

Modular PLC  
XC-CPU201-...(-XV)  
XC-CPU202-...-XV



**EATON**

*Powering Business Worldwide*

Alle Marken- und Produktnamen sind Warenzeichen oder eingetragene Warenzeichen der jeweiligen Titelhälter.

### **Störfallservice**

Bitte rufen Sie Ihre lokale Vertretung an:

<http://www.eaton.eu/aftersales>

oder

Hotline After Sales Service:

+49 (0) 180 5 223822 (de, en)

[AfterSalesEGBonn@eaton.com](mailto:AfterSalesEGBonn@eaton.com)

### **Originalbetriebsanleitung**

Die deutsche Ausführung dieses Dokuments ist die Originalbetriebsanleitung.

### **Übersetzung der Originalbetriebsanleitung**

Alle nichtdeutschen Sprachausgaben dieses Dokuments sind Übersetzungen der Originalbetriebsanleitung.

1. Auflage 2003, Redaktionsdatum 12/03
2. Auflage 2004, Redaktionsdatum 12/03
3. Auflage 2004, Redaktionsdatum 04/04
4. Auflage 2004, Redaktionsdatum 06/04
5. Auflage 2004, Redaktionsdatum 08/04
6. Auflage 2004, Redaktionsdatum 11/04
7. Auflage 2005, Redaktionsdatum 03/05
8. Auflage 2005, Redaktionsdatum 11/05
9. Auflage 2006, Redaktionsdatum 09/06
10. Auflage 2006, Redaktionsdatum 12/06
11. Auflage 2007, Redaktionsdatum 04/07
12. Auflage 2008, Redaktionsdatum 01/08
13. Auflage 2010, Redaktionsdatum 10/10
14. Auflage 2013, Redaktionsdatum 08/13
15. Auflage 2014, Redaktionsdatum 06/14

© 2010 by Eaton Industries GmbH, 53105 Bonn

Autor: Norbert Mausolf

Redaktion: Antje Panten-Nonnen

Alle Rechte, auch die der Übersetzung, vorbehalten.

Kein Teil dieses Handbuches darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder einem anderen Verfahren) ohne schriftliche Zustimmung der Firma Eaton Industries GmbH, Bonn, reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Änderungen vorbehalten.



## **Gefahr!** **Gefährliche elektrische Spannung!**

---

### **Vor Beginn der Installationsarbeiten**

- Gerät spannungsfrei schalten
- Gegen Wiedereinschalten sichern
- Spannungsfreiheit feststellen
- Erden und kurzschließen
- Benachbarte, unter Spannung stehende Teile abdecken oder abschranken.
- Die für das Gerät angegebenen Montagehinweise (AWA/IL) sind zu beachten.
- Nur entsprechend qualifiziertes Personal gemäß EN 50110-1/-2 (VDE 0105 Teil 100) darf Eingriffe an diesem Gerät/System vornehmen.
- Achten Sie bei Installationsarbeiten darauf, dass Sie sich statisch entladen, bevor Sie das Gerät berühren.
- Die Funktionserde (FE) muss an die Schutz Erde (PE) oder den Potenzialausgleich angeschlossen werden. Die Ausführung dieser Verbindung liegt in der Verantwortung des Errichters.
- Anschluss- und Signalleitungen sind so zu installieren, dass induktive und kapazitive Einstreuungen keine Beeinträchtigung der Automatisierungsfunktionen verursachen.
- Einrichtungen der Automatisierungstechnik und deren Bedienelemente sind so einzubauen, dass sie gegen unbeabsichtigte Betätigung geschützt sind.
- Damit ein Leitungs- oder Aderbruch auf der Signalseite nicht zu undefinierten Zuständen in der Automatisierungseinrichtung führen kann, sind bei der E/A-Kopplung hard- und softwareseitig entsprechende Sicherheitsvorkehrungen zu treffen.
- Bei 24-Volt-Versorgung ist auf eine sichere elektrische Trennung der Kleinspannung zu achten. Es dürfen nur Netzgeräte verwendet werden, die die Forderungen der IEC 60364-4-41 bzw. HD 384.4.41 S2 (VDE 0100 Teil 410) erfüllen.
- Schwankungen bzw. Abweichungen der Netzspannung vom Nennwert dürfen die in den technischen Daten angegebenen Toleranzgrenzen nicht überschreiten, andernfalls sind Funktionsausfälle und Gefahrenzustände nicht auszuschließen.

- NOT-AUS-Einrichtungen nach IEC/EN 60204-1 müssen in allen Betriebsarten der Automatisierungseinrichtung wirksam bleiben. Entriegeln der NOT-AUS-Einrichtungen darf keinen Wiederanlauf bewirken.
- Einbaugeräte für Gehäuse oder Schränke dürfen nur im eingebauten Zustand, Tischgeräte oder Portables nur bei geschlossenem Gehäuse betrieben und bedient werden.
- Es sind Vorkehrungen zu treffen, dass nach Spannungseinbrüchen und -ausfällen ein unterbrochenes Programm ordnungsgemäß wieder aufgenommen werden kann. Dabei dürfen auch kurzzeitig keine gefährlichen Betriebszustände auftreten. Gegebenenfalls ist NOT-AUS zu erzwingen.
- An Orten, an denen in der Automatisierungseinrichtung auftretende Fehler Personen- oder Sachschäden verursachen können, müssen externe Vorkehrungen getroffen werden, die auch im Fehler- oder Störfall einen sicheren Betriebszustand gewährleisten beziehungsweise erzwingen (z. B. durch unabhängige Grenzwertschalter, mechanische Verriegelungen usw.).

# Inhaltsverzeichnis

<b>0</b>	<b>Zu diesem Handbuch .....</b>	<b>7</b>
0.1	Änderungsprotokoll .....	7
0.2	Lesekonventionen .....	8
0.2.1	Warnhinweise vor Sachschäden .....	8
0.2.2	Warnhinweise vor Personenschäden .....	8
0.2.3	Tipps .....	8
0.3	Weiterführende Dokumentationen .....	9
<b>1</b>	<b>Aufbau der Steuerung XC200 .....</b>	<b>10</b>
1.1	Baugruppenträger .....	10
1.1.1	Leistungsumfang der CPU .....	11
1.1.2	Funktionsbereiche .....	11
1.1.3	Spannungsversorgung .....	12
1.1.4	Lokale Ein-/Ausgänge .....	12
1.1.5	Prozessoreinheit mit Schnittstellen .....	14
1.1.6	Echtzeituhr .....	14
1.1.7	Batterie .....	14
1.1.8	Multi Media Card (MMC), Secure Digital Card (SD), USB-Stick ...	15
1.1.9	CPU-Laufwerke .....	15
1.1.10	Programmierschnittstelle ETH232 .....	16
1.1.11	Splitten der ETH232-Schnittstelle .....	16
1.1.12	CAN-/easyNet-Schnittstelle .....	17
1.1.13	Verhalten der Teilnehmer am CAN-Bus .....	18
1.1.14	Zusatzfunktionen der CPU (lokale Eingänge) .....	19
<b>2</b>	<b>Montage der CPU .....</b>	<b>21</b>
2.0.1	Demontage der CPU .....	21
<b>3</b>	<b>Projektierung .....</b>	<b>22</b>
3.1	Schaltschrankaufbau .....	22
3.1.1	Belüftung .....	22
3.1.2	Geräteanordnung .....	23
3.2	Störungen vermeiden .....	23
3.2.1	Leitungsführung und Verdrahtung .....	23
3.2.2	Schutzbeschaltung von Störquellen .....	24
3.2.3	Schirmung .....	24
3.3	Blitzschutzmaßnahmen .....	24
3.3.1	Äußerer Blitzschutz .....	24
3.3.2	Innerer Blitzschutz .....	24
3.4	Anschlüsse .....	25
3.4.1	Spannungsversorgung anschließen .....	25
3.4.2	Ein-/Ausgänge (CPU) anschließen .....	25
3.4.3	Inkrementalgeber anschließen .....	26
3.4.4	Vor-/Rückwärtszähler anschließen .....	27
3.4.5	Interrupt-Aktoren anschließen .....	27

3.4.6	PC anschließen .....	28
3.5	Schnittstellenbelegungen .....	29
3.5.1	USB-Schnittstelle .....	29
3.5.2	Programmierschnittstelle XC200 .....	29
3.5.3	CAN/easyNet-Schnittstelle.....	30
<b>4</b>	<b>Betrieb .....</b>	<b>31</b>
4.1	Startverhalten.....	31
4.1.1	Startverhalten XC-CPU201 .....	32
4.1.2	Startverhalten XC-CPU202.....	33
4.1.3	Startverhalten in der CODESYS einstellen.....	34
4.2	Programmstart.....	35
4.2.1	Programmstart (STOP → RUN) .....	35
4.2.2	Programmstopp (RUN → STOP).....	35
4.3	Ausschalten/Unterbrechen der Spannungsversorgung .....	36
4.3.1	CPU-Betriebszustandsanzeige .....	36
4.3.2	Test und Inbetriebnahme (Debugging) .....	36
4.3.3	Breakpoint/Einzelschritt-Betrieb.....	37
4.3.4	Einzelzyklus-Betrieb .....	37
4.3.5	Forcen .....	37
4.3.6	Zustandsanzeige .....	37
4.3.7	Reset .....	38
4.4	Programme und Projekte .....	39
4.4.1	Programme laden.....	39
4.4.2	Allgemeines zu RETAIN PERSISTENT .....	40
4.4.3	Bootprojekt abspeichern und löschen.....	41
4.5	Betriebssystem aktualisieren.....	43
4.5.1	XC-CPU201 .....	43
4.5.2	XC-CPU202 .....	47
<b>5</b>	<b>Programmbearbeitung, Multitasking und Systemzeiten .....</b>	<b>50</b>
5.1	Taskkonfiguration.....	50
5.1.1	Zyklische Task „Basic“ anlegen.....	51
5.1.2	Ereignisgesteuerte Task „Param“ anlegen und Programmaufruf festlegen.....	52
5.2	Systemereignisse .....	53
5.2.1	POU einem Systemereignis zuordnen.....	53
5.3	Multitasking .....	54
5.3.1	Ein-/Ausgangsabbilder aktualisieren.....	54
5.3.2	Verhalten des CAN-Stacks bei Multitasking.....	57
5.4	Tasküberwachung durch Watchdog .....	57
5.4.1	Mehrere Tasks mit gleicher Priorität.....	60
5.5	Direkter Peripheriezugriff .....	60
5.5.1	ReadBitDirect.....	62
5.5.2	ReadWordDirect .....	62
5.5.3	ReadDWordDirect.....	63

5.5.4	Write...Direct .....	63
5.5.5	WriteBitDirect .....	64
5.5.6	WriteWordDirect .....	64
5.5.7	GetSlotPtr.....	65
5.5.8	Fehlercode bei direktem Peripheriezugriff .....	65
5.6	Betriebszustände .....	65
5.7	Web-Visualisierung.....	66
5.8	Grenzwerte für die Speichernutzung .....	66
5.9	Ein-/Ausgänge und Merker adressieren .....	68
5.9.1	„Adressen automatisch“ aktivieren .....	68
5.9.2	„Adressüberschneidungen prüfen“ aktivieren.....	69
5.9.3	Ungerade Wort-Adressen .....	69
5.9.4	Adressbereich .....	69
5.9.5	Adressen von Eingabe-/Ausgabe-Modulen und Diagnoseadresse frei vergeben oder verändern.....	70
5.9.6	„Adressen berechnen“ durchführen.....	70
5.10	Diagnose .....	71
<b>6</b>	<b>Verbindungsaufbau PC – XC200 .....</b>	<b>72</b>
6.1	Verbindungsaufbau über RS232-Schnittstelle.....	72
6.2	Kommunikationsparameter des PC festlegen/ändern.....	72
6.2.1	Kommunikationsparameter der CPU ändern.....	73
6.3	Verbindungsaufbau über Ethernet .....	74
6.3.1	Kommunikationskanal und Adresse auswählen.....	74
6.4	Abfragen/Ändern der IP-Adresse .....	75
<b>7</b>	<b>Systemparameter über Datei Startup.ini vorgeben .....</b>	<b>77</b>
7.1	Übersicht.....	77
7.1.1	Parameter in der Datei Startup.ini .....	77
7.2	Aufbau einer ini-Datei.....	78
7.3	Startup.ini-Datei erstellen.....	78
7.4	Eintrag der ini-Datei HOST_NAME.....	78
7.4.1	Einschalten der Steuerung bei gesteckter Speicherkarte mit Datei XCSTARTUP.ini.....	79
7.4.2	Parameter ändern.....	79
7.4.3	Startup.ini-Datei löschen .....	79
<b>8</b>	<b>Programmieren über CAN(open)-Netzwerk (Routing) .....</b>	<b>80</b>
8.1	Voraussetzungen.....	80
8.2	Routingeneigenschaften der Steuerung .....	81
8.2.1	Hinweise .....	82
8.2.2	Adressierung .....	83
8.2.3	Kommunikation mit der Zielsteuerung .....	84
8.2.4	Steuerungskombinationen zum Routing .....	86
8.2.5	Anzahl der Kommunikationskanäle .....	86

<b>9</b>	<b>RS232-Schnittstelle im Transparent-Modus.....</b>	<b>87</b>
9.1	Programmieren der RS232-Schnittstelle im Transparent-Modus	88
<b>10</b>	<b>Konfiguration und Parametrierung der Ein-/ Ausgänge.....</b>	<b>89</b>
10.1	Ein-/Ausgänge allgemein.....	89
10.1.1	Lokale Digital-Ein-/Ausgänge.....	89
10.2	Ein-/Ausgänge für Zusatzfunktionen.....	92
10.2.1	Inkrementalgeber.....	92
10.2.2	Funktionalität der Ein-/Ausgänge.....	92
10.2.3	Darstellung der Ein-/Ausgänge des Inkrementalgebers.....	94
10.2.4	Zähler.....	95
10.2.5	Darstellung der Ein-/Ausgänge des 32-Bit-Zählers.....	96
10.2.6	Darstellung der Ein-/Ausgänge der zwei 16-Bit-Zähler.....	96
10.3	Interruptverarbeitung.....	97
10.3.1	EnableInterrupt.....	98
10.3.2	Parametrierung.....	98
10.3.3	Beispiel zur Interruptverarbeitung.....	99
<b>11</b>	<b>Bibliotheken, Funktionsbausteine und Funktionen.....</b>	<b>101</b>
11.1	Bibliotheken handhaben.....	101
11.1.1	Weitere Systembibliotheken installieren.....	102
11.2	XC200-spezifische Funktionen.....	102
11.2.1	CAN_Uutilities.....	103
11.2.2	Event-Funktionen.....	104
11.2.3	XIOC-Funktionen.....	107
11.2.4	Zusätzliche Funktionen der Library XC200_Util2.lib für die XC-CPU201.....	108
11.2.5	Zusätzliche Funktionen der Library XC200_Util2.lib für die XC-CPU202.....	114
<b>12</b>	<b>Browser-Befehle.....</b>	<b>130</b>
12.1	Browser-Befehle aufrufen.....	132
12.2	Auf Kommunikationsparameter zugreifen.....	132
12.3	Auslastung der CPU anzeigen (plcload).....	133
12.3.1	Auslastung des CAN-Busses anzeigen (canload).....	133
12.3.2	Auf Speicherobjekte zugreifen.....	134
12.3.3	Fehler- und Ereignisliste nach Aufruf von Browser-Befehlen.....	135
<b>13</b>	<b>Anhang.....</b>	<b>137</b>
13.1	Eigenschaften des Ethernet-Kabels.....	137
13.2	Eigenschaften des CAN-Kabels.....	138
13.3	Transparentmodus: Textausgabe über RS232 (Beispiel).....	139
13.4	Zugriff auf die CPU-Laufwerke/Speicherkarte.....	141
13.4.1	Bibliothek SysLibFile.lib.....	141
13.4.2	Modi zum Öffnen einer Datei.....	141
13.4.3	Beispiele zu den Funktionen „SysFile...“.....	142

13.5	Abmessungen .....	143
13.6	Technische Daten .....	144
13.7	Technische Daten – Filter.....	149
	<b>Stichwortverzeichnis.....</b>	<b>151</b>

# Inhaltsverzeichnis

## 0 Zu diesem Handbuch

### 0.1 Änderungsprotokoll

Gegenüber den früheren Ausgaben hat es folgende wesentliche Änderungen gegeben:

Redaktionsdatum	Seite	Stichwort	Neu	Änderung
12/03 (Nachdruck)	38	Datenremanenz, 1. Absatz		✓
04/04	66	Grenzwerte für die Speichernutzung	✓	
	64	WriteBitDirect		✓
06/04	25, 86, 91	Externer 24-V-DC-Netzfilter für Spannungsversorgung der XC200	✓	
08/04	38	Datenremanenz, Hinweis	✓	
	42	Download von Programmen	✓	
	65	RS232-Schnittstelle des XIOC-SER im Transparent-Modus (COM2/3/4/5)	✓	
	90	Elektromagnetische Verträglichkeit		✓
11/04	15, 85	Multi Media Card (MMC), Secure Digital Card (SD), USB-Stick	✓	
	16	Splitten der ETH232-Schnittstelle		✓
	37	Zustandsanzeige		
	45	Verbindungsaufbau über RS232-Schnittstelle	✓	
	102	XC200-spezifische Funktionen	✓	
	108	Zusätzliche Funktionen der Library XC200_Util2.lib für die XC-CPU201		✓
	87	RS232-Schnittstelle im Transparent-Modus	✓	
03/05	16	Splitten der ETH232-Schnittstelle		✓
	17	Abbildung 20		✓
	67	Segmentgröße der XC-CPU201-EC256k	✓	
	68	Ein-/Ausgänge und Merker adressieren	✓	
	71	Diagnose	✓	
	80	Programmieren über CAN(open)-Netzwerk (Routing)	✓	
09/05	43	Systemparameter über Startup.ini einstellen	✓	
11/05		komplette Überarbeitung des Handbuchs		
09/06	77	Kapitel „Systemparameter über Datei Startup.ini vorgeben“		✓
	112	Kapitel „Das Netzwerk easyNet“	✓	
12/06	50	Kapitel „Programmbearbeitung, Multitasking und Systemzeiten“		✓
	80	Kapitel „Programmieren über CAN(open)-Netzwerk (Routing)“		✓
04/07	112	Kapitel „Das Netzwerk easyNet“		✓
01/08		Kapitel 13: „Das Netzwerk easyNet“ und Kapitel 14: „Programmieren über easyNet (Routing)“ sind nun im Handbuch MN05006004Z-DE.		✓
01/08 (Redaktionsdatum unverändert)	18	Abb. 7		✓
10/10		Hinzunahme XC-CPU202	✓	
08/13	145	Stromaufnahme typisch 1,4 0,85 A		✓
06/14		Firmware-Update für XC-CPU202		✓

## 0.2 Lesekonventionen



### Kurzschreibweise XC200

In diesem Handbuch wird an denjenigen Stellen, bei denen die Gerätevarianten XC-CPU201 und XC-CPU202 gleichwertig gemeint sind, die abkürzende Schreibweise XC200 verwendet. Ist speziell nur der Typ XC-CPU201 bzw. XC-CPU202 gemeint, so wird dieser explizit genannt.

In diesem Handbuch werden Symbole mit folgender Bedeutung verwendet:

- ▶ zeigt Handlungsanweisungen an.

### 0.2.1 Warnhinweise vor Sachschäden

#### **ACHTUNG**

Warnt vor möglichen Sachschäden.

### 0.2.2 Warnhinweise vor Personenschäden



#### **VORSICHT**

Warnt vor gefährlichen Situationen mit möglichen leichten Verletzungen.



#### **WARNUNG**

Warnt vor gefährlichen Situationen, die möglicherweise zu schweren Verletzungen oder zum Tod führen.



#### **GEFAHR**

Warnt vor gefährlichen Situationen, die zu schweren Verletzungen oder zum Tod führen.

### 0.2.3 Tipps



Weist auf nützliche Tipps hin.

- ▶ zeigt Handlungsanweisungen an

Die Formulierung „Wählen Sie <Datei r Neu>“ bedeutet: Aktivieren Sie den Befehl „Neu“ im Menü „Datei“.

### 0.3 Weiterführende Dokumentationen

In diesem Handbuch wird an verschiedenen Stellen auf vertiefende Beschreibungen in anderen Handbüchern hingewiesen. Diese Handbücher werden mit Titel und Dokumentationsnummer (z. B. MN05002002Z-DE) angegeben. Alle Handbücher liegen als PDF-Datei vor. Sofern sie nicht auf der Produkt-CD mitgeliefert wurden, steht die aktuelle Version der PDF-Datei im Internet zum Download zur Verfügung.

Für ein schnelles Auffinden geben Sie bitte unter der Internetadresse

<http://www.eaton.eu> | **Kundensupport** | **Download Center**  
**Dokumentation**

in der Schnellsuche die Dokumentationsnummer ein.

## 1 Aufbau der Steuerung XC200

### 1.1 Baugruppenträger

## 1 Aufbau der Steuerung XC200

Die Steuerung XC200 ist für den Einsatz in Maschinen- und Anlagensteuerungen konzipiert. Mit einer RS232-/Ethernet-Schnittstelle zum Anschluss eines Programmiergeräts, der zentralen Erweiterung mit XIOC-Signalmodulen und der dezentralen Ankopplung von CAN-Geräten bildet diese Steuerung die Basis zum Aufbau eines umfassenden Automatisierungssystems.

Die Steuerung besteht aus:

- Baugruppenträger → Seite 10
- CPU mit Netzteil und lokalen Ein-/Ausgängen → Seite 11
- XIOC-Signalmodulen → separates Handbuch „Hardware und Projektierung“ (MN05002002Z-DE).

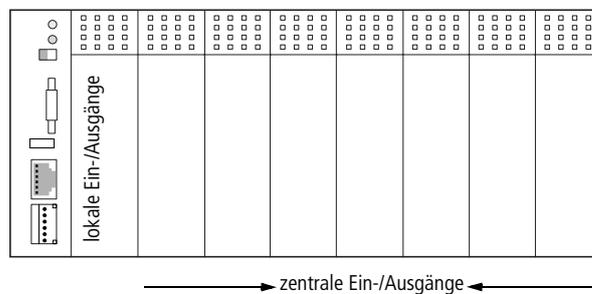


Abbildung 1: Aufbau der XC-CPU201 mit XIOC-Modulen



Für die Programmierung der XC200 ist die Programmier-Software CODESYS ab Version 2.3 erforderlich.

### 1.1 Baugruppenträger

Es gibt Basisbaugruppenträger und Erweiterungsbaugruppenträger.

Der Basisbaugruppenträger XIOC-BP-XC besitzt zwei Steckplätze für die Aufnahme der CPU. Der Typ XIOC-BP-XC1 stellt drei Steckplätze zur Verfügung, sodass neben der CPU auch noch Platz für ein XIOC-Signalmodul vorhanden ist.

Ein Basisbaugruppenträger kann mit mehreren Erweiterungsbaugruppenträgern erweitert werden. Erweiterungsbaugruppenträger werden mit XIOC-Signalmodulen bestückt.

Über eine integrierte Busschiene stellt der Baugruppenträger die Verbindung zwischen der CPU und den Modulen her.



Ausführliche Hinweise zu den Baugruppenträgern und zu den XIOC-Signalmodulen finden Sie im Handbuch „XI/OC-Signalmodule – Hardware und Projektierung“ (MN05002002Z-DE).

### 1.1.1 Leistungsumfang der CPU

Um die verschiedenen Anwendungen besser abzudecken, wird die CPU mit unterschiedlichem Leistungsumfang angeboten. Dies betrifft die Geschwindigkeit, die Größe des Anwenderspeichers sowie die Funktion des integrierten Webservers.

Es stehen folgende Typen zur Verfügung:

- XC-CPU201-EC256K-8DI-6DO(-XV),
- XC-CPU201-EC512K-8DI-6DO(-XV),
- XC-CPU202-EC4M-8DI-6DO-XV.

„EC256K“, „EC512K“ und „EC4M“ sind ein Maß für die Größe des Anwenderspeichers. „XV“ kennzeichnet eine Visualisierungs-CPU mit integriertem Webserver.

Für die Größe des Anwenderprogramms gelten folgende Speicherwerte:

	XC-CPU201-...-8DI-6DO(-XV)		XC-CPU202...
	... EC256K	... EC512K	... EC4M
Programmcode	512 kByte	2048 kByte – ab Betriebssystem 1.04.01	4 MB
Programmdaten – davon	256 kByte	512 kByte	2 MB
Merker	16 kByte	16 kByte	16 kByte
Retaindaten	32 kByte	32 kByte	64 kByte

### 1.1.2 Funktionsbereiche

Die CPU ist in drei Funktionsbereiche gegliedert:

- Spannungsversorgung → Seite 12,
- Lokale Ein- und Ausgänge → Seite 12,
- Prozessoreinheit mit Schnittstellen → Seite 14.



Abbildung 2: XC-CPU201 – eine CPU der XC200-Serie

- a Prozessoreinheit mit Schnittstellen
- b Spannungsversorgung mit lokalen Ein-/Ausgängen

# 1 Aufbau der Steuerung XC200

## 1.1 Baugruppenträger

### 1.1.3 Spannungsversorgung

Für die Versorgung der Prozessoreinheit und der lokalen Ein-/Ausgänge sind zwei getrennte Spannungszuführungen vorhanden: Zum einen besteht ein 24-V-Anschluss für die Prozessoreinheit (Beschriftung: 24V/0V) und zum anderen ein 24-V-Anschluss für die lokalen Ein-/Ausgänge (Beschriftung: 24VQ/0VQ).

Eine Abschaltlogik schaltet bei einem Spannungseinbruch der 24-V-Versorgungsspannung (Schaltschwelle ca. 10 V) die 5-V-Versorgungsspannung für die Signalmodule (zentrales I/O) ab.

### 1.1.4 Lokale Ein-/Ausgänge

In der rechten Hälfte der CPU befindet sich hinter der Frontabdeckung der 18-polige Klemmenblock. Daran schließen Sie die Spannungsversorgung der CPU und der lokalen Ein-/Ausgänge sowie die Sensoren und Aktoren an.

Die acht Digital-Eingänge (I0.0 bis I0.7) und sechs Halbleiter-Ausgänge (Q0.0 bis Q0.5) sind für 24-V-Signale ausgelegt und verfügen über eine gemeinsame Spannungsversorgung 0VQ/24VQ, die zum Bus hin potentialgetrennt ist.

Die Ausgänge Q0.0 bis Q0.5 können mit 500 mA, einer Einschaltdauer (ED) von 100 % und einem Gleichzeitigkeitsfaktor (g) von 1 belastet werden.

Die Ausgänge sind kurzschlussfest. Ein Kurzschluss sollte jedoch nicht über eine längere Zeit anstehen.

Siehe auch:

- Zusatzfunktionen der CPU (lokale Eingänge) → Seite 19

### 1.1.4.1 Klemmenbelegung

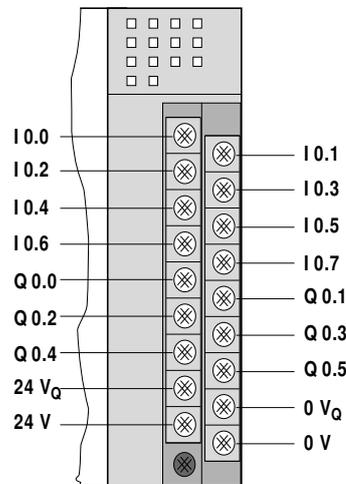


Abbildung 3: Anschlüsse des Netzteils und lokale Ein-/Ausgänge

I 0.0 bis I 0.7: lokale Digital-Eingänge

Q 0.0 bis Q 0.5: lokale Digital-Ausgänge

0V<sub>Q</sub>/+24V<sub>Q</sub>: Versorgungsspannung der lokalen Ein-/Ausgänge

0V/+24V: Versorgungsspannung der Prozessoreinheit

### 1.1.4.2 LED-Anzeigen

Die LEDs zeigen den Signalzustand der Ein- und Ausgänge an. Eine leuchtende LED repräsentiert ein High-Signal an der entsprechenden Anschlussklemme.

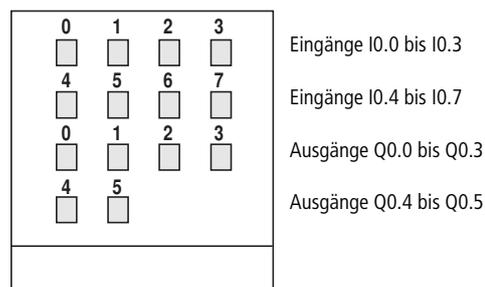


Abbildung 4: LEDs der lokalen Ein-/Ausgänge

Die oberen beiden LED-Zeilen zeigen den Signalzustand der acht Digital-Eingänge des CPU-Moduls (I 0.0 bis I 0.7), die unteren beiden LED-Zeilen zeigen den Signalzustand der sechs Digital-Ausgänge (Q 0.0 bis Q 0.5) an.

# 1 Aufbau der Steuerung XC200

## 1.1 Baugruppenträger

### 1.1.5 Prozessoreinheit mit Schnittstellen

Zur Prozessoreinheit gehören:

- Echtzeituhr → Seite 14,
- Batterie → Seite 14,
- Multi Media Card (MMC), Secure Digital Card (SD), USB-Stick → Seite 15,
- CPU-Laufwerke → Seite 15,
- USB-Schnittstelle → Seite 29,
- Programmierschnittstelle ETH232 → Seite 16,
- CAN-/easyNet-Schnittstelle → Seite 17,
- Zusatzfunktionen der CPU (lokale Eingänge) → Seite 19.

### 1.1.6 Echtzeituhr

Die XC200 besitzt eine Echtzeituhr, die Sie im Anwenderprogramm über Funktionen aus der Bibliothek SysLibRTC.lib ansprechen können.

Mögliche Funktionen sind:

- Anzeige der Batteriezustands,
- Darstellungsmodus der Stunden (12-/24-Stunden-Anzeige),
- Auslesen und Setzen der Echtzeituhr.

Eine Beschreibung der Funktionen finden Sie in der Datei SysLibRTC.pdf.

Darüber hinaus können Sie die Echtzeituhr über folgende Browser-Befehle setzen bzw. abfragen:

- `setrtc` (Setzen der Echtzeituhr) → Seite 131,
- `getrtc` (Abfragen der Echtzeituhr) → Seite 130.

### 1.1.7 Batterie

Zur Sicherung remanenter Daten und für den Betrieb der Echtzeituhr wird eine Lithium-Batterie vom Typ 1/2 AA (3,6 V) verwendet. Der Batterieschacht befindet sich an der linken Außenseite der CPU hinter einer Abdeckplatte. Der Ladezustand der Batterie wird überwacht. Unterschreitet die Batteriespannung einen fest vorgewählten Grenzwert, wird eine Sammelfehlermeldung ausgegeben.

Die Pufferzeiten betragen:

- Worst Case: 3 Jahre bei Dauerpufferung,
- typisch: 5 Jahre bei Dauerpufferung.

#### **ACHTUNG**

Nehmen Sie einen Batteriewechsel nur bei eingeschalteter Versorgungsspannung vor. Es können andernfalls Daten verlorengehen.

Die Bestellbezeichnung der Batterie lautet: XT-CPU-BAT1

### 1.1.8 Multi Media Card (MMC), Secure Digital Card (SD), USB-Stick

MMC, SD und USB dienen als Massenspeicher. Sie können darauf Rezepturdaten, allgemeine Daten und das Anwenderprogramm laden. Das Betriebssystem unterstützt Speichertypen mit dem FAT16-Dateisystem.

Sie können anstelle einer MMC-Speicherkarte auch eine SD-Speicherkarte verwenden.

Ab der Betriebssystemversion 01.03. der XC-CPU201 können Sie das Betriebssystem auf die MMC übertragen, um es von dort in andere Steuerungen zu laden (Betriebssystem aktualisieren). Ebenso ist ab dieser Betriebssystemversion in der XC-CPU201 der Einsatz eines USB-Sticks zur Datenspeicherung möglich.

#### **ACHTUNG**

Das Dateisystem der Speicherkarten ist nicht transaktions-sicher. Stellen Sie daher sicher, dass vom Programm alle Dateien geschlossen worden sind, bevor Sie eine Karte ziehen/stecken oder die Spannung ausschalten.

Siehe auch:

- Bei der XC-CPU-202: → Seite 42
- Betriebssystem/Bootprojekt von der MMC der XC-CPU201 löschen → Seite 46
- Belegung der USB-Schnittstelle → Seite 29



Das Löschen von Dateien erfolgt analog dem Löschen des Betriebssystems.

### 1.1.9 CPU-Laufwerke

Die XC200 verfügt über folgende Laufwerke:

- intern
  - Systemspeicher (disk\_sys)
- extern (optional)
  - Multi Media Card MMC oder Secure Digital Card SD (disk\_mmc)
  - USB-Stick (disk\_usb).

Auch wenn eine SD-Karte verwendet wird, heißt das externe CPU-Laufwerk weiterhin disk\_mmc.

Im transaktionssicherem Systemspeicher sind das Bootprojekt und das Betriebssystem in komprimierter Form spannungsausfallsicher abgelegt. Im Betriebszustand werden das Bootprojekt und die relevanten Teile des Betriebssystems „entpackt“ in den Arbeitsspeicher kopiert. Die remanenten Daten werden im batteriegepufferten SRAM-Speicher abgelegt.

## 1 Aufbau der Steuerung XC200

### 1.1 Baugruppenträger



„Transaktionssicher“ bedeutet, dass ein Spannungsausfall während der Bearbeitung einer Datei in der Regel das Dateisystem sowie die geöffnete Datei nicht zerstört. Möglich ist allerdings, dass die Daten verlorengehen, die Sie in die zuletzt geöffnete Datei geschrieben haben.

Abbildung 5 zeigt das Zusammenwirken der verschiedenen Speichersysteme/Laufwerke der XC-CPU200.

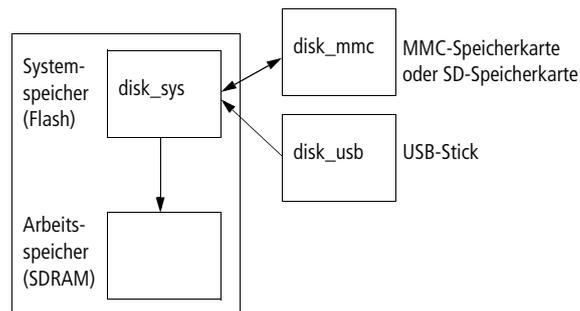


Abbildung 5: Speicherorganisation der XC-CPU200

Siehe auch:

- Datenzugriff auf die Speicherkarte
  - mit Hilfe von Browser-Befehlen wie z. B. copyprojtommc, um das Anwenderprogramm auf die MMC zu kopieren → Seite 130
  - Funktionen wie SysFileOpen oder SysFileRead → Seite 141
- Grenzwerte für die Speichernutzung → Seite 66

#### 1.1.10 Programmierschnittstelle ETH232

Über die Programmierschnittstelle ETH232 der CPU erfolgt die Kommunikation zwischen Steuerung und dem Programmiergerät. Sie besteht aus einer Ethernet-Schnittstelle und einer RS232-Schnittstelle.

Die Ethernet-Schnittstelle nutzen Sie zum Programmieren und zum Debuggen, da sie vom Betriebssystem schneller abgearbeitet wird. Diese Schnittstelle ist netzwerkfähig und galvanisch getrennt.

Bei der XC-CPU202 können Sie über die Ethernet-Schnittstelle auch das Betriebssystem der Steuerung aktualisieren.

Über die RS232-Schnittstelle können Sie ebenfalls programmieren.

Ab der Betriebssystemversion V01.03 bei der XC-CPU201 können Sie die RS232-Schnittstelle auch in den Transparentmodus schalten, um eine Punkt-zu-Punkt-Verbindung ohne Handshake-Leitungen aufzubauen.

#### 1.1.11 Splitten der ETH232-Schnittstelle

Mit Hilfe der Kabelweiche XT-RJ45-ETH-RS232 können Sie bei der XC200 gleichzeitig über die RS232- und die Ethernet-Schnittstelle kommunizieren.

Mit dem Kabel EASY-NT-30/80/130 stellen Sie die Verbindung zwischen der CPU und der Kabelweiche her. Von der Steckerbuchse „IN“ der Kabelweiche führen Sie das Kabel dann zu der ETH232-Steckerbuchse der CPU.

An die Ethernet-Schnittstelle der Kabelweiche können Sie zum Beispiel das Programmiergerät und an die RS232-Schnittstelle (im Transparent-Modus) einen Drucker anschließen. Die Stiftbelegung der RS232- und der Ethernet-Steckerbuchse der Kabelweiche entspricht der Stiftbelegung der ETH232-Steckerbuchse der CPU.

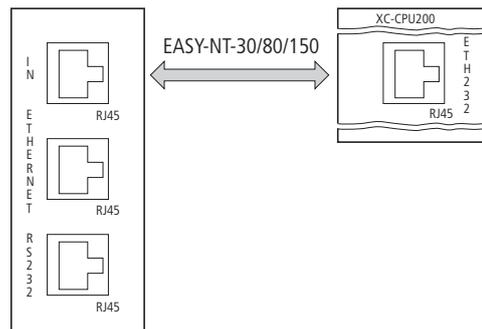


Abbildung 6: Verbindung XC-CPU200 mit XT-RJ45-ETH-RS232

Siehe auch:

- PC anschließen → Seite 28
- Belegung der Programmierschnittstelle → Seite 29
- Verbindungsaufbau PC – XC200 → Seite 72.

