgedeckt werden.
Sind die SCHLEICHER Automatisierungssysteme Teil der elektrischen Ausrüstung einer Maschine, müssen sie vom Maschinenhersteller in das Verfahren zur Konformitätsbewertung einbezogen werden. Hierzu ist die Norm DIN EN 60204-1 zu beachten (Sicherheit von Maschinen, allgemeine Anforderungen an die elektrische Ausrüstung von Maschinen).

Von den Automatisierungssystemen gehen bei bestimmungsgemäßer Verwendung und ordnungsgemäßer Unterhaltung im Normalfall keine Gefahren in Bezug auf Sachschäden oder für die Gesundheit von Personen aus. Es können jedoch durch angeschlossene Stellelemente wie Motoren, Hydraulikaggregate usw. bei unsachgemäßer Projektierung, Installation, Wartung und Betrieb der gesamten Anlage oder Maschine, durch Nichbeachten von Anweisungen in dieser Betriebsanleitung und bei Eingriffen durch ungenügend qualifiziertes Personal Gefahren entstehen.

19.1 Bestimmungsgemäße Verwendung

Die Automatisierungssysteme sind nach dem Stand der Technik und den anerkannten sicherheitstechnischen Regeln gebaut. Dennoch können bei ihrer Verwendung Gefahren für Leib und Leben des Benutzers oder Dritter bzw. Beeinträchtigungen von Maschinen, Anlagen oder anderen Sachwerten entstehen.

Das Automatisierungssystem darf nur in technisch einwandfreiem Zustand sowie bestimmungsgemäß, sicherheits- und gefahrenbewußt unter Beachtung der Betriebsanleitung benutzt werden. Der einwandfreie und sichere Betrieb der Steuerung setzt sachgemäßen Transport, sachgerechte Lagerung und Montage sowie sorgfältige Bedienung und Wartung voraus. Insbesondere Störungen, die die Sicherheit beeinträchtigen können, sind umgehend beseitigen zu lassen.

Die Automatisierungssysteme sind ausschließlich zur Steuerung von Maschinen und Anlagen vorgesehen. Eine andere oder darüber hinausgehende Benutzung gilt nicht als bestimmungsgemäß. Für daraus resultierende Schäden haftet der Hersteller nicht.

Zur bestimmungsgemäßen Verwendung der Automatisierungssysteme sind die in dieser Betriebsanleitung beschriebenen Anweisungen zum mechanischen und elektrischen Aufbau, zur Inbetriebnahme und zum Betrieb zu beachten.

19.2 Personalauswahl und -qualifikation

Alle Projektierungs-, Programmier-, Installations-, Inbetriebnahme-, Betriebs- und Wartungsarbeiten in Verbindung mit dem Automatisierungssystem dürfen nur von geschultem Personal ausgeführt werden (z.B. Elektrofachkräfte, Elektroingenieure).

Das Projektierungs- und Programmierpersonal muß mit den Sicherheitskonzepten der Automatisierungstechnik vertraut sein.

Das Bedienpersonal muß im Umgang mit der Steuerung unterwiesen sein und die Bedienungsanweisungen kennen.

Das Installations-, Inbetriebnahme- und Wartungspersonal muß eine Ausbildung besitzen, die zu Eingriffen am Automatisierungssystem berechtigt.

19.3 Projektierung, Programmierung, Installation, Inbetriebnahme und Betrieb

Das Automatisierungssystem ist in seiner Anwendung zumeist Bestandteil größerer Systeme oder Anlagen, in denen Maschinen gesteuert werden. Bei Projektierung, Installation und Inbetriebnahme der Automatisierungssysteme im Rahmen der Steuerung von Maschinen müssen deshalb durch den Maschinenhersteller und Anwender die Sicherheitsbestimmungen der Maschinenrichtlinie 89/392/EWG beachtet werden. Im spezifischen Einsatzfall geltende nationale Unfallverhütungsvorschriften wie z.B. VBG 4.0.