### 1.1.12 CAN-/easyNet-Schnittstelle

Die CAN-/easyNet-Schnittstelle ist potentialgetrennt. Die Anschlüsse der Schnittstellen sind gleich. Die CPU kann am CAN-Bus sowohl als Netzwerk-(NMT-)Master als auch als NMT-Slave (Device) betrieben werden. Das CANopen- und das easyNet-Protokoll können die CPU parallel ausführen.

Siehe auch:

- Ausführliche Informationen zur Projektierung und Programmierung von CAN-Teilnehmern  
a Anwenderhinweis AN2700K27
- Bedienungshandbuch easy800 Steuerrelais (MN04902001Z-DE; frühere Bezeichnung AWB2528-1423)
- Das Netzwerk easyNet a Seite 73

### Busabschlusswiderstände

Beim ersten oder letzten Teilnehmer am Strang müssen Busabschlusswiderstände eingesetzt werden.

Teilnehmer	Busabschlusswiderstand
XC-CPU201, MFD4	120 Ω (extern)
easy800, MFD	EASY-NT-R
XC-CPU202	schaltbar

# 1 Aufbau der Steuerung XC200

## 1.1 Baugruppenträger

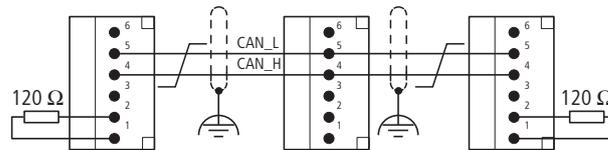


Abbildung 7: Beispiel: Netz mit Busabschlusswiderständen an der XC-CPU201

Die Klemmen 1 und 4, 2 und 5 sowie 3 und 6 sind intern verbunden.

Bei der XC-CPU202 ist der Busabschluss schaltbar ausgeführt. Der Schalter befindet sich oberhalb des Batterieschalters (Auslieferungszustand: ON).

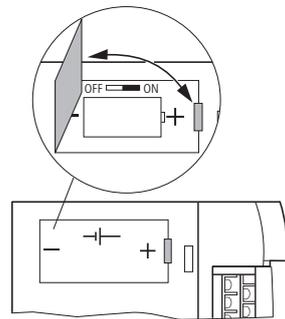


Abbildung 8: XC-CPU202, XT-CPU-BAT1

### 1.1.13 Verhalten der Teilnehmer am CAN-Bus

Teilnehmer/Bus überwachen: CAN-Telegramme werden direkt aus dem Anwenderprogramm gesendet und empfangen. Eine Unterbrechung des CAN-Busses wird nur erkannt, wenn die entsprechenden CAN-Teilnehmer von der Steuerung her überwacht werden (Nodeguarding-Funktion).

Start/Stop-Verhalten:

Wenn Sie den Betriebsartenschalter (BAS) in die Stellung STOP setzen, werden alle Ausgänge der dezentralen Geräte am Zyklusende auf „0“ gesetzt.

Spannung einschalten:

Die Reihenfolge, in der Sie die Spannungsversorgung der einzelnen CAN-Teilnehmer zuschalten, hat keine Auswirkung auf die Funktionalität des CAN-Busses. Je nach Parametrierung „wartet“ die Steuerung auf nicht vorhandene Teilnehmer oder startet diese zu demjenigen Zeitpunkt, zu dem der Teilnehmer an das CAN-Netz angeschaltet wird.

Kommunikation mit CAN-Teilnehmern:

Die Kommunikation mit den CAN-Teilnehmern und deren Konfiguration wird in den folgenden Anwenderhinweisen und Handbüchern beschrieben:

- Ankopplung einer XION-Station an die XC100/200 über CAN (AN27K18D)
- Kommunikation zwischen zwei Steuerungen mit Netzwerkvariablen über CAN (AN27K19D)

- Kopplung mehrerer autarker Steuerungen (CAN-Device) über CAN (AN27K20D)
- Projektierung von CAN-Teilnehmern (AN27K27D)
- Bibliotheksbeschreibung: CANUser.lib/CANUser\_Master.lib (MN05010001Z-DE).

Siehe auch:

- Belegung der CAN/easyNet-Schnittstelle → Seite 30
- Eigenschaften des CAN-Kabels → Seite 138

### 1.1.14 Zusatzfunktionen der CPU (lokale Eingänge)

Die Eingänge I0.0 bis I0.5 können Sie parametrieren als:

- Inkrementalgeber-Eingänge (I0.0 bis I0.3)
- Zähler-Eingänge (32 Bit I0.0, I0.1)
- Zähler-Eingänge (16 Bit I0.0, I0.1 und I0.2, I0.3)
- Interrupt-Eingänge (I0.4 und I0.5)

Bei diesen Funktionen werden die Eingangssignale in der CPU vorverarbeitet.

#### 1.1.14.1 Inkrementalgeber-Eingang (32 Bit)

Die Funktion steht einmal zur Verfügung. Auf die Eingänge I0.0 und I0.1 werden die Inkrementalsignale A und B des Gebers geführt, auf den Eingang I0.2 das Referenzsignal, das der Geber einmal pro Umdrehung erzeugt. An den Eingang I0.3 wird der Schalter angeschlossen, der im geschlossenen Zustand das Referenzfenster bildet, in dem das Referenzsignal I0.2 verarbeitet wird.

Die Inkrementalsignale A und B sind zur Ermittlung der Zählrichtung um 90 Grad phasenverschoben. Es werden die steigenden und fallenden Flanken ausgewertet (4-Fach-Auswertung). Die maximale Eingangsfrequenz beträgt 50 kHz. Damit ergibt sich eine Gesamtfrequenz von 200 kHz.

Siehe auch:

- Inkrementalgeber anschließen → Seite 26
- Inkrementalgeber parametrieren → Seite 92

#### 1.1.14.2 Vor-/Rückwärtszähler (32 Bit)

Die Funktion steht einmal zur Verfügung. Der Zähl Eingang I0.0 nimmt die Impulse mit einer maximalen Frequenz von 50 kHz auf. Das Richtungssignal am Eingang I0.1 gibt an, ob der Zählwert bei Eintreffen eines Zählimpulses inkrementiert oder dekrementiert wird. Das Richtungssignal ist ein statisches Signal, das vor den Zählimpulsen anstehen muss. Der Zählwert wird mit jedem Zählimpuls in- bzw. dekrementiert, bis der Sollwert erreicht ist.

Nach Erreichen des Sollwertes wird ein Interrupt ausgelöst, mit dem in eine Programmroutine (POU) verzweigt wird. Das Verhalten nach Erreichen des Sollwertes ist von der Zählweise abhängig:

# 1 Aufbau der Steuerung XC200

## 1.1 Baugruppenträger

**Inkrementieren:** Wird der Sollwert erreicht, wird der parametrisierte Interrupt aktiviert. Beim nächsten Zählimpuls beginnt der Zähler bei 0. Die Interruptquelle geben Sie im Steuerungskonfigurator an.

**Dekrementieren:** Wird der Sollwert erreicht, wird der parametrisierte Interrupt aktiviert. Beim nächsten Zählimpuls beginnt der Zähler ab demvorgeählten Sollwert. Die Interruptquelle geben Sie im Steuerungskonfigurator an.

Siehe auch:

- Interruptverarbeitung → Seite 97
- Eingabe des Sollwertes in der Steuerungskonfiguration → Seite 95
- Vor-/Rückwärtszähler anschließen → Seite 27

### 1.1.14.3 Vor-/Rückwärtszähler (16 Bit)

Die Funktion steht zweimal zur Verfügung. Sie entspricht dem Vor-/Rückwärtszähler (32 Bit).

Eingänge	
Zähler 1:	
I0.0	Impulseingang
I0.1	Richtungseingang
Zähler 2:	
I0.2	Impulseingang
I0.3	Richtungseingang

### 1.1.14.4 Interrupt-Eingänge

Die Digital-Eingänge I0.4 und I0.5 können als Interrupt-Eingänge parametrisiert werden. Ausgewertet wird jeweils die L/H-Flanke bzw. die H/L-Flanke (parametrierbar) der Eingangssignale.



Wenn Sie eine XC100-Steuerung durch eine XC200-Steuerung ersetzen, liegen die Interrupt-Eingänge auf anderen physikalischen Eingangsadressen!

Siehe auch:

- Zeitliche Anforderung an die Interrupt-Eingänge: „Technische Daten – Eingangsverzögerung – Schneller Digital-Eingang“ → Seite 147
- Die Programmierung der „Interrupt-Funktion“ ist auf Seite 97 beschrieben
- Vor-/Rückwärtszähler anschließen → Seite 27

## 2 Montage der CPU



Ausführliche Hinweise zur Montage des Baugruppenträgers und zu den XIOC-Modulen finden Sie im Handbuch „XI/OC-Signalmodule – Hardware und Projektierung“ (MN05002002Z-DE).

- ▶ Stecken Sie die Lasche an der Unterseite der CPU in das Loch des Baugruppenträgers ①.
- ▶ Drücken Sie die Oberseite der CPU in den Baugruppenträger, bis Sie ein Klicken hören ②.

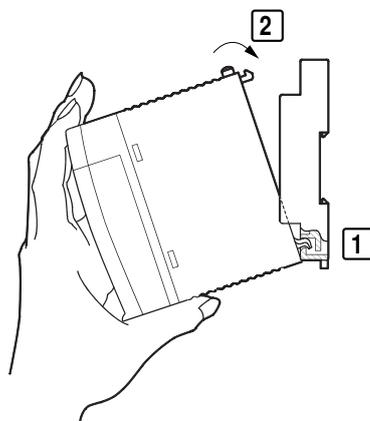


Abbildung 9: Montage der CPU

### 2.0.1 Demontage der CPU

- ▶ Drücken Sie den Verriegelungsknopf ①.
- ▶ Ziehen Sie bei gedrücktem Knopf das Oberteil der CPU nach vorne ②.
- ▶ Heben Sie die CPU an und ziehen Sie sie heraus ③.

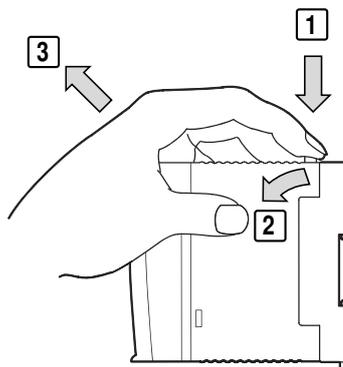


Abbildung 10: Demontage der Module

## 3 Projektierung

### 3.1 Schaltschrankaufbau

Die Anordnung der Komponenten im Schaltschrank hat wesentlichen Einfluss auf die ungestörte Anlagen- oder Maschinenfunktion. Bei der Planung, Entwurfsphase sowie bei der Ausführung ist darauf zu achten, dass Leistungs- und Steuerteil getrennt voneinander angeordnet werden.

Zum Leistungsteil zählen unter anderem:

- Schütze,
- Koppelbausteine,
- Transformatoren,
- Frequenzumrichter,
- Stromrichter.

Um eine elektromagnetische Beeinflussung wirksam auszuschließen, ist es zweckmäßig, eine Aufteilung in Bereiche von unterschiedlichen Leistungs- und Störniveaus vorzunehmen. Bei kleinen Schaltschränken genügt oft schon eine Abschottung durch Trennbleche, um Störeinflüsse zu reduzieren.

#### 3.1.1 Belüftung

Um eine ausreichende Belüftung zu gewährleisten, sind Mindestabstände von 50 mm zu passiven Komponenten einzuhalten. Handelt es sich bei den benachbarten Komponenten um aktive Elemente (z. B. Laststromversorgung, Transformatoren), müssen Sie einen Minimalabstand von 75 mm einhalten. Die in den technischen Daten angegebenen Werte müssen eingehalten werden.

### 3.1.2 Geräteanordnung

Bauen Sie die Baugruppenträger und die Steuerung waagrecht in den Schaltschrank ein.

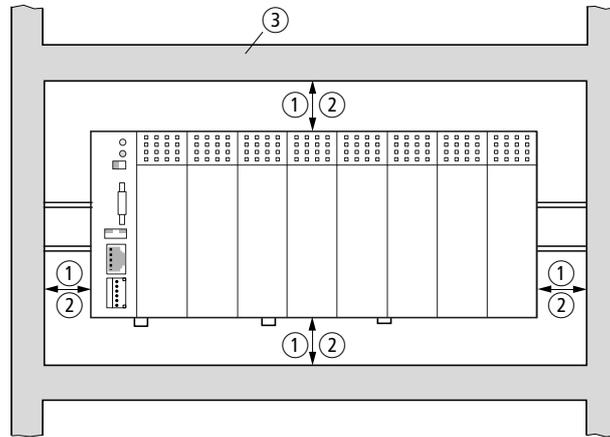


Abbildung 11: Schaltschrankaufbau

- a Abstand > 50 mm
- b Abstand > 75 mm zu aktiven Elementen
- c Kabelkanal

### 3.2 Störungen vermeiden

#### 3.2.1 Leitungsführung und Verdrahtung

Es existieren folgende Kategorien von Leitungen:

- Starkstromleitungen (z. B. Leistungsleitungen, die hohe Ströme führen, oder Leitungen zu Stromrichtern, Schützen, Magnetventilen),
- Steuer- und Signalleitungen (z. B. Digitaleingabeleitungen),
- Mess- und Signalleitungen (z. B. Feldbusleitungen).



Verlegen Sie Starkstrom-, Steuer- und Signalleitungen immer so weit wie möglich voneinander entfernt. Auf diese Weise vermeiden Sie kapazitive und induktive Einkopplungen. Ist eine getrennte Leitungsführung nicht möglich, sollte in erster Linie die Störleitung geschirmt werden.

Achten Sie auf eine richtige Leitungsführung innerhalb und außerhalb des Schaltschranks, um die Störungen so gering wie möglich zu halten:

- ▶ Vermeiden Sie parallel geführte Abschnitte von Leitungen unterschiedlichen Leistungsniveaus.
- ▶ Trennen Sie grundsätzlich Wechselspannungsleitungen von Gleichspannungsleitungen.

## 3 Projektierung

### 3.3 Blitzschutzmaßnahmen

- ▶ Halten Sie die folgenden Mindestabstände ein:
  - zwischen Starkstromleitungen und Signalleitungen mindestens 10 cm;
  - zwischen Starkstrom und Daten- bzw. Analogleitungen mindestens 30 cm.
- ▶ Achten Sie bei der Leitungsführung darauf, dass Hin- und Rückleiter eines Stromkreises zusammen geführt werden. Durch den gegenseitigen Stromfluss wird die Summe aller Ströme bei diesem Leitungspaar null. Die entstehenden Felder werden kompensiert.

#### 3.2.2 Schutzbeschaltung von Störquellen

- ▶ Bringen Sie alle Schutzbeschaltungen so dicht wie möglich an der Störquelle (Schütz, Relais, Ventil) an.



Geschaltete Induktivitäten sollten grundsätzlich geschützt werden.

#### 3.2.3 Schirmung

- ▶ Verwenden Sie zum Anschluss an die Datenschnittstellen Kabel mit Schirm.

Generell gilt:

Je kleiner die Kopplungsimpedanz ist, desto besser ist die Schirmwirkung.

### 3.3 Blitzschutzmaßnahmen

#### 3.3.1 Äußerer Blitzschutz

Alle gebäudeübergreifenden Leitungen müssen eine Abschirmung erhalten. Metallrohre sind hierfür am besten geeignet. Verwenden Sie für Signalleitungen Schutzelemente gegen Überspannungen wie beispielsweise Varistoren oder andere Überspannungsableiter. Nehmen Sie diese Maßnahme möglichst bei Kabeleintritt in das Gebäude, spätestens aber am Schaltschrank vor.

#### 3.3.2 Innerer Blitzschutz

Der innere Blitzschutz erfasst alle Maßnahmen, die die Auswirkungen des Blitzstromes und seiner elektrischen und magnetischen Felder auf metallische Installationen und elektrische Anlagen in einer baulichen Anlage reduzieren. Dabei handelt es sich um:

- den Blitzschutz-Potentialausgleich,
- die Abschirmung,
- den Einsatz von Überspannungsschutzgeräten.

## 3.4 Anschlüsse

### 3.4.1 Spannungsversorgung anschließen

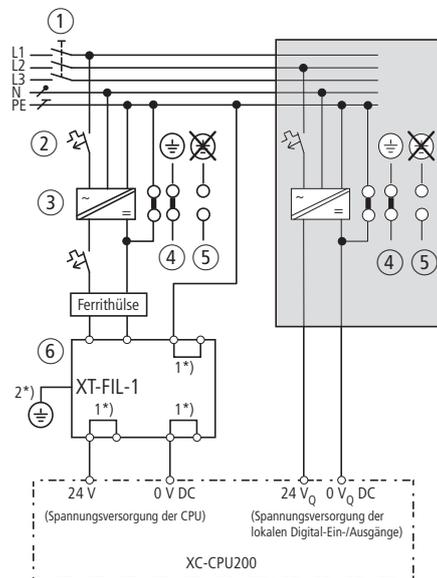


Abbildung 12: Verdrahtungsbeispiel für Netzteil

- a Hauptschalter
- b Leitungsschutzorgan
- c 24-V-DC-Versorgungsspannung
- d Geerdeter Betrieb
- e Bei ungeerdetem Betrieb muss eine Isolationsüberwachung eingesetzt werden (IEC 204-1, EN 60204-1, DIN EN 60204-1)
- f 24-V-DC-Netzfilter; stellt sicher, dass bei einer Bemessungsspannung von 24 V DC ein Strom von bis zu 2,2 A (maximal) zur Verfügung steht. Mit dem Einsatz des Filters werden die Vorgaben des EMVG erfüllt.  
Hinweis: Der Filter ist nicht Bestandteil der CPU und muss deshalb getrennt bestellt werden:  
Typ: XT-FIL-1, Artikel-Nr.: 285316 (Lieferant: Eaton Industries GmbH)  
→ Abmessungen auf Seite 143  
→ Technische Daten auf Seite 149.

1\*) intern gebrückt

2\*) zusätzliche PE-Verbindung über Kontaktfeder auf der Rückseite

### 3.4.2 Ein-/Ausgänge (CPU) anschließen

Die Abbildung zeigt den Anschluss der Ein-/Ausgänge und deren Spannungsversorgung.

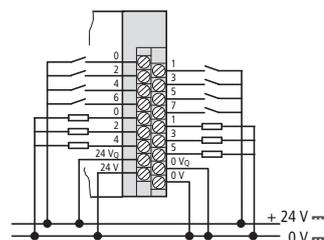


Abbildung 13: Ein-/Ausgänge der CPU anschließen

### 3 Projektierung

#### 3.4 Anschlüsse



Verdrahtungsbeispiele zu den XIOC-Modulen finden Sie im Handbuch „Hardware und Projektierung der XI/OC-Signalmodule“ (MN05002002Z-DE).

#### 3.4.3 Inkrementalgeber anschließen

Der Inkremental-Drehgeber wird, wie in der nachfolgenden Abbildung dargestellt, an die Steuerung angeschlossen.

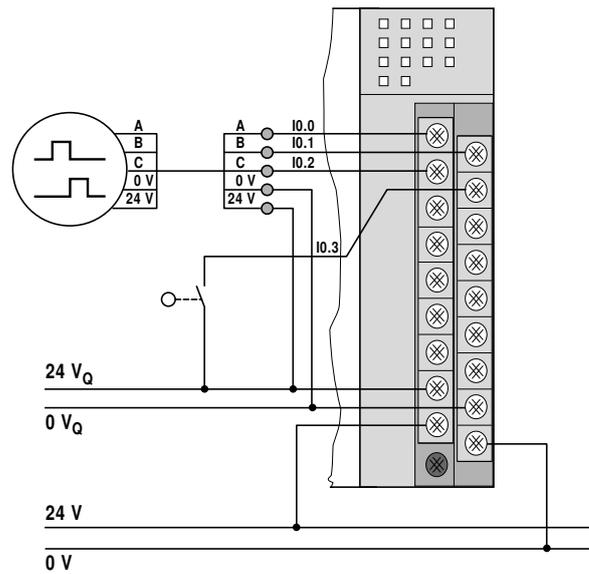


Abbildung 14: Anschluss Inkremental-Drehgeber mit Referenzfenster-Schalter

### 3.4.4 Vor-/Rückwärtszähler anschließen

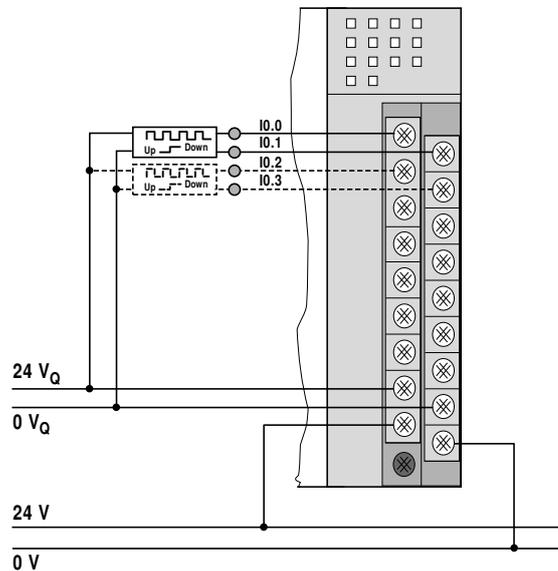


Abbildung 15: Anschluss Impulsgeber mit Signal zum In-/Dekrementieren

### 3.4.5 Interrupt-Aktoren anschließen

Die Eingänge I0.4 und I0.5 können Sie als Interrupt-Eingänge nutzen.

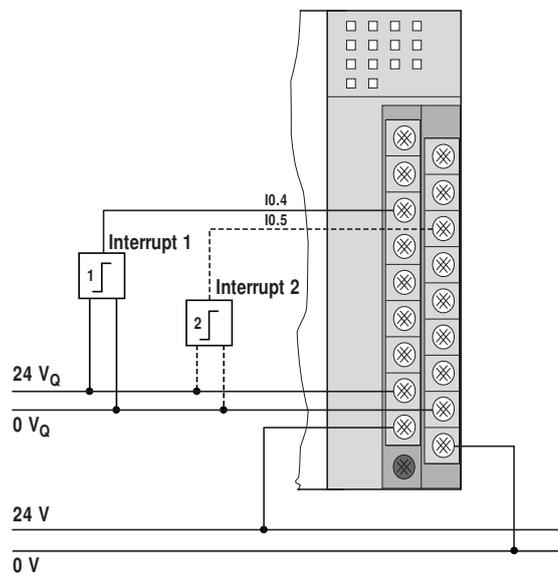


Abbildung 16: Anschluss der Interrupt-Eingänge



Bitte beachten Sie, wenn Sie eine XC100-Steuerung durch eine XC200-Steuerung ersetzen, dass die Interrupt-Eingänge auf anderen physikalischen Eingangsadressen liegen!

### 3.4.6 PC anschließen

#### 3.4.6.1 Ethernet-Verbindung

Physikalisch ist die Programmiergeräte-Schnittstelle als RJ45-Schnittstelle (Buchse) ausgebildet. Somit können handelsübliche Ethernet-(Patch-)Kabel mit RJ45-Steckern eingesetzt werden.

##### Direktverbindung PC – XC200

Die XC200 können Sie direkt mit dem (Programmier-)PC über ein Cross-Over-Ethernet-Kabel verbinden, → Abbildung 17, 18.

Cross-Over-Kabel haben folgenden Aufbau:

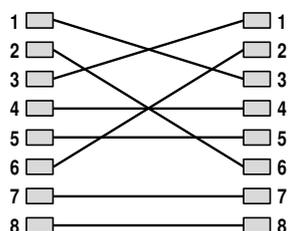


Abbildung 17: Verbindungsaufbau Cross-Over-Kabel 8-adrig

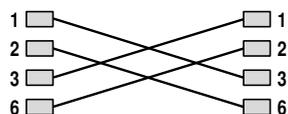


Abbildung 18: Verbindungsaufbau Cross-Over-Kabel 4-adrig

Folgende Cross-Over-Kabel stehen zur Verfügung:

- XT-CAT5-X-2 2 m lang (Art.-Nr. 256487)
- XT-CAT5-X-5 5 m lang (Art.-Nr. 256488)

##### Verbindung PC – XC200 über Hub/Switch

Wenn Sie zwischen die Verbindung PC – XC200 einen Hub oder Switch einsetzen, müssen Sie für die Verbindung zwischen PC – Hub/Switch und Hub/Switch – XC200 ein Ethernet-Standardkabel verwenden, dessen Adern 1:1 verbunden sind.

Für die Programmierung über die USB-Schnittstelle eines PC steht das Kabel EU4A-RJ45-USB-CAB1 (Art.-Nr. 115735) zur Verfügung.



Beachten Sie bitte, dass durch die Doppelbelegung der RJ45-Schnittstelle mit RS232 und Ethernet auf den Anschlüssen 4 und 7 wegen der RS232-Schnittstelle „GND-Potential“ liegt. Aus diesem Grund empfehlen wir den Einsatz von vieradrigen Kabeln für die Anbindung der XC200 an das Ethernet.

Siehe auch:

- Eigenschaften des Ethernet-Kabels → Seite 137

### 3.4.6.2 RS232-Verbindung

Zur Herstellung einer Verbindung zwischen XC200 und PC verwenden Sie das Programmierkabel XT-SUB-D/RJ45 (Art.-Nr. 262186).

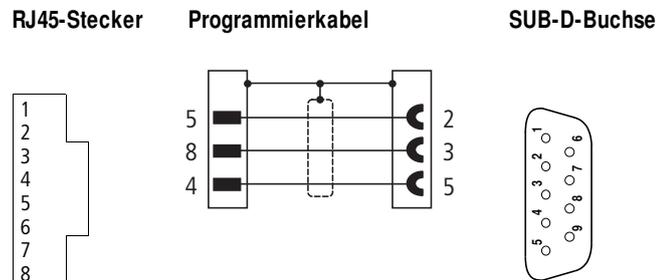


Abbildung 19: Aderbelegung RS232-Programmierkabel

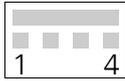
Siehe auch:

- Verbindungsaufbau PC – XC200 → Seite 72
- RS232-Schnittstelle im Transparent-Modus → Seite 87

## 3.5 Schnittstellenbelegungen

### 3.5.1 USB-Schnittstelle

Tabelle 1: Belegung der USB-Schnittstelle

	Signal
	
1	+5 V $\overline{\text{---}}$
2	USB-
3	USB+
4	GND

### 3.5.2 Programmierschnittstelle XC200

Tabelle 2: Belegung der Programmierschnittstelle

RJ45-Buchse	Signal		
	RS232	Ethernet	
	8	RxD	–
	7	GND	–
	6	–	Rx-
	5	TxD	–
	4	GND	–
	3	–	Rx+
	2	–	Tx-
	1	–	Tx+

### 3 Projektierung

#### 3.5 Schnittstellenbelegungen



Bei der XC-CPU202 ist die Ethernet-Buchse um 180 Grad gedreht eingebaut. Die Pinbelegung ist jedoch identisch mit der der XC-CPU201.

### 3.5.3 CAN/easyNet-Schnittstelle

Tabelle 3: Belegung der CAN/easyNet-Schnittstelle

	Klemme	Signal	
		CAN	easyNet
	6	GND	GND
	5	CAN_L	ECAN_L
	4	CAN_H	ECAN_H
	3	GND	GND
	2	CAN_L	ECAN_L
	1	CAN_H	ECAN_H



Steckertyp: 6-poliger, steckbarer Federzugklemmenblock,  
 Leiteranschlüsse: bis 0,5 mm<sup>2</sup>  
 Die Klemmen 1 und 4, 2 und 5 sowie 3 und 6 sind intern verbunden.

## 4 Betrieb

### 4.1 Startverhalten

Auf der CPU können mehrere unterschiedliche Anwenderprogramme/Bootprojekte abgespeichert sein. Diese können sich sowohl auf der MMC/SD/USB als auch auf dem Systemspeicher disk\_sys befinden. Die CPU führt jedoch nur ein Anwenderprogramm aus.

Die folgenden beiden Ablaufdiagramme (Abb. 20 und Abb. 21) zeigen, welches Programm zum Einsatz kommt. Die Diagramme zeigen auch die Aktualisierung des Betriebssystems (BTS) mit Hilfe der MMC/SD/USB.

Nach der Spannungswiederkehr wird nur ein in der XC200 gespeichertes Bootprojekt entsprechend der Stellung des Betriebsartenschalters und der parametrisierten Startbedingungen gestartet.

4 Betrieb  
4.1 Startverhalten

4.1.1 Startverhalten XC-CPU201

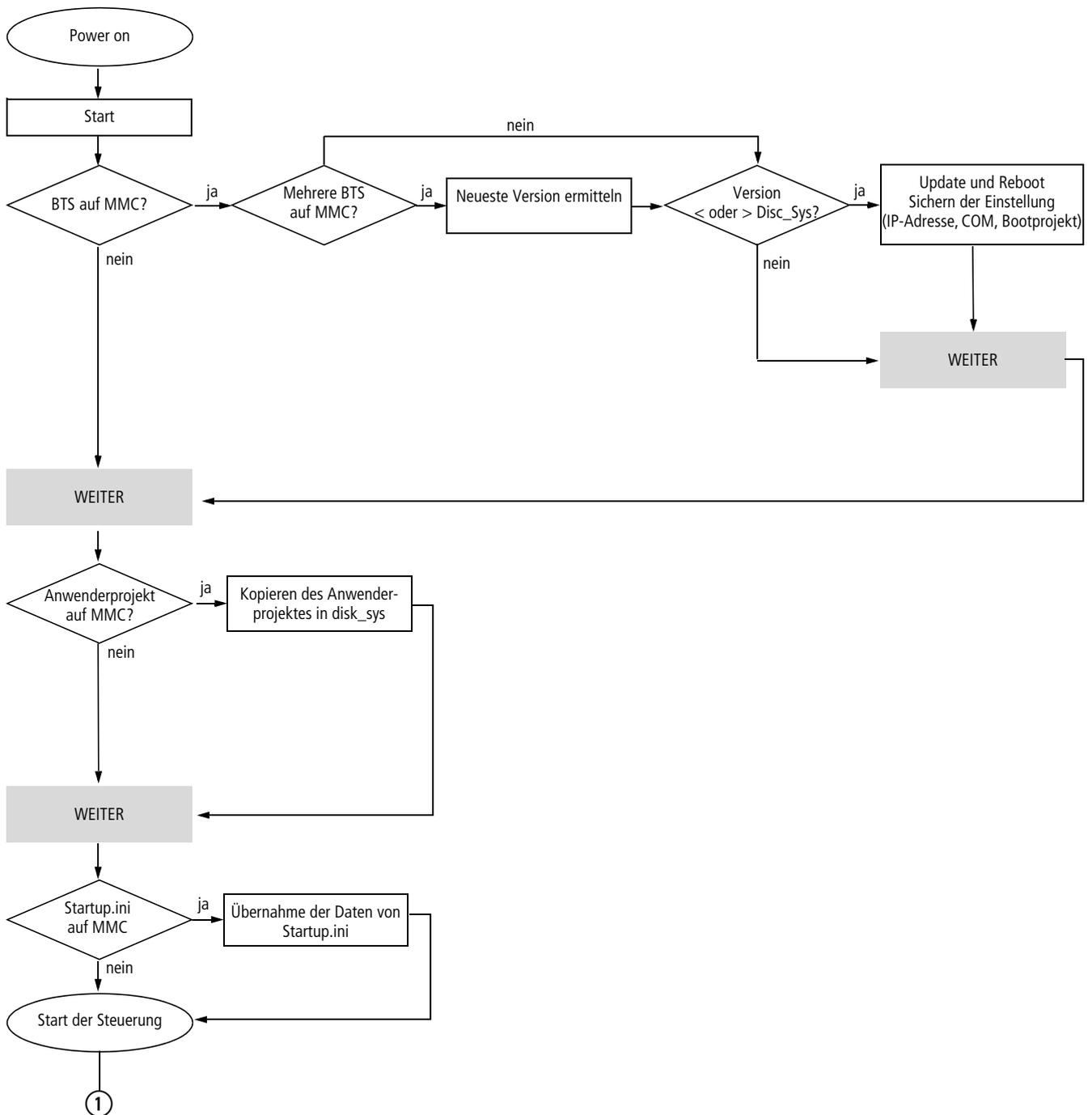
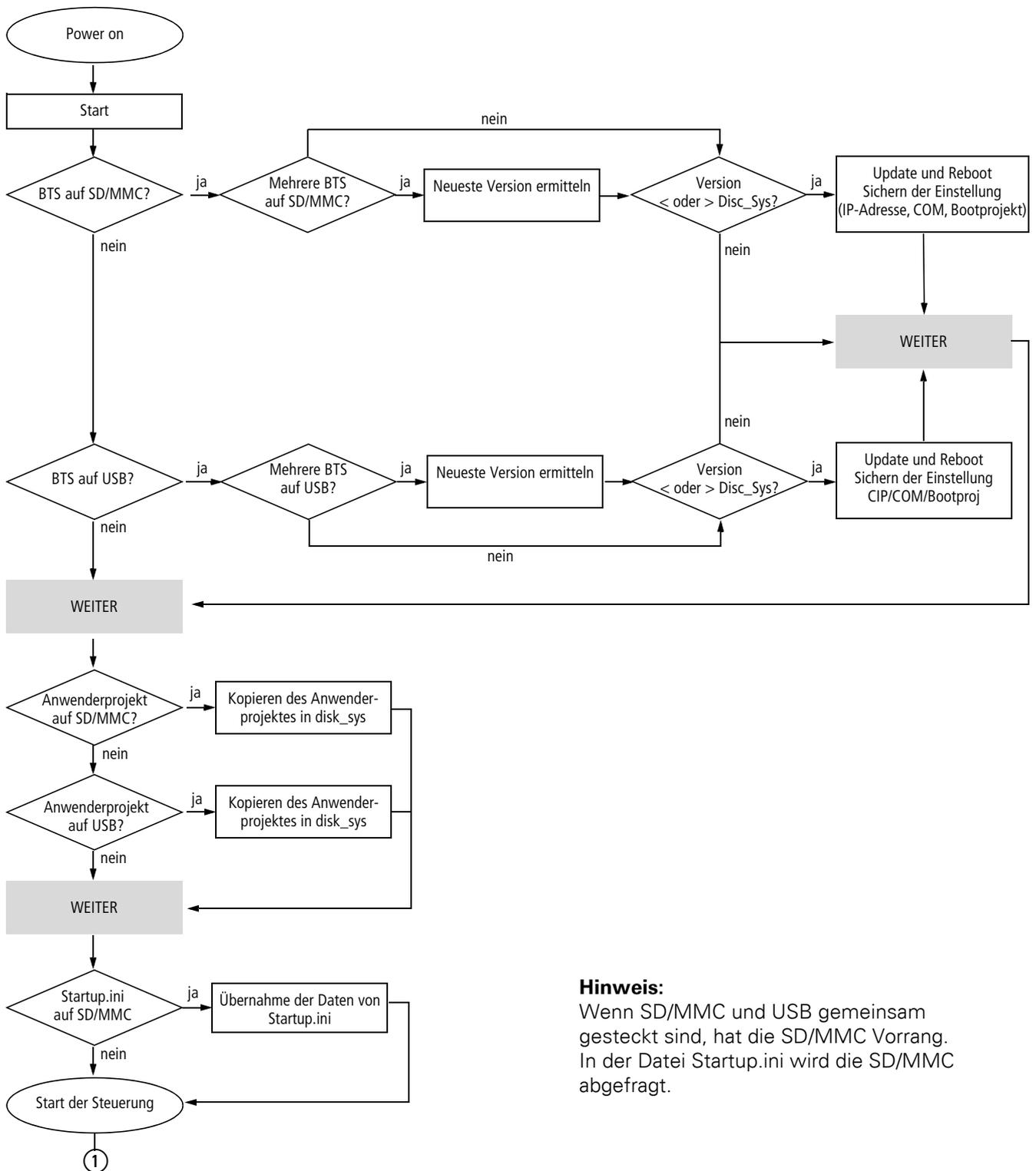


Abbildung 20: Ablauf Boot mit MMC/SD

### 4.1.2 Startverhalten XC-CPU202

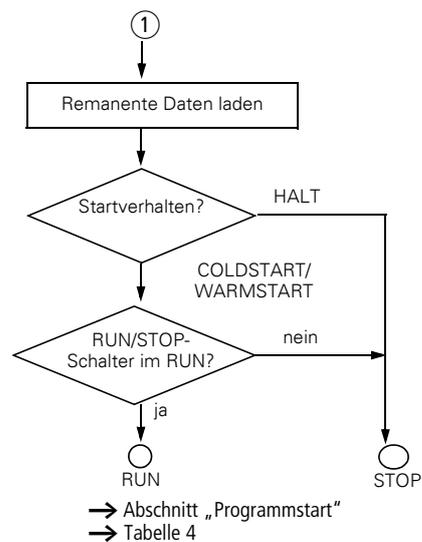


**Hinweis:**  
Wenn SD/MMC und USB gemeinsam gesteckt sind, hat die SD/MMC Vorrang. In der Datei Startup.ini wird die SD/MMC abgefragt.

Abbildung 21: Ablauf Boot mit SD/MMC und USB

## 4 Betrieb

### 4.1 Startverhalten



#### 4.1.3 Startverhalten in der CODESYS einstellen

Die Einstellung des Startverhaltens bestimmt im Wesentlichen die Behandlung der remanenten Variablen. Die folgende Einstellung wird nur dann berücksichtigt, wenn die Spannungsversorgung eingeschaltet wird.

Zur Einstellung wählen Sie im Steuerungskonfigurator im Register „Weitere Parameter“ im Drop-Down-Menü „STARTVERHALTEN“ eine der folgenden Startbedingungen aus:

- HALT,
- COLDSTART (Kaltstart),
- WARMSTART.