Alle sicherheitstechnischen Vorrichtungen der gesteuerten Maschine sind so auszuführen, daß sie unabhängig von der Steuerung funktionieren. Not-Aus-Einrichtungen müssen in allen Betriebsarten der Steuerung wirksam bleiben. Im Not-Aus-Fall müssen die Versorgungsspannungen aller von der Steuerung angesteuerten Schaltelemente abgeschaltet werden. Hierzu kann ein Sicherheitsrelais (z.B. SCHLEICHER Typ SNO 2002-17) eingesetzt werden.

Es sind Vorkehrungen zu treffen, daß nach Spannungseinbrüchen und -ausfällen ein unterbrochenes Steuerungsprogramm ordnungsgemäß wieder aufgenommen werden kann. Dabei dürfen auch kurzzeitig keine gefährlichen Betriebszustände auftreten. Gegebenenfalls ist Not-Aus zu erzwingen.
Damit ein Leitungsbruch auf der Signalseite nicht zu undefinierten Zuständen in der Steuerung führen kann, sind bei der E/A-Kopplung hard- und softwareseitig entsprechende Sicherheitsvorkehrungen zu treffen. Einrichtungen der Steuerungstechnik und deren Bedienelemente sind so einzubauen, daß sie gegen unbeabsichtigte Betätigung ausreichend geschützt sind.

19.4 Wartung und Instandhaltung

Werden Meß- oder Prüfarbeiten am aktiven Gerät erforderlich, dann sind die Festlegungen und Durchführungsanweisungen der Unfallverhütungsvorschrift VBG 4.0 zu beachten. Es ist geeignetes Elektrowerkzeug zu verwenden.

Reparaturen an Steuerungskomponenten dürfen nur von SCHLEICHER autorisierten Reparaturstellen vorgenommen werden. Unbefugtes Öffnen und unsachgemäße Eingriffe oder Reparaturen können zu Körperverletzungen oder Sachschäden führen.

Vor Öffnen des Gerätes ist immer die Verbindung zum speisenden Netz zu trennen (Netzstecker ziehen oder Trennschalter öffnen).

Steuerungsmodule dürfen nur im spannungslosen Zustand gewechselt werden. Demontage und Montage sind gemäß der mechanischen Aufbaurichtlinien vorzunehmen.

Beim Auswechseln von Sicherungen dürfen nur Typen verwendet werden, die in den technischen Daten spezifiziert sind.

Beim Austausch von Batterien dürfen nur Typen verwendet werden, die in den technischen Daten spezifiziert sind. Batterien sind in jedem Fall nur als Sondermüll zu entsorgen.

19.5 Gefahren durch elektrische Energie

Nach Öffnen des Systemschrankes oder nach Entfernen des Gehäuses von Systemkomponenten werden bestimmte Teile des Automatisierungssystems zugänglich, die unter gefährlicher Spannung stehen können.

Der Anwender muß dafür sorgen, daß unbefugte und unsachgemäße Eingriffe unterbunden werden (z.B. verschlossener Schaltschrank).

Das Personal muß gründlich mit allen Gefahrenquellen und Maßnahmen zur Inbetriebnahme und Wartung gemäß den Angaben in der Betriebsanleitung vertraut sein.

19.6 Umgang mit verbrauchten Batterien

Die in den Automatisierungssystemen verwendeten Batterien sind, nach deren Verbrauchsende, dem Gemeinsamen Rücknahmesystem Batterien (GRS) oder öffentlich-rechtlichen Entsorgungsträgern zuzuführen.

Batterien sollen nur im entladenen Zustand zurückgegeben werden. Der entladene Zustand ist erreicht, wenn eine Funktionsbeeinträchtigung des Gerätes wegen unzureichender Batteriekapazität vorliegt.

Bei nicht vollständig entladenen Batterien muß Vorsorge gegen mögliche Kuzschlüsse getroffen werden. Das kann durch Isolieren der Batteriepole mit Klebestreifen erreicht werden.

20 Abbildungsverzeichnis und Index

Abbildung 1: Übersicht XCx 300 ... 9
Abbildung 2: Übersicht XCx 500 ... 10
Abbildung 3: Übersicht XCx 540 ... 11
Abbildung 4: Montageabstände ... 13
Abbildung 5: Montageabmaße und -abstände XCx 300 ... 13
Abbildung 6: Abmaße der Befestigungslöcher für Schraubmontage der XCx 500/540 ... 15
Abbildung 7: Kontaktschieber RIO mit XCx 300 ... 16
Abbildung 8: Endklammern an der XCx 300 ... 16
Abbildung 9: Anschluss der Spannungsversorgung XCx 500 und XCx 540 ... 18
Abbildung 10: Anschluss der Spannungsversorgung XCx 300 ... 19
Abbildung 11: Montagelage XRIO ... 20
Abbildung 12: Montageabmaße und -abstände XRIO mit Buskoppler ... 20
Abbildung 13: Montage XRIO ... 21
Abbildung 14: Endklammern XRIO ... 21
Abbildung 15: Demontage XRIO ... 21
Abbildung 16: Kontaktschieber RIO mit XRIO-Buskoppler ... 22
Abbildung 17: Schneidenbreite Schraubendreher ... 23
Abbildung 18: Federkraftklemmen des Kopplers ... 23
Abbildung 19: Federkraftklemmen der Erweiterungsmodule ... 24
Abbildung 20: Anschluss der Spannungsversorgung XRIO an der XCx 500 und XCx 540 ... 26
Abbildung 21: Kontaktfedern im Klemmfuß ... 27
Abbildung 22: Buskabel XRIO ... 28
Abbildung 23: Anschluss des Service-PC an der XCx ... 36
Abbildung 24: XRIO-Anschluss an der XCx ... 57
Abbildung 25: SLM-Modul Frontansicht ... 98
Abbildung 26: Buskoppler RIO EC X2 mit 5 RIO-Erweiterungsmodulen ... 101
Abbildung 27: Buskoppler RIO EC X2 mit 4 RIO-Erweiterungsmodulen ... 103
Abbildung 28: Tasksystem ... 126
Abbildung 29: Task und Watchdog ... 135
Abbildung 30: SLM-Modul Frontansicht ... 150
Abbildung 31: Schnittstellen XP-SLM ... 155
Abbildung 32: Applikationsbeispiel XP-SLM ... 157
Abbildung 33: Verdrahtung XP-SLM an MultiAx ... 158
Abbildung 34: Steckerbelegung XP-SLM MC (FNC) ... 159
Abbildung 35: Verdrahtung XP-SLM an M'Ax ... 160
Abbildung 36: Verdrahtung XP-SLM an M'Ax RJ45 ... 160
Abbildung 37: Steckerbelegung XP-SLM MC/EIA485 ... 161
Abbildung 38: Steckerbelegung XP-SLM MC/EIA485 ... 161
Abbildung 39: SERCOS-Modul Frontansicht ... 162
Abbildung 40: SERCOS-Modul Applikationsbeispiel ... 164
Abbildung 41: PROFIBUS-DP Master Frontansicht ... 166
Abbildung 42: PROFIBUS-DP Slave Frontansicht ... 168

Index

A

Abisolierlänge
RIO ... 25
Abmessungen
XCx ... 185
XRIO ... 20
Activ-Error-Buffer
über Schleicher-Dialog einsehen ... 97
Adressierung
XRIO I/O-Module ... 103
Analog-Achsen ... 173
Anlauf mit leerer Puffer-Batterie ... 78
Anschluss
CANopen ... 33
EtherNet ... 32
Programmiergerät ... 32
USB ... 33
Versorgungsspannung, direkte E/A und BUSY-Kontakt ... 34
X RIO ... 33
X1 34
X3 siehe USB-Anschluss ... 33
X4 siehe X RIO Anschluss ... 33
X5 siehe CANopen Feldbusanschluss ... 33
X6 ETH siehe EtherNet-Anschluss ... 32
X7 siehe Programmiergeräteschnittstelle ... 32
Anschlüsse ... 29
PROFIBUS-DP Modul XF-DPM ... 167
PROFIBUS-DP Modul XF-DPS ... 169
SERCOS-Modul XP-SRC ... 163
SLM-Modul XP-SLM ... 151
Anschlußquerschnitte
RIO ... 25
Anwenderbibliotheken siehe Bibliotheken ... 141
Anwender-Task ... 127
Anzeige, LED's auf der Frontseite ... 30
Applikationsbeispiel
SERCOS-Modul XP-SRC ... 164
SLM-Modul XP-SLM ... 157
Web-Server ... 77
Aufbau
der Fehlermeldungen ... 181