##### 4.1.3.1 HALT

Das Anwenderprogramm wird nicht gestartet, unabhängig von der Schalterstellung des RUN/STOP-Schalters.

##### 4.1.3.2 COLDSTART/WARMSTART

Voraussetzung: Der RUN/STOP-Schalter steht in der Stellung RUN.

Die Variablen werden gemäß Tabelle 4 initialisiert, bevor die Steuerung gestartet wird.

Tabelle 4: Verhalten der Variablen nach COLDSTART/WARMSTART

Art der Variablen	Verhalten der Variablen nach ...	
	COLDSTART	WARMSTART
<b>Nicht remanent</b>	Aktivieren der Initialwerte	Aktivieren der Initialwerte
<b>Retain<sup>1)</sup></b>	Aktivieren der Initialwerte	Werte bleiben erhalten
<b>Persistent</b>	Werte bleiben erhalten	Aktivieren der Initialwerte
<b>Retain Persistent</b>	Werte bleiben erhalten	Werte bleiben erhalten

1) Physikalische Operanden wie I, Q oder M können nicht als „Retain“-Variablen deklariert werden.

## 4.2 Programmstart

Beim Programmstart prüft die CPU, ob die konfigurierten Ein-/Ausgänge mit den physikalisch vorhandenen übereinstimmen. Sie prüft auch, ob das parametrisierte Modul oder physikalisch ein anderer Modultyp vorhanden ist. Erkennt sie einen falschen Modultyp, wechselt die CPU in den Zustand NOT READY.

### 4.2.1 Programmstart (STOP → RUN)

Sie haben folgende Möglichkeiten, das Programm zu starten:

	Programm im Arbeitsspeicher vorhanden	Programm soll geladen werden
<b>Voraussetzung</b>	<ul style="list-style-type: none"> <li>• CPU im STOP</li> <li>• RUN/STOP-Schalter im STOP</li> </ul>	<ul style="list-style-type: none"> <li>• CPU im STOP</li> <li>• RUN/STOP-Schalter im RUN</li> </ul>
<b>Aktion</b>	<ul style="list-style-type: none"> <li>• RUN/STOP-Schalter in RUN schalten oder</li> <li>• im Online-Betrieb den Befehl „Start“ wählen</li> </ul>	<ul style="list-style-type: none"> <li>• Programm laden</li> <li>• im Online-Betrieb den Befehl „Start“ wählen</li> </ul>
<b>Ergebnis für alle Variablen</b>	CPU im RUN Werte bleiben beim Start erhalten	CPU im RUN Initialwerte werden aktiviert

### 4.2.2 Programmstopp (RUN → STOP)

Ein Wechsel des RUN/STOP-Schalters in die Stellung STOP führt dazu, dass die CPU nach Beendigung des Programmzyklus (Beendigung aller aktiven Tasks) in den Zustand STOP wechselt.

Mit Beendigung der Tasks werden die in den I/O-Tasks benutzten Ausgänge auf 0 gesetzt, a Kapitel „Programmbearbeitung, Multitasking und Systemzeiten“ auf Seite 50.

Sie haben zwei Möglichkeiten, das Programm zu stoppen:

- Im Online-Betrieb den Befehl STOP anwählen.
- Den RUN/STOP-Schalter in die Stellung STOP setzen.

## 4 Betrieb

### 4.3 Ausschalten/Unterbrechen der Spannungsversorgung

#### 4.3 Ausschalten/Unterbrechen der Spannungsversorgung

Das Ausschalten bzw. Unterbrechen der (CPU) Spannung führt bei einem laufenden Programm zu einem sofortigen Abbruch des Programmzyklus bzw. der Tasks. Die Daten sind nicht mehr konsistent!

Alle Ausgänge, die in den I/O-Tasks benutzt werden, werden auf 0 gesetzt bzw. abgeschaltet. Kapitel „Programmbearbeitung, Multitasking und Systemzeiten“ auf Seite 50. Das Verhalten permanenter Variablen geht aus Tabelle 4 hervor.

Bei einem erneuten Einschalten der Spannung wird das restliche Programm des Zyklus nicht beendet!

Ist die Konsistenz der Daten für eine Anwendung zwingend erforderlich, sind weitere Maßnahmen zu projektieren, wie beispielsweise der Einsatz einer unterbrechungsfreien Stromversorgung mit Akku-Pufferung. Der Start der Steuerung erfolgt wie in Abbildung 20 und Abbildung 21 gezeigt.

#### 4.3.1 CPU-Betriebszustandsanzeige

Der Betriebszustand der CPU wird an den LEDs RUN/STOP und SF angezeigt:

CPU-Zustand	RUN/STOP-LED	SF-LED
RUN	ein	aus
STOP	blinkt	aus
NOT READY	blinkt	ein

Der Zustand NOT READY wird durch die LEDs RUN/STOP und SF signalisiert. Er tritt ein, wenn während des Starts ein Fehler aufgetreten ist. Die CPU verbleibt dann im Zustand STOP. Nach Beseitigung des Fehlers kann die CPU neu gestartet werden.

#### 4.3.2 Test und Inbetriebnahme (Debugging)

Die Steuerung unterstützt folgende Test- und Inbetriebnahmemöglichkeiten:

- Breakpoint/Einzelschritt-Betrieb,
- Einzelzyklus-Betrieb,
- Forcen,
- Online-Änderung, → Handbuch für SPS-Programmierung mit CODE-SYS, Kapitel „Online-Funktionen“,
- Zustandsanzeige/Powerflow.

### 4.3.3 Breakpoint/Einzelschritt-Betrieb

Innerhalb des Applikationsprogramms können Breakpoints gesetzt werden. Bei der Ausführung einer mit einem Breakpoint versehenen Anweisung wird das Programm an dieser Stelle angehalten. Die folgenden Anweisungen können im Einzelschrittbetrieb ausgeführt werden. Die Task-Überwachung ist deaktiviert.

#### **ACHTUNG**

Die zu diesem Zeitpunkt gesetzten Ausgänge bleiben gesetzt!

### 4.3.4 Einzelzyklus-Betrieb

Im Einzelzyklus-Betrieb wird ein einzelner Programmzyklus in Echtzeit ausgeführt. Die Ausgänge sind während des Zyklus freigegeben. Am Ende des Zyklus wird das Ausgangsabbild gelöscht und die Ausgänge abgeschaltet. Die Task-Überwachung ist aktiv.

### 4.3.5 Forcen

Alle Variablen des Anwenderprogramms können zwangsgesetzt werden. Ein lokaler Ausgang wird nur dann zwangsgesetzt, wenn die zugehörige Variable zwangsgesetzt wird und die CPU sich im Zustand RUN befindet.

### 4.3.6 Zustandsanzeige

Um die Zustände der konfigurierten Ein-/Ausgänge in einer intervallgesteuerten Task im Steuerungskonfigurator sichtbar zu machen, sind die Ein-/Ausgänge zu referenzieren. Dazu reicht in der Programmiersprache ST bereits die nachfolgende Syntax aus, um auch einzelne I/O-Bits anzeigen zu können.

Beispiel:

```
%IB0; (Referenzieren der Eingänge I0.0 - I0.7)
%QB Referenzieren der Ausgänge Q0.0 - Q0.7)
0;
```

in AWL:

```
LD %IB0
ST Defaultbyte
LD Defaultbyte
ST %QB0
```

## 4 Betrieb

### 4.3 Ausschalten/Unterbrechen der Spannungsversorgung

#### 4.3.7 Reset

Es existieren drei Arten von Reset-Befehlen:

- Reset-Warm,
- Reset-Kalt,
- Reset-Ursprung.

Aus Tabelle 5 geht hervor, welcher Befehl anzuwenden ist, um einen remanenten Variablenbereich zu initialisieren. Die Befehle beeinflussen auch den CPU-Zustand.

##### 4.3.7.1 Reset warm

Das Programm wird angehalten. Die Variablen werden initialisiert. Das Programm kann wieder gestartet werden.

##### 4.3.7.2 Reset kalt

Das Programm wird angehalten. Die Variablen werden initialisiert. Das Programm kann wieder gestartet werden.

##### 4.3.7.3 Reset Ursprung

Das Programm in der Steuerung sowie das Bootprojekt werden gelöscht. Die Variablen werden initialisiert. Die Steuerung wird in den Zustand NOT READY gesetzt.

Tabelle 5: Verhalten der Variablen nach einem Reset

Art der Variablen	Resetbefehl		Ursprung <sup>1)</sup>
	Reset warm	Reset kalt	
<b>Nicht remanent</b>	Aktivieren der Initialwerte	Aktivieren der Initialwerte	Aktivieren der Initialwerte
<b>Retain<sup>2)</sup></b>	Werte bleiben erhalten		
<b>Persistent</b>	Aktivieren der Initialwerte	Werte bleiben erhalten	
<b>Retain Persistent</b>	Werte bleiben erhalten	Werte bleiben erhalten	

1) Nach „Reset Ursprung“ muss das Programm neu geladen werden. Im Online-Betrieb können Sie nun den „Start“-Befehl eingeben.

2) Physikalische Operanden wie I, Q oder M können nicht als „Retain“-Variablen deklariert werden.

## 4.4 Programme und Projekte

### 4.4.1 Programme laden

Um neu erstellte oder geänderte Programme zu laden, müssen Sie sich einloggen. Es erscheint die Frage „Soll das neue Programm geladen werden?“. Nach der Bestätigung beginnt der Ladevorgang.



Beachten Sie, dass durch den Ladevorgang die „Retain“-Variablen initialisiert werden, aber die „PERSISTENT“-Variablen ihren Wert behalten.

Der Download von Programmen wird überwacht. Nach Überschreiten der Default-Übertragungszeit bricht die Kommunikation ab und es erscheint die Fehlermeldung: „Kommunikationsfehler (#0). Es wird ausgelaggt“.

Dies geschieht, wenn die Programme sehr groß sind oder die Anzahl der „Persistent“-Variablen und /oder „Retain-Persistent“-Variablen größer als 5000 ist. Die Anzahl ist unabhängig vom Datentyp. Zur Beseitigung des Problems können Sie die Übertragungszeit beispielsweise auf 30000 ms verlängern. Die Übertragungszeit kann in CODESYS eingestellt werden.

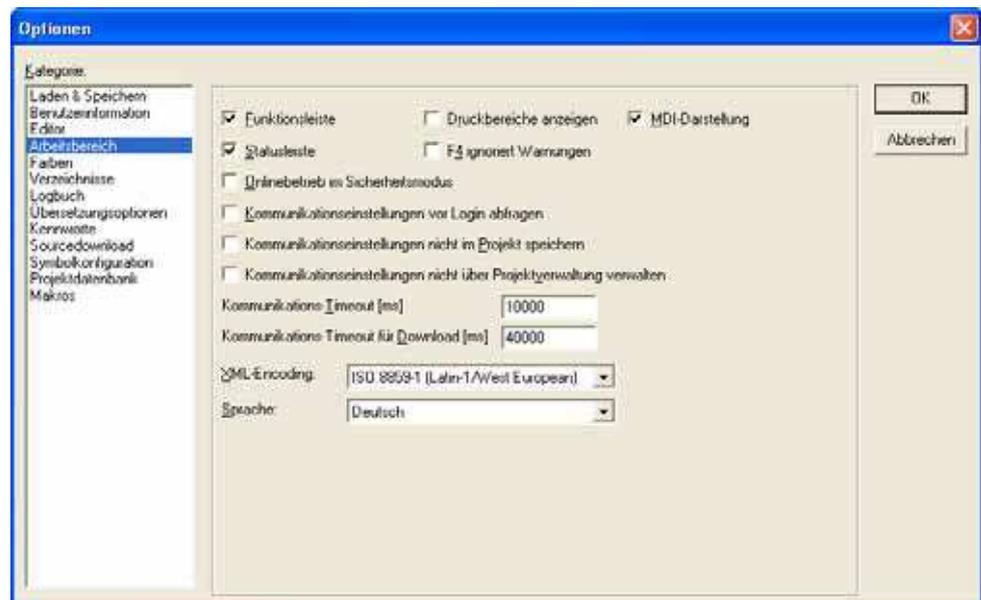


Abbildung 22: Einstellen der Übertragungszeit

Um das Programm sicher abzuspeichern, müssen Sie vom Anwenderprogramm ein Bootprojekt erstellen. Mit dem Befehl „Bootprojekt erzeugen“ wird das Programm vom PC in den Systemspeicher geladen und als Bootprojekt nullspannungssicher gespeichert.

Um ein Bootprojekt zu erstellen, sind folgende Schritte notwendig:

- ▶ Wechseln Sie in den Ordner „Online“.
- ▶ Wählen Sie den Befehl „Einloggen“.
- ▶ Wählen Sie den Befehl „Bootprojekt erzeugen“.

## 4 Betrieb

### 4.4 Programme und Projekte

#### 4.4.2 Allgemeines zu RETAIN PERSISTENT

Die Daten der Variablen, die als RETAIN PERSISTENT deklariert sind, bleiben (im Speicher der XC200) beim Laden eines neuen Programms über MMC/SD oder CODESYS erhalten, wenn folgende Bedingungen erfüllt sind:

- Für das geladene Programm wurde ein Bootprojekt erzeugt.
- Die Namen der Variablen des geladenen Programms und des neuen Programms sind identisch.
- Der Datentyp der Variablen des geladenen Programms und des neuen Programms sind identisch oder ineinander konvertierbar.

Grundsätzlich gilt:

Die Daten von allen Standard-Datentypen werden 1:1 übernommen. Strings werden entsprechend der deklarierten Stringlänge vollständig übernommen oder abgeschnitten.

Sind im neuen Programm den Variablennamen andere Datentypen zugeordnet worden, so erfolgt beim Laden des Programms eine automatische Konvertierung der Daten durch das Betriebssystem der XC200.

In der Regel werden je nach Typgröße Nullen aufgefüllt (SINT → DWORD) bzw. die oberen Bytes abgeschnitten (DWORD → BYTE). Es existieren aber auch einige Daten, die nicht konvertierbar sind, z. B.: (WORD → UINT). Dafür ist das Ergebnis immer NULL.

Die folgende Tabelle zeigt die Konvertierungen:

		im SPS-Speicher vorhandene Variablenamen und Datentypen (Src)											
rp_typN	N =	1..11	1..11	1..11	1..11	1..11	1..11	1..11	1..11	1..11	1..11	1..11	1..11
N =	Typ	SINT	INT	ENUM	DINT	BYTE	USINT	WORD	UINT	DWOR	UDINT	REAL	LREAL
Auf MMC-Programm vorhandene Variablenamen und Datentypen (Dest)	1, 11 SINT	=	X	X	X	X	X	X	X	X	X	X	X
	2 INT	X	=	0	X	X	X	X	X	X	X	X	X
	ENUM	X	0	=	X	X	X	X	X	X	X	X	X
	3 DINT	X	X	X	=	X	X	X	X	X	X	X	X
	4 BYTE	X	X	X	X	=	0	0	0	X	X	X	X
	5 USINT	X	X	X	X	0	=	0	0	X	X	X	X
	6 WORD	X	X	0	X	X	X	=	0	0	X	X	X
	7 UINT	X	X	0	X	X	X	0	=	X	X	X	X
	8 DWORD	X	X	X	X	X	X	X	X	=	0	X	X
	9 UDINT	X	X	X	X	X	X	X	X	0	=	X	X
10 REAL	X	X	X	X	X	X	X	X	X	X	=	X	
LREAL	X	X	X	X	X	X	X	X	X	X	X	=	
Programmabschnitte		1	2	3	4	5	6	7	8	9	10	11	

rp = RETAIN PERSISTENT

X = Konvertierung erfolgt. Die Daten werden an den neuen Datentyp (aus MMC-Programm) angepasst. Nullen werden vorangestellt oder höherwertige Bytes abgenitten

= keine Konvertierung erforderlich. Daten sind identisch.

0 = Konvertierungsergebnis ist NULL.

1..11 blau = Variablenamen unterschiedlich, aber vom gleichen Typ (SPS-Programm)

1..11 gelb = Variablenamen identisch mit 1..11 blau, aber unterschiedlicher Typ (MMC-Programm)

RETAIN PERSISTENT-Daten werden gelöscht , wenn

- das neue Programm keine identischen Variablenamen enthält,
- der Befehl „Reset Ursprung“ ausgeführt wird,
- die Batterie gezogen wurde.

### 4.4.3 Bootprojekt abspeichern und löschen

#### Bei der XC-CPU201

##### 1. Bootprojekt auf MMC abspeichern

- ▶ Klicken Sie auf den Ordner „Ressourcen → PLC Browser“ und geben Sie den Befehl copyprojtommc ein.

Das Bootprojekt wird auf der MMC im Unterverzeichnis „project“ unter dem Namen „Default.prg“ abgespeichert. Zusätzlich wird eine Datei mit Namen „Default.chk“ erzeugt. Mit den Browser-Befehlen filecopy oder filename können Sie das Bootprojekt kopieren (z. B. für eine Sicherungskopie) und den Namen der Datei verändern. In der Software CODESYS ist jedoch nur das Bootprojekt mit dem Namen „Default“ aktiv.

##### 2. Bootprojekt auf MMC löschen

Klicken Sie auf den Ordner „Ressourcen“ → PLC Browser und geben Sie für die XC-CPU201-EC256K den folgenden Befehl ein:

```
filedelete \\disk_mmc\MOELLER\XC-CPU201-EC256K-8DI-6DO\ project\default.prg
```

#### **Bei der XC-CPU-202:**

##### **1. Bootprojekt auf SD/MMC abspeichern**

Klicken Sie auf den Ordner „Ressourcen | PLC Browser“ und geben Sie den Befehl `copyprojtommc` ein.

Das Bootprojekt wird auf der MMC/SD im Unterverzeichnis „project“ unter dem Namen `Default.prg` abgespeichert. Zusätzlich wird eine Datei `Default.chk` erzeugt.

Mit den Browser-Befehlen `filecopy` und `filerename` können Sie das Bootprojekt kopieren (z. B. für eine Sicherungskopie) bzw. den Namen der Datei verändern. In der Software CODESYS ist jedoch nur das Bootprojekt mit dem Namen „Default“ aktiv.

##### **2. Bootprojekt auf USB abspeichern**

Klicken Sie auf den Ordner „Ressourcen | PLC Browser“ und geben Sie den Befehl `copyprojtoUSB` ein.

Das Bootprojekt wird auf dem USB-Stick im Unterverzeichnis „project“ unter dem Namen `Default.prg` abgespeichert. Zusätzlich wird eine Datei `Default.chk` erzeugt.

Mit den Browser-Befehlen `filecopy` oder `filerename` können Sie das Bootprojekt kopieren (z. B. für eine Sicherungskopie) und den Namen der Datei verändern. In der Software CODESYS ist jedoch nur das Bootprojekt mit dem Namen „Default“ aktiv.

##### **3. Bootprojekt auf SD/MMC löschen**

Klicken Sie auf den Ordner „Ressourcen → PLC Browser“ und geben Sie für die XC-CPU202 den folgenden Befehl ein:

```
filedelete \\disk_mmc\\CONTROL\\XC-CPU202-EC4M-8DI-6DO\\project\\default.prg
```

##### **4. Bootprojekt auf USB-Stick löschen**

Klicken Sie auf den Ordner «Ressourcen → PLC Browser» und geben Sie für die XC-CPU202 den folgenden Befehl ein:

```
filedelete \\disk_usb\\CONTROL\\XC-CPU202-EC4M-8DI-6DO\\project\\default.prg
```

## 4.5 Betriebssystem aktualisieren

Bei der XC200 haben Sie die Möglichkeit, das Betriebssystem durch eine aktuelle Version zu ersetzen. Eaton bietet die jeweils aktuelle Betriebssystemversion im Internet an: <http://www.eaton.com/moeller> → **Support**

→ Wenn Sie ein aktuelles Betriebssystem auf eine ältere Hardware-Version übertragen, werden eventuell nicht alle Funktionen des Betriebssystems von der Hardware unterstützt.

### 4.5.1 XC-CPU201

#### 4.5.1.1 Betriebssystem vom PC in die XC-CPU201 übertragen

→ Beim Einladen eines Betriebssystems in die Steuerung werden das bestehende Betriebssystem und das Anwenderprogramm gelöscht!

→ Zum Laden des Betriebssystems ist die Baudrate fest auf den Wert 115200 Bit/s eingestellt.

Vorgehensweise:

- ▶ Stellen Sie eine serielle Verbindung über die RS232-Schnittstelle des PC mit der XC-CPU201 her. Informationen hierzu finden Sie in den Abschnitten „PC anschließen“ auf Seite 28 und „Verbindungsaufbau PC – XC200“ auf Seite 72.
- ▶ Aktivieren Sie in der Software CODESYS im Fenster „Steuerungskonfiguration“ die Registerkarte „Weitere Parameter“ und klicken Sie dort auf die Schaltfläche „Start“.

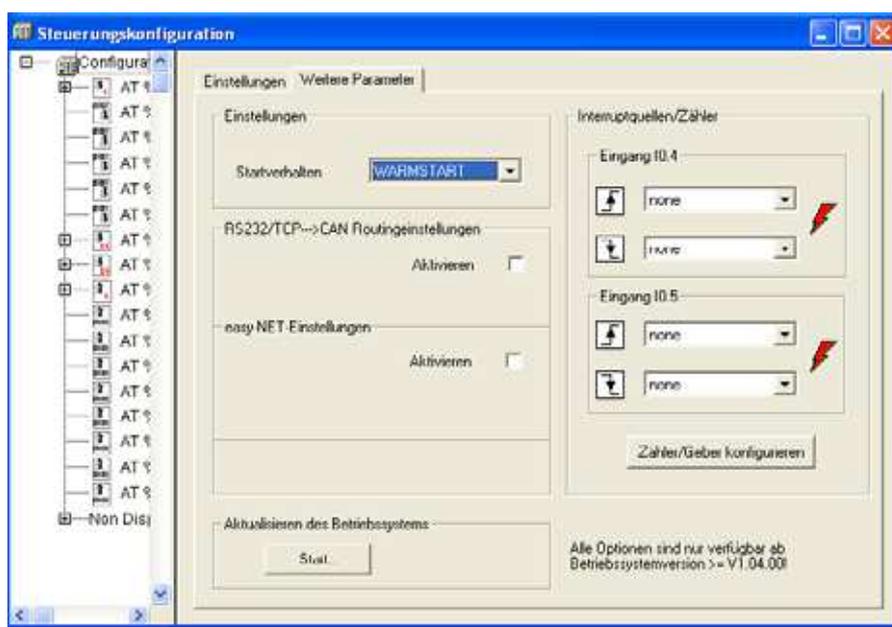


Abbildung 23: Betriebssystem der XC-CPU201 aktualisieren

## 4 Betrieb

### 4.5 Betriebssystem aktualisieren

Es öffnet sich das Fenster für das „Download Tool“.

- Klicken Sie auf die Schaltfläche „Öffnen“ und geben Sie den Pfad an, in dem das Update des Betriebssystems liegt.

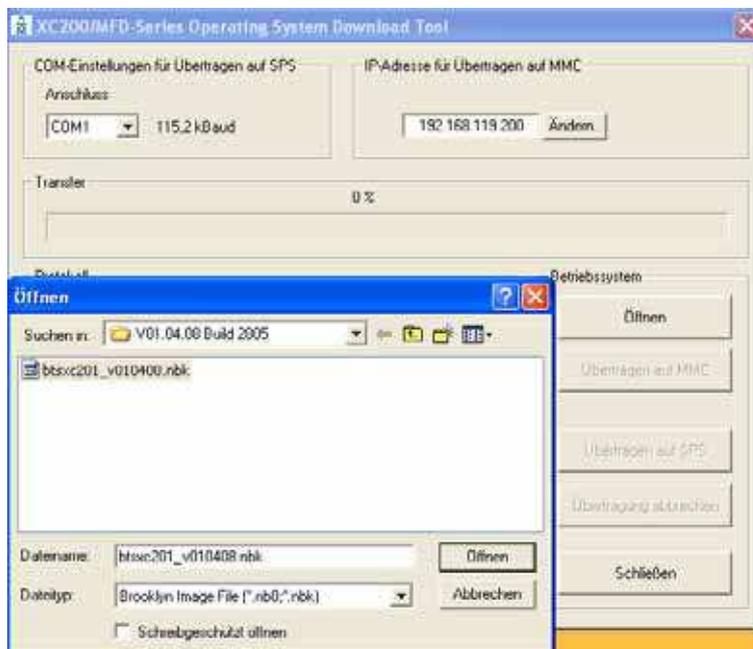


Abbildung 24: Betriebssystem für XC-CPU201 auswählen

- Öffnen Sie die zu übertragende Betriebssystem-Datei. Sie erhalten das nachfolgende Fenster:

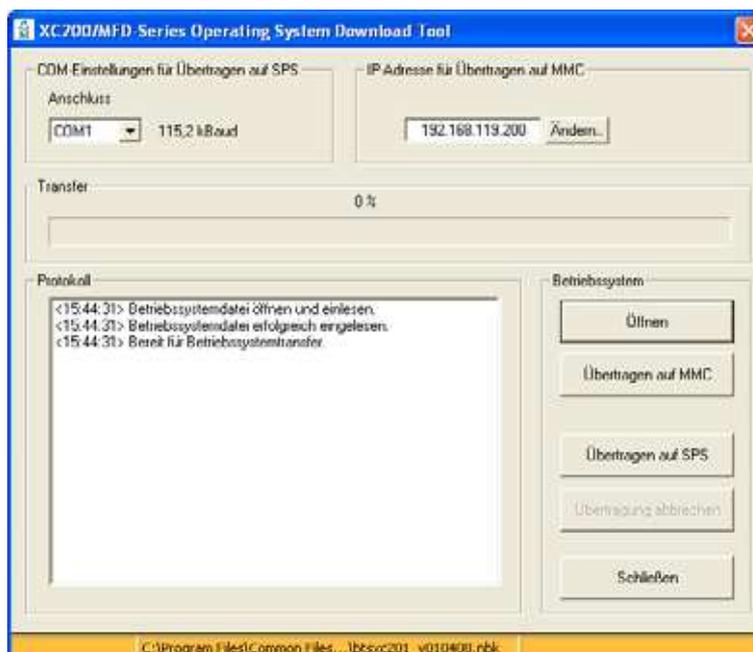


Abbildung 25: Download des Betriebssystems der XC-CPU201

- ▶ Klicken Sie auf die Schaltfläche „Übertragen auf SPS“.

Es erscheint der Fenstereintrag „Verbinden mit SPS. Booten Sie die Steuerung erneut.“

- ▶ Schalten Sie die Steuerspannung der XC-CPU201 aus und warten Sie einige Sekunden. Auf diese Weise stellen Sie sicher, dass die gespeicherte Restspannung komplett abgebaut wird.
- ▶ Schalten Sie die Steuerspannung der XC-CPU201 wieder ein.

Der Transfer des Betriebssystems in die XC-CPU201 wird gestartet. Dies kann mehrere Minuten dauern. Achten Sie auf die Signalzustände der Betriebs-LEDs:

Während des Transfers leuchtet die rote SF-LED. Wenn die Transferanzeige 100 % anzeigt, geht die SF-Anzeige nach einer kurzen Verzögerung aus. Nach ca. 1 Minute leuchtet sie wieder; die grüne RUN/STOP-LED blinkt. Die Wartezeit ergibt sich aus der Programmierung der internen Flash-Speicher (vergleichbar mit dem Booten eines PC).

Sie erhalten weitere Einträge im Download-Fenster. Der Fortschritt des Downloads wird Ihnen zusätzlich über den Zustandsbalken im Transferfeld angezeigt.

Bitte greifen Sie nicht in den Download-Prozess ein, bis im Download-Fenster der Eintrag „Bereit für Betriebssystemtransfer“ ein zweites Mal erscheint und die grüne LED blinkt. Erst wenn beide Attribute vorhanden sind, ist der Download vollständig abgeschlossen.



Abbildung 26: Download des Betriebssystems der XC-CPU201 beendet

- ▶ Beenden Sie den Download mit der Schaltfläche „Schließen“.

## 4 Betrieb

### 4.5 Betriebssystem aktualisieren

#### 4.5.1.2 Betriebssystem vom PC auf die MMC der XC-CPU201 übertragen

Dies ist nur über eine Ethernet-Verbindung möglich. Zudem muss die XC-CPU201 ein Betriebssystem ab Version 01.03.00 enthalten.

Das Betriebssystem befindet sich nach dem Transfer im Verzeichnis:

**disc\_mmc\moeller\XC-CPU201**

##### **PC → MMC**

Der Vorgang erfolgt analog zum Übertragen des Betriebssystems vom PC in die Steuerung. Allerdings klicken Sie hier auf die Schaltfläche „Übertragen auf MMC“ (siehe Abbildung 25).

##### **MMC → Steuerung**

Soll das Betriebssystem einer XC-CPU201 über die MMC aktualisiert werden, muss die Steuerung über ein Betriebssystem ab der Version 01.03.00 verfügen. Das Betriebssystem wird beim Einschaltvorgang aktualisiert.

#### 4.5.1.3 Betriebssystem/Bootprojekt von der MMC der XC-CPU201 löschen

Sie können das Betriebssystem/Bootprojekt vom PC aus, z. B. mit dem Internet Explorer, löschen:

- ▶ Stellen Sie hierzu die Verbindung über die Default-Adresse `ftp://192.168.119.200` der XC-CPU201 her.
- ▶ Öffnen Sie das Verzeichnis **disc\_mmc\moeller\XC-CPU201**.

In diesem Verzeichnis sind alle Betriebssystem-Dateien abgelegt und können dort auch gelöscht werden.

## 4.5.2 XC-CPU202

Die nachfolgende Beschreibung (Firmware-Update) gilt für Versionen ab Version 2.4.13 (Ziel-Firmware-Version).

Die Aktualisierung des Betriebssystems der XC-CPU202 erfolgt innerhalb des XSOFTECODESYS-2 Programmiersystems.

Vorgehensweise:

- ▶ Stellen Sie eine Verbindung über die Ethernet-Schnittstelle des PC mit der XC-CPU202 her.  
Informationen hierzu finden Sie in → Abschnitt 3.4.6, „PC anschließen“, Seite 28 und → Kapitel 6, „Verbindungsaufbau PC – XC200“, Seite 72.
- ▶ Aktivieren Sie in der XSOFTECODESYS-2 im Fenster „Steuerungskonfiguration“ die Registerkarte „Firmware“.

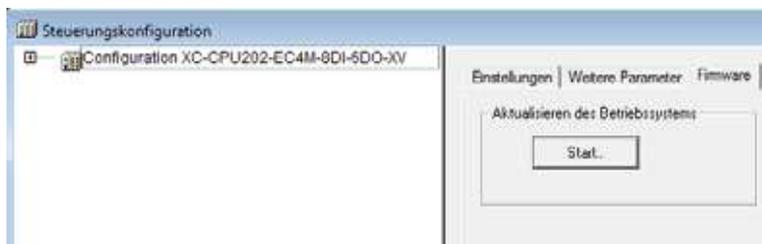


Abbildung 27: Fenster „Steuerungskonfiguration“

- ▶ Klicken Sie auf die Schaltfläche „Start“ (Aktualisieren des Betriebssystems).
- ▶ Wählen Sie die Firmware-Datei für die XC202 aus und klicken Sie auf die Schaltfläche „Öffnen“.
- ▶ Klicken Sie auf die Schaltfläche „Weiter“, um den Setup-Assistenten zu starten.
- ▶ Wählen Sie die Option „FTP Installation“ und klicken Sie dann auf die Schaltfläche „Weiter“.

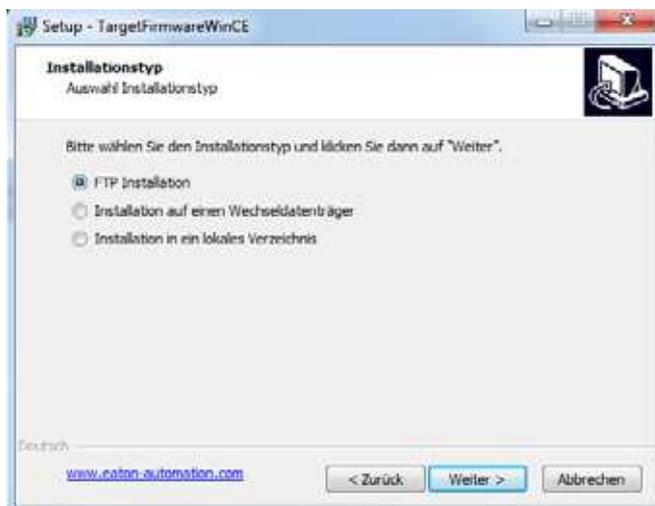


Abbildung 28: Option „FTP Installation“

## 4 Betrieb

### 4.5 Betriebssystem aktualisieren

- ▶ Geben Sie die IP-Adresse der Steuerung ein und klicken Sie auf die Schaltfläche „Weiter“.  
Hinweis: Die Netzwerkeinstellung im Auslieferungszustand lautet:  
IP-Adresse: 192.168.119.202

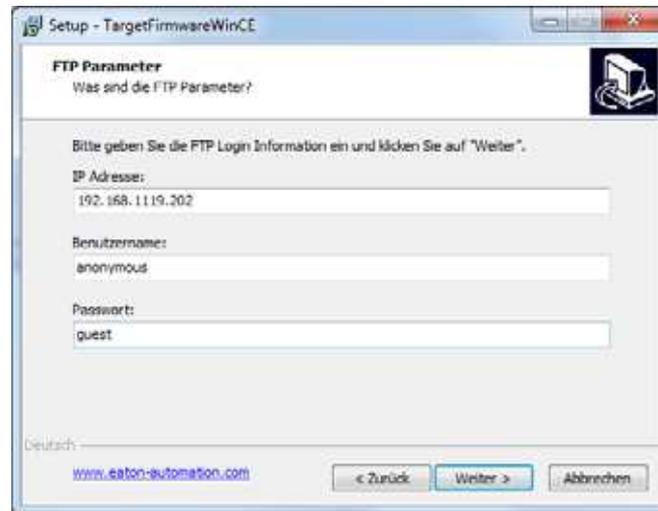


Abbildung 29: FTP-Parameter

- ▶ Wählen Sie den Gerätetyp „XC202“ aus und klicken Sie auf die Schaltfläche „Weiter“.
- ▶ Wählen Sie die Komponente „Betriebssystem PLC“ aus und klicken Sie dann auf die Schaltfläche „Weiter“.
- ▶ Klicken Sie auf die Schaltfläche „Installieren“, um den Firmware-Download zu starten.
- ▶ Warten Sie, bis im Fenster „Update finished“ angezeigt wird. Schließen Sie das Fenster anschließend mit „RETURN“.
- ▶ Klicken auf die Schaltfläche „Fertigstellen“, um den Setup-Assistenten zu beenden.

Die Installation auf einen Wechseldatenträger und die Installation in ein lokales Verzeichnis (→ Abbildung 28, Seite 47) erfolgen prinzipiell analog der FTP-Installation.

Vorgehensweise:

- ▶ Stecken Sie den USB-Stick bzw. die SD-Karte am PC ein.
- ▶ Wählen Sie im Fenster „Setup - TargetFirmwareWinCE“ (a Abb. 28) „Installation auf einen Wechseldatenträger“ aus.
- ▶ Wählen Sie das Verzeichnis des Wechseldatenträgers aus und klicken Sie auf „Weiter“.
- ▶ Wählen Sie den Gerätetyp „XC202“ aus und klicken Sie auf die Schaltfläche „Weiter“.
- ▶ Wählen Sie die Komponente „Betriebssystem PLC“ aus und klicken Sie auf die Schaltfläche „Weiter“.
- ▶ Klicken Sie auf die Schaltfläche „Installieren“, um den Firmware-Download zu starten.
- ▶ Klicken Sie auf die Schaltfläche „Fertigstellen“, um den Setup-Assistenten zu beenden.

Auf dem Wechseldatenträger befinden sich nun die Firmware-Dateien im Verzeichnis **CONTROL/XC-CPU202**.

- ▶ Stecken Sie den USB-Stick oder die SD-Karte auf die XC-CPU202.

Es wird geprüft, ob sich ein anderes Betriebssystem auf dem Wechseldatenträger befindet. Ist dies der Fall, wird das Firmware-Update gestartet. Sie erkennen dies daran, dass die LEDs RUN/STOP wechselweise (grün) und SF (rot) blinken.



Bei einem Update zwischen einer Firmware-Version 01.00.xx und einer Firmware-Version 2.4.xx wird das Dateisystem des lokalen Speichers „disk\_sys“ gewechselt. Dieser Wechsel kann bis zu 10 Minuten dauern.

Das Schreiben auf den SPS-internen Speicher wird an der XC-CPU202 durch ein wechselweises Blinken der beiden LEDs RUN/STOP (grün) und SF (rot) angezeigt.



Die neue Firmware-Version ist erst nach einem Neustart der Steuerung aktiv.

#### **ACHTUNG**

Bei einem Update zwischen einer Firmware-Version 01.00.xx und einer Firmware-Version 2.4.xx wird das Dateisystem erneuert. Alle auf dem lokalen Speicher „disk\_sys“ abgelegten Daten (SPS-Programm / Registry-Einstellungen / Eigene Daten) werden gelöscht.

Bei einem Update innerhalb der Firmware-Versionen 1.00.xx oder 2.4.xx bleiben diese Daten erhalten.

## 5 Programmbearbeitung, Multitasking und Systemzeiten

### 5.1 Taskkonfiguration

Die Projektbearbeitung kann über Tasks gesteuert werden. Jeder Task können Sie eine Folge von Programmen zuordnen, die beim Ausführen der Task abgearbeitet werden sollen.