B

Batterie leer ... 78
batterieloser Betrieb ... 78
Bedienbereiche
XCS ... 90
Bedienelemente ... 29
Bedienoberfläche
Bedienbereiche XCS ... 90
Befestigung
XCx 500 und 540 durch Aufschrauben ... 15
XCx auf Hutschiene ... 14
XRIO auf Hutschiene ... 21
BETRIEB Betriebszustand ... 138
Betriebsartenschalter ... 35
Betriebszustände der SPS ... 138
Bibliothek
CANopen_Vxxx ... 143
CFB_Vxxx ... 143
CNC_Vxxx ... 144
Date_Time ... 144
MMI ... 145
PLC_Vxxx ... 145
Profibus_VxxxSerial ... 146
SchleicherLib_Vxxx ... 146
Serial ... 146
XCx7_Vxxx ... 146
Browser für Visualisierung mit Web-Server ... 77
Buskabel
XRIO ... 28
Buskoppler RIO EC X2 ... 98

C

CAN HANDLER FEHLER ... 123
CANopen ... 64
Deklaration des I/O-Treibers ... 66
Erweiterungsmodul XF-CAN ... 125
Fehlermeldungen ... 123
Feldbuskarte ... 69
Firmware-Bibliotheken in MULTIPROG ... 124
Konfiguration mit ProCANopen ... 69
Lifeguarding ... 114
Netzvariablen ... 115
Nodeguarding ... 113
Parametrier- und Diagnosefunktionen ... 120
PDO ... 113
SDO ... 113
SDO Funktionsbausteine ... 119
Spezifikationen ... 64
SPS-Adressen ... 115
Statusvariable ... 115
Zugriff auf direkte Netzvariablen ... 118
Zugriff auf Netzvariablen ... 116
CF siehe Compact Flash ... 32
CNC
Inbetriebnahme und Programmierung ... 148
Koppelspeicher ... 148
Multi-Task-System ... 149
Compact Flash ... 32

D

Daten, Gültigkeit der XRIO-Prozessdaten ... 103
Datenübertragungsrate
CANopen einstellen ... 121
Default-Task ... 131
Deklaration
des I/O-Treibers CANopen ... 66
OPC-Variablen ... 82
von Netzwerkvariablen ... 68
von Variablen zur Visualisierung mit dem Web-Server ... 77
Diagnose
XRIO ... 63
Diagnose CANopen
Code 1 Knotennummer ... 121
Code 2 Datenübertragungsrate ... 121
Code 20 CANopen-Hochlaufverzögerung einstellen ... 122
Code 7 CANopen Fehlernummer ... 122
Code 8 CANopen-Firmware-Version ... 122

E

Echtzeit Uhr
Funktionsbausteine zum lesen/setzen ... 144
EIN Betriebszustand ... 138
Einbauabstände siehe Montageabstände ... 20
Einbaulage siehe Montagelage ... 13

Einstellungen
Ethernet-Verbindung in MULTIPROG ... 49
Elektrische Installation
Anschluss der Spannungsversorgung XCx ... 18
Anschluss der Spannungsversorgung XRIO ... 26
XRIO ... 23
Endklammer für Hutschienenmontage RIO ... 21
Ereignis-Task ... 128
Erweiterungsmodule
für XRIO ... 106
XF-CAN ... 125
XF-DPM ... 166
XF-DPS ... 168
XP-SLM ... 150
XP-SRC ... 162
XRIO ... 111
Ethernet
Anschluss für das Web ... 76
TCP/IP-Verbindung zur Programmierung ... 36