Eine Task ist definiert durch einen Namen, eine Priorität und einen Typ, der festlegt, unter welchen Bedingungen eine Task startet. Taskbedingung und Priorität legen fest, in welcher zeitlichen Abfolge die Tasks abgearbeitet werden.

Als Taskbedingung können Sie „Zyklisch“ oder „Ereignisgesteuert“ festlegen. Eine zyklische Task wird nach Ablauf der parametrisierten Intervallzeit erneut gestartet. Eine ereignisgesteuerte Task wird nur dann bearbeitet, wenn das Ereignis eintritt. Darüber hinaus haben Sie die Möglichkeit, Systemereignisse wie z. B. „Start“, „Stop“ oder „Reset“ mit der Ausführung eines Programms zu koppeln.

Die Prioritäten einer Task können zwischen „0“ und „31“ parametrisiert werden, wobei „0“ die höchste und „31“ die niedrigste Priorität darstellt.

Prinzipiell wird vor jedem Taskaufruf das Ausgangsabbild auf die physikalischen Ausgänge geschrieben und das Abbild der Eingänge gelesen (Aktualisieren des Ein-/Ausgangs-Abbildes). Danach wird die Task ausgeführt. Zusätzlich werden alle anfallenden Systemaktivitäten vor bzw. nach dem Task-Aufruf abgearbeitet. Hierunter fallen beispielsweise die Kommunikation mit CODESYS oder Online-Änderungen.

Das Aktualisieren des Ein-/Ausgangsabbild von mehreren Tasks wird im Abschnitt „Multitasking“ auf Seite 54 beschrieben.

Alle IEC-Tasks, auch die mit der höchsten Priorität, können durch einen Interrupt oder eine ereignisgesteuerte Task unterbrochen werden.

Für jede Task kann eine Zeitüberwachung (Watchdog) aktiviert werden.



Eine ausführliche Beschreibung der Taskkonfiguration finden Sie im Handbuch Handbuch für SPS-Programmierung mit CODESYS.

Im Anschluss werden die steuerungsspezifischen Einstellungen anhand eines Beispiels erläutert.

### Task erstellen (Beispiel)

Zuerst erstellen Sie die zyklische Task „Basic“ mit dem zugeordneten Programm „Basic\_prog“. Dann können Sie die ereignisgesteuerte Task „Param“ mit dem Programm „Param\_prog“ hinzufügen. Im Programm „Basic\_prog“ wird ein Ereignis programmiert, das die Task „Param“ aufruft.

Um eine Task zu erstellen, sind folgende Schritte notwendig:

- Task anlegen
- Programmaufruf festlegen
- Programm erstellen.

#### 5.1.1 Zyklische Task „Basic“ anlegen

- ▶ Öffnen Sie den Ordner „Taskkonfiguration“ im Register „Ressourcen“.
- ▶ Klicken Sie mit der rechten Maustaste auf den Ordner „Taskkonfiguration“ und wählen Sie im Kontextmenü den Befehl „Task anhängen“.
- ▶ Geben Sie im Feld „Name“ zum Beispiel „Basic“ ein.
- ▶ Parametrieren Sie die Task im Dialogfenster wie in Abbildung 30.
- ▶ Klicken Sie auf den Ordner „Taskkonfiguration“, so wird die Konfiguration übernommen.

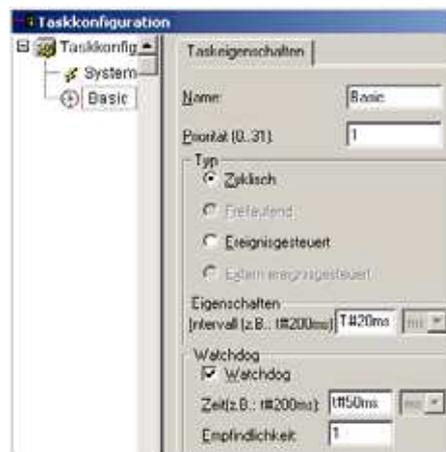


Abbildung 30: Parametrieren der zyklischen Task

##### 5.1.1.1 Programmaufruf festlegen

Mit dem Programmaufruf legen Sie fest, welches Programm mit der Task „Basic“ aufgerufen werden soll.

- ▶ Klicken Sie mit der rechten Maustaste auf das Uhrensymbol der zuvor erstellten Task „Basic“ und wählen Sie im Kontextmenü den Befehl „Programmaufruf anhängen“.
- ▶ Geben Sie den Namen „Basic\_prog“ im Fenster „Programmaufruf“ an.
- ▶ Klicken Sie auf die Schaltfläche am Ende des Eingabefeldes und bestätigen Sie den Programmnamen im Fenster „Eingabehilfe“.

## 5 Programmbearbeitung, Multitasking und Systemzeiten

### 5.1 Taskkonfiguration

#### 5.1.1.2 Programm erstellen

- ▶ Wechseln Sie in das Register „Bausteine“, klicken Sie dann mit der rechten Maustaste auf den Default-Programmbaustein PLC\_PRG und wählen Sie den Befehl „Objekt umbenennen“. Nennen Sie den Baustein „Basic\_Prog“.
- ▶ Sie können nun ein Programm eingeben. Im Programmbeispiel (Abbildung 31) wird die Variable „count“ hochgezählt. Beim Zählerstand = 9 wird a = TRUE.



Abbildung 31: Programmbaustein für eine zyklische Task anlegen

#### 5.1.2 Ereignisgesteuerte Task „Param“ anlegen und Programmaufruf festlegen

Die Vorgehensweise entspricht der Erstellung einer zyklischen Task.

- ▶ Legen Sie eine Task vom Typ „ereignisgesteuert“ mit dem Namen „Param“ gemäß Abbildung 32 an.
- ▶ Geben Sie als Ereignis die Boolesche Variable „a“ an.
- ▶ Geben Sie den Programmaufruf „Param\_prog“ ein.

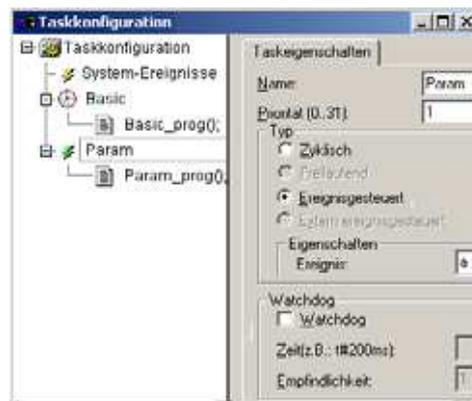


Abbildung 32: Ereignisgesteuerte Task anlegen

### 5.1.2.1 Programm erstellen

- ▶ Wechseln Sie in das Register „Bausteine“ und fügen Sie ein Objekt (POU) mit dem Namen „Param\_prog“ ein.
- ▶ Sie können nun ein Programm eingeben. Im Programmbeispiel „Param\_prog“ (Abbildung 33) wird die Variable „value“ um den Wert 1 erhöht. Das Programm „Param\_prog“ wird abgearbeitet, wenn die Variable a = TRUE wird.

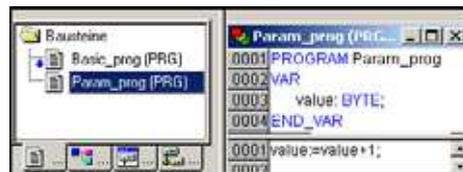


Abbildung 33: Programmbaustein für eine ereignisgesteuerte Task

## 5.2 Systemereignisse

Mit Hilfe eines Systemereignisses kann eine POU aufgerufen werden. Sie kann beim Start der Steuerung zum Initialisieren von Modulen mit neuen Parametern verwendet werden. Die Systemereignisse sind unabhängig von der Task!

### 5.2.1 POU einem Systemereignis zuordnen

- ▶ Aktivieren Sie in der Taskkonfiguration unter „System-Ereignisse“ das Ereignis, z. B. „Start“ und geben Sie den Namen der POU (z. B. „Power\_prog“) ein, die abgearbeitet werden soll.

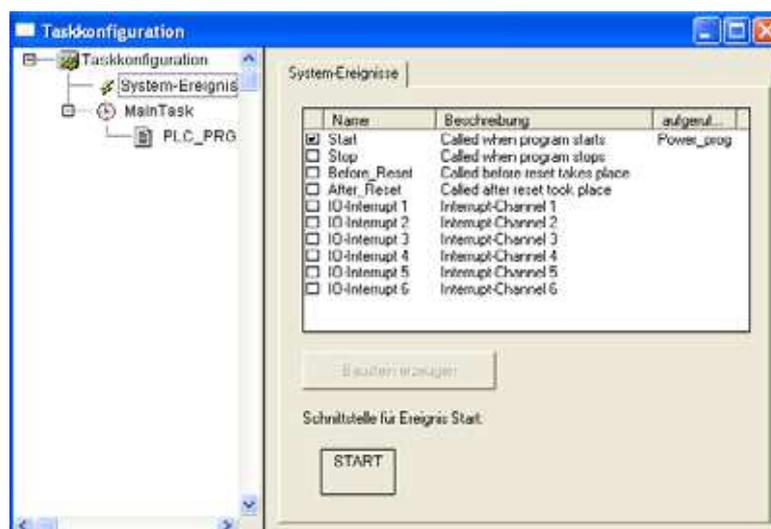


Abbildung 34: POU dem Systemereignis zuordnen

## 5 Programmbearbeitung, Multitasking und Systemzeiten

### 5.3 Multitasking

- ▶ Wechseln Sie in das Register „Ressourcen → Bausteine“ und fügen Sie das Objekt (POU) „Power\_prog“ ein.
- ▶ Programmieren Sie die Anwendung.



Abbildung 35: POU programmieren



Weitere Informationen zu den Systemereignissen finden Sie in der Online-Hilfe des Programmiersystems.

### 5.3 Multitasking

Das XC200-Laufzeitsystem ist ein Multitasking-System. Dies bedeutet, dass mehrere Tasks parallel abgearbeitet werden können.

#### 5.3.1 Ein-/Ausgangsabbilder aktualisieren

Werden in mehreren Tasks die lokalen und zentralen Ein-/Ausgänge programmiert, erfolgt das Update (Aktualisierung der Ein-/Ausgangspegel) des Ein-/Ausgangsabbilds nach besonderen Regeln:

Das System beginnt – z. B. nach dem Start – in der ersten Task nach programmierten Eingängen zu suchen. Der Begriff „erste Task“ bezeichnet die erste Task in der Taskkonfiguration, unabhängig von der Priorität und der Zykluszeit der einzelnen Tasks. In der Abbildung 37 ist der Name der ersten Task „Prog1“. Erkennt das System einen Eingang, der durch die Konfiguration mit einem Eingangsmodul, z. B. XIOC-16DI, verbunden ist, werden alle Eingänge dieses Moduls im Abbild aktualisiert. Sind in dieser Task noch weitere Eingänge vorhanden, die anderen Modulen zugeordnet sind, werden die Eingänge dieser Module ebenfalls aktualisiert (Modulweise Update-Prozedur). Werden beispielsweise die Eingänge %IX6.0 und %IX7.1 des Eingangsmoduls „1“ von verschiedenen Tasks angesprochen, werden die Eingänge dieses Moduls nur von der ersten Task aktualisiert.

### 5.3.1.1 Beispiele

Die Beispiele basieren auf folgender Konfiguration:

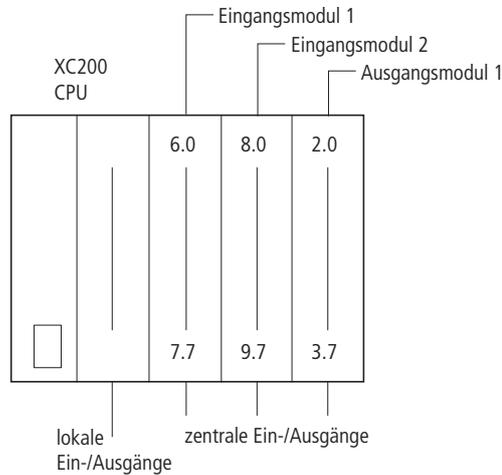


Abbildung 36: Konfiguration der XC200

Die Taskkonfiguration sieht wie folgt aus::



Abbildung 37: Taskkonfiguration für die Beispiele

#### Beispiel 1:

Tabelle 6: Taskangaben für Beispiel 1

Taskname	Priorität	Zykluszeit
Prog 1	2	50 ms
Prog 2	1	10 ms

In der ersten Task „Prog1“ werden im Programm „progtes(1)“ die Eingänge %IX6.0 des Eingangsmoduls 1 und %IX8.3 des Eingangsmoduls 2 programmiert. Vor dem Start der ersten Task „Prog1“ werden die Eingänge dieser Module aktualisiert.

In der zweiten Task „Prog2“ wird im Programm „progtes(2)“ der Eingang %IX7.1 des Eingangsmoduls 1 programmiert. Vor dem Start der 2. Task „Prog2“ werden die Eingänge dieses Eingangsmoduls nicht aktualisiert, da dies nur in der ersten Task erfolgt.

## 5 Programmbearbeitung, Multitasking und Systemzeiten

### 5.3 Multitasking

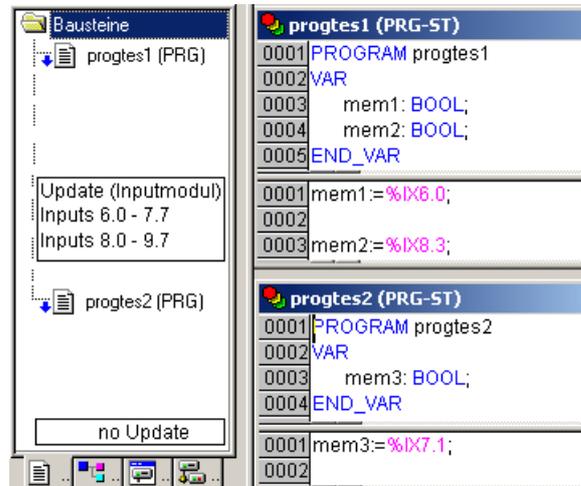


Abbildung 38: Programm für Beispiel 1

#### Beispiel 2:

Tabelle 7: Taskangaben für Beispiel 2

Taskname	Priorität	Zykluszeit
Prog 1	2	50 ms
Prog 2	1	20 ms

In Beispiel 2 werden in der ersten Task der Eingang 6.1 und in der zweiten Task der Eingang 8.4 sowie der Ausgang 3.4 programmiert. Zu Beginn der ersten Task erfolgt ein Update der Eingänge 6.0 bis 7.7 des Eingangsmoduls 1.

Zu Beginn der zweiten Task folgen die Eingänge 8.0 bis 9.7 des Eingangsmoduls 2 und die Ausgänge 2.0 bis 3.7 des Ausgangsmoduls 1.

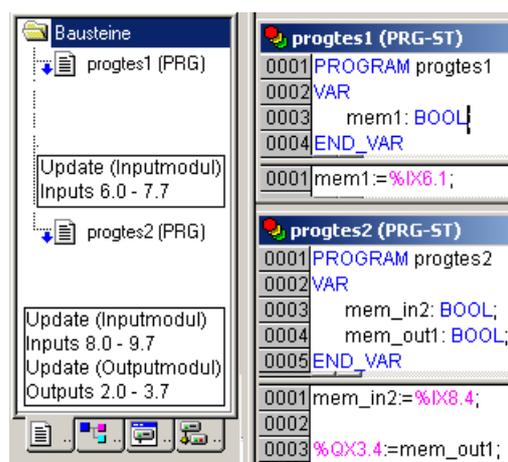


Abbildung 39: Programm für Beispiel 2

#### 5.3.1.2 Task mit konsistenten I/O erstellen

Vermeiden Sie einen Zugriff auf physikalische Ausgänge aus mehreren Tasks heraus. Um einen eindeutigen Steuerungsablauf sicherzustellen, erstellen Sie für die lokalen/zentralen Ein-/Ausgänge eine Task, in der alle Eingänge in globale Variablen kopiert und am Ende des Intervalls alle Ausgänge von globalen Variablen auf die Ausgangsmodule geschrieben werden (I/O-Update-Task). Innerhalb dieser Task sind dann die I/Os konsistent. Die globalen Variablen können dann anstelle der I/Os in anderen Tasks verwendet werden.



Bei der XC200-Steuerung sind maximal 10 Tasks möglich. Die Parametrierung einer Task als „freilaufend“ wird nicht unterstützt.

Beachten Sie bei der Parametrierung der Watchdog-Zeit, dass die vom Interrupt aufgerufene POU die Tasklaufzeiten entsprechend verlängern.

#### 5.3.2 Verhalten des CAN-Stacks bei Multitasking

Vor jeder Task, in der CAN-Variablen verwendet werden, findet ein Aufruf eines CAN-Stacks statt. In einem Multitasking-System können sich einzelne Tasks in Abhängigkeit von ihrer Priorität beliebig unterbrechen. Dieses Verhalten kann zu einer Inkonsistenz des CAN-Stacks führen, wenn dieser von einer höherprioritären Task aufgerufen wird, bevor die Bearbeitung des CAN-Stacks durch die unterbrochene Task beendet wurde.



Der CAN-Stack der XC200 ist nicht multitaskingfähig! Es darf nur eine Anwender-Task erstellt werden, in der CAN-Variablen verwendet werden.

#### 5.4 Tasküberwachung durch Watchdog

Mit Hilfe des Watchdogs kann die Abarbeitungszeit einer Task zeitlich überwacht werden. Zur Festlegung der Überwachungszeit gilt folgende Beziehung:

Abarbeitungszeit < Intervallzeit der Task < Watchdog(-Zeit)

Überschreitet die Abarbeitungszeit die Intervallzeit, wird das Ende der zweiten Intervallzeit abgewartet, bis die Task neu gestartet wird.

→ Watchdog deaktiv.

Der Watchdog unterbricht die Programmbearbeitung, wenn die Abarbeitungszeit der Task die Watchdog-Zeit überschreitet.

Zusätzlich kann die Häufigkeit (Empfindlichkeit) eingestellt werden, die die Anzahl der Überschreitungen zulässt. In diesem Fall werden die Ausgänge der Steuerung abgeschaltet und das Anwenderprogramm in den Zustand „Halt“ gesetzt. Danach muss das Anwenderprogramm mit „Reset“ zurückgesetzt werden.



Wird der Watchdog deaktiviert, erfolgt keine Tasküberwachung!



#### VORSICHT

Wenn Sie eine Task ohne Watchdog parametrieren oder den Watchdog nachträglich deaktivieren, können alle bis zu dem Zeitpunkt angesteuerten Ausgänge weiterhin aktiv geschaltet bleiben. Das ist zum Beispiel der Fall, wenn die Task wegen einer Endlosschleife (Programmierfehler) und/oder fehlender Ablaufbedingung (Weiterschaltbedingung) nicht beendet werden kann. Die Ausgänge behalten dabei ihr „High“-Potential bis zum Betriebsartwechsel von RUN in STOP oder bis zum Abschalten der Steuerspannung für die Ausgänge.

#### 5.4.0.1 Watchdog konfigurieren

In der Taskkonfiguration können Sie folgende Einstellungen auswählen:

- Watchdog ein/aus
- Watchdog-Zeit
- Watchdog-Empfindlichkeit.

Diese Einstellungen gelten sowohl für zeit- als auch für ereignisgesteuerte Tasks.

##### Watchdog aktiv

Der Watchdog wird zu Beginn eines jeden Abarbeitungszyklus gestartet und am Ende der Task wieder zurückgesetzt.



Zur Festlegung der Watchdog-Zeit bei mehreren Tasks gilt folgende Regelung: Jede Watchdog-Zeit muss größer sein als die Summe der Taskintervallzeiten.

Ist die Abarbeitungszeit länger als die Watchdog-Zeit (Empfindlichkeit = 1) – zum Beispiel wegen einer Endlosschleife im Programm – wird der Watchdog aktiv. Ist der Abarbeitungszyklus kürzer als die Watchdog-Zeit, erfolgt keine Aktivierung des Watchdogs.

Das Auslösen des Watchdogs ist weiterhin abhängig von der Watchdog-Empfindlichkeit. Die Watchdog-Empfindlichkeit bestimmt, nach wie vielen unmittelbar aufeinanderfolgenden Watchdog-Zeitüberschreitungen der Watchdog ausgelöst wird.

Der Watchdog wird ausgelöst:

- bei einer Watchdog-Empfindlichkeit von „1“ gleich bei der ersten Überschreitung der Watchdog-Zeit,
- bei einer Watchdog-Empfindlichkeit von „x“ erst beim „x“-ten unmittelbar aufeinander folgendem Überschreiten der Watchdog-Zeit.

Wenn Sie beispielsweise bei einer Task die Watchdog-Zeit von „10 ms“ und die Watchdog-Empfindlichkeit von „5“ eintragen, wird die Task also spätestens nach  $5 \times 10 \text{ ms} = 50 \text{ ms}$  beendet.

### 5.4.0.2 Beispiel: Watchdog aktiv

Das Zusammenwirken von Intervallzeit (IZ), Tasklaufzeit (TZ), Watchdog-Zeit (WZ) und Watchdog-Empfindlichkeit verdeutlicht folgendes Konfigurationsbeispiel:

- Watchdog ein
- Watchdog-Zeit (WZ) = 15 ms
- Watchdog-Empfindlichkeit = 2

Die Intervallzeit (IZ) der Task beträgt 10 ms.

Variante ①: Der Watchdog wird nicht ausgelöst, da die Zeit stets unter der angegebenen Watchdog-Zeit bleibt.

Variante ②: Der Watchdog wird 15 ms nach Beginn des zweiten Intervalls ausgelöst ⚡, da beide Zeiten länger als die angegebene Watchdog-Zeit sind und unmittelbar aufeinander folgen.

Variante ③: Der Watchdog wird 15 ms nach Beginn der zweiten aufeinander folgenden Task ausgelöst, die länger als die angegebene Watchdog-Zeit ist.

Variante ④ – Endlosschleife: Der Watchdog wird ausgelöst ⚡, weil die Task länger dauert als die Watchdog-Zeit multipliziert mit der Watchdog-Empfindlichkeit ( $2 \times 15 \text{ ms} = 30 \text{ ms}$ ).

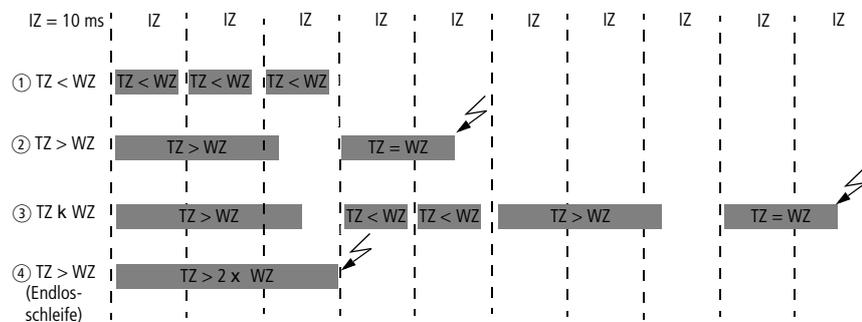


Abbildung 40: Watchdog aktiv – mehrere Tasks mit verschiedener Priorität

### 5.4.0.3 Watchdog deaktiviert

Bei deaktiviertem Watchdog wird die Ausführungszeit einer Task nicht überwacht. Wird eine Task bei ausgeschaltetem Watchdog nicht innerhalb der vorgewählten Intervallzeit beendet, wird diese Task im darauf folgenden Zyklus nicht aufgerufen bzw. gestartet. Eine Task wird erst dann wieder gestartet, wenn sie im Zyklus zuvor beendet wurde.

## 5 Programmbearbeitung, Multitasking und Systemzeiten

### 5.5 Direkter Peripheriezugriff

#### 5.4.0.4 Beispiel: Watchdog deaktiviert

Die Intervallzeit (IZ) beträgt 10 ms.

Variante ①: Die Intervall-Zeit (IZ) einer Task wurde auf 10 ms festgelegt. Die tatsächliche Tasklaufzeit (TZ) beträgt 15 ms. Die Task wird beim ersten Aufruf gestartet, aber erst während des zweiten Zyklus beendet. Somit wird diese Task im zweiten Zyklus nicht erneut angestoßen. Erst im dritten Zyklus – nach 20 ms – kann diese Task erneut gestartet werden. Die Task läuft somit nicht alle 10 ms sondern nur in einem Zeitintervall von  $2 \times 10$  ms.

Variante ②: Der laufende Zyklus wird nicht beendet.

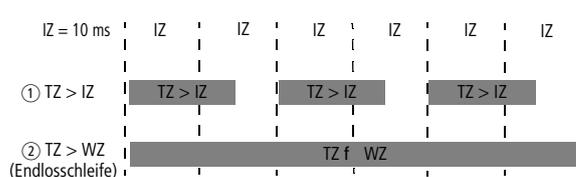


Abbildung 41: Watchdog deaktiviert

#### 5.4.1 Mehrere Tasks mit gleicher Priorität

Sie können mehreren Tasks die gleiche Priorität zuordnen. Die Tasks werden dann nach dem „Time Slice“-Prinzip gestückelt und als Teilintervalle quasi zeitgleich ausgeführt (Round Robin).

### 5.5 Direkter Peripheriezugriff

Die Funktion „Direkter Peripheriezugriff“ ermöglicht es, direkt auf die lokalen und zentralen Ein- und Ausgangs-Signale der Steuerung zuzugreifen. Der I/O-Zugriff erfolgt nicht über das Ein- und Ausgangs-Abbild. Die lokalen und zentralen Ein- und Ausgangs-Signale sind die Ein-/Ausgänge der CPU und die der mit XIOC-Signalbaugruppen zentral erweiterten XC-200-Steuerung. XIOC-Signalbaugruppen, die über ein Bussystem eingebunden werden können, sind über „Direkter Peripheriezugriff“ nicht erreichbar.

Die Adressierung ist abhängig von der Steckplatznummer „0 bis 15“ der Signalbaugruppen. Eine weitere Differenzierung erfolgt innerhalb des Steckplatzes und bezieht sich auf die Bitnummer „0 bis max. 63“ der Ein-/Ausgänge.

Der Zugriff erfolgt je nach Funktionalität der XIOC-Signalbaugruppen als Bit- oder Word- bzw. als Lese- oder Schreib-Zugriff. Die Zugriffsparameter zeigt Tabelle 8.

Zur physikalischen Anbindung werden die Ein-/Ausgänge, die über den „Direkten Peripheriezugriff“ angesprochen werden, genauso verdrahtet, wie die normalen Ein-/Ausgänge.

Tabelle 8: Übersicht „Direkter Peripheriezugriff“

Baugruppe	I/O-Bit-Zugriff			I/O-Word-Zugriff			I/O-Slot
	Read	Write	Parameter/Baugruppe	Read	Write	Parameter/Baugruppe	
XC-CPU201-EC256K-8DI-6DO	✓	✓	DI: 0 bis 7, DO: 0 bis 5	✓	✓	0	0
XC-CPU201-EC256K-8DI-6DO-XV	✓	✓	DI: 0 bis 7, DO: 0 bis 5	✓	✓	0	0
XC-CPU201-EC512K-8DI-6DO	✓	✓	DI: 0 bis 7, DO: 0 bis 5	✓	✓	0	0
XC-CPU201-EC512K-8DI-6DO-XV	✓	✓	DI: 0 bis 7, DO: 0 bis 5	✓	✓	0	0
XC-CPU202-EC4M-XV	✓	✓	DI: 0 bis 7, DO: 0 bis 5	✓	✓	0	0
XIOC-8DI	✓	–	0 bis 7	✓	–	0	1 bis 15
XIOC-16DI	✓	–	0 bis 15	✓	–	0	1 bis 15
XIOC-8DO	–	✓	0 bis 7	–	✓	0	1 bis 15
XIOC-16DO	–	✓	0 bis 15	–	✓	0	1 bis 15
XIOC-16DO-S	–	✓	0 bis 15	–	✓	0	1 bis 15
XIOC-12DO-R	–	✓	0 bis 11	–	✓	0	1 bis 15
XIOC-16DX	–	✓	0 bis 15	✓	✓	0	1 bis 15
XIOC-8AI-I2	–	–	–	✓	–	0 bis 7	1 bis 15
XIOC-8AI-U1	–	–	–	✓	–	0 bis 7	1 bis 15
XIOC-8AI-U2	–	–	–	✓	–	0 bis 7	1 bis 15
XIOC-4T-PT	–	–	–	✓	–	0 bis 3	1 bis 15
XIOC-4AI-T	–	–	–	✓	–	0 bis 3	1 bis 15
XIOC-2AO-U1-2AO-I2	–	–	–	–	✓	0 bis 3	1 bis 15
XIOC-4AO-U1	–	–	–	–	✓	0 bis 3	1 bis 15
XIOC-4AO-U2	–	–	–	–	✓	0 bis 3	1 bis 15
XIOC-2AO-U2	–	–	–	–	✓	0 bis 1	1 bis 15
XIOC-4AI-2AO-U1	–	–	–	✓	✓	AI: 0 bis 3/ AO: 0 bis 1	1 bis 15
XIOC-2AI-1AO-U1	–	–	–	✓	✓	AI: 0 bis 1/ AO: 0	1 bis 15
XIOC-1CNT-100KHZ	–	–	–	–	–	–	1 bis 15
XIOC-2CNT-100KHZ	–	–	–	–	–	–	1 bis 15
XIOC-2CNT-2AO-INC	–	–	–	✓	✓	–	1 bis 15
XIOC-NET-DP-M	–	–	–	–	–	–	1 bis 3

### 5.5.1 ReadBitDirect

Mit dieser Funktion können Sie ein Bit eines Eingangsmoduls direkt lesen. Der Zustand des Eingangs-Bits wird in der Variablen abgelegt, auf die der parametrisierte Pointer „ptr\_xValue“ zeigt. Die Pointer-Variable wird nicht verändert, falls bei der Bearbeitung ein Fehler auftritt.

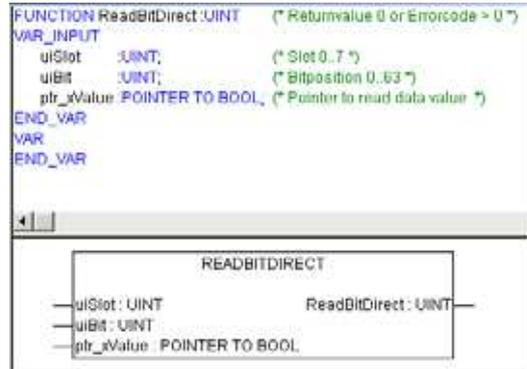


Abbildung 42: Funktion „ReadBitDirect“

#### 5.5.1.1 Parameter der Funktion „ReadBitDirect“

uiSlot	Steckplatznummer des Signalmoduls. Mögliche Parameter siehe Tabelle 8 auf Seite 61
uiBit	Bitposition innerhalb des Eingangswertes des Signalmoduls. Mögliche Parameter siehe Tabelle 8 auf Seite 61.
ptr_xValue	Zeiger auf den Variablenwert
ReadBitDirect	Anzeige des Fehlercodes siehe Tabelle 9 auf Seite 65.

### 5.5.2 ReadWordDirect

Mit dieser Funktion kann ein Wort eines Eingangsmoduls direkt gelesen werden. Der Zustand des Eingangs-Wortes wird in der Variablen abgelegt, auf die der parametrisierte Pointer „ptr\_wValue“ zeigt.

Die Pointer-Variable wird nicht verändert, falls bei der Bearbeitung ein Fehler auftritt.

#### 5.5.2.1 Parameter der Funktion „ReadWordDirect“

uiSlot	Steckplatznummer des Signalmoduls. Mögliche Parameter siehe Tabelle 8 auf Seite 61.
uiOffset	Wortoffset innerhalb des Signalmoduls. Mögliche Parameter siehe Tabelle 8 auf Seite 61
ptr_wValue	Zeiger auf den Variablenwert
ReadWordDirect	Anzeige des Fehlercodes Tabelle 9 auf Seite 65.

### 5.5.3 ReadDWordDirect

Mit dieser Funktion können Sie ein Doppelwort eines Eingangsmoduls oder einer Eingangsfunktion wie zum Beispiel den Zählerwert des 32Bit-Counters direkt lesen. Der Zustand des Doppel-Wortes wird in der Variablen abgelegt, auf die der parametrierte Pointer „ptr\_dwValue“ zeigt.

Die Pointer-Variable wird nicht verändert, falls bei der Bearbeitung ein Fehler auftritt.

#### 5.5.3.1 Parameter der Funktion „ReadDWordDirect“

uiSlot	Steckplatznummer des Signalmoduls Mögliche Parameter siehe Tabelle 8 auf Seite 61
uiOffset	Wordoffset innerhalb des Signalmoduls Mögliche Parameter siehe Tabelle 8 auf Seite 61
ptr_dwValue	Zeiger auf den Variablenwert
ReadDWordDirect	Anzeige des Fehlercodes Tabelle 9 auf Seite 65

### 5.5.4 Write...Direct

Grundsätzlich sollten Ausgänge der SPS nur aus einer Task oder von einem Interrupt verändert werden. Arbeiten Sie bitte innerhalb von Interrupts immer mit direkten Zugriffsfunktionen, da Ereignisse kein Abbild haben.

Werden in einer Anwendung Ausgänge von verschiedenen Tasks oder Ereignissen verändert, sind folgende Regeln zu beachten:

- Wird ein Ausgangsbit mit der **WriteBitDirect**-Funktion bei einem Ereignis (Interrupt- oder Ereignistask) bearbeitet, darf das Ausgangswort „Q-WORD“, in dem das Bit liegt, in keiner anderen Task deklariert oder referenziert werden!  
Die anderen Bits des Ausgangswortes dürfen aber als Ausgangsbit „Q-BOOL“ weiterhin in anderen Tasks zugewiesen werden.
- Wird ein Ausgangsbit zur schnellen Verarbeitung mit Hilfe der Funktion **WriteBitDirect** verändert und wird dieses Bit auch an einer anderen Stelle (Task, Event oder Interrupt) bearbeitet, muss an allen Stellen die Funktion **WriteBitDirect** verwendet werden (keine Q-BOOL-Deklaration und keine Referenzierung).

#### 5.5.4.1 Beispiel

Deklaration Ausgangsvariable:

```
Q-BOOL (z.B.Qbit3)      AT%QX1.2:BOOL;
Q-WORD (z.B.Qword0)    AT%QW0:WORD;
```

Referenzierung (Zuweisung in der Anwendung):

```
Qbit3:=TRUE;
Qword0:=16#Test;
```

### 5.5.5 WriteBitDirect

Mit dieser Funktion kann ein Bit eines Ausgangsmoduls direkt angesteuert werden. Neben dem physikalischen Ausgang wird auch das zugehörige Ausgangsabbild aktualisiert. Das Schreiben auf den Ausgang ist ohne Einschränkungen nur für die lokalen sechs Ausgänge der XC200-CPU mit dem Steckplatz „0“ möglich.

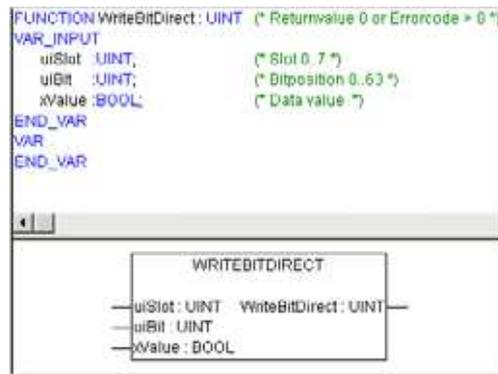


Abbildung 43: Funktion „WriteBitDirect“

#### 5.5.5.1 Parameter der Funktion „WriteBitDirect“

uiSlot	Steckplatznummer des Signalmoduls. Mögliche Parameter siehe Tabelle 8 auf Seite 61.
uiBit	Ausgangsbit innerhalb des Signalmoduls. Mögliche Parameter siehe Tabelle 8 auf Seite 61.
xValue	Der Zeiger zeigt auf die Variable, in der der Wert für das Ausgangs-Bit liegt.
WriteBitDirect	Anzeige des Fehlercodes siehe Tabelle 9 auf Seite 65

### 5.5.6 WriteWordDirect

Mit dieser Funktion kann ein Wort eines Ausgangsmoduls direkt geschrieben werden. Neben dem physikalischen Ausgang wird zum Zeitpunkt des Zugriffs auch das zugehörige Ausgangsabbild aktualisiert.

Eine weitere Aktualisierung des Ausgang-Wortes erfolgt am Zyklusende.

#### 5.5.6.1 Parameter der Funktion „WriteWordDirect“

uiSlot	Steckplatznummer des Signalmoduls. Mögliche Parameter siehe Tabelle 8 auf Seite 61.
uiOffset	Ausgangswort innerhalb des Signalmoduls. Mögliche Parameter siehe Tabelle 8 auf Seite 61.
wValue	Der Zeiger zeigt auf die Variable, in der der Wert für das Ausgangs-Wort liegt.
WriteWordDirect	Anzeige des Fehlercodes siehe Tabelle 9 auf Seite 65

### 5.5.7 GetSlotPtr



Diese Funktion steht Ihnen nicht zur Verfügung!

### 5.5.8 Fehlercode bei direktem Peripheriezugriff

Alle Funktionen prüfen soweit wie möglich die Gültigkeit der Aufrufparameter. Es wird geprüft, ob der Zugriff in Abhängigkeit vom parametrisierten Signalmodul und der physikalischen Existenz des Signalmoduls erfolgt. Bei einem festgestellten Fehler wird der Zugriff nicht durchgeführt und ein Fehlercode ausgegeben. Die Datenfelder für die Wertübergabe bleiben unverändert. Die Funktionen **DisableInterrupt** und **EnableInterrupt** generieren keinen Fehlercode.