F

Federkraftklemmen
RIO-Erweiterungsmodule ... 24
XRIO Busoppler ... 23
Fehlermeldungen ... 181
Fehlermeldungen CANopen ... 123
Fensteraufbau
Schleicher-Dialog ... 87
Firmware-Funktionsbausteine siehe Bibliothek PLC_Vxxx ... 145
Firmware-Task Prioritäten ... 134
Flags
XRIO ... 103
Flash-Speicher siehe Compact Flash ... 32
FTP ... 40
Funktionsbaustein
CLEAR_ERROR ... 145
CLOSE_PROFILE ... 145
CO_NET_CAN_SYNC ... 143
CO_NET_GET_KERNEL_STATUS ... 143
CO_NET_GET_LOCAL_NODE_ID ... 143
CO_NET_GET_STATE ... 143
CO_NET_NMT ... 143
CO_NET_PING ... 143
CO_NET_RECV_EMY ... 143
CO_NET_RECV_EMY_DEV ... 143
CO_NET_RECV_ERR ... 143
CO_NET_RECV_ERR_DEV ... 143
CO_NET_RESTART_ALL ... 143
CO_NET_RESTART_CAN ... 143
CO_NET_SDO_READ ... 143
CO_NET_SDO_WRITE ... 143
CO_NET_SENDL2 ... 143
CO_NET_SHUTDOWN ... 143
CONNECT_V ... 143
DP_NET_GET_MSG ... 146
DP_NET_GET_STATE ... 146
DP_NET_PUT_MSG ... 146
FLUSH_PROFILE ... 145
für Diagnose XRIO ... 63
GET_DATE ... 144
GET_MTS ... 145
GET_PROFILE_INT ... 145
GET_PROFILE_REAL ... 145
GET_PROFILE_STRING ... 145
GET_TIME ... 144
IBSM ... 146
MC_ANALOG ... 144
MC_ANALOG_1_AXIS ... 144
MC_CAN ... 144
MC_DP ... 144
MC_DP_1_AXIS ... 144
NEW_PROFILE ... 145
OPEN_PROFILE ... 145
PORT_CLOSE ... 146
PORT_OPEN ... 146
PORT_READ ... 146
PORT_STATE ... 146
PORT_WRITE ... 146
PPF_COP_COMM ... 145
PUT_ERROR ... 145
PUT_ERROR2 ... 145
READ_AXIS_PAGE ... 146
READ_FILE ... 145
READ_Q_PARAM_* ... 144
READ_RP ... 146
READ_SERC_PARAM ... 144
SAVE_Q_PARAM_* ... 144
SAVE_R_PARAM_* ... 144
SEND_MAIL ... 145
SET_DATE ... 144
SET_SERC_COMMAND ... 144
SET_SERC_PHASE ... 144
SET_TIME ... 144
UBA_ERR_CTRL ... 146
URCV_V ... 143
USEND_V ... 143
UZB_VR ... 146
WRITE_AXIS_PAGE ... 146
WRITE_FILE ... 145
WRITE_PROFILE_INT ... 145
WRITE_PROFILE_REAL ... 145
WRITE_PROFILE_STRING ... 145
WRITE_Q_PARAM_* ... 144
WRITE_RP ... 146
WRITE_SERC_PARAM ... 144
XFIO_CONFIG ... 145
XRIO_STATE ... 145
Funktionsbausteine und Bibliotheken ... 141

G

Grundinitialisierung ... 79
Gültigkeit der XRIO-Prozessdaten ... 103
Gültigkeit der XRIO-Prozessdaten, Gültigkeit der ... 103

H

HALT Betriebszustand ... 138
Hutschiene
RIO ... 21
XCx ... 14
Hutschienenmontage
Endklammer RIO ... 21
RIO ... 21
XCx ... 14

I

Inbetriebnahme ... 36
Initialisierung siehe Grundinitialisierung ... 79
Initialisierungsdatei ProConOS.ini ... 147
Installation
mechanische XCx ... 13
mechanische XRIO ... 20

MULTIPROG ... 41
OPC-Server ... 41, 81
Schleicher-Dialog ... 81
IP-Adresse
Ändern auf der XCx ... 40
Auslesen aus der XCx ... 38
Einstellen/Auslesen auf dem PC ... 39
überprüfen mit ping ... 39

K

Knotennummer
CANopen einstellen ... 121
Kombikanäle RIO
Spannungsversorgung ... 27
Konfiguration
CANopen mit ProCANopen ... 69
XRIO ... 100
Konfigurationsdaten
PROFIBUS-DP Modul ... 170
Kontaktschieber RIO ... 22
Koppelspeicher ... 148
bei der Inbetriebnahme ... 52