Folgende Rückgabewerte sind möglich:

Tabelle 9: Fehlercodes beim direkten Peripherie-Zugriff

IO_ACCESS_NO_ERROR:	kein Fehler
IO_ACCESS_INVALIDE_SLOTNUMBER	Steckplatz = 0 oder größer 15
IO_ACCESS_INVALIDE_OFFSET	Bit-/Word-Offset ist zu groß
IO_ACCESS_DENIED	ungültiger Zugriff, z. B. Schreibzugriff auf Eingangsmodul, Lesezugriff auf Ausgangsmodul oder Zugriff auf nicht vorhandenen Adressbereich (Offset zu groß)
IO_ACCESS_NO_MODULE	Kein Modul auf parametrisiertem Steckplatz vorhanden
IO_ACCESS_INVALIDE_Buffer	Kein oder falscher Zeiger auf die Ausgangsvariable
IO_ACCESS_INVALIDE_Value	Ereignis ist nicht „0“ oder „1“ bei WriteBitDirect

## 5.6 Betriebszustände

In der folgenden Übersicht finden Sie die Zustandsdefinitionen der CPU. Die LED-Anzeigen der jeweiligen Zustände sind mit aufgeführt.

Tabelle 10: Zustandsdefinitionen der XC200 mit LED-Anzeige

Zustand	Anzeige		Definition
	RUN/STOP	SF	
Boot	aus (blinkt bei Start)	ein	Serieller Bootloader startet und bootet und/oder updatet das Betriebssystem. Windows CE wird aus dem Flash geladen und entpackt in den Speicher kopiert und gestartet.
Betriebssystem Start	aus (blinkt bei Start)	aus	Windows CE Systemstart und System-Test wird durchgeführt. Start von Applikationen <ul style="list-style-type: none"> <li>• HTTP-Server</li> <li>• FTP-Server</li> <li>• Telnet-Server</li> <li>• PLC-Runtime</li> <li>• Webserver</li> </ul>
STOP	blinkt	aus	SPS im Zustand STOP

Zustand	Anzeige		Definition
	RUN/STOP	SF	
RUN	ein	aus	SPS im Zustand RUN
RUN   STOP	blinkt	ein	Fehler im RUN-Zustand „Zykluszeit Überschreitung“. Das System wird in den Halt-Zustand gesetzt. Es wird automatisch ein „Reset-Warm“-Kommando ausgeführt. Das Auftreten des Fehlers wird in der „Error List“ protokolliert. (Auslesen mit Browser-Befehl <code>geterrorlist</code> )
NOTREADY	blinkt	ein	Kein Start möglich. Ein gravierender Fehler verhindert einen Start (siehe „Error Liste“) z. B.: <ul style="list-style-type: none"> <li>• kein Program geladen</li> <li>• Feldbusfehler</li> <li>• Konfigurationsfehler</li> <li>• Prüfsummenfehler</li> <li>• ...</li> </ul>
ShutDown	blinkt	blinkt	Warten auf das Abschalten der Versorgungsspannung (nach Browser-Befehl <code>shutdown</code> )

### 5.7 Web-Visualisierung

Die Beschreibung der Web-Visualisierung finden Sie im Handbuch für SPS-Programmierung mit CODESYS in Abschnitt 7.4, „Web-Visualisierung“.

Der XC-CPU201-spezifische Aufruf der Web-Visualisierung lautet:

`http://192.168.119.200:8080/webvisu.htm`

Der XC-CPU202-spezifische Aufruf der Web-Visualisierung lautet:

`http://192.168.119.202:8080/webvisu.htm`

Voraussetzung ist, dass Sie die Defaulteinstellung der IP-Adresse nicht geändert haben.

Falls Sie die IP-Adresse verändert haben, so ersetzen Sie die IP-Adresse im „`http://...`“-Aufruf bitte mit der von Ihnen gewählten Adresse.

**ACHTUNG**

Es dürfen maximal 10 Clients auf die XC200 zugreifen!

### 5.8 Grenzwerte für die Speichernutzung

Der Datenspeicher der XC200 für Daten ist in Speichersegmente gegliedert. Die Speichergröße der einzelnen Segmente geht aus Abbildung 44 hervor. Die globalen Daten verfügen über mehrere Segmente. Die gewünschte Anzahl geben Sie in Abhängigkeit von der Programmgröße vor.

Die Segmentgröße für die verschiedenen Steuerungstypen erkennen Sie unter `⟨Ressourcen | Zielsystemeinstellungen | Speicheraufteilung⟩`:

Beispiel:



Abbildung 44: Segmentgröße der XC-CPU201-EC256k

Die hexadezimalen Werte der anderen Steuerungstypen sind in die dezimalen Werte umzurechnen.

Damit Sie den zur Verfügung stehenden Speicherbereich der globalen Daten optimal und effizient nutzen können, empfehlen wir Ihnen, schon bei der Neuanlage eines Projekts die nachfolgenden Einstellungen vorzunehmen:

Steuerungstyp	Anzahl der Datensegmente (Global)
XC-CPU201-EC256K	2
XC-CPU201-EC512K	4
XC-CPU202-EC4M	1

Per Default steht die Anzahl der Segmente auf 1.

Die Anzahl der Segmente verändern Sie wie folgt:

- ▶ Wählen Sie im Verzeichnis «Projekt → Optionen → Übersetzungsoptionen» das Feld „Anzahl Datensegmente“ an und tragen Sie entsprechend dem ausgewählten Steuerungstyp die oben angegebene Anzahl der Segmente ein.

## 5 Programmbearbeitung, Multitasking und Systemzeiten

### 5.9 Ein-/Ausgänge und Merker adressieren

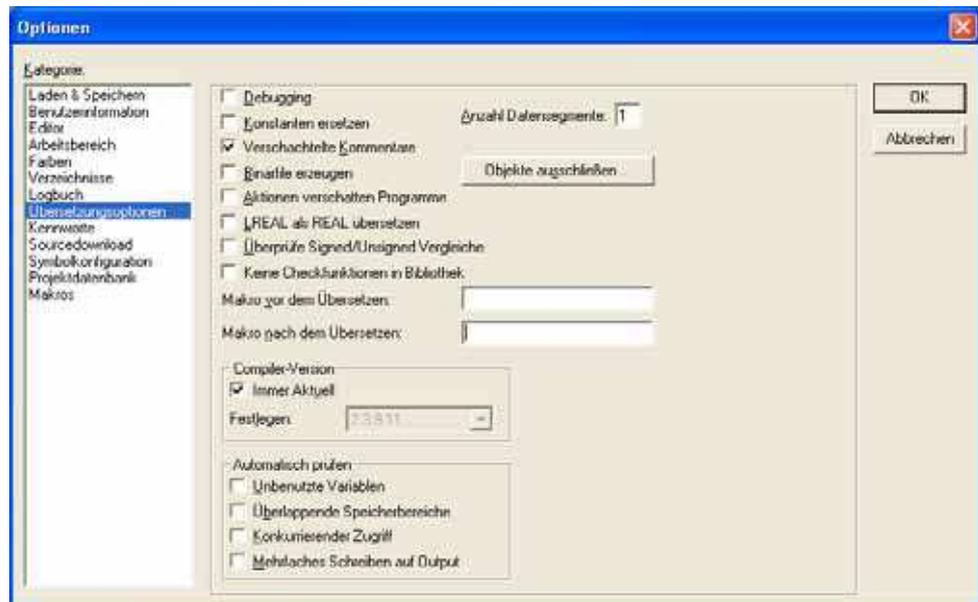


Abbildung 45: Speicherverwaltung: Anzahl der Datensegmente ändern

### 5.9 Ein-/Ausgänge und Merker adressieren

Wenn Sie die Steuerungskonfiguration eines neuen Projekts öffnen, erhalten Sie die aktuelle Ansicht der Defaulteinstellung zur Adressierung. In dieser Einstellung werden Adressen automatisch vergeben und Adressüberschneidungen gemeldet.

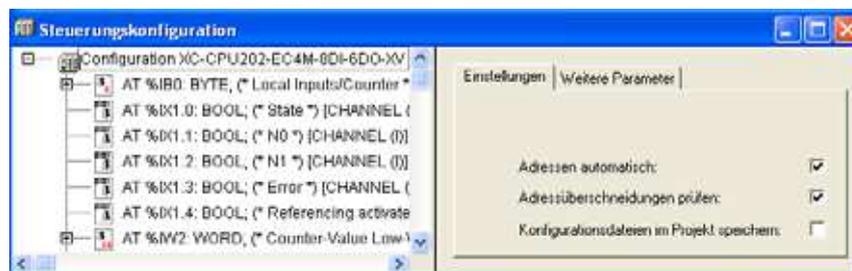


Abbildung 46: Defaulteinstellung zur Adressierung

Fügen Sie der Steuerung im Konfigurator ein Modul hinzu, vergibt der Konfigurator für dieses Modul eine Adresse. Weitere Module erhalten die folgenden Adressen in aufsteigender Reihenfolge. Sie können auch die Adressvergabe frei gestalten. Rufen Sie jedoch später die Funktion „Adressen berechnen“ auf, werden die Adressen in aufsteigender Reihenfolge neu geordnet.

#### 5.9.1 „Adressen automatisch“ aktivieren

Die Adressen werden beim Ändern oder beim Einfügen eines Moduls automatisch vergeben bzw. geändert. Das erfolgt sowohl bei einem zentral angeordneten Modul als auch bei einem Modul, das Bestandteil eines dezentralen PROFIBUS-DP-Slaves oder CAN-Teilnehmers ist.

Wenn Sie ein Modul einfügen, werden die Adressen aller nachfolgender Module (unabhängig vom Strang) um die Adressanzahl des eingefügten Moduls verschoben und das eingefügte Modul bekommt eine Adresse. Module, die sich in der Konfiguration vor dem eingefügten Modul befinden, werden nicht geändert. Entfernen Sie den Haken im Betätigungsfeld „Adressen automatisch“, so bleiben die Adressen bei Änderungen/Erweiterungen bestehen.

#### 5.9.2 „Adressüberschneidungen prüfen“ aktivieren

Ist die Prüfung der Adressüberschneidung aktiviert, werden doppelt vergabene Adressen erkannt und beim Übersetzen eine Fehlermeldung erzeugt. Diese Einstellung sollten Sie nicht verändern.

#### 5.9.3 Ungerade Wort-Adressen

Wenn Sie einem wortadressierbaren Modul im Feld „Eingabeadresse“ eine ungerade Offset-Adresse zuordnen (z. B. IB5), erscheint im Steuerungskonfigurator automatisch die folgende gerade Word-Adresse (IW6). Dies geschieht automatisch und ist unabhängig von der Einstellung „Adressüberschneidung prüfen“.

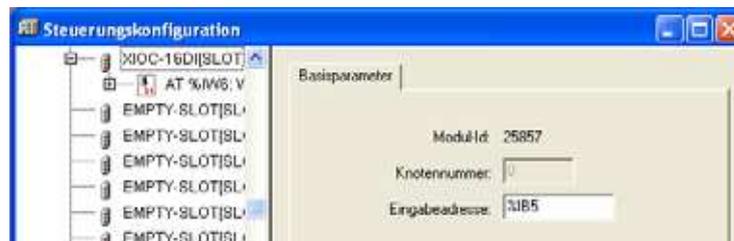


Abbildung 47: Ungerade Adressen

#### 5.9.4 Adressbereich

Adressen können Sie nur innerhalb der gültigen Bereiche vergeben. Die Bereichsangaben finden Sie unter «Zielsystemeinstellungen → Speicheraufteilung | Größe».

Die Adressen werden beim Übersetzen geprüft. Achten Sie bitte grundsätzlich darauf, dass die Adressen des konfigurierten Moduls im Programm benutzt (referenziert) werden. Überschreitet die Adresse den Bereich, so wird ein Fehler gemeldet.

Tabelle 11: Adressbereiche

Steuerung	Input			Output			Merker		
	Größe	Max. Byteadr.	Max. Wordadr.	Größe	Max. Byteadr.	Max. Wordadr.	Größe	Max. Byteadr.	Max. Wordadr.
	[kByte]			[kByte]			[kByte]		
XC101-64k	2	2047	2046	2	2047	2046	4	4095	4094
XC101-128k	4	4095	4094	4	4095	4094	8	8191	8190
XC101-256k	16	16383	16382	16	16383	16382	16	16383	16382
XC201-256k	4	4095	4094	4	4095	4094	16	16383	16382
XC201-512k	4	4095	4094	4	4095	4094	16	16383	16382
XC202-EC4M	4	4095	4094	4	4095	4094	16	16383	16382

#### 5.9.5 Adressen von Eingabe-/Ausgabe-Modulen und Diagnoseadresse frei vergeben oder verändern

Je nach Modul können Sie die Eingabe-, Ausgabe- und die Diagnose(Merker)-Adressen vergeben/verändern.

Damit die Änderungen im Steuerungskonfigurator sichtbar werden, müssen Sie nach dem Editieren einer Adresse einmal in den Steuerungskonfigurator klicken oder ein anderes Modul auswählen. Beim Übersetzen werden sie auf jeden Fall übernommen.

#### 5.9.6 „Adressen berechnen“ durchführen

Mit der Funktion „Adressen berechnen“, die Sie entweder über das Kontextmenü oder über die Menüleiste unter „Extras“ ausführen können, werden alle dem Modul zugehörigen Adressen neu berechnet. Handelt es sich um ein Bus-Master-Modul, findet die Berechnung auch für die Module statt, die Bestandteil eines Slave am Busstrang sind. Die frei eingetragenen Adressen von untergeordneten Modulen werden überschrieben, wenn die Adresse eines übergeordneten Moduls berechnet wird. Haben Sie Adressen geändert und wollen Sie „Adressen berechnen“ durchführen, so müssen Sie zunächst die Änderung aktivieren. Klicken Sie dazu auf den Knoten, um die Struktur aufzuklappen oder setzen Sie den Cursor in das Feld „Steuerungskonfiguration“ und betätigen Sie die linke Maustaste.

Wenn Sie den Schriftzug „Configuration XC-CPU...“ markieren und „Adressen berechnen“ aufrufen, werden alle Adressen neu berechnet.



Geben Sie Adressen möglichst in aufsteigender Reihenfolge und in zusammenhängenden Blöcken ein.

## 5.10 Diagnose

Mit Hilfe von Diagnose-Funktionsbausteinen können Sie eine Diagnose durchführen. Es bestehen folgende Möglichkeiten:

Art der Diagnose	Funktionsbaustein	Bibliothek	Dokumentation
Überprüfung der XIOC-Module: <ul style="list-style-type: none"> <li>• Stimmt die Konfiguration im Konfigurator mit der Hardware überein?</li> <li>• Ist die Modulfunktion in Ordnung?</li> </ul> Bemerkung: Diese Überprüfungen erfolgen einmalig beim Einschalten oder nach dem Laden bzw. Starten des Programms	XDiag_SystemDiag	xSysDiag.lib	MN05010002Z-DE (frühere Bezeichnung AWB2786-1456)
Überprüfung des Moduls XIOC-NET-DP-M und der Teilnehmer am DP-Strang	XDiag_SystemDiag XDiag_ModuleDiag	XSysDiag.lib	MN05010002Z-DE (frühere Bezeichnung AWB2786-1456)
	DiagGetState	BusDiag.lib	MN05002002Z-DE (frühere Bezeichnung AWB2725-1452)
Überprüfung des Moduls XIOC-NET-DP-S	XDiag_SystemDiag XDiag_ModuleDiag	xSysDiag.lib	MN05010002Z-DE (frühere Bezeichnung AWB2786-1456)
Der DP-Slave stellt dem Master zusätzliche Diagnosedaten zur Verfügung.	XDPS_SendDiag	xSysNetDPSDiag.lib	MN05002002Z-DE (frühere Bezeichnung AWB2725-1452)

## 6 Verbindungsaufbau PC – XC200

### 6.1 Verbindungsaufbau über RS232-Schnittstelle

## 6 Verbindungsaufbau PC – XC200

Die Verbindung zwischen PC und CPU können Sie herstellen über

- die RS232-Schnittstelle,
- die Ethernet-Schnittstelle.

In diesem Kapitel erfahren Sie, welche Einstellungen Sie hierzu in der Software CODESYS vornehmen müssen.

Siehe auch:

- PC anschließen → Seite 28

### 6.1 Verbindungsaufbau über RS232-Schnittstelle

Zur Herstellung der Verbindung zwischen PC und CPU müssen die Kommunikationsparameter beider Geräte übereinstimmen.

- Passen Sie hierzu zunächst die Kommunikationsparameter des PC an die Standardparameter der CPU an a Abschnitt „Kommunikationsparameter des PC festlegen/ändern“.

Die CPU besitzt folgende Standardparameter:

Baudrate	38400
Parity	No
Stopbits	1
Motorola Byte	No



Falls Sie beim Einloggen eine Fehlermeldung erhalten, sind die Standardeinstellungen der CPU bereits verändert worden. Versuchen Sie es in diesem Fall mit einer Baudrate von 57600 Bit/s.

- Nach dem Einloggen können Sie die Parameter der CPU neu festlegen (a Abschnitt „Kommunikationsparameter der CPU ändern“).

### 6.2 Kommunikationsparameter des PC festlegen/ändern

In der Software CODESYS legen Sie die Kommunikationsparameter der Schnittstelle fest. Sie können die COM1- oder COM2-Schnittstelle des PC verwenden.

- ▶ Wählen Sie den Menüpunkt «Online → Kommunikationsparameter» aus.
- ▶ Legen Sie den Port (COM1- oder COM2-Schnittstelle) fest a Abschnitt „Parameter ändern“
- ▶ Übernehmen Sie die weiteren Parameter aus der Abbildung 48.
- ▶ Bestätigen Sie die Parameter mit „OK“.
- ▶ Loggen Sie sich in die Steuerung ein.



Abbildung 48: Kommunikationsparameter des PC festlegen



Weitere Hinweise zu den Kommunikationsparametern finden Sie in der Online-Hilfe des Programmiersystems.

Ab Betriebssystem V01.03.xx der XC-CPU201 kann der Kommunikationskanal serial (RS232) (Level 2 Route) ausgewählt und eine Target ID angegeben werden. Wenn Sie für die Target ID eine „0“ eintragen, wird mit der lokalen Steuerung kommuniziert.

### 6.2.0.1 Parameter ändern

Um Parameter wie beispielsweise die Baudrate oder die Port-Angabe zu verändern, gehen Sie wie folgt vor:

- ▶ Klicken Sie doppelt auf den entsprechenden Wert, z. B. 38400. Das Feld wird grau unterlegt.
- ▶ Geben Sie den gewünschten Wert an.

Mit weiteren Doppelklicks in diesem Feld können Sie die gewünschte Baudrate, z. B. 57600 Bit/s, auswählen.

### 6.2.1 Kommunikationsparameter der CPU ändern

- ▶ Wählen Sie in „Ressourcen“ den „PLC-Browser“ aus.
- ▶ Wählen Sie den Browser-Befehl „setcomconfig“ aus und fügen Sie nach einem Leerzeichen die gewünschte Baudrate hinzu.
- ▶ Quittieren Sie die Vorwahl mit RETURN.
- ▶ Wählen Sie den Browser-Befehl save registry.
- ▶ Wählen Sie den Browser-Befehl reboot. Ist der Reboot erfolgt, wird die neue Baudrate in der XC200 aktiviert.

Greifen Sie nun auf die CPU zu (z. B. durch Anmelden), erhalten Sie folgende Fehlermeldung:



Abbildung 49: Kommunikationsfehler

## 6 Verbindungsaufbau PC – XC200

### 6.3 Verbindungsaufbau über Ethernet

Um mit der CPU kommunizieren zu können, müssen Sie die Kommunikationsparameter des PC anpassen a Abschnitt „Kommunikationsparameter des PC festlegen/ändern“.

#### 6.3 Verbindungsaufbau über Ethernet

Nachdem Sie den PC mit der CPU durch ein Kabel verbunden haben, wählen Sie in der Software CODESYS den Kommunikationskanal TCP/IP aus und geben die IP-Adresse der CPU an. Die XC-CPU201 hat die Default-Adresse 192.168.119.200; die XC-CPU202 hat die Default-Adresse 192.168.119.202

Die Auswahl der Übertragungsgeschwindigkeit der Ethernet-Verbindung erfolgt im Autosensing(detect)-Mode. Komponenten mit dieser Eigenschaft erkennen automatisch, ob es sich um eine 10- oder eine 100-MBit-Verbindung handelt.

##### 6.3.1 Kommunikationskanal und Adresse auswählen

- ▶ Rufen Sie das Menü «Online → Kommunikationsparameter» auf.



Abbildung 50: Kanalauswahl

- ▶ Betätigen Sie die Schaltfläche „Neu...“.
- ▶ Wählen Sie aus der Übersicht den Kommunikationskanal TCP/IP (Level 2 Route) aus und ändern Sie den Namen „Lokal“ beispielsweise in „Ethernet-Test“.
- ▶ Bestätigen Sie mit „OK“.



Abbildung 51: IP-Adresse eingeben

- ▶ Führen Sie einen Doppelklick auf das Feld „localhost“ aus und tragen Sie die Default-Adresse 192.168.119.200 bzw. 192.168.119.202 ein.
- ▶ Bestätigen Sie Ihre Angaben, indem Sie zunächst auf ein beliebiges anderes Feld und erst dann auf OK klicken.



Abbildung 52: Kommunikationsparameter mit IP-Adresse

- ▶ Kompilieren Sie das Programm und loggen Sie sich ein.

## 6.4 Abfragen/Ändern der IP-Adresse

Für das Abfragen und Ändern der IP-Adresse stehen Ihnen die Browser-Befehle `setipconfig` und `getipconfig` zur Verfügung (a Abschnitt „Browser-Befehle“ auf Seite 130).

Nachdem Sie die IP-Adresse geändert haben, starten Sie die XC200 neu. Die DHCP-Funktion (DHCP = Dynamic Host Configuration Protocol) ist nicht aktiviert.

Achten Sie darauf, dass die IP-Adresse des Programmiergeräts (= PC) der gleichen Adressfamilie wie die der Steuerung angehört. Dies bedeutet, dass die IP-Adresse des Programmiergerätes und der XC200 in folgenden Zifferngruppen übereinstimmt:

### 6.4.0.1 Beispiel 1

IP-Adresse XC200:192.168.119.xxx  
IP-Adresse PC: 192.168.119.yyy

### 6.4.0.2 Beispiel 2

IP-Adresse XC200:192.168.100.xxx  
IP-Adresse PC: 192.168.100.yyy

In den Beispielen 1 und 2 gelten folgende Bedingungen:

- xxx ungleich yyy,
- die Adressen müssen in den Grenzen 1 und 254 liegen,
- die Adressen müssen der gleichen Adressfamilie angehören.

Kommt keine Verbindung zu Stande, können Sie den Übertragungsweg mit der „PING“-Funktion überprüfen, um sicher zu sein, dass der Verbindungsaufbau nicht am Übertragungsweg scheitert. Hierzu sind folgende Schritte erforderlich:

- ▶ Öffnen Sie das DOS-Fenster über das Feld „START“ und den Befehl „Ausführen“.
- ▶ Geben Sie in das Eingabefeld „CMD“ ein und bestätigen Sie mit „OK“.

## 6 Verbindungsaufbau PC – XC200

### 6.4 Abfragen/Ändern der IP-Adresse

Sie erhalten ein Fenster mit einer Laufwerksangabe und einem blinkenden Cursor hinter der Laufwerksbezeichnung.

- ▶ Geben Sie hier konkret für dieses Beispiel den folgenden Text ein:  
ping 192.168.119.200 für die XC-CPU201 bzw. ping 192.168.119.202 für die XC-CPU202. Bestätigen Sie mit „OK“.

Ist das Routing in Ordnung, erhalten Sie eine Antwort mit Angabe einer Response-Zeit. Andernfalls werden Sie auf eine Zeitüberschreitung beim Verbindungsaufbau hingewiesen.

Die nachfolgende Abbildung zeigt das Ergebnis eines ordnungsgemäßen Verbindungsaufbaus.



```
C:\WINNT\system32\CMD.exe
Microsoft Windows [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\>ping 192.168.119.60

Ping wird ausgeführt für 192.168.119.60 mit 32 Bytes Daten:

Antwort von 192.168.119.60: Bytes=32 Zeit<10ms TTL=128

Ping-Statistik für 192.168.119.60:
    Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0
              (0% Verlust),
    Ca. Zeitangaben in Millisek.:
    Minimum = 0ms, Maximum = 0ms, Mittelwert = 0ms

C:\>
```

Abbildung 53: PING-Antwort bei einer ordnungsgemäßen Ethernet-Verbindung

## 7 Systemparameter über Datei Startup.ini vorgeben

### 7.1 Übersicht

Projektunabhängige Systemparameter können von Ihnen erstellt und auf der Speicherkarte gespeichert werden. Sie werden dort in der Startup.ini-Datei zusammengefasst. Die Speicherkarte können Sie auch in andere Steuerungen stecken. Beim Start übernimmt die Steuerung die Parameter. Die Startup.ini-Datei wird immer mit allen steuerungsspezifischen Einträgen erstellt (→ Tabelle 12). Der Begriff Startup.ini-Datei ist allgemein gültig. Der Dateiname der Startup.ini-Datei für die XC200 lautet XCSTARTUP.ini.

#### 7.1.1 Parameter in der Datei Startup.ini

Einige Parameter, wie zum Beispiel die Baudrate der COM-Schnittstelle, sind vom System bereits eingetragen, damit eine Kommunikation zwischen PC und Steuerung stattfinden kann. Die Parameter können Sie später anpassen.

Tabelle 12: Voreingetragene Defaultparameter in der Datei XCSTARTUP.ini

```
TARGET=XC-CPU202
HOST_NAME=NoNameSet
IP_ENABLE_DHCP=01)
IP_ADDRESS=192.168.119.202
IP_SUBNETMASK=255.255.255.0
COM_BAUDRATE=38400
CAN_ROUTING_CHANNEL=1
```

1) Hinweis: Der Parameter „IP\_ENABLE\_DHCP“ existiert nur bei einer XC-CPU202.

Tabelle 13: Beispiel: Inhalt der Datei XCSTARTUP.ini

```
[STARTUP]
TARGET=XC-CPU202
zur Ethernet-Verbindung:
HOST_NAME=NoNameSet
IP_ADDRESS=192.168.119.200
IP_SUBNETMASK=255.255.255.0
IP_GATEWAY=
IP_DNS=
IP_WINS=
zur Programmierschnittstelle RS232:
COM_BAUDRATE=4800, 9600, 19200, 38400, 57600
zur CAN-Schnittstelle:
CAN1_BAUDRATE=10, 20, 50,100, 125, 250, 500
CAN1_NODEID=1-127
CAN_ROUTEID=1-127
CAN_ROUTING_CHANNEL=1
für Adressen der PROFIBUS-Slaves:
NET_DPS1_BUSADDRESS= (DP-S in Slot 1)
NET_DPS2_BUSADDRESS= (DP-S in Slot 2)
NET_DPS3_BUSADDRESS= (DP-S in Slot 3)
```

## 7 Systemparameter über Datei Startup.ini vorgeben

### 7.2 Aufbau einer ini-Datei

#### 7.2 Aufbau einer ini-Datei

Eine ini-Datei ist eine Textdatei mit einem festgelegten Datenformat. Ab einer namentlich benannten Sektion wie z. B. [STARTUP] werden die Systemparameter aufgeführt, gefolgt von einem Gleichheitszeichen und deren Wert. Die Zeile wird mit CR/LF (Carriage/Return-Taste) abgeschlossen.

```
COM1_BAUDRATE=38400  
(Carriage/Return)
```

Die mit Semikolon beginnenden Zeilen interpretiert die Steuerung als Kommentar und überspringt sie beim Einlesen:

```
; CAN_NODEID=2
```

Die Parameter können Sie mit einem Text-Editor ändern oder erstellen, wenn Sie die Speicherkarte in den Speicherkarten-Slot des PC stecken.

- Auf der Speicherkarte der XC-CPU201 wird die Datei XCSTARTUP.ini gespeichert im Verzeichnis:  
**disk\_mmc\WOELLER\XC-CPU201-EC256K-8DI-6DO\PROJEKT**
- Auf der Speicherkarte der XC-CPU202 wird die Datei XCSTARTUP.ini gespeichert im Verzeichnis:  
**disk\_mmc\CONTROL\XC-CPU202-EC4M-8DI-6DO-XV\PROJEKT**

#### 7.3 Startup.ini-Datei erstellen

Grundsätzlich arbeitet die Steuerung beim ersten Einschalten (Grundzustand) mit Default-Systemparametern, den STARTUP-Daten, unabhängig davon, ob die Steuerung ein Projekt oder ein Bootprojekt oder keins von beiden enthält! Laden Sie ein Projekt in die Steuerung, die sich im Grundzustand befindet, so arbeitet die Steuerung sofort mit den Systemparametern des Projekts.

Mit dem Browser-Befehl `createstartupini` übertragen Sie aus der Steuerung entweder die STARTUP-Daten oder – wenn diese ein Projekt enthält – die Systemparameter auf die Speicherkarte. Dabei wird die Startup.ini-Datei erstellt, die diese Daten enthält. Voraussetzung: Die Speicherkarte muss gesteckt, formatiert und leer, d. h. ohne Startup.ini-Datei sein.

Eine bereits vorhandene Datei können Sie nicht über das Browser-Kommando `createstartupini` verändern oder überschreiben. Falls Sie das Kommando trotzdem eingegeben, erscheint eine Warnung. Um eine neue Datei zu erstellen, müssen Sie zuerst die bestehende Datei löschen a Abschnitt „Startup.ini-Datei löschen“ auf Seite 79.

#### 7.4 Eintrag der ini-Datei HOST\_NAME

Mit Hilfe des Parameters „Gerätename (HOST\_NAME)“ kann die Steuerung über das Ethernet mit diesem Gerätenamen angesprochen werden. Außerdem kann sie mit ihrer IP-Adresse angesprochen werden. Der Gerätename ist bei der XC-CPU201 ab der Betriebssystemversion 1.04 verfügbar. Er erhält vom System den Eintrag „NoNameSet“. Wird dieser nicht verändert, so kann das an das Ethernet gekoppelte Gerät nur über seine IP-Adresse

## 7 Systemparameter über Datei Startup.ini vorgeben

### 7.4 Eintrag der ini-Datei HOST\_NAME

angesprochen werden. Mit Hilfe des Browser-Befehls `settargetname` tragen Sie einen neuen Gerätenamen ein. Der Gerätename muss für jedes Gerät eindeutig sein. Er kann auch als Kommunikationsparameter eingesetzt werden (in der Abbildung 54: alias), wenn Sie in der Programmiersoftware im Menü `Online → Kommunikationsparameter` die Parameter festlegen. Die Parameter geben die Eigenschaften der Programmierverbindung zwischen PC und Steuerung an.



Abbildung 54: Kommunikationsparameter

Der Gerätename kann mit dem Browser-Befehl `gettargetname` ausgelesen werden.

#### 7.4.1 Einschalten der Steuerung bei gesteckter Speicherkarte mit Datei XCSTARTUP.ini

Beim Einschalten der Steuerung werden die Daten der Startup.ini-Datei der Speicherkarte in die Steuerung übertragen. Diese Systemparameter bleiben auch nach dem Laden eines neuen Programms aktiv.

#### 7.4.2 Parameter ändern

Alle Parameter bleiben solange erhalten, bis Sie den Browser-Befehl `removestartupini` eingeben und dann die Steuerung aus- bzw. einschalten. Die Steuerung arbeitet nun mit den Parametern des Projekts.

#### 7.4.3 Startup.ini-Datei löschen

Es stehen folgende Browser-Befehle zur Verfügung, die auf die Speicherkarte zugreifen.

- `removestartupini`: Löscht grundsätzlich die Systemparameter in der Steuerung. Wenn eine Speicherkarte gesteckt ist, wird auch die ini-Datei auf der Speicherkarte gelöscht. Beim nächsten Einschalten werden die Parameter aus dem Projekt übernommen.
- `removeprojfrommmc`: Löscht das Boot-Projekt und die ini-Datei auf der Speicherkarte. Die Systemparameter in der Steuerung bleiben erhalten.

Das Verhalten der Datei Startup.ini beim Menü-Befehl „Reset Ursprung“, beim Befehl „Werkseinstellung“ im Menü der Steuerung sowie dem Browser-Befehl `factoryset` wird im Abschnitt „Reset“ auf Seite 38 beschrieben. Wenn Sie den Befehl „RESET-Ursprung“ im Online-Betrieb ausführen, werden das Betriebssystem und das Projekt auf dem Systemspeicher `disk_sys` gelöscht. Die Datei XCSTARTUP.ini bleibt erhalten.

## 8 Programmieren über CAN(open)-Netzwerk (Routing)

### 8.1 Voraussetzungen

## 8 Programmieren über CAN(open)-Netzwerk (Routing)

Als „Routing“ bezeichnet man die Möglichkeit, eine Online-Verbindung von einem Programmiergerät (PC) zu einer beliebigen (routingfähigen) Steuerung in einem CAN-Netzwerk aufzubauen, ohne dass das Programmiergerät direkt mit der Zielsteuerung verbunden sein muss. Es kann auch an eine andere Steuerung im Netzwerk angeschlossen werden. Über die Routing-Verbindung können Sie alle Aktionen durchführen, die auch bei einer direkten Online-Verbindung zwischen Programmiergerät und Steuerung zur Verfügung stehen:

- Programm-Download
- Online-Änderungen
- Programmtest (Debugging)
- Erzeugen von Bootprojekten
- Dateien in die Steuerung schreiben
- Dateien aus der Steuerung lesen

Das Routing hat den Vorteil, dass man von einer Steuerung, die mit dem Programmiergerät verbunden ist, Zugriff auf alle routingfähigen Steuerungen am CAN-Bus erhält. Durch die Projektauswahl bestimmen Sie, mit welcher Steuerung Sie kommunizieren möchten. So lassen sich dezentral angeordnete Steuerungen leicht bedienen.

Allerdings ist die Datenübertragung von Routing-Verbindungen deutlich langsamer als bei Direktverbindungen (seriell oder über TCP/IP). Dies macht sich beispielsweise durch langsamere Aktualisierungszeiten von Visualisierungselementen (Variablen) oder langsamere Download-Geschwindigkeiten bemerkbar.

### 8.1 Voraussetzungen

Um das Routing einsetzen zu können, müssen folgende Voraussetzungen erfüllt sein:

- Sowohl die routende Steuerung als auch die Zielsteuerung müssen das Routing unterstützen.
- Beide Steuerungen müssen über den CAN-Bus verbunden sein.
- Die Steuerungen müssen über die gleiche aktive CAN-Baudrate verfügen.
- Auf beiden Steuerungen muss eine gültige Routing-Node-ID eingestellt sein.
- Das Routing mit der XC-CPU201 ist ab der Betriebssystemversion V1.03.02 möglich.

## 8.2 Routingeigenschaften der Steuerung

Die Steuerung unterstützt das Routing über den CAN-Bus.

Das Routing kann ohne vorherigen Download eines Anwenderprogramms ausgeführt werden (Default: 125 kBaud, Node-Id 127). Die Zielsteuerung muss hierfür nicht als CAN-Master bzw. CAN-Device konfiguriert werden.

Sie können beispielsweise ein Programm vom PC über eine Steuerung der XC-Gerätefamilie in die XC200 laden. Ordnen Sie in diesem Fall der XC200 (Zielsteuerung) eine Routing-Node-Id zu.

### 8.2.0.1 Routing über XC200

Falls Sie bei einer Verbindung zwischen XC200 und PC über TCP/IP einen Programmtransfer oder ein Routing ausführen, müssen Sie die Blockgröße der zu übertragenden Daten einstellen. Die Blockgröße (4 kByte oder 128 kByte) ist abhängig von der Art des Transfers (Programmtransfer oder Routing) sowie vom Betriebssystem → Tabelle 14.

Tabelle 14: Blockgröße für die Datenübertragung

	Programm-/ Datei-Transfer		Routing		
	XC-CPU201 BTS-Version < V1.03.02	XC-CPU201 BTS-Version ≧ V1.03.02	XC-CPU201 BTS-Version < V1.03.02	XC-CPU201 BTS-Version ≧ V1.03.02	XC-CPU202 BTS-Version ≧ V1.00.07
Blockgröße Default 128 kByte	128 kByte	128/4 kByte	Routing nicht möglich	4 kByte	4 kByte

BTS = Betriebssystem

#### ACHTUNG

Der Programm-Download mit einer Blockgröße von 4 kByte auf eine XC-CPU201 mit einer Betriebssystemversion < V1.03.02 führt zu einem Fehlverhalten!

Wird ein Programm-Download vorgenommen, ändert sich die Fortschrittsanzeige auf dem Bildschirm des Programmiergeräts nur sprunghaft (ca. alle 10 Sekunden).

Die Einstellung der Blockgröße (Änderung des Wertes in der Registry) wird im Folgenden beschrieben:



Diese Einstellung kann nur mit Administratorrechten durchgeführt werden (Zugriff auf die Registry).

Einstellen der Blockgröße:

- ▶ Schließen Sie alle CODESYS-Anwendungen.
- ▶ Schließen Sie den CODESYS-Gateway-Server.

## 8 Programmieren über CAN(open)-Netzwerk (Routing)

### 8.2 Routingeigenschaften der Steuerung



Abbildung 55: CODESYS-Gateway-Server schließen

- ▶ Ändern Sie die Blockgröße auf den gewünschten Wert.

Zum Eintragen der Blockgröße in die Registry stehen im CODESYS-Installationsverzeichnis folgende \*.reg-Dateien zur Verfügung:

BlockSizeDefault.reg	Trägt eine Blockgröße (Defaultwert) von 20000 <sub>hex</sub> = 128 kByte in die Registry ein.
BlockSizeRout.reg	Trägt eine Blockgröße von 1000 <sub>hex</sub> = 4 kByte in die Registry ein.

Auch über die Anwendung `BlockSizeEditor.exe` lässt sich die Blockgröße anpassen.

Die Download-Blockgröße wird über den folgenden Registry-Eintrag eingestellt:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\3S-Smart Software Solutions GmbH\Gateway Server\Drivers\Standard\Settings\Tcp/Ip (Level 2 Route)]
"BlockSize"=dword:00020000
```

Die Default-Blockgröße beträgt 20000<sub>hex</sub> (= 128 kByte), die Blockgröße für das Routing beträgt 1000<sub>hex</sub> (= 4 kByte).

#### 8.2.1 Hinweise

- Werden größere Dateien in die Zielsteuerung geschrieben bzw. aus der Steuerung gelesen, kann es nach Abschluss des Übertragungsvorgangs zu einer Unterbrechung der Online-Verbindung kommen. Ein erneutes Ankoppeln ist möglich.
- Wird ein Programm mit geänderter Routing-Node-ID über eine routende Steuerung in die Zielsteuerung geladen, übernimmt die Zielsteuerung die geänderte Routing-Node-ID; die Kommunikationsverbindung wird jedoch abgebrochen. Ein Wiederankoppeln mit korrigierter Routing-Node-ID ist möglich.
- Enthält eine Steuerung ein Programm ohne gültige Routing-Parameter (Baudrate/Node-ID), kann auf diese Steuerung nicht über eine Routing-Verbindung zugegriffen werden. Das Löschen der Parameter kann zum Beispiel durch einen RESET URSPRUNG erfolgt sein, wenn der PC mit der Programmiersoftware direkt mit der Zielsteuerung verbunden war. Erfolgt der RESET URSPRUNG über die „routende Steuerung“, bleiben die Parameter erhalten.
- Das Routing ist unabhängig von der Konfiguration (Master/Device): Es ist möglich, auf eine Zielsteuerung zuzugreifen, die weder als Master noch als Device konfiguriert wurde. Sie muss lediglich die Grundparameter, wie Node-Id und Baudrate sowie ein einfaches Programm enthalten.

#### 8.2.2 Adressierung

Steuerungen am CAN-Bus können als Master oder als Slave (Device) konfiguriert werden. Zur eindeutigen Identifikation (bei der Basiskommunikation) erhalten die Steuerungen eine Node-ID/Knotennummer (Adresse). Möchten Sie mit Hilfe der Routing-Funktion auf eine (Ziel-)Steuerung zugreifen, müssen Sie der routenden und der Ziel-Steuerung eine weitere Routing-ID zuordnen. Als Verbindung zwischen PC und XC200 kann die RS232- oder die Ethernet-Schnittstelle genutzt werden.

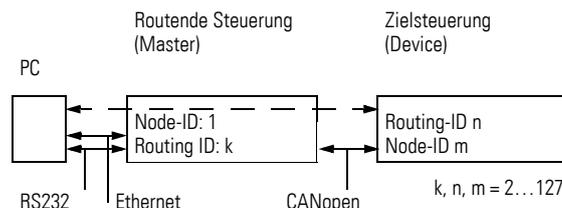


Abbildung 56: Routing über XC..., EC4P, ...

Tabelle 15: Beispiel zur Einstellung der Node-Id, Baudrate

Steuerung	Funktion	Node-ID	Routing-ID	Baudrate	a	Abb.
Routende Steuerung	Master	1	127	125 kB	58	
Zielsteuerung	Device	3	54	125 kB	59	



Für Device-Steuerungen gilt: Die Routing-ID muss **ungleich** der Node-ID (Basiskommunikation) sein!  
 Ausnahme: XC100 mit Betriebssystemversion f V2.0:  
 Hier muss die Routing-ID **gleich** der Node-ID sein!

Die Routing-ID beim Master stellen Sie in der Steuerungskonfigurator im Register „Weitere Parameter“ ein:



Abbildung 57: CAN-Master-Routing-Einstellungen

## 8 Programmieren über CAN(open)-Netzwerk (Routing)

### 8.2 Routingeigenschaften der Steuerung

Die ID zur Basiskommunikation legen Sie im Ordner „CanMaster“ im Register „CAN Parameter“ (Abbildung 58) fest.



Abbildung 58: CAN-Master: Node-ID zur Basiskommunikation

### 8.2.3 Kommunikation mit der Zielsteuerung

- ▶ Verkabeln Sie den PC mit der routenden Steuerung.
- ▶ Wählen Sie das Projekt der Zielsteuerung aus, mit der Sie kommunizieren möchten.
- ▶ Bestimmen Sie zunächst die Kommunikationsparameter für die Verbindung zwischen dem PC und der Steuerung, die mit dem PC verbunden ist.
- ▶ Geben Sie die Target-Id (Target-Id = Node-Id!) der Zielsteuerung (wie im Beispiel) an und melden Sie sich an.

Sie können folgende Funktionen ausführen:

- Programm-Download
- Online-Änderung
- Programmtest (Debugging)
- Bootprojekt erzeugen
- Sourcecode-Ablage

#### Hinweis zur Projekterstellung

Der Zielsteuerung ordnen Sie zwei Node-IDs zu:

- eine ID zur Basiskommunikation
- eine ID zum Routing.

Die Routing-ID (Node-ID, z. B. 54) und die Baudrate der Zielsteuerung (z. B. XC200) legen Sie in der Steuerungskonfiguration im Fenster „Weitere Parameter“ fest → Abbildung 57. Klicken Sie zunächst in das Kästchen „Aktivieren“ im Feld „RS232/TCP → CAN Routing“. Die Aktivierung ist erforderlich, damit die Steuerung über den CAN-Bus kommunizieren kann. Geben Sie dann die Node-ID/Knotennummer und die Baudrate am CAN-Bus in die entsprechenden Eingabefelder ein.

## 8 Programmieren über CAN(open)-Netzwerk (Routing)

### 8.2 Routingeigenschaften der Steuerung



Um eine zügige Datenübertragung sicherzustellen, sollte das Routing mit einer CAN-Baudrate von mindestens 125 kBit/s erfolgen.

Die ID zur Basiskommunikation legen Sie im Ordner „CanDevice“ im Register „CAN-Einstellungen“ fest → Abbildung 59.



Abbildung 59: CAN-Device-Parameter

ID und Baudrate werden beim Projekt-Download in die Steuerung übertragen.

#### Beispiel

Das folgende auf Abbildung 60 basierende Beispiel zeigt den Zugriff auf ein Steuerungsprogramm.

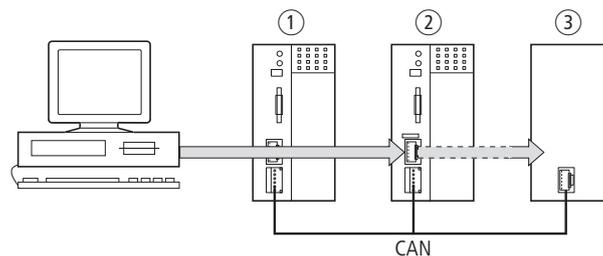


Abbildung 60: Diagnosemöglichkeiten

- XC100 mit Node-ID 1
- XC200 mit Node-ID 2, Routing-ID 127
- Steuerung (z.B. XC..., EC4P...) mit Node-ID 3 und Routing-ID 54

Sie haben den PC an die Steuerung mit der Node-Id „2“ angeschlossen und wollen auf die Zielsteuerung mit der Routing-Id „54“ zugreifen.

- ▶ Öffnen Sie das Projekt der Zielsteuerung, deren Programm Sie bearbeiten oder testen wollen.
- ▶ Parametrieren Sie zunächst die Hardware-Verbindung PC ↔ Steuerung (Node-Id 2).
- ▶ Wählen Sie das Menü <Online → Kommunikationsparameter> an.
- ▶ Klicken Sie unter Kanäle „lokal“ die Taste „Neu“ an.

Es erscheint das Fenster „Neuer Kanal“.

- ▶ Wählen Sie im Fenster „Gerät“ den Kanal aus: Serial [RS232] [Level 2 Route] oder TCP/IP [Level 2 Route].
- ▶ Im Feld „Name“ können Sie einen neuen Namen (z. B. „Rout\_232“) vergeben.
- ▶ Bestätigen Sie mit „OK“. Sie gelangen zum Ausgangsfenster zurück.

## 8 Programmieren über CAN(open)-Netzwerk (Routing)

### 8.2 Routingeigenschaften der Steuerung



Abbildung 61: Kanal parametrieren

Sie haben nun die Parameter für die Hardware-Verbindung zwischen PC und Steuerung (Node-Id 2) festgelegt.

- ▶ Rufen Sie die Kommunikationsparameter im Menü „Online“ nochmals auf und wählen Sie die Steuerung aus, die Sie programmieren bzw. testen möchten.
- ▶ Tragen Sie dazu die Target-ID, im Beispiel die Nummer 54, ein. Die Target-Id ist identisch mit der Routing-Id! Zur Eingabe der Target-ID klicken Sie auf das Feld in der Spalte „Wert“ rechts neben dem Begriff Target-ID. Geben Sie dort die Zahl 54 ein und bestätigen Sie mit „OK“.
- ▶ Melden Sie sich an und führen Sie die Aktion aus.

### 8.2.4 Steuerungskombinationen zum Routing

Folgende Steuerungen unterstützen das Routing:

Von P	XC100, XC121	XC-CPU201 <sup>1)</sup>	EC4P	XC-CPU202
<b>Nach O</b>				
XC100, XC121	x	x	x	x
XC-CPU201 <sup>1)</sup>	x	x	x	x
XC-CPU202	x	x	x	x
EC4P	x	x	x	x

1) XC-CPU201 ab Betriebssystemversion V01.03.01

### 8.2.5 Anzahl der Kommunikationskanäle

In Abhängigkeit von der Steuerung (Kommunikationskanal), die mit dem PC verbunden ist, können Sie mehrere Kommunikationskanäle öffnen, z. B. PC ↔ Steuerung 2, PC ↔ Steuerung 3. So kann die Zustandsanzeige von Steuerung 2 und 3 gleichzeitig durchgeführt werden.

Tabelle 16: Art und Anzahl der Kommunikationskanäle

Kommunikationskanal	Steuerung	Max. Kanalanzahl
TCP/IP Level 2 Route	XC200	5
Serial RS232 Level2 Route	XC.../EC4P	1

## 9 RS232-Schnittstelle im Transparent-Modus

Im Transparent-Modus erfolgt der Datenaustausch zwischen der XC200-Steuerung und Datenendgeräten (z. B. Terminals, Drucker, PCs, Messgeräte) ohne Interpretation der Daten. Dazu ist die serielle Schnittstelle RS232 der CPU oder der XIOC-SER-Module per Anwenderprogramm in den Transparent-Modus zu schalten. Für die RS232-Schnittstelle der XC-CPU201 gilt dies ab Betriebssystemversion 01.03.xx.

	XC-CPU201	XC-CPU202
RS232 der CPU	COM1	LocalCOM
RS232 der XIOC-SER bzw. XIOC-TC1	COM2...5	COM2...5

xSysCom200.lib für XIOC-SER und XIOC-TC1  
SysLibCom.lib für RS232 der CPUs

➔ Befindet sich die RS232-Schnittstelle der XC-CPU201 im Transparent-Modus, ist eine Programmierung über diese Schnittstelle nicht möglich. Sie können jedoch das Programm über die Ethernet-Schnittstelle testen (sogenannter gemeinsamer Betrieb).

➔ Bei der XC-CPU202 ist ein solcher gemeinsamer Betrieb nicht möglich. Per Default ist die RS232 auf den Programmiermodus eingestellt. Mit den Browser-Befehlen **Serial Programming Off** und **Serial Programming On** oder den Bibliotheksfunktionen (FUN) **Disable Com Programming** und **Enable Com programming** kann die RS232-Schnittstelle vom Programmiermodus auf den Transparentbetrieb umgeschaltet werden.

Zeichenformate im Transparentmodus sind: 8E1, 8O1, 8N1, 8N2.

Diese Funktionalität wird bei der XC200 durch die Libraries `xSysCom200.lib` oder `SysLibCom.lib` erbracht. Eine dieser beiden Libraries muss daher in den Bibliotheksverwalter eingebunden sein.

Die Library `SysLibCom.lib` wurde eingeführt (ab Version 01.03.xx), um die Kompatibilität zwischen der XC200 und anderen XControl-Geräten zu gewährleisten.

Die beiden Libraries enthalten Funktionen zum Öffnen und Schließen der Schnittstelle, zum Senden und Empfangen von Daten und zum Setzen der Schnittstellenparameter.

Mit der Funktion `SysComWriteControl` aus der Library `xSysCom200.lib` werden die Kontrollleitungen der RS232 der XIOC-SER-Module angesteuert und mit der Funktion `SysComReadControl` werden sie überwacht.

Die RS232-Schnittstelle der CPU hat im Gegensatz zur RS232-Schnittstelle des XIOC-SER-Moduls keine Kontrollleitungen.

## 9 RS232-Schnittstelle im Transparent-Modus

### 9.1 Programmieren der RS232-Schnittstelle im Transparent-Modus

Die Datentypen der Libraries sind nicht identisch. So ist die Auswahl der Baudrate unterschiedlich:

xSysCom200.lib: 300, ... , 115200

SysLibCom.lib: 4800, ... , 115200

Die RS232-Schnittstelle der CPU wird (im Gegensatz zur Schnittstelle des XIOC-SER-Moduls) über das Betriebssystem angesprochen. Daher kann die Ausführung der Schnittstellen-Funktionen bis zu 50 ms dauern. Die Task, in der die RS232-Schnittstelle angesprochen wird, sollte eine Intervallzeit von mindestens 50 ms besitzen und im Multitaskingbetrieb eine niedrige Priorität (hoher Wert) erhalten, damit sie zeitkritische Tasks nicht verdrängt.

Bei den Funktionen (x)SysComRead/Write werden daher nur Teile der gewünschten Datenlänge abgearbeitet. Zur kompletten Übertragung von Datenblöcken sind wiederholte Aufrufe mit angepassten Offsetwerten in mehreren Task-Intervallen durchzuführen. Die Anzahl der Aufrufe hängt von der Baudrate und der Datenmenge ab.

Die Leistungsfähigkeit der RS232 der CPU ist von der Belastung der SPS (PLCLoad) und von der gewählten Baudrate abhängig. Aufgrund der hohen Intervallzeiten der COM1-Task kann sie von zeitkritischen Tasks verdrängt werden. Bei einem Datenempfang mit hohen Baudraten kann es zu einem Zeichenverlust kommen!

### 9.1 Programmieren der RS232-Schnittstelle im Transparent-Modus

Mit Hilfe des Anwenderprogramms können Sie auf die Daten der RS232-Schnittstelle zugreifen. Dazu stehen Ihnen die Funktionen aus den Libraries xSysCom200.lib oder SysLibCom.lib zur Verfügung. Beachten Sie bitte, dass nur eine der beiden Libraries in den Bibliotheksverwalter eingebunden sein darf! In beiden Libraries finden Sie zahlreiche Funktionen – beispielsweise zum Öffnen und Schließen der Schnittstelle. Die Funktionen der Libraries werden hier nebeneinander gezeigt: Links die Funktionen aus der Library xSysCom200.lib, rechts die aus der Library SysLibCom.lib.

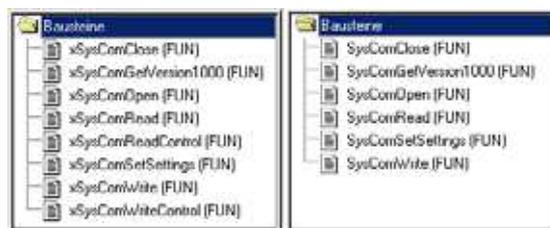


Abbildung 62: Funktionsübersicht (links: xSysCom200.lib, rechts: SysLibCom.lib)

Die Funktionen sind im Handbuch „Funktionsbausteine zur easySoft-CoDe-Sys“ (MN05002002Z-DE) beschrieben.

Siehe auch: Transparentmodus: Textausgabe über RS232 (Beispiel) → Seite 139

## 10 Konfiguration und Parametrierung der Ein-/Ausgänge

### 10.1 Ein-/Ausgänge allgemein

In der Steuerungskonfiguration werden die lokalen Ein-/Ausgänge IX0.0 bis IX0.7, QX0.0 bis QX0.5 und die Ein-/Ausgänge IX1.0 bis IW4 und QX1.0 bis QX1.7 der Zusatzfunktionen, wie zum Beispiel Zähler, angezeigt. Die Ein-/Ausgänge der Zusatzfunktionen werden erst aktiv, nachdem Sie eine Funktion im Register „Weitere Parameter“ ausgewählt haben.

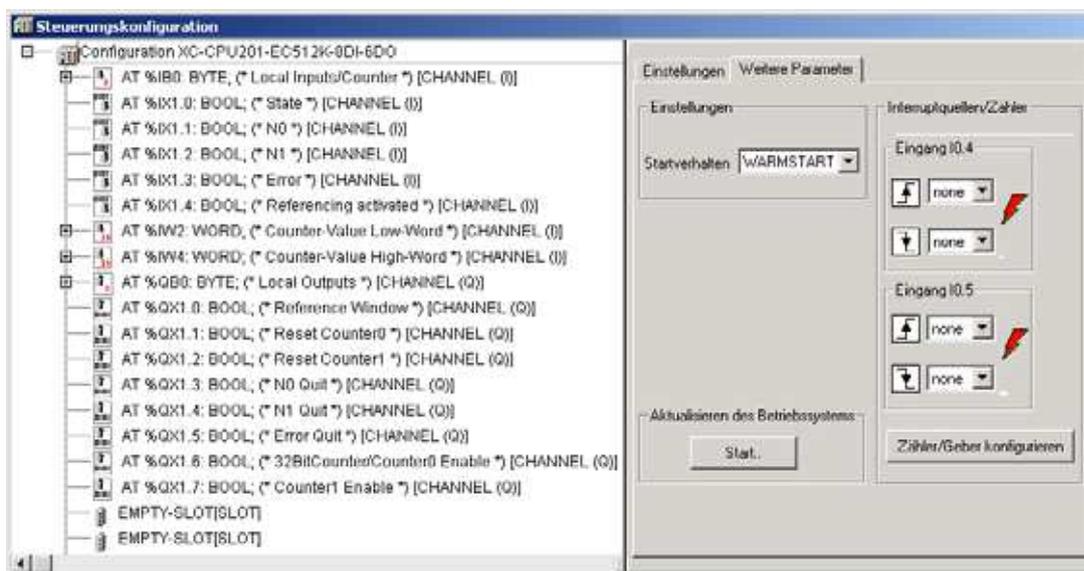


Abbildung 63: Steuerungskonfiguration

Die Processing Unit überträgt Zustände und Ergebnisse auf die virtuellen Eingänge. Die benötigten Ein- und Ausgänge für die Inkrementalgeber-Funktion sind in Abbildung 69 bzw. auf Seite 93 dargestellt, die benötigten Ein- und Ausgänge für die Zähler-Funktionen finden Sie auf Seite 96.

Zur Erweiterung der lokalen Ein- und Ausgänge fügen Sie XIOC-Module ein, indem Sie den Ordner „EMPTY SLOT“ anklicken. Mit dem Befehl „Element ersetzen“ wählen Sie aus der Liste ein Modul aus. Der neue Modulname wird anstelle von EMPTY-SLOT angezeigt.

#### 10.1.1 Lokale Digital-Ein-/Ausgänge

Jede physikalische Veränderung der Module auf den Steckplätzen des Baugruppenträgers (Steckplatzaustausch oder Tausch gegen eine andere Funktionalität) wird von der CPU erkannt, da der Ein-/Ausgangs-Offset verändert wird und damit die Zuweisung der Ein- und Ausgangsparameter zu Fehlzugriffen führt. Wenn Sie bei der Konfiguration für spätere Optimierungen freie Steckplätze reserviert haben und diese freien Steckplätze nachträglich belegt werden, führt auch dies zu einer Ungleichheit und Veränderung des Ein-/Ausgangs-Offsets zwischen Konfiguration und Programm.

## 10 Konfiguration und Parametrierung der Ein-/Ausgänge

### 10.1 Ein-/Ausgänge allgemein

#### ACHTUNG

- Passen Sie bei jeder Veränderung der Konfiguration die Ein- und Ausgänge im Programm an.
- Bei einer Ungleichheit von Konfiguration und Programm oder bei einem konfiguriertem aber nicht vorhandenem Modul erfolgt kein Betriebsartwechsel in den Zustand RUN.

Eine Differenz zwischen der Konfiguration und der physikalischen Existenz bzw. Nichtexistenz von Signalmodulen wird als „FehlerEreignis“ im gepufferten Speicherbereich eingetragen. Der Browser-Befehl `geterrorlist` gibt diesen Fehler als „Allgemeiner IO-Access-Error“ aus. Eine eindeutige Steckplatzzuordnung ist dabei nicht möglich.

Die nachfolgenden Abbildungen zeigen die Veränderungen der Zuweisung der Ein- und Ausgangsparameter beim Vertauschen oder Hinzufügen bzw. Entfernen von Signalmodulen.

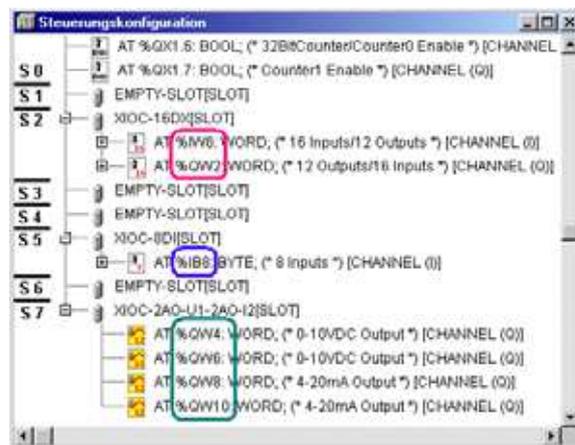


Abbildung 64: Aktuelle Konfiguration

S0, S1, ..., S7 = Steckplatz (Slot)-Nummer im Baugruppenträger

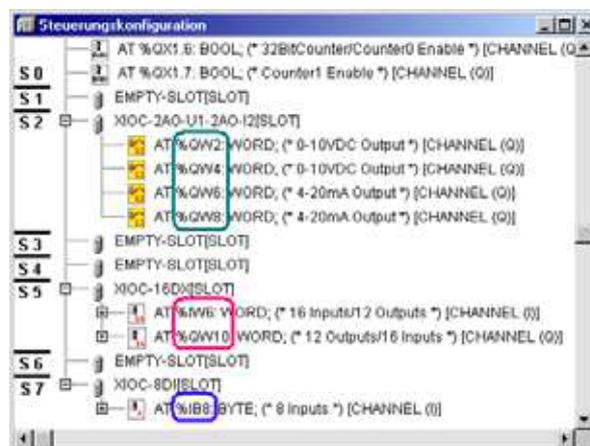


Abbildung 65: Konfigurationsänderung durch Vertauschen der Module

(S0, S1, ..., S7 = Steckplatz(Slot)-Nummer des Baugruppenträgers)

## 10 Konfiguration und Parametrierung der Ein-/Ausgänge

### 10.1 Ein-/Ausgänge allgemein

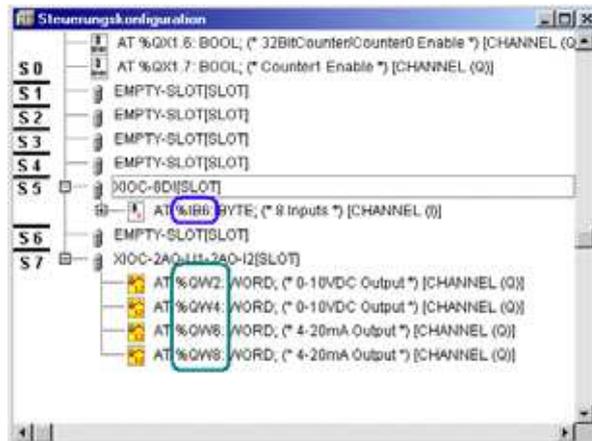


Abbildung 66: Konfigurationsänderung durch Entfernen eines Moduls  
(S0, S1, ..., S7 = Steckplatz(Slot)-Nummer im Baugruppenträger)

Die Abbildungen 64 bis 66 zeigen die Veränderungen der Ein-/Ausgangsparameter der Signalmodule in Abhängigkeit der Steckplätze und sind in Tabelle 17 zusammengefasst.

Tabelle 17: Ein-/Ausgangsparameter bei Änderung der Konfiguration

Abbildung	Steckplatz	Modultyp	Eingangsparameter	Ausgangsparameter
64	2	XIOC-16DX	%IW 6	%QW 2
	5	XIOC-8DI	%IB 8	–
	7	XIOC-2AO-U1-2AO-I2	–	%QW 4 %QW 6 %QW 8 %QW 10
65	2	XIOC-2AO-U1-2AO-I2	–	%QW 2 %QW 4 %QW 6 %QW 8
	5	XIOC-16DX	%IW 6	%QW 10
	7	XIOC-8DI	%IB 8	–
66	2	Steckplatz nicht belegt	–	–
	5	XIOC-8DI	%IB 6	–
	7	XIOC-2AO-U1-2AO-I2	–	%QW 2 %QW 4 %QW 6 %QW 8

## 10.2 Ein-/Ausgänge für Zusatzfunktionen

### 10.2.1 Inkrementalgeber

Die Parametrierung erfolgt in der Steuerungskonfiguration.

- ▶ Aktivieren Sie im Fenster „Steuerungskonfiguration“ die Registerkarte „Weitere Parameter“ und klicken Sie auf die Schaltfläche „Zähler/Geber konfigurieren“.



Abbildung 67: Vorwahl Inkrementalgeber

- ▶ Wählen Sie hier „Inkrementalgeber“ aus. Das Fenster verändert sich zu:



Abbildung 68: Parametrierung „Inkrementalgeber“

- ▶ Ist die Konfiguration vollständig, so betätigen Sie bitte die Schaltfläche „Übernehmen“.

### 10.2.2 Funktionalität der Ein-/Ausgänge

Wenn Sie die Zusatzfunktion „Inkrementalgeber“ ausgewählt haben, erhalten die Eingänge I0.0 bis I0.3 eine neue Funktion. Die Eingänge I0.4 bis I0.7 behalten ihre Standardfunktion. Die Funktionen der virtuellen Ein- und Ausgänge für den Inkrementalgeber gehen aus der folgenden Abbildungen hervor.

## 10 Konfiguration und Parametrierung der Ein-/Ausgänge

### 10.2 Ein-/Ausgänge für Zusatzfunktionen

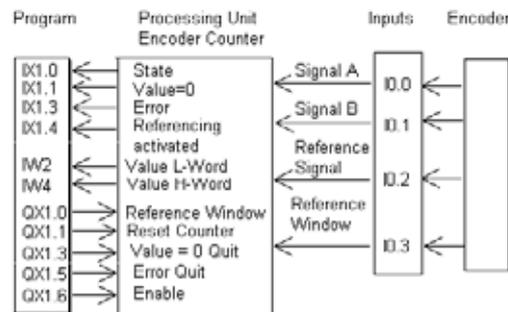


Abbildung 69: Ein-/Ausgänge für Inkrementalgeber

### Referenzierung

In vielen Positioniersteuerung wird zum Start ein Referenzpunkt angefahren. Der Referenziervorgang kann über die Hardware (Reference Window-Signal vom Geber auf I0.3) oder über die Software (QX1.0) gesteuert werden. Die Auswahl treffen Sie in der Steuerkonfiguration. Hat eines der beiden Signale H-Pegel, wird dies am Eingang IX1.4 angezeigt. Erfolgt in diesem Zustand ein Impuls an I0.2 (Referenzsignal vom Geber), so wird der Zählerstand auf den Referenzwert gesetzt, den Sie in der Steuerungskonfiguration angegeben haben.



Wählen Sie das Referenzfenster so groß, dass das Referenzsignal nur einmal ansteht, aber dennoch zuverlässig ausgewertet werden kann.

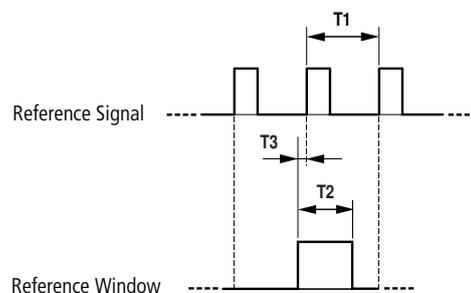


Abbildung 70: Beziehung zwischen Referenzsignal und Referenzfenster

T1 Impuls-Wiederholzeit zweier aufeinander folgender Referenzsignale bei einer Umdrehung des Inkremental-Drehgebers

T2 Maximal zulässige Zeitdauer des Referenzfensters. Muss hinreichend kleiner sein als T1, so dass kein zweiter Referenzimpuls erkannt wird.

T3 Muss so lange sein, dass die L/H-Flanke des Referenzimpulses sicher erkannt werden kann.

T2 und T3 sind abhängig von der Impulsfolgefrequenz des Referenzimpulses und applikationsabhängig eventuell experimentell zu ermitteln.

## 10 Konfiguration und Parametrierung der Ein-/Ausgänge

### 10.2 Ein-/Ausgänge für Zusatzfunktionen

#### 10.2.3 Darstellung der Ein-/Ausgänge des Inkrementalgebers

##### Reale Eingänge

AT %IX0.0: BOOL; (*Bit0*)	Signal A
AT %IX0.1: BOOL; (*Bit1*)	Signal B
AT %IX0.2: BOOL; (*Bit2*)	Referenz-Signal
AT %IX0.3: BOOL; (*Bit3*)	Referenzieren freigeben <sup>1)</sup>
AT %IX0.4: BOOL; (*Bit4*)	lokaler Eingang
AT %IX0.5: BOOL; (*Bit5*)	lokaler Eingang
AT %IX0.6: BOOL; (*Bit6*)	lokaler Eingang
AT %IX0.7: BOOL; (*Bit7*)	lokaler Eingang

##### Darstellung der virtuellen Ein-/Ausgänge in der Steuerungskonfiguration

AT %IX1.0: BOOL; (*State*) [CHANNEL(I)]	H = Referenzieren durchgeführt
AT %IX1.1: BOOL; (*N0*) [CHANNEL(I)]	L = kein Nulldurchgang; H = Nulldurchgang des Zählerstandes
AT %IX1.2: BOOL; (*N1*) [CHANNEL(I)]	
AT %IX1.3: BOOL; (*Error*) [CHANNEL(I)]	L = kein Fehler H = interner Fehler (Flanken von Signal A und B gleichzeitig)
AT %IX1.4: BOOL; (*Referencing activated*)	H = Referenzieren wurde freigeben
AT %IW2: WORD; (*Counter-Value Low-Word*) [CHANNEL(I)]	Zählerstand Low-Word
AT %IW4: WORD; (*Counter-Value High-Word*) [CHANNEL(I)]	Zählerstand High-Word
AT %QX1.0: BOOL; (*Reference Window*) [CHANNEL(Q)]	Referenzieren freigeben <sup>2)</sup>
AT %QX1.1: BOOL; (*Reset Counter0*) [CHANNEL(Q)]	Reset auf Referenzwert
AT %QX1.2: BOOL; (*Reset Counter1*) [CHANNEL(Q)]	
AT %QX1.3: BOOL; (*N0 Quit*) [CHANNEL(Q)]	Quittierung Null-Durchgang
AT %QX1.4: BOOL; (*N1 Quit*) [CHANNEL(Q)]	
AT %QX1.5: BOOL; (*Error Quit*) [CHANNEL(Q)]	Fehler-Quittierung
AT %QX1.6: BOOL; (*32BitCounter/Counter0 Enable*) [CHANNEL(Q)]	L = Eingangsimpulse sperren H = Eingangsimpulse freigeben
AT %QX1.7: BOOL; (*Counter1 Enable*) [CHANNEL(Q)]	

1) Voraussetzung: Im Konfigurator wurde die Konfigurierungsart „Hardware“ ausgewählt.

2) Voraussetzung: Im Konfigurator wurde die Konfigurierungsart „Software“ ausgewählt.

### 10.2.4 Zähler

Zur Erfassung der Zählimpulse wählen Sie zwischen den folgenden Funktionen aus:

- 1 x 32 Bit Vor/Rück-Zähler oder
- 2 x 16 Bit Vor/Rück-Zähler.

Die Parametrierung erfolgt in der Steuerungskonfiguration:

- ▶ Aktivieren Sie im Fenster „Steuerungskonfiguration“ die Registerkarte „Weitere Parameter“ und klicken Sie auf die Schaltfläche „Zähler/Geber konfigurieren“.
- ▶ Wählen Sie die Eingangsauswahl „1 x 32 Bit Vor/Rück-Zähler oder 2 x 16 Bit Vor/Rück-Zähler“ an und klicken Sie auf die Schaltfläche „Übernehmen“.

Es öffnet sich ein weiteres Fenster zur Konfiguration.

- ▶ Geben Sie hier die „Interruptquelle“ und den „Sollwert“ an.



Abbildung 71: Parametrierung Zählereingang 1 x 32 Bit

- ▶ Ist die Konfiguration vollständig, so betätigen Sie bitte die Schaltfläche „Übernehmen“.

Siehe auch:

- Interruptverarbeitung → Seite 97

## 10 Konfiguration und Parametrierung der Ein-/Ausgänge

### 10.2 Ein-/Ausgänge für Zusatzfunktionen

#### 10.2.5 Darstellung der Ein-/Ausgänge des 32-Bit-Zählers

<b>Reale Eingänge</b>	
AT %IX0.0: BOOL; (*Bit0*)	Eingang für Zählimpuls
AT %IX0.1: BOOL; (*Bit1*)	Eingang für Richtungssignal
<b>Darstellung der virtuellen E/As im Steuerungskonfigurator</b>	
AT %IX1.0: BOOL; (*State*) [CHANNEL(I)]	
AT %IX1.1: BOOL; (*N0*) [CHANNEL(I)]	L = kein Nulldurchgang, H = Nulldurchgang
AT %IX1.2: BOOL; (*N1*) [CHANNEL(I)]	
AT %IX1.3: BOOL; (*Error*) [CHANNEL(I)]	H = Fehler
AT %IW2: WORD; (*Counter-Value Low-Word*) [CHANNEL(I)]	Zählerstand Low-Word
AT %IW4: WORD; (*Counter-Value High-Word*) [CHANNEL(I)]	Zählerstand High-Word
AT %QX1.0: BOOL; (*Reference Window*) [CHANNEL(Q)]	
AT %QX1.1: BOOL; (*Reset Counter0*) [CHANNEL(Q)]	Reset auf Null
AT %QX1.2: BOOL; (*Reset Counter1*) [CHANNEL(Q)]	
AT %QX1.3: BOOL; (*N0 Quit*) [CHANNEL(Q)]	Quittierung Null-Durchgang
AT %QX1.4: BOOL; (*N1 Quit*) [CHANNEL(Q)]	
AT %QX1.5: BOOL; (*Error Quit*) [CHANNEL(Q)]	Fehler-Quittierung
AT %QX1.6: BOOL; (*32BitCounter/Counter0 Enable*) [CHANNEL(Q)]	L = Zählimpuls sperren, H = Zählimpuls freigeben
AT %QX1.7: BOOL; (*Counter1 Enable*) [CHANNEL(Q)]	

#### 10.2.6 Darstellung der Ein-/Ausgänge der zwei 16-Bit-Zähler

<b>Reale Eingänge</b>	
AT %IX0.0: BOOL; (*Bit0*)	Eingang für Zählimpuls (Zähler 0)
AT %IX0.1: BOOL; (*Bit1*)	Eingang für Richtungssignal (Zähler 0)
AT %IX0.2: BOOL; (*Bit2*)	Eingang für Zählimpuls (Zähler 1)
AT %IX0.3: BOOL; (*Bit3*)	Eingang für Richtungssignal (Zähler 1)
<b>Darstellung der virtuellen E/As im Steuerungskonfigurator</b>	
AT %IX1.0: BOOL; (*State*) [CHANNEL(I)]	
AT %IX1.1: BOOL; (*N0*) [CHANNEL(I)]	L = kein Nulldurchgang, H = Nulldurchgang
AT %IX1.2: BOOL; (*N1*) [CHANNEL(I)]	L = kein Nulldurchgang, H = Nulldurchgang
AT %IX1.3: BOOL; (*Error*) [CHANNEL(I)]	H = Fehler
AT %IW2: WORD; (*Counter-Value Low-Word*) [CHANNEL(I)]	Zählerstand Zähler 0
AT %IW4: WORD; (*Counter-Value High-Word*) [CHANNEL(I)]	Zählerstand Zähler 1
AT %QX1.0: BOOL; (*Reference Window*) [CHANNEL(Q)]	
AT %QX1.1: BOOL; (*Reset Counter0*) [CHANNEL(Q)]	Reset auf Null Zähler 0
AT %QX1.2: BOOL; (*Reset Counter1*) [CHANNEL(Q)]	Reset auf Null Zähler 1
AT %QX1.3: BOOL; (*N0 Quit*) [CHANNEL(Q)]	Quittierung Null für Zähler 0
AT %QX1.4: BOOL; (*N1 Quit*) [CHANNEL(Q)]	Quittierung Null für Zähler 1
AT %QX1.5: BOOL; (*Error Quit*) [CHANNEL(Q)]	Fehler-Quittierung
AT %QX1.6: BOOL; (*Counter0 Enable*) [CHANNEL(Q)]	0: L = Zählimpuls sperren, H = Zählimpuls freigeben
AT %QX1.7: BOOL; (*Counter1 Enable*) [CHANNEL(Q)]	1: L = Zählimpuls sperren, H = Zählimpuls freigeben

## 10.3 Interruptverarbeitung

Beim Auftreten eines Interrupts führt das Betriebssystem die mit der Interruptquelle verknüpfte Programm-Organisations-Einheit (POU = Program Organisation Unit) aus.



### VORSICHT

Die Ausführung der Interrupt-POU wird zeitlich **nicht** überwacht. Versehentlich programmierte Endlosschleifen werden nicht mehr verlassen!

Es werden maximal sechs Interruptquellen (IO-Interrupt1, ..., IO-Interrupt6) unterstützt, die sich durch die Zahl am Ende des Namens unterscheiden.

Interrupt-Erzeuger sind:

- Eingang I0.4 L → H-Flanke
- Eingang I0.4 H → L-Flanke
- Eingang I0.5 L → H-Flanke
- Eingang I0.5 H → L-Flanke
- 32 Bit Zähler, Istwert = Sollwert oder
- 16 Bit Zähler (1), Istwert = Sollwert
- 16 Bit Zähler (2), Istwert = Sollwert

Die vom Interrupt aufgerufene POU wird immer komplett abgearbeitet und kann nicht durch einen neuen Interrupt unterbrochen werden. Ein neuer Interrupt wird erst nach Beendigung des aktuellen ausgeführt.

### ACHTUNG

Alle bis zu diesem Zeitpunkt angesteuerten Ausgänge (H-Signal) bleiben aktiv und werden nicht abgeschaltet.

Die Interrupts werden im Zustand RUN der CPU freigegeben und im STOP gesperrt. In der Konfiguration nicht freigegebene Interruptquellen lösen keinen Interrupt aus. Ist einer freigegebenen Interruptquelle keine POU zugewiesen, wird der Interrupt erkannt und ausgeführt, ohne eine POU abzuarbeiten.

Zu häufiges Auftreten von Interrupts während eines Programmdurchlaufs kann zum Überschreiten der programmierten Taskzeit führen und einen RESET durch den Watchdog auslösen.

Die Interrupts können aus dem Programm heraus gesperrt und wieder freigegeben werden. Hierzu existieren die Funktionen **Disableinterrupt** und **Enableinterrupt**. Ein Aufrufparameter in der Software CODESYS bestimmt, ob ein einzelner Interrupt oder alle Interrupts gesperrt/freigegeben werden. Die Freigabe eines gesperrten Interrupts muss mit dem gleichen Parameter erfolgen wie die Sperrung.

Die beiden Funktionen **Disableinterrupt** und **Enableinterrupt** sind Bestandteil der Library XC200\_Util.lib. Diese Library müssen Sie – falls noch nicht vorhanden – in den Bibliotheksverwalter der CODESYS einbinden.

## 10 Konfiguration und Parametrierung der Ein-/Ausgänge

### 10.3 Interruptverarbeitung

#### 10.3.0.1 DisableInterrupt

Mit dieser Funktion schalten Sie einen parametrisierten physikalischen Interrupt aus dem Anwenderprogramm heraus inaktiv.



Abbildung 72: Funktion „DisableInterrupt“

#### 10.3.1 EnableInterrupt

Mit dieser Funktion geben Sie den zuvor inaktiv geschalteten physikalischen Interrupt wieder als aktiven Interrupt frei.



Abbildung 73: Funktion „EnableInterrupt“

#### 10.3.2 Parametrierung

Parametrierung und Priorisierung der Interrupts erfolgen in der „Steuerungs- und Taskkonfiguration“ der CODESYS (Register „Ressourcen“ aktivieren und Ordner „Taskkonfiguration I Systemereignisse“ aufrufen). Hier können Sie jedem Interrupt eine POU zuweisen.

### 10.3.3 Beispiel zur Interruptverarbeitung

Eine Task „Basic“ enthält eine POU „PLC\_PRG“. Eine weitere POU „Fastprog“ soll abgearbeitet werden, wenn eine L → H-Flanke am Eingang I0.5 einen Interrupt erzeugt.

- ▶ Erstellen Sie die POU „PLC\_PRG“ und „Fastprog“ gemäß Abbildung 74.

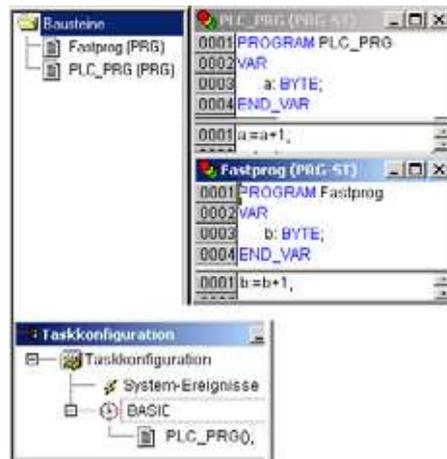


Abbildung 74: Steuerungs- und Task-Konfiguration

- ▶ Wechseln Sie in die Steuerungskonfiguration und weisen Sie dem Eingang I0.5 (L → H-Flanke) z. B. die Interruptquelle „IO-Interrupt3“ aus dem Drop-Down-Menü zu.



Abbildung 75: Zuweisung I0.5 → Interruptquelle

- ▶ Wechseln Sie in die Taskkonfiguration und öffnen Sie den Ordner „System-Ereignisse“.

## 10 Konfiguration und Parametrierung der Ein-/Ausgänge

### 10.3 Interruptverarbeitung

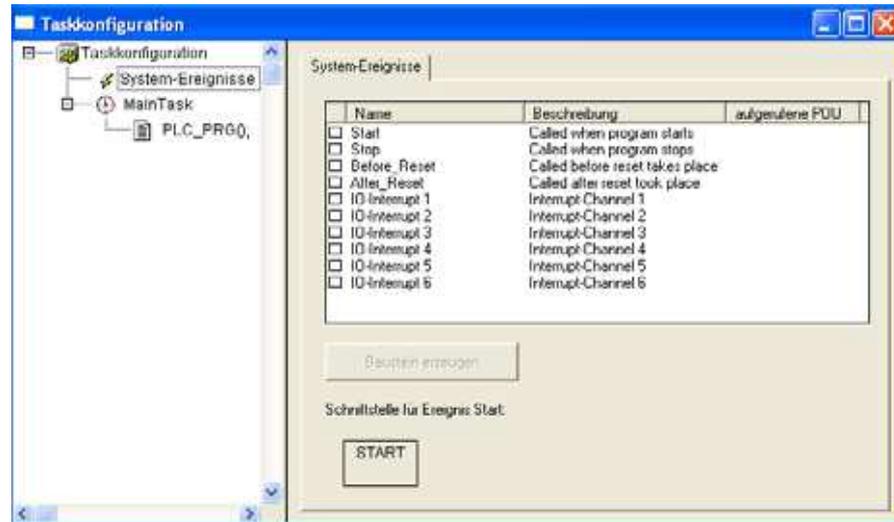


Abbildung 76: System-Ereignisse

- ▶ Geben Sie den IO-Interrupt 3 frei, indem Sie auf das Kästchen links neben dem Namen „IO-Interrupt 3“ klicken. Die Bestätigung wird durch einen Haken dargestellt.
- ▶ Markieren Sie den Bereich von Spalte „ aufgerufene POU“ und Zeile „IO-Interrupt 3“.
- ▶ Stellen Sie den Cursor auf den markierten Bereich und betätigen Sie die Funktionstaste F2.



Abbildung 77: Zuweisung Interruptquelle → POU

Das Fenster „Eingabehilfe“ wird geöffnet, indem alle vordefinierten Programme gelistet sind.

- ▶ Wählen Sie die POU „Fastprog“ aus und bestätigen Sie mit „OK“.
- ▶ Speichern Sie das Projekt. Sie können es nun testen.

Mit jeder steigenden Flanke am Eingang IO.5 wird die Variable „b“ um eins erhöht.

## 11 Bibliotheken, Funktionsbausteine und Funktionen

Die Bibliotheken enthalten IEC-Funktionsbausteine und Funktionen, die Sie beispielsweise für folgende Aufgaben anwenden können:

- Datenaustausch über den CAN-Bus
- Bedienung der Echtzeituhr
- Ermittlung der Busbelastung des CAN-Bus
- Interrupt ausführen
- Daten über die Schnittstellen senden/empfangen

Die Bibliotheken finden Sie in den folgenden Verzeichnissen:

- **Lib\_Common** – für alle Steuerungen
- **Lib\_CPU100** u. **Lib\_CPU200** – für die Steuerungen XC100 bzw. XC200
- **Lib\_XN\_PLC\_CANopen** – für die Steuerung XN-PLC

### 11.1 Bibliotheken handhaben

Beim Öffnen eines Projekts werden die beiden Bibliotheken Standard.lib und SYSLIBCALLBACK.lib in den Bibliotheksverwalter kopiert. Sind für die Anwendung weitere Bibliotheken notwendig, müssen Sie diese nachinstallieren.

Die im Bibliotheksverwalter stehenden Bibliotheken werden nach dem Speichern dem Projekt zugeordnet. Wenn Sie das Projekt wieder öffnen, werden auch diese Bibliotheken aufgerufen.

Die folgende Übersicht listet die Dokumente auf, in denen die Funktionsbausteine und Funktionen beschrieben sind.

Dokument	Bibliothek
AWB 2700-1437	Standard.lib Util.lib
MN05003004Z-DE (frühere Bezeichnung AWB 2724-1453)	XC100_Util.lib
MN05003001Z-DE (frühere Bezeichnung AWB 2724-1491)	XC200_Util.lib
AWB 2724-1566	XN_PLC_Util.lib
Online-Hilfe oder PDF-Dateien	SysLib...lib
MN05010002Z-DE (frühere Bezeichnung AWB 2786-1456)	XS40_MoellerFB.lib/ Visu.lib/...
AN2700K20	3S_CanOpenDevice.lib 3S_CanOpenManager.lib
AN2700K19	3S_CANopenNetVar.lib
AN2700K27	XC_SysLibCan.lib XN_PLC_SysLibCan.lib
MN05010001Z-DE (frühere Bezeichnung AWB 2786-1554)	CANUser.lib CANUser_Master.lib

## 11 Bibliotheken, Funktionsbausteine und Funktionen

### 11.2 XC200-spezifische Funktionen

#### 11.1.1 Weitere Systembibliotheken installieren

Zum Nachinstallieren von Bibliotheken sind folgende Schritte erforderlich:



Abbildung 78: Bibliotheken nachinstallieren

- ▶ Klicken Sie im angelegten Projekt auf das Register „Ressourcen“.
- ▶ Klicken Sie das Verzeichnis „Bibliotheksverwalter“ mit der linken Maustaste doppelt an.
- ▶ Klicken Sie auf den Menüpunkt «Einfügen → Weitere Bibliothek... Einfg».

Im neuen Fenster werden die zur Verfügung stehenden – vom Zielsystem abhängigen – Bibliotheken dargestellt.

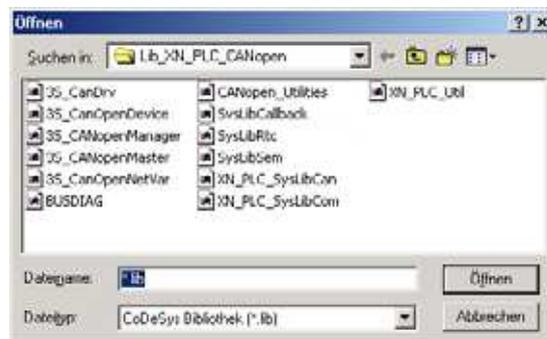


Abbildung 79: Bibliothek auswählen

- ▶ Wählen Sie die nachzuinstallierende Bibliothek aus und klicken Sie auf die Schaltfläche „Öffnen“.

Die Bibliothek erscheint danach im Bibliotheksverwalter.

### 11.2 XC200-spezifische Funktionen

Die XC200-spezifischen Funktionen sind in der Bibliothek XC200\_Util.lib enthalten. Ab Betriebssystemversion V01.03.xx der XC-CPU201 wurde die Bibliothek XC200\_Util2.lib mit zusätzlichen Funktionen eingeführt. Diese zusätzlichen Funktionen werden ab Seite 108 beschrieben.

Die Funktionen der Bibliothek XC200\_Util.lib teilen sich in folgende Gruppen auf:

- CAN-Funktionen (CAN\_Uilities)
- Event-Funktionen (EVENT)
- XIOC-Funktionen (XIOC)

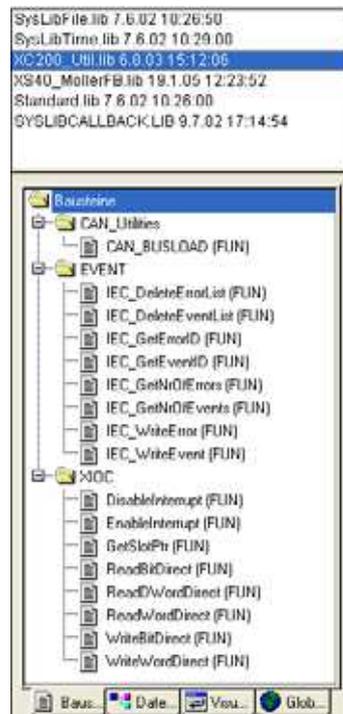


Abbildung 80: XC200-spezifische Funktionen der Bibliothek XC200\_Util.lib

### 11.2.1 CAN\_Uilities

Die Funktion CAN\_BUSLOAD ist in der Bibliothek XC200\_Util.lib im Ordner „CAN\_Uilities“ enthalten.

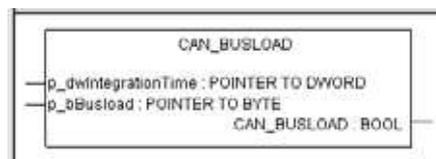


Abbildung 81: Funktion „CAN\_BUSLOAD“

Die Funktion kann in einem Anwenderprogramm zyklisch aufgerufen werden. Wenn ein Lesezyklus erfolgreich beendet wurde, liefert die Funktion den Wert TRUE zurück und schreibt die ermittelten Werte für die Integrationszeit und Busauslastung auf die übergebenen Adressen.

Ist die Berechnung der Buslast noch nicht beendet oder ist der CAN-Controller nicht initialisiert, liefert die Funktion als Rückgabewert FALSE.

Ein Lesezyklus beträgt 500 ms.

Siehe auch: Auslastung des CAN-Busses anzeigen (canload) → Seite 133

## 11 Bibliotheken, Funktionsbausteine und Funktionen

### 11.2 XC200-spezifische Funktionen

#### 11.2.2 Event-Funktionen

Events sind besondere Ereignisse aus dem Betriebssystem oder der Applikation. Diese Ereignisse werden in einem Ringspeicher gespeichert. Die nachstehenden Funktionen erlauben den Lese- und Schreibzugriff auf diesen Event-(Ring-)Speicher.

##### 11.2.2.1 IEC\_DeleteErrorList

Diese Funktion löscht alle in der Error-Liste eingetragenen Fehlermeldungen.