L

LED-Anzeigen ... 30
PROFIBUS-DP Modul XF-DPM ... 167
PROFIBUS-DP Modul XF-DPS ... 169
SERCOS-Modul XP-SRC ... 163
SLM-Modul XP-SLM ... 151
Lifeguarding CANopen ... 114
Log-Book
über Schleicher-Dialog aufrufen ... 97

M

MC_ANALOG ... 173
Mechanische Installation
XCx ... 13, 20
Messtaster über direkte E/A ... 34
Module
XF-CAN Erweiterungsmodul für XCx ... 125
XF-DPM PROFIBUS-DP Erweiterungsmodul für XCx ... 166
XF-DPS PROFIBUS-DP Erweiterungsmodul für XCx ... 168
XP-SLM Erweiterungsmodul für XCx ... 150
XP-SRC Erweiterungsmodul für XCx ... 162
XRIO Erweiterungsmodul für XCx ... 111
Montageabmaße
XCx ... 13
XRIO ... 20
Montageabstände
XCx ... 13
XRIO ... 20
Montagelage
RIO ... 20
XCx ... 13
MULTIPROG
Einstellungen ... 49
Installation ... 41
neues Projekt öffnen ... 47
Projekt zur XCx senden ... 50
Multi-Task-System
der XCx ... 126
Hinweise zur CNC ... 149

N

Netzvariablen
CANopen ... 115
Node ID siehe Knotennummer ... 121
Nodeguarding CANopen ... 113

O

OPC-Server
Installation ... 41, 81
Version 1.x ... 81, 83
Version 2.0 ... 81, 84
OPC-Variablen
Deklaration ... 82

P

Parametrier- und Diagnosefunktionen CANopen ... 120
Parametrierung CANopen
Code 1 Knotennummer ... 121
Datenübertragungsrate CANopen ... 121
PDD Process Data Directory ... 77
PDO CANopen ... 113
ping ... 39
Potential-Weiterschaltungsklemmen ... 27
ProCANopen
Einbindung in MULTIPROG ... 71
erste Verbindung ... 72
Installation ... 70
Process Data Directory ... 77
ProConOS ... 147
ProConOS.ini ... 147
PROFIBUS-DP Modul
Konfigurationsdaten ... 170
PROFIBUS-DP Modul XF-DPM ... 166
Anschlüsse ... 167
LED-Anzeigen ... 167
Technische Daten ... 167
PROFIBUS-DP Modul XF-DPS ... 168
Anschlüsse ... 169
LED-Anzeigen ... 169
Technische Daten ... 169
Programm
Zuweisen zu Task ... 136
Programmiergeräteschnittstelle ... 32
Programmierung der SPS ... 138
Projekt
öffnen in MULTIPROG ... 47

R

RIO
Erweiterungsmodule Abisolierlänge ... 25
Erweiterungsmodule Anschlußquerschnitt ... 25
RIO A10-10 ... 173
RIO EC X2 für XRIO ... 98
RS232
Schnittstelle zur Inbetriebnahme ... 36
RS422-Schnittstelle Anschluss X1 ... 34

S

Schleicher-Dialog
Activ-Error-Buffer einsehen ... 97
Aktivierung des Log-Book ... 97
Fensteraufbau ... 87