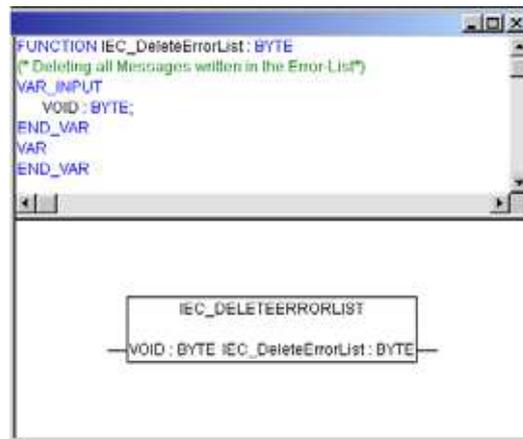


Abbildung 82: Funktion „IEC\_DeleteErrorList“ mit Deklarationsteil

##### 11.2.2.2 IEC\_DeleteEventList

Diese Funktion löscht alle in der Event-Liste eingetragenen Fehlermeldungen.

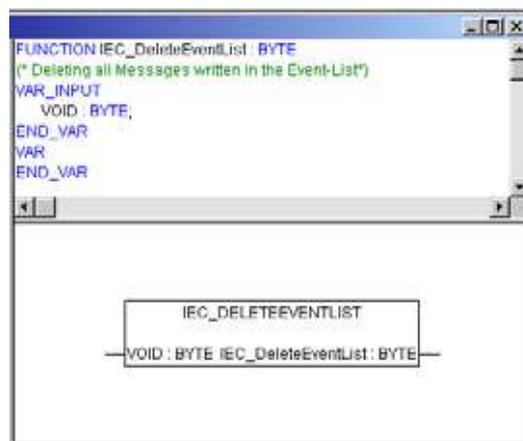


Abbildung 83: Funktion „IEC\_DeleteEventList“ mit Deklarationsteil

### 11.2.2.3 IEC\_GetErrorID

Diese Funktion gibt die Modul-ID und Error-ID der angeforderten Fehlermeldung zurück.

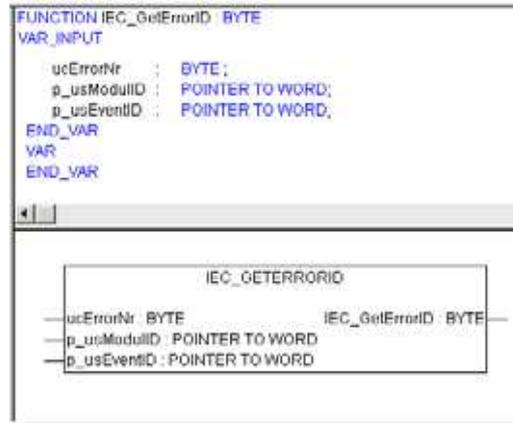


Abbildung 84: Funktion „IEC\_GetErrorID“ mit Deklarationsteil

Die Beschreibung der Fehlermeldungen und die Fehlernummer finden Sie in der Online-Dokumentation der CODESYS-Software zur Funktion IEC\_GetErrorID.

### 11.2.2.4 IEC\_GetEventID

Diese Funktion gibt die Modul-ID und Event-ID der angeforderten Ereignismeldung zurück.

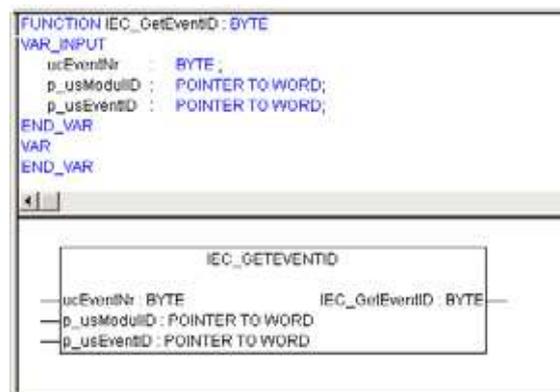


Abbildung 85: Funktion „IEC\_GetEventID“ mit Deklarationsteil

Die Beschreibung der Ereignismeldungen und die Ereignisnummer finden Sie in der Online-Dokumentation der CODESYS-Software zur Funktion IEC\_GetEventID.

### 11.2.2.5 IEC\_GetNrOfErrors

Diese Funktion gibt die Anzahl der eingetragenen Fehlermeldungen zurück.

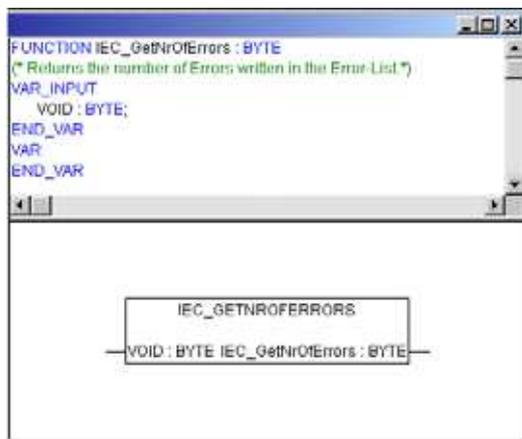


Abbildung 86: Funktion „IEC\_GetNrOfErrors“

### 11.2.2.6 IEC\_GetNrOfEvents

Diese Funktion gibt die Anzahl der derzeit eingetragenen Ereignismeldungen zurück.

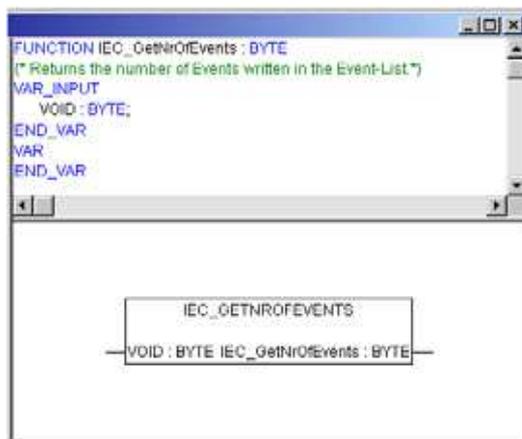


Abbildung 87: Funktion „IEC\_GetNrOfEvents“

### 11.2.2.7 IEC\_WriteError

Diese Funktion schreibt eine Fehlermeldung in die Fehlerliste der Steuerung.



Abbildung 88: Funktion „IEC\_WriteError“

### 11.2.2.8 IEC\_WriteEvent

Diese Funktion schreibt eine Ereignismeldung in die Ereignisliste der Steuerung.



Abbildung 89: Funktion „IEC\_WriteEvent“

## 11.2.3 XIOC-Funktionen

Die XIOC-Funktionen beinhalten Funktionen zur Interrupt-Verarbeitung (→ Seite 97) und zur Programmierung des direkten Peripheriezugriffs (→ Seite 60).

## 11 Bibliotheken, Funktionsbausteine und Funktionen

### 11.2 XC200-spezifische Funktionen

#### 11.2.4 Zusätzliche Funktionen der Library XC200\_Util2.lib für die XC-CPU201

Die Funktionen der Bibliothek XC200\_Util2.lib gehen aus der folgenden Übersicht hervor:

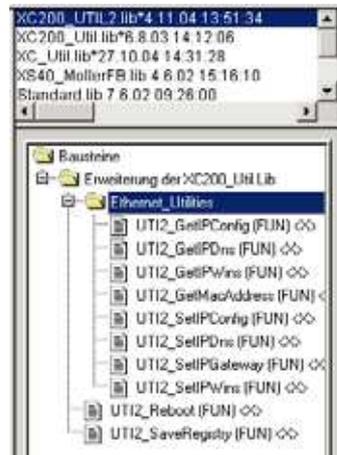


Abbildung 90: Übersicht der XC200\_Util2.lib für die XC-CPU201

#### 11.2.4.1 Ethernet\_Utilities

##### UTI2\_GetIPConfig

Ausgabe der IP-, Subnetmask- und IPGateway-Adresse

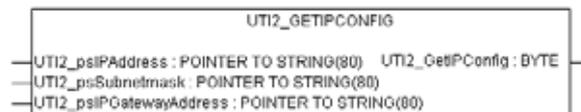


Abbildung 91: UTI2\_GetIPConfig

Tabelle 18: Eingangsvariablen für UTI2\_GetIPConfig

Eingangsvariable	Bedeutung
UTI2_psIPAddress	Zeiger auf einen String, in den die ausgelesene IP-Adresse eingeschrieben wird
UTI2_psSubnetmask	Zeiger auf einen String, in den die ausgelesene Subnetzmaske eingeschrieben wird
UTI2_psIPGatewayAddress	Zeiger auf einen String, in den die ausgelesene Adresse des Standard-Gateways eingeschrieben wird

Tabelle 19: Rückgabewerte für UTI2\_GetIPConfig

Rückgabewert	Bedeutung
1	Auslesen erfolgreich
< 0	Auslesen fehlgeschlagen (allgemeiner Fehler)
-4	Kein gültiger Zeiger übergeben

### UT12\_GetIPDns

Ausgabe der IP-Adresse des momentan in der „Registry“ eingetragenen DNS-Servers

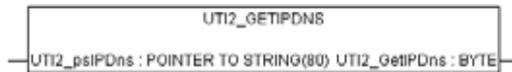


Abbildung 92: UT12\_GetIPDns

Tabelle 20: Eingangsvariablen für UT12\_GetIPDns

Eingangsvariable	Bedeutung
UT12_psIPDns	Zeiger auf einen String, in den die ausgelesene IP-Adresse eingeschrieben wird

Tabelle 21: Rückgabewerte für UT12\_GetIPDns

Rückgabewert	Bedeutung
1	Auslesen erfolgreich
< 0	Auslesen fehlgeschlagen
-4	Kein gültiger Zeiger übergeben

### UT12\_GetIPWins

Ausgabe der IP-Adresse des momentan in der „Registry“ eingetragenen WINS-Servers

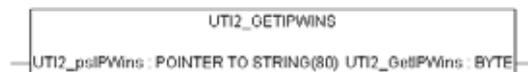


Abbildung 93: UT12\_GetIPWins

Tabelle 22: Eingangsvariablen für UT12\_GetIPWins

Eingangsvariable	Bedeutung
UT12_psIPWins	Zeiger auf einen String, in den die ausgelesene IP-Adresse hineingeschrieben wird

Tabelle 23: Rückgabewerte für UT12\_GetIPWins

Rückgabewert	Bedeutung
1	Auslesen erfolgreich
< 0	Auslesen fehlgeschlagen
-4	Kein gültiger Zeiger übergeben

## 11 Bibliotheken, Funktionsbausteine und Funktionen

### 11.2 XC200-spezifische Funktionen

#### UT12\_GetMacAddress

Ausgabe der MAC-Adresse (MAC = Media Access Control)

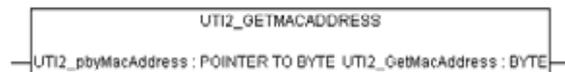


Abbildung 94: UT12\_GetMacAddress

Tabelle 24: Eingangsvariablen für UT12\_GetMacAddress

Eingangsvariable	Bedeutung
UT12_pbyMacAddress	Zeiger auf ein Array von 5 Byte-Werten, in das die ausgelesene MAC-Adresse eingetragen wird

Tabelle 25: Rückgabewerte für UT12\_GetMacAddress

Rückgabewert	Bedeutung
1	Auslesen erfolgreich
< 0	Auslesen fehlgeschlagen (allgemeiner Fehler)
-4	Kein gültiger Zeiger übergeben

#### UT12\_SetIPConfig

Setzen der IP- und Subnetmask-Adresse



Abbildung 95: UT12\_SetIPConfig

#### **ACHTUNG**

Ein neu eingetragener Wert muss durch einen „SaveRegistry“- oder einen „Reboot“-Befehl nichtflüchtig gespeichert werden. Der neu eingetragene Wert wird erst nach einem Neustart der Steuerung übernommen.

Tabelle 26: Eingangsvariablen für UT12\_SetIPConfig

Eingangsvariable	Bedeutung
UT12_psIPAddress	Zeiger auf eine String-Variablen, die die zu schreibende IP-Adresse enthält
UT12_psSubnetmask	Zeiger auf eine String-Variablen, die den einzutragenden Wert für die Subnetzmaske enthält

Tabelle 27: Rückgabewerte für UTI2\_SetIPConfig

Rückgabewert	Bedeutung
1	Schreiben erfolgreich
< 0	Schreiben fehlgeschlagen (allgemeiner Fehler)
-4	Kein gültiger Zeiger übergeben

### UTI2\_SetIPDns

Setzen der IP-Adresse eines DNS-Servers in der Registry  
(muss gegebenenfalls mit UTI2\_SaveRegistry gesichert werden)

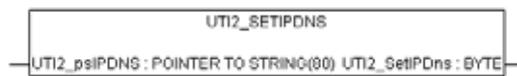


Abbildung 96: UTI2\_SetIPDns

Tabelle 28: Eingangsvariablen für UTI2\_SetIPDns

Eingangsvariable	Bedeutung
UTI2_psIPDNS	Zeiger auf eine String-Variable, die die zu schreibende IP-Adresse enthält

Tabelle 29: Rückgabewerte für UTI2\_SetIPDns

Rückgabewert	Bedeutung
1	Schreiben erfolgreich
< 0	Schreiben fehlgeschlagen
-4	Kein gültiger Zeiger übergeben

### UTI2\_SetIPGateway

Setzen der IPGateway-Adresse

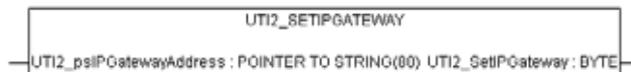


Abbildung 97: UTI2\_SetIPGateway

#### **ACHTUNG**

Ein neu eingetragener Wert muss durch einen „SaveRegistry“- oder einen „Reboot“-Befehl nichtflüchtig gespeichert werden. Der neu eingetragene Wert wird erst nach einem Neustart der Steuerung übernommen.

Tabelle 30: Eingangsvariablen für UTI2\_SetIPGateway

Eingangsvariable	Bedeutung
UTI2_psIPGatewayAddress	Zeiger auf eine String-Variable, die den einzutragenden Wert für die Gateway-Adresse enthält

## 11 Bibliotheken, Funktionsbausteine und Funktionen

### 11.2 XC200-spezifische Funktionen

Tabelle 31: Rückgabewerte für UTI2\_SetIPGateway

Rückgabewert	Bedeutung
1	Schreiben erfolgreich
< 0	Schreiben fehlgeschlagen (allgemeiner Fehler)
-4	Kein gültiger Zeiger übergeben

#### UTI2\_SetIPWins

Setzen der IP-Adresse eines WINS-Servers in der Registry  
(muss gegebenenfalls mit UTI2\_SaveRegistry gesichert werden)

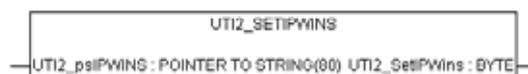


Abbildung 98: UTI2\_SetIPWins

Tabelle 32: Eingangsvariablen für UTI2\_SetIPWins

Eingangsvariable	Bedeutung
UTI2_psIPGatewayAddress	Zeiger auf eine String-Variable, die die zu schreibende IP-Adresse enthält

Tabelle 33: Rückgabewerte für UTI2\_SetIPWins

Rückgabewert	Bedeutung
1	Schreiben erfolgreich
< 0	Schreiben fehlgeschlagen
-4	Kein gültiger Zeiger übergeben

#### UTI2\_Reboot

Neustart mit Registry-Speicherung



Abbildung 99: UTI2\_Reboot

Tabelle 34: Eingangsvariablen für UTI2\_Reboot

Eingangsvariable	Bedeutung
UTI2_Dummy	Ein Dummy-Byte, dass in der Funktion nicht ausgewertet wird

Tabelle 35: Rückgabewerte für UTI2\_Reboot

Rückgabewert	Bedeutung
1	Dummy-Rückgabewert / Reboot erfolgt im Anschluss.

### UT12\_SaveRegistry Sichern der Registry



Abbildung 100:UT12\_SaveRegistry

Tabelle 36: Eingangsvariablen für UT12\_SaveRegistry

Eingangsvariable	Bedeutung
UT12_Dummy	Ein Dummy-Byte, dass in der Funktion nicht ausgewertet wird

Tabelle 37: Rückgabewerte für UT12\_SaveRegistry

Rückgabewert	Bedeutung
1	Funktion erfolgreich ausgeführt
-1	Fehler

## 11 Bibliotheken, Funktionsbausteine und Funktionen

### 11.2 XC200-spezifische Funktionen

#### 11.2.5 Zusätzliche Funktionen der Library XC200\_Util2.lib für die XC-CPU202

Die Funktionen der Bibliothek XC200\_Util2.lib gehen aus der folgenden Übersicht hervor:

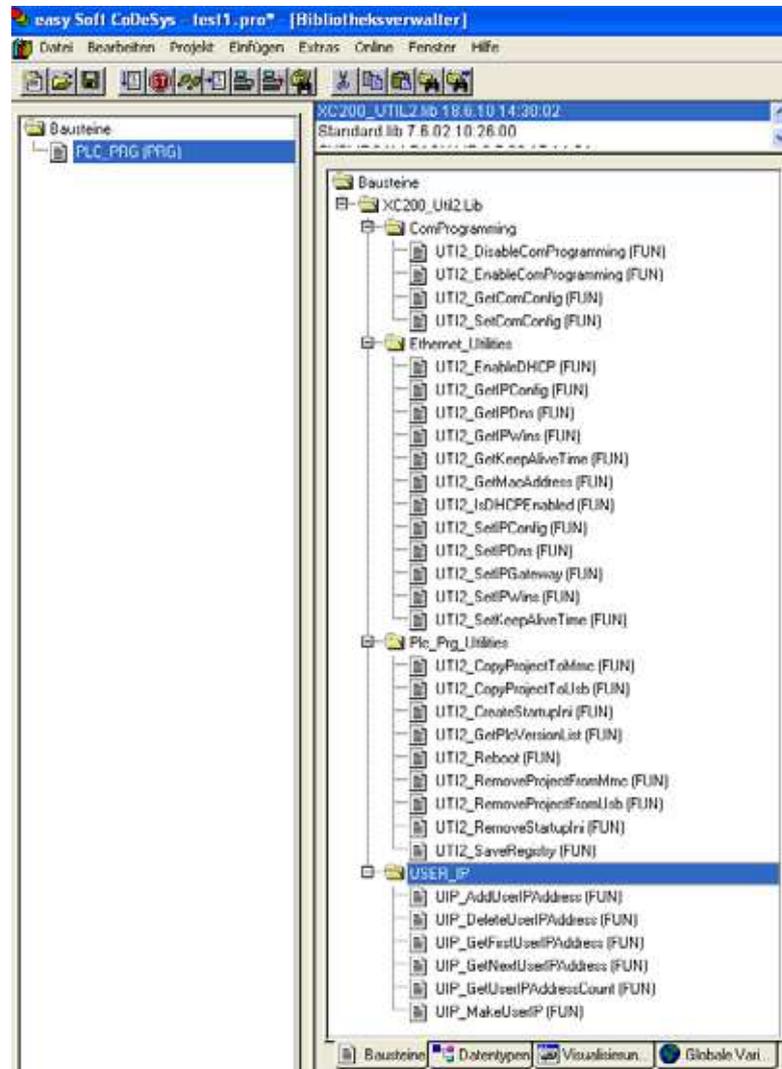


Abbildung 101:Übersicht der XC200\_UTIL2.lib für die XC-CPU202

### 11.2.5.1 ComProgramming

#### UT12\_DisableComProgramming

Deaktiviert die serielle Schnittstelle als Programmierschnittstelle (lokale COM2).



Abbildung 102:UT12\_DisableComProgramming

Tabelle 38: Eingangsvariablen für UT12\_DisableComProgramming

Eingangsvariable	Bedeutung
keine	–

Tabelle 39: Rückgabewerte für UT12\_DisableComProgramming

Rückgabewert	Bedeutung
TRUE	Funktion erfolgreich ausgeführt
FALSE	Fehler

#### UT12\_EnableComProgramming

Aktiviert die serielle Schnittstelle als Programmierschnittstelle (lokale COM2).



Abbildung 103:UT12\_EnableComProgramming

Tabelle 40: Eingangsvariablen für UT12\_EnableComProgramming

Eingangsvariable	Bedeutung
keine	–

Tabelle 41: Rückgabewerte für UT12\_EnableComProgramming

Rückgabewert	Bedeutung
TRUE	Funktion erfolgreich ausgeführt
FALSE	Fehler

#### UT12\_GetComConfig

Zeigt die Schnittstellenparameter der lokalen COM-Schnittstelle.



Abbildung 104:UT12\_GetComConfig

## 11 Bibliotheken, Funktionsbausteine und Funktionen

### 11.2 XC200-spezifische Funktionen

Tabelle 42: Eingangsvariablen für UTI2\_GetComConfig

Eingangsvariable	Bedeutung
keine	–

Tabelle 43: Rückgabewerte für UTI2\_GetComConfig

Rückgabewert	Bedeutung
DWORDCOM	Baudrate des lokalen COM-Ports (4800, 9600, 19200, 38400, 57600, 115200)

#### UTI2\_SetComConfig

Setzen der Baudrate der lokalen seriellen Schnittstelle

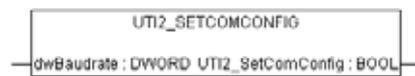


Abbildung 105: UTI2\_SetComConfig

Tabelle 44: Eingangsvariablen für UTI2\_SetComConfig

Eingangsvariable	Bedeutung
dwBaudrate	(4800, 9600, 19200, 38400, 57600, 115200)

Tabelle 45: Rückgabewerte für UTI2\_SetComConfig

Rückgabewert	Bedeutung
TRUE	
FALSE	

### 11.2.5.2 Ethernet\_Utilities

#### UTI2\_EnableDHCP

Aktiviert die DHCP-Funktion der SPS.

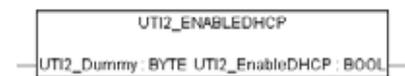


Abbildung 106: UTI2\_EnableDHCP

Tabelle 46: Eingangsvariablen für UTI2\_EnableDHCP

Eingangsvariable	Bedeutung
Dummy BYTE	

Tabelle 47: Rückgabewerte für UTI2\_EnableDHCP

Rückgabewert	Bedeutung
TRUE	immer wahr

Um mit DHCP zu arbeiten, müssen Sie die Funktion „UTI2\_SaveRegistry“ aufrufen und die Steuerung neu booten. Nach dem Booten fordert die Steuerung eine IP-Adresse vom DHCP-Server an.

Um die DHCP-Funktion zu deaktivieren, müssen Sie die Funktion „UTI2\_SetIPConfig“ oder den Browser-Befehl setipconfig aufrufen.

### UTI2\_GetIPConfig

Ausgabe der IP-, Subnetmask- und IPGateway-Adresse

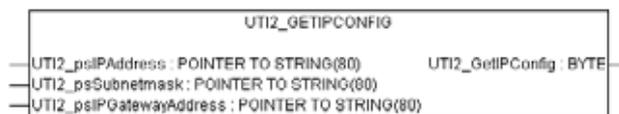


Abbildung 107: UTI2\_GetIPConfig

Tabelle 48: Eingangsvariablen für UTI2\_GetIPConfig

Eingangsvariable	Bedeutung
UTI2_psIPAddress	Zeiger auf einen String, in den die ausgelesene IP-Adresse eingeschrieben wird
UTI2_psSubnetmask	Zeiger auf einen String, in den die ausgelesene Subnetzmaske eingeschrieben wird
UTI2_psIPGatewayAddress	Zeiger auf einen String, in den die ausgelesene Adresse des Standard-Gateways eingeschrieben wird

Tabelle 49: Rückgabewerte für UTI2\_GetIPConfig

Rückgabewert	Bedeutung
1	Auslesen erfolgreich
< 0	Auslesen fehlgeschlagen
-4	Kein gültiger Zeiger übertragen

### UTI2\_GetIPDns

Ausgabe der IP-Adresse des momentan in der „Registry“ eingetragenen DNS-Servers

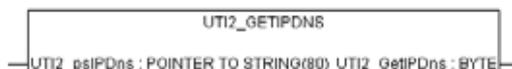


Abbildung 108: UTI2\_GetIPDns

## 11 Bibliotheken, Funktionsbausteine und Funktionen

### 11.2 XC200-spezifische Funktionen

Tabelle 50: Eingangsvariablen für UTI2\_GetIPDns

Eingangsvariable	Bedeutung
UTI2_psIPDns	Zeiger auf einen String, in den die ausgelesene IP-Adresse eingeschrieben wird

Tabelle 51: Rückgabewerte für UTI2\_GetIPDns

Rückgabewert	Bedeutung
1	Auslesen erfolgreich
< 0	Auslesen fehlgeschlagen
-4	Kein gültiger Zeiger übergeben

#### UTI2\_GetIPWins

Ausgabe der IP-Adresse des momentan in der „Registry“ eingetragenen WINS-Servers



Abbildung 109: UTI2\_GetIPWins

Tabelle 52: Eingangsvariablen für UTI2\_GetIPWins

Eingangsvariable	Bedeutung
UTI2_psIPWins	Zeiger auf einen String, in den die ausgelesene IP-Adresse hineingeschrieben wird

Tabelle 53: Rückgabewerte für UTI2\_GetIPWins

Rückgabewert	Bedeutung
1	Auslesen erfolgreich
< 0	Auslesen fehlgeschlagen
-4	Kein gültiger Zeiger übergeben

#### UTI2\_GetKeepAliveTime

Ausgabe der KeepAliveTime in Sekunden



Abbildung 110: UTI2\_GetKeepAliveTime

Tabelle 54: Eingangsvariablen für UTI2\_GetKeepAliveTime

Eingangsvariable	Bedeutung
Dummy BYTE	Wird in der Funktion nicht ausgewertet.

Tabelle 55: Rückgabewerte für UTI2\_GetKeepAliveTime

Rückgabewert	Bedeutung
KeepAliveTime	KeepAliveTime in Sekunden

### UTI2\_GetMacAddress

Ausgabe der MAC-Adresse (MAC = Media Access Control)

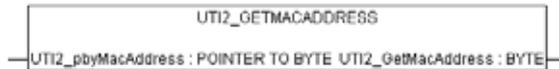


Abbildung 111: UTI2\_GetMacAddress

Tabelle 56: Eingangsvariablen für UTI2\_GetMacAddress

Eingangsvariable	Bedeutung
UTI2_pbyMacAddress	Zeiger auf ein Array von 5 Byte-Werten, in das die ausgelesene MAC-Adresse eingetragen wird

Tabelle 57: Rückgabewerte für UTI2\_GetMacAddress

Rückgabewert	Bedeutung
1	Auslesen erfolgreich
< 0	Auslesen fehlgeschlagen (allgemeiner Fehler)
-4	Kein gültiger Zeiger übergeben

### UTI2\_IsDHCPEnabled

Lesen des DHCP-Status

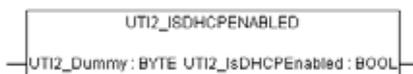


Abbildung 112: UTI2\_IsDHCPEnabled

Tabelle 58: Eingangsvariablen für UTI2\_IsDHCPEnabled

Eingangsvariable	Bedeutung
Dummy BYTE	Wird in der Funktion nicht ausgewertet.

Tabelle 59: Rückgabewerte für UTI2\_IsDHCPEnabled

Rückgabewert	Bedeutung
TRUE	DHCP aktiv
FALSE	DHCP inaktiv

## 11 Bibliotheken, Funktionsbausteine und Funktionen

### 11.2 XC200-spezifische Funktionen

#### UT12\_SetIPConfig

Setzen der IP- und Subnetmask-Adresse

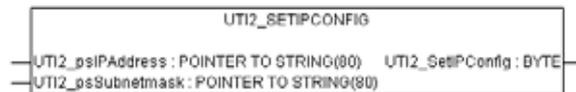


Abbildung 113:UT12\_SetIPConfig

Tabelle 60: Eingangsvariablen für UT12\_SetIPConfig

Eingangsvariable	Bedeutung
UT12_psIPAddress	Zeiger auf eine String-Variable, die die zu schreibende IP-Adresse enthält
UT12_psSubnetmask	Zeiger auf eine String-Variable, die den einzutragenden Wert für die Subnetzmaste enthält

Tabelle 61: Rückgabewerte für UT12\_SetIPConfig

Rückgabewert	Bedeutung
1	Schreiben erfolgreich
< 0	Schreiben fehlgeschlagen (allgemeiner Fehler)
-4	Kein gültiger Zeiger übergeben

#### UT12\_SetIPDns

Setzen der IP-Adresse eines DNS-Servers in der Registry  
(muss gegebenenfalls mit `UT12_SaveRegistry` gesichert werden)

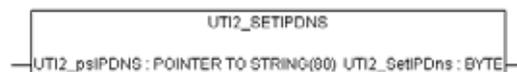


Abbildung 114:UT12\_SetIPDns

Tabelle 62: Eingangsvariablen für UT12\_SetIPDns

Eingangsvariable	Bedeutung
UT12_psIPDns	Zeiger auf eine String-Variable, die die zu schreibende IP-Adresse enthält

Tabelle 63: Rückgabewerte für UT12\_SetIPDns

Rückgabewert	Bedeutung
1	Schreiben erfolgreich
< 0	Schreiben fehlgeschlagen
-4	Kein gültiger Zeiger übergeben

### UT12\_SetIPGateway

Setzen der IPGateway-Adresse

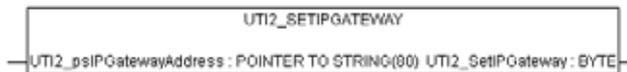


Abbildung 115:UT12\_SetIPGateway

Tabelle 64: Eingangsvariablen für UT12\_SetIPGateway

Eingangsvariable	Bedeutung
UT12_psIPGatewayAddress	Zeiger auf eine String-Variable, die den einzutragenden Wert für die Gateway-Adresse enthält

Tabelle 65: Rückgabewerte für UT12\_SetIPGateway

Rückgabewert	Bedeutung
1	Schreiben erfolgreich
< 0	Schreiben fehlgeschlagen (allgemeiner Fehler)
-4	Kein gültiger Zeiger übergeben

### UT12\_SetIPWins

Setzen der IP-Adresse eines WINS-Servers in der Registry  
(muss gegebenenfalls mit UT12\_SaveRegistry gesichert werden)

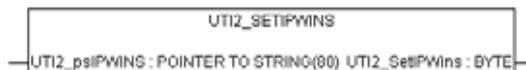


Abbildung 116:UT12\_SetIPWins

Tabelle 66: Eingangsvariablen für UT12\_SetIPWins

Eingangsvariable	Bedeutung
UT12_psIPGatewayAddress	Zeiger auf eine String-Variable, die die zu schreibende IP-Adresse enthält

Tabelle 67: Rückgabewerte für UT12\_SetIPWins

Rückgabewert	Bedeutung
1	Schreiben erfolgreich
< 0	Schreiben fehlgeschlagen
-4	Kein gültiger Zeiger übergeben

## 11 Bibliotheken, Funktionsbausteine und Funktionen

### 11.2 XC200-spezifische Funktionen

#### UT12\_SetKeepAliveTime

Setzt die KeepAliveTime in Sekunden.



Abbildung 117:UT12\_SetKeepAliveTime

Tabelle 68: Eingangsvariablen für UT12\_SetKeepAliveTime

Eingangsvariable	Bedeutung
5 – 500	Gültige Werte in Sekunden

Tabelle 69: Rückgabewerte für UT12\_SetKeepAliveTime

Rückgabewert	Bedeutung
TRUE	Wert gültig
FALSE	Wert außerhalb des gültigen Bereichs

### 11.2.5.3 Plc\_Prg\_Uutilities

#### UT12\_CopyProjectToMmc

Kopiert ein Projekt auf MMC/SD.



Abbildung 118:UT12\_CopyProjectToMmc

Tabelle 70: Eingangsvariablen für UT12\_CopyProjectToMmc

Eingangsvariable	Bedeutung
keine	–

Tabelle 71: Rückgabewerte für UT12\_CopyProjectToMmc

Rückgabewert	Bedeutung
TRUE	Funktion erfolgreich ausgeführt.
FALSE	Fehler

#### UT12\_CopyProjectToUsb

Kopiert ein Projekt auf den USB-Stick.



Abbildung 119:UT12\_CopyProjectToUsb

Tabelle 72: Eingangsvariablen für UTI2\_CopyProjectToUsb

Eingangsvariable	Bedeutung
keine	–

Tabelle 73: Rückgabewerte für UTI2\_CopyProjectToUsb

Rückgabewert	Bedeutung
TRUE	Funktion erfolgreich ausgeführt
FALSE	Fehler

### UTI2\_CreateStartupIni

Erstellt die Datei Startup.ini auf disk\_sys und disk\_mmc.



Abbildung 120:UTI2\_CreateStartupIni

Tabelle 74: Eingangsvariablen für UTI2\_CreateStartupIni

Eingangsvariable	Bedeutung
keine	–

Tabelle 75: Rückgabewerte für UTI2\_CreateStartupIni

Rückgabewert	Bedeutung
TRUE	Funktion erfolgreich ausgeführt
FALSE	Fehler

### UTI2\_GetPlcVersionList

Anzeigen der Geräte-Versionsliste



Abbildung 121:UTI2\_GetPlcVersionList

Tabelle 76: Eingangsvariablen für UTI2\_GetPlcVersionList

Eingangsvariable	Bedeutung
pVersionList	Zeiger auf UTI2_VersionList

Tabelle 77: Rückgabewerte für UTI2\_GetPlcVersionList

Rückgabewert	Bedeutung
TRUE	Funktion erfolgreich ausgeführt
FALSE	Fehler

## 11 Bibliotheken, Funktionsbausteine und Funktionen

### 11.2 XC200-spezifische Funktionen

#### UT12\_Reboot

Neustarten der Steuerung



Abbildung 122:UT12\_Reboot

Tabelle 78: Eingangsvariablen für UT12\_Reboot

Eingangsvariable	Bedeutung
UTI_Dummy	Setzt Variable auf 0.

Tabelle 79: Rückgabewerte für UT12\_Reboot

Rückgabewert	Bedeutung
1	Dummy-Rückgabewert / Reboot erfolgt im Anschluss.

#### UT12\_RemoveProjectFromMmc

Entfernen des Backup-Projekts von der MMC/SD



Abbildung 123:UT12\_RemoveProjectFromMmc

Tabelle 80: Eingangsvariablen für UT12\_RemoveProjectFromMmc

Eingangsvariable	Bedeutung
keine	–

Tabelle 81: Rückgabewerte für UT12\_RemoveProjectFromMmc

Rückgabewert	Bedeutung
TRUE	Funktion erfolgreich ausgeführt
FALSE	Fehler

### UT12\_RemoveProjectFromUsb

Entfernen des Backup-Projekts vom USB-Stick



Abbildung 124:UT12\_RemoveProjectFromUsb

Tabelle 82: Eingangsvariablen für UT12\_RemoveProjectFromUsb

Eingangsvariable	Bedeutung
keine	–

Tabelle 83: Rückgabewerte für UT12\_RemoveProjectFromUsb

Rückgabewert	Bedeutung
TRUE	Funktion erfolgreich ausgeführt
FALSE	Fehler

### UT12\_RemoveStartupIni

Entfernt die Datei Startup.ini von der disk\_sys und der disk\_mmc.



Abbildung 125:UT12\_RemoveStartupIni

Tabelle 84: Eingangsvariablen für UT12\_RemoveStartupIni

Eingangsvariable	Bedeutung
keine	–

Tabelle 85: Rückgabewerte für UT12\_RemoveStartupIni

Rückgabewert	Bedeutung
TRUE	Funktion erfolgreich ausgeführt
FALSE	Fehler

## 11 Bibliotheken, Funktionsbausteine und Funktionen

### 11.2 XC200-spezifische Funktionen

#### UT12\_SaveRegistry

Sichert Änderungen in der „Registry“ nichtflüchtig.



Abbildung 126:UT12\_SaveRegistry

Tabelle 86: Eingangsvariablen für UT12\_SaveRegistry

Eingangsvariable	Bedeutung
UT12_Dummy	Setzt Variable auf 0.

Tabelle 87: Rückgabewerte für UT12\_SaveRegistry

Rückgabewert	Bedeutung
1	Funktion erfolgreich ausgeführt
-1	Fehler

### 11.2.5.4 USER\_IP

#### UIP\_AddUserIPAddress

Fügt dem System eine neue IP-Adresse hinzu.

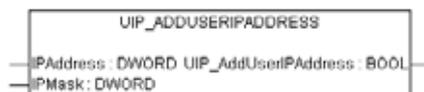


Abbildung 127:UIP\_AddUserIPAddress

Tabelle 88: Eingangsvariablen für UIP\_AddUserIPAddress

Eingangsvariable	Bedeutung
IPAddress	Hinzuzufügende IP-Adresse
IPMask	Hinzuzufügende zugehörige IP-Maske

Tabelle 89: Rückgabewerte für UIP\_AddUserIPAddress

Rückgabewert	Bedeutung
TRUE	IP-Adresse wurde erfolgreich hinzugefügt.
FALSE	Tabelle voll; Adresse bereits in User-IP-Tabelle vorhanden, Adresse bereits in Betriebssystemtabelle vorhanden

### UIP\_DeleteUserIPAddress

Löscht eine IP-Adresse aus dem System.



Abbildung 128: UIP\_DeleteUserIPAddress

Tabelle 90: Eingangsvariablen für UIP\_DeleteUserIPAddress

Eingangsvariable	Bedeutung
IPAddress	Identifiziert den Eintrag, der aus der User-IP-Tabelle gelöscht werden soll.

Tabelle 91: Rückgabewerte für UIP\_DeleteUserIPAddress

Rückgabewert	Bedeutung
TRUE	Zugehöriger User-IP-Tabelleneintrag wurde gelöscht.
FALSE	Zugehöriger User-IP-Tabelleneintrag war nicht vorhanden.

### UIP\_GetFirstUserIPAddress

Liest die erste eingetragene User-IP-Adresse aus der User-IP-Tabelle.

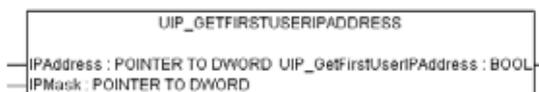


Abbildung 129: UIP\_GetFirstUserIPAddress

Tabelle 92: Eingangsvariablen für UIP\_GetFirstUserIPAddress

Eingangsvariable	Bedeutung
IPAddress	Zeigt auf Platzhalter für den zu ermittelnden User-Tabelleneintrag.
IPMask	Zeigt auf Platzhalter für den zu ermittelnden User-Tabelleneintrag.

Tabelle 93: Rückgabewerte für UIP\_GetFirstUserIPAddress

Rückgabewert	Bedeutung
TRUE	Gültigen Eintrag gefunden.
FALSE	Die User-IP-Adresstabelle enthält keinen Eintrag.

## 11 Bibliotheken, Funktionsbausteine und Funktionen

### 11.2 XC200-spezifische Funktionen

#### UIP\_GetNextUserIPAddress

Liest die „nächste“ eingetragene User-IP-Adresse aus der User-IP-Tabelle.

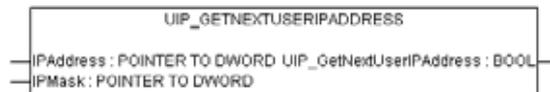


Abbildung 130: UIP\_GetNextUserIPAddress

Tabelle 94: Eingangsvariablen für UIP\_GetNextUserIPAddress

Eingangsvariable	Bedeutung
IPAddress	Zeiger auf DWORD-Platzhalter für den zu ermittelnden User-Tabelleneintrag
IPMask	Zeiger auf DWORD-Platzhalter für den zu ermittelnden User-Tabelleneintrag

Tabelle 95: Rückgabewerte für UIP\_GetNextUserIPAddress

Rückgabewert	Bedeutung
TRUE	Gültigen Eintrag gefunden
FALSE	Die User-IP-Adresstabelle enthält keinen weiteren Eintrag.

#### UIP\_GetUserIPAddressCount

Liefert die Anzahl der momentan in der User-IP-Tabelle vorhandenen User-IP-Einträge.



Abbildung 131: UIP\_GetUserIPAddressCount

Tabelle 96: Eingangsvariablen für UIP\_GetUserIPAddressCount

Eingangsvariable	Bedeutung
keine	–

Tabelle 97: Rückgabewerte für UIP\_GetUserIPAddressCount

Rückgabewert	Bedeutung
0 bis maximale Anzahl (= 3)	Anzahl der momentan in der User-IP-Adresstabelle eingetragenen Werte

### UIP\_MakeUserIP

Wandelt IP-String in IP\_DWORD (Bigendianness)



Abbildung 132: UIP\_MakeUserIP

Tabelle 98: Eingangsvariablen für UIP\_MakeUserIP

Eingangsvariable	Bedeutung
IPAdress	Zeiger auf IP-Adresse in Hochkamma (z. B. '192.168.119.11')

Tabelle 99: Rückgabewerte für UIP\_MakeUserIP

Rückgabewert	Bedeutung
IP-Adresse in Hexadezimal-Darstellung	IP als DWORD (z. B. 16#C0A8770B – korrespondierend zu obiger IP-Adresse 192.168.119.11)
0	Fehler im Eingangsstring oder ungültiger String

## 12 Browser-Befehle

### 11.2 XC200-spezifische Funktionen

## 12 Browser-Befehle

Der PLC-Browser ist ein textbasierter Steuerungs-(Terminal-)Monitor. Kommandos zur Abfrage bestimmter Informationen aus der Steuerung werden in einer Eingabezeile eingegeben und als String an die Steuerung geschickt. Der Antwortstring wird in einem Ergebnisfenster des Browsers dargestellt. Diese Funktionalität kann zu Diagnose- und Debugging-Zwecken genutzt werden. Folgend die für das Zielsystem XC200 verfügbaren Browser-Befehle.

Tabelle 100: Browser-Befehle (in alphabetischer Reihenfolge)

Befehl	Beschreibung
?	Liste der implementierten Kommandos holen
caninfo	Display CAN Controller Informationen
canload	Anzeige der Auslastung des CAN-Feldbusses
clearerrorlist	Fehlerliste löschen
cleareventlist	Ereignisliste löschen
copyprojtommc <sup>1)</sup>	(Boot-)Projekt auf Multi Media Card kopieren (inkl. Verzeichnisstruktur/Projekt-Verzeichnis)
copyprojtousb <sup>1)</sup>	(Boot-)Projekt auf USB-Laufwerk kopieren (inkl. Verzeichnisstruktur/Projekt-Verzeichnis)
createstartupini	Datei Startupini auf disk_sys und disk_mmc erstellen
delpwd	Passwort für Online-Zugriff löschen
dpt	Datenzeiger-Tabelle ausgeben
enabledhcp	Aktiviert die DHCP-Funktion der SPS.
filecopy <sup>1)</sup>	Datei kopieren
filedelete <sup>1)</sup>	Datei löschen
filedir <sup>1)</sup>	Verzeichnisliste [Erstes Verzeichnis in der Liste]
filerename <sup>1)</sup>	Datei umbenennen
getbattery	Batteriestatus anzeigen
getcomconfig	Baudrate der seriellen Schnittstelle 1 anzeigen
getcommunicationport	Lesen der Schnittstellenparameter für die TCP/IP-Kommunikation
geterrorlist	Fehlerliste anzeigen
geteventlist	Ereignisliste anzeigen
getipconfig	Ethernet-Adresse anzeigen
getipdns	Aktuelle DNS-Adresse anzeigen
getipgateway	Gateway-Adresse anzeigen
getipwins	Aktuelle WINS-Adresse anzeigen
getlanguage	Dialogsprache für Fehlerliste anzeigen
getmacaddress	MAC-Adresse anzeigen [80-80-99-2-x-x]
getprgprop	Programminformation lesen
getprgstat	Programmstatus lesen
getrtc	Datum und Uhrzeit anzeigen [JJ:MM:TT] [HH:MM:SS]
getswitchpos	Status des Betriebsschalters anzeigen
gettargetname	Gerätenamen anzeigen
getversion	Versionsinformation anzeigen

Befehl	Beschreibung
isdhcpenabled	Abfrage ob DHCP aktiviert
memdisk_sys	Zeigt den freien Speicher auf disk_sys an.
pid	Projekt-ID ausgeben
pinf	Projektinformationen ausgeben
plcload <sup>1)</sup>	Anzeige Systemleistung: CPU-Nutzung
ppt	Bausteineiger-Tabelle ausgeben
reboot	Änderungen übernehmen (Registry speichern) und SPS neu starten
reflect <sup>1)</sup>	Aktuelle Kommandozeile spiegeln, zu Testzwecken
reload	Bootprojekt erneut laden
removeprojfrommmc	Löschen des Backup-Projekts von der MMC/SD
removestartupini	Datei Startup.ini von disk_sys und disk_mmc löschen
resetprg	Anwenderprogramm reset
resetprgcold	Anwenderprogramm reset kalt
resetprgorg	Anwenderprogramm reset Ursprung
restoreretain	Remanente Daten von Datei restaurieren [Dateiname]
rtsinfo	runtime system information (version, IO drivers)
saveregistry	Änderungen übernehmen
saveretain	Remanente Daten in Datei sichern [Dateiname]
serialProgrammingOn	„Enable serial programming with CoDeSys on XC202 local COM2 port“ – Gibt die lokale COM als Programmierschnittstelle frei. Auslieferungszustand. Serielle Schnittstelle im Programmiermodus
serialProgrammingOff	„Disable serial programming with CoDeSys on XC202 local COM2 port“ – Sperrt die lokale COM als Programmierschnittstelle. Auslieferungszustand. Serielle Schnittstelle im Programmiermodus
setcomconfig <sup>1)</sup>	Baudrate der seriellen Schnittstelle setzen [setcomconfig 4800,9600,19200, 38400, 57600,115200]
setcommunicationport	Schnittstellenparameter für die TCP/IP-Kommunikation setzen
setipconfig <sup>1)</sup>	Ethernet-Adresse einstellen [setipconfig adr1.adr2.adr3.adr4 mask1.mask2.mask3.mask4]; z. B. setipconfig 192.168.119.010 255.255.255.000
setipdns	DNS-Adresse setzen [setipdns adr1.adr2.adr3.adr4]
setipgateway <sup>1)</sup>	Gateway-Adresse einstellen [adr1.adr2.adr3.adr4]; z. B.: setipgateway 192.168.119.010
setipwins	WINS-Adresse setzen [setipwins adr1.adr2.adr3.adr4]
setlanguage	Dialogsprache für Fehlerliste festlegen [deu/eng/fra/ita]
setpwd	Passwort für Online-Zugriff aktivieren
setrtc <sup>1)</sup>	Datum und Uhrzeit einstellen [JJ:MM:TT] [HH:MM:SS]; z. B.: setrtc 03:07:24 10:46:33
settargetname <sup>1)</sup>	Gerätename setzen [Gerätename]; z. B.: settargetname test
shutdown	Änderungen übernehmen (Registry speichern) und SPS abschalten
startprg	Anwenderprogramm starten
stopprg	Anwenderprogramm stoppen
tsk	IEC-Taskliste mit Taskinformationen ausgeben
tskclear	Clear IEC task information: Cyclecount, accumulated, max. and min. cycle
updatefrommmc	update windows image von /disk_mmc/MOELLER/XC-CPU201/btsxc201_Vxxxx.nbk

1) Für diese Browser-Befehle können Sie in der CODESYS-Software eine Hilfe mit erweiterten Informationen aufrufen. Geben Sie hierzu in der Befehlszeile des PLC-Browsers vor dem Befehl ein Fragezeichen mit anschließendem Leerzeichen ein, z. B.: ? plcload

## 12 Browser-Befehle

### 12.1 Browser-Befehle aufrufen

#### 12.1 Browser-Befehle aufrufen

- ▶ Aktivieren Sie in der Software CODESYS die Registerkarte „Ressourcen“ und wählen Sie hier den Ordner „PLC-Browser“ aus.
- ▶ Klicken Sie im Fenster oben rechts auf die Schaltfläche „...“.
- ▶ Wählen Sie den gewünschten Browser-Befehl mit einem Doppelklick aus. Ergänzen Sie den Befehl – falls erforderlich – mit weiteren Angaben, wie z. B. der Baudrate bei `setcomconfig`, → Tabelle 100.
- ▶ Ergänzen Sie den Befehl in der Eingabezeile unter Umständen.
- ▶ Drücken Sie die Eingabetaste (Return).

Das Ergebnis wird angezeigt.

#### 12.2 Auf Kommunikationsparameter zugreifen

Einstellungen der Kommunikationsparameter über Browser-Befehle wie Geräte-Name, Ethernet-Adresse, Gateway-Adresse oder Baudrate der seriellen Schnittstelle werden mit den folgenden Befehlen im Datenbankeintrag in der Windows-CE REGISTRY nur modifiziert und nicht direkt übernommen bzw. gespeichert. Die Übernahme der Funktion erfolgt erst beim nächsten Start von Windows CE.

- `setcomconfig`
- `setipconfig`
- `setipgateway`
- `settargetname`

Nach der Ausführung eines dieser Browser-Befehle ist ein Speichern der Registry erforderlich. Dazu stehen folgende Browser-Befehle zur Verfügung.

- `saveregistry` (Speichert die Registry.)
- `shutdown` (Speichert die Registry und wartet auf „Spannung aus“.)
- `reboot` (Speichert die Registry und erzeugt einen „Software-Reset“.)

Die Befehle `setcomconfig`, `setipconfig`, `setipgateway` und `settargetname` müssen Sie in der Befehlszeile des PLC-Browsers ergänzen, zum Beispiel mit der Baudrate bei `setcomconfig` → Tabelle 100. Schließen Sie Ihre Angabe mit der Eingabetaste (Return) ab. Eine Antwort erhalten Sie im grau hinterlegten Fenster.

Der Browser-Befehl `setipconfig` generiert automatisch einen `settargetname`. Der Targetname setzt sich zusammen aus der Kurzbezeichnung des Zielsystems und dem letzten Ziffernblock der IP-Adresse, z. B.: `Xc201_Nr010`.

Der Targetname wird in Abhängigkeit von der IP-Adresse und dem Zielsystem automatisch generiert. Er kann über den Befehl `gettargetname` aufgerufen werden.

### 12.3 Auslastung der CPU anzeigen (plcload)

Der Browser-Befehl `plcload` informiert Sie über die aktuelle Systemauslastung der CPU.

Eine Auslastung von mehr als 95 Prozent kann zu einem Ausfall der seriellen sowie der Ethernet-Kommunikation und/oder zu einer Beeinträchtigung des Echtzeitverhaltens führen.

#### 12.3.1 Auslastung des CAN-Busses anzeigen (canload)

Der PLC-Browser-Befehl `canload` ist Bestandteil der Library `XC200_Util.lib`. Er zeigt die Auslastung des CAN-Busses an.

Beispiele zur Anzeige:

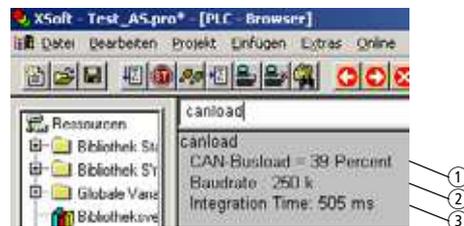


Abbildung 133: Auslastung des CAN-Busses (Beispiel 1)

- a Auslastung des CAN-Busses im letzten Integrationsintervall.
- b Aktuelle Baudrate des CAN-Busses
- c Zeit, über die die Auslastung auf dem CAN-Bus integriert wurde.

Die Integrationszeit ist standardmäßig auf 500 ms eingestellt und lässt sich nicht über den Browser ändern.

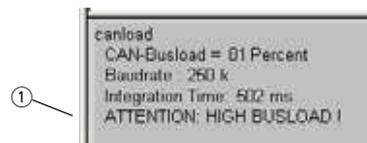


Abbildung 134: Auslastung des CAN-Busses mit Warnmeldung (Beispiel 2)

- a Warnmeldung, → Tabelle 101

Tabelle 101: Mögliche Warnmeldungen

Warnmeldung	Bedeutung
ATTENTION: HIGH BUSLOAD	Auslastung des CAN-Busses f 75 %
CAN-Bus not activated	Der CAN-Bus ist nicht aktiv.
CAN-Busload = Invalid Calculation	Die Überwachung der Busbelastung ist fehlgeschlagen.

## 12 Browser-Befehle

### 12.3 Auslastung der CPU anzeigen (plcload)

#### 12.3.2 Auf Speicherobjekte zugreifen

Diese Befehle haben als Parameter den Namen der Speicherkarte, die Verzeichnisstruktur und den Dateinamen. Achten Sie bitte bei der Eingabe der Befehle auf entsprechende Sonderzeichen.

- filecopy
- filerename
- filedelete
- filedir

#### Beispiele für XC-CPU201

```
filedir (ohne Parameterangabe ist die Defaulteinstellung: \\disk_sys\project)
filedir \\disk_sys
filedir \\disk_sys\project
filedir \\disk_mmc\MOELLER\XC-CPU201-EC512k-8DI-6DO
filedir \\disk_mmc\MOELLER\XC-CPU201-EC512k-8DI-6DO\project\aaa.prg
filedir \\disk_usb\MOELLER\XC-CPU201-EC512k-8DI-6DO
filedir \\disk_usb\MOELLER\XC-CPU201-EC512k-8DI-6DO\project\bbb.prg
filecopy \\disk_sys\project\default.prg\disk_sys\project\yyy.prg
filerename \\disk_sys\project\yyy.prg\disk_sys\project\xxx.prg
filecopy \\disk_sys\project\default.prg\disk_mmc\MOELLER\XC-CPU201-EC512k-8DI-6DO\project\default.prg
filedelete \\disk_mmc\MOELLER\XC-CPU201-EC512k-8DI-6DO\project\default.prg
```



Steht die CPU „XC-CPU201-EC256K-8DI-6DO“ zur Verfügung, wird der Anweisungsteil „512“ durch „256“ ersetzt. Bei der XC-CPU202 wird \CONTROL anstelle von \MOELLER benutzt.

### 12.3.3 Fehler- und Ereignisliste nach Aufruf von Browser-Befehlen

Die Dialogsprache für die Fehler- und Ereignislisten wird in den Sprachen Deutsch, Englisch, Französisch und Italienisch zur Verfügung gestellt.

Die aktive Sprache wird mit `getlanguage` angezeigt; die Umstellung der Sprache erfolgt mit `setlanguage`.

#### Beispiel für die Sprachumstellung

Soll die Fehler- und Ereignisliste in Deutsch angezeigt werden, so ist der Browser-Befehl `setlanguage deu` einzugeben. Die Eingabe beenden Sie mit „Return“. Sie erhalten das nachfolgend dargestellte Fenster.



Abbildung 135: Browser-Befehl „setlanguage“

Nachfolgend eine Übersicht der Meldungen, die in den Browser-Fehler- und Ereignislisten vorkommen können. Die Modul-ID weist darauf hin, welcher Programmtyp den Fehler meldet:

Modul-ID	Programmtyp
1	RTS (Laufzeitsystem)
2	CST (Moeller-spezifische Anpassung)
3	XIO (XIOC)
4	CAN
5	IEC

Die Event-ID gibt die Fehlernummer des Programms an. Für jede Modul-ID kann diese Fehlernummer von 0 aus starten.

Modul-ID	Event-ID	Fehlermeldung
2	1	Stop Programm
2	2	Start Programm
2	3	Reset Warm
2	4	Reset Kalt
2	5	Reset Hard
2	6	Batterie entleert
2	7	Kein Program geladen
2	8	Task-Überwachung
4	10	CAN-Controller gestartet
4	20	CAN-Controller gestoppt
4	30	Überlauf

## 12 Browser-Befehle

### 12.3 Auslastung der CPU anzeigen (plcload)

Modul-ID	Event-ID	Fehlermeldung
4	31	Überlauf
4	40	Überlauf
4	41	Überlauf
4	42	Überlauf
4	50	Kritischer Fehler CAN
4	60	CAN-Controller in Status Error-Warning
4	70	CAN-Controller in Status Bus-Off
1	16	Tasküberwachung fehlgeschlagen
1	17	Hardware-Überwachung fehlgeschlagen
1	18	Busfehler
1	19	Prüfsummenfehler
1	20	Feldbusfehler
1	21	I/O-Aktualisierung fehlgeschlagen
1	22	Zykluszeit überschritten
1	80	Ungültige Anweisung
1	81	Zugriffsverletzung
1	82	Privilegierte Anweisung
1	83	Seitenfehler
1	84	Stack-Überlauf
1	85	Ungültige Disposition
1	86	Ungültige Zugriffskennung
1	87	Zugriff auf geschützte Seite
1	256	Zugriff auf ungerade Adresse
1	257	Array-Grenze überschritten
1	258	Division durch Null
1	259	Überlauf
1	260	Nicht übergehbare Ausnahme
1	336	Gleitkommaeinheit: Allgemeiner Fehler
1	337	Gleitkommaeinheit: Nicht normalisierter Operand
1	338	Gleitkommaeinheit: Division durch Null
1	339	Gleitkommaeinheit: Ungenaueres Ergebnis
1	340	Gleitkommaeinheit: Ungültige Anweisung
1	341	Gleitkommaeinheit: Überlauf
1	342	Gleitkommaeinheit: Stack-Überprüfung fehlgeschlagen
1	343	Gleitkommaeinheit: Unterlauf

## 13 Anhang

### 13.1 Eigenschaften des Ethernet-Kabels

Verwenden Sie für die Verkabelung des Ethernet-Netzwerkes nur ein dafür geeignetes Kabel. Das Kabel muss mindestens der Kategorie Cat-5 entsprechen. Cat-5-Kabel sind für eine Datenübertragungsgeschwindigkeit von 10 und 100 MBit/s geeignet.

Tabelle 102: Eigenschaften des Ethernet-Kabels

	UTP <sup>1)</sup>	STP <sup>2)</sup>	SSTP <sup>3)</sup>
Übertragungsmedium	Unshielded Twisted Pair	Shielded Twisted Pair	Shielded Twisted Pair
Übertragungsgeschwindigkeit	10 MBit/s 100 MBit/s	10 MBit/s 100 MBit/s	10 MBit/s 100 MBit/s
Aufbau	je zwei Adern verseilt	je zwei Adern verseilt	je zwei Adern verseilt
	ohne Schirm	mit Gesamtschirm	mit Gesamtschirm, jedes Aderpaar zusätzlich geschirmt
Flexibilität	mittel	mittel	mittel
Abschirmung	keine	einfach	zweifach
Topologie	Punkt-zu-Punkt	Punkt-zu-Punkt, Linie, Stern	Punkt-zu-Punkt, Linie, Stern
Maximale Segmentlänge	100 m	100 m	100 m

- 1) Vom Einsatz in industriellen Anwendungen ist auf Grund der schlechten EMV-Eigenschaften dringend abzuraten.
- 2) Die Leiterpaare sind von einem Gesamtschirm umgeben. Die Aufgabe des Gesamtschirmes ist es, die äußeren Störeinflüsse abzufangen. Auf Grund der hohen Übersprechwerte zwischen den einzelnen Leiterpaaren ist dieses Kabel für den industriellen Einsatz (bedingt) geeignet.
- 3) Dieses Kabel besitzt gegenüber dem STP-Kabel für jedes Leiterpaar einen separaten Innenschirm. Dadurch werden die Übersprechwerte wesentlich reduziert und das Kabel weist einen guten Schutz gegen EMV-Störungen auf. Aufgrund dieser Eigenschaften ist das SSTP-Kabel sehr gut für den industriellen Einsatz geeignet.

Die maximale Segmentlänge beträgt 100 m. Ist die Ausdehnung des Netzwerkes größer, sind geeignete Infrastrukturkomponenten einzusetzen. Hierfür kommen Transceiver, Hubs und Switches in Frage.

Maßgebend bei der Auswahl des Kabels sind die Umgebungsbedingungen (Störbeeinflussung, Flexibilität, Übertragungsgeschwindigkeit) am Einsatzort.

Die Installationsrichtlinien für die (Ethernet-)Verkabelung sind in der ISO/IEC 11801 und EN50173 beschrieben.

### 13.2 Eigenschaften des CAN-Kabels

Bitte verwenden Sie nur ein für CAN zugelassenes Kabel mit folgenden Eigenschaften:

- Wellenwiderstand 100 bis 120  $\Omega$
- Kapazitätsbelag < 60 pF/m

Die Anforderungen an Kabel, Stecker und Busabschlusswiderstand sind in der ISO 11898 spezifiziert. Nachfolgend sind einige Anforderungen und Festlegungen für das CAN