Installation ... 81
Übersicht ... 80
XCS ... 90, 92
Schlüsselschalter siehe Betriebsartenschalter ... 35
SDO CANopen ... 113
SDO Funktionsbausteine ... 119
SERCOS-Modul XP-SRC ... 162
Anschlüsse ... 163
Applikationsbeispiel ... 164
LED-Anzeigen ... 163
Technische Daten ... 163
Sicherheitshinweise ... 187
Bestimmungsgemäße Verwendung ... 187
Darstellung Warnhinweise ... 7
Inbetriebnahme ... 188
Installation ... 188
Instandhaltung ... 188
Not-Aus-Einrichtung ... 188
Personalauswahl ... 187
Programmierung ... 188
Projektierung ... 188
Unfallverhütungsvorschrift ... 188
Wartung ... 188
SLM-Modul XP-SLM
Applikationsbeispiel ... 157
Steckerbelegung ... 159, 161
Technische Daten ... 154
Verdrahtungsbeispiel ... 158
SLM-Modul XP-SLM ... 150
Anschlüsse ... 151
LED-Anzeigen ... 151
Spannungsversorgung
Anschluss XCx ... 18
Anschluss XRIO ... 26
SPG siehe Systemprogramme ... 129
SPS
Adressen für CANopen ... 115
Betriebssystem ProConOS ... 147
Betriebszustände ... 138
Betriebszustände wechseln ... 139
Grundlagen ... 138
Koppelspeicher ... 148
Programmierung ... 138
Startverhalten ... 140
Systemvariablen ... 140
Startverhalten der SPS ... 140
Statusvariable
CANopen ... 115
Steckerbelegung
SLM-Modul XP-SLM ... 159, 161
XRIO ... 28
Steuerungsanlauf mit leerer Puffer-Batterie ... 78
Steuerungsmenü
XCS ... 90
STOP Betriebszustand ... 138
sysinfo.txt Konfiguration XRIO ... 101
Systemprogramme SPG's ... 129
System-Task ... 129
Systemvariablen der SPS ... 140

T

Task
Anwender-Task ... 127
Default-Task ... 131
Ereignis-Task ... 128
erstellen ... 136
Firmware-Task Prioritäten ... 134
Information ... 132
Prioritäten ... 134
Prioritätsstufen Übersicht ... 126
System-Task ... 129
Task-Information ... 132
Überwachungstask ... 126
Watchdog ... 135
zyklische Task ... 127
TCP/IP-Verbindung ... 36
Technische Daten
PROFIBUS-DP Modul XF-DPM ... 167
PROFIBUS-DP Modul XF-DPS ... 169
SERCOS-Modul XP-SRC ... 163
SLM-Modul XP-SLM ... 154
XCx ... 183
Treiber
XRIO ... 104

Ü

Übersicht
XCx -Varianten ... 8
Überwachungstask ... 126

U

USB (Universal Serial Bus) ... 33

V

Variablen
CANopen Zugriff auf dirakte Netzvariablen ... 118
CANopen Zugriff auf Netzvariablen ... 116
Deklaration der OPC-Variablen ... 82
Deklaration von Netzwerkvariablen ... 68
zur Visualisierung mit Web-Server ... 77
Variablendeklaration
in den Beispielprogrammen der Funktionsbausteine 124, 142
Verbinden der RIO-Module untereinander ... 22
Verdrahtungsbeispiel
SLM-Modul XP-SLM ... 158
Verkabelung
XCx ... 18
XRIO ... 26
Versorgungsspannung
Anschluss XCx ... 18
Anschluss XRIO ... 26

W

Warenzeichenvermerke ... 186
Watchdog ... 135
Watchdog Taskausführung ... 135
Web-Server
Applikationsbeispiel ... 77
Deklaration von Variablen zur Visualisierung ... 77
Funktionen und Konzept ... 76
notwendige Browser ... 77
Schleicher Applet ... 76

X

X1 Anschluss ... 34
X2 Anschluss ... 34
X3 siehe USB-Anschluss ... 33
X4 siehe X RIO Anschluss ... 33
X5 siehe CANopen Feldbusanschluss ... 33

X6 ETH siehe EtherNet-Anschluss 32
X7 siehe Programmiergeräteschnittstelle..... 32
XCx
Abmessungen..... 185
Übersicht..... 8
Verkabelung..... 18
XF-DPM siehe PROFIBUS-DP Module 166
XF-DPS siehe PROFIBUS-DP Modul..... 168
XP-SLM siehe SLM-Modul XP-SLM..... 150
XP-SRC siehe SERCOS-Modul..... 162
XRIO
Adressierung..... 103
aus dem System RIO 106
Buskabel 28
Buskoppler Abisolierlänge 25
Buskoppler Anschlußquerschnitt 25
Buskoppler RIO EC X2..... 98
Datei sysinfo.txt 101
Diagnose..... 63
Erweiterungsmodul 111
Flags 103
Konfiguration Erkennung der..... 100
Steckerbelegung..... 28
Treiber 104
Verkabelung..... 26
Zugriff auf die I/O-Ebene 57, 98

Z

Zuweisen
Programm zu Task 136
zyklische Task 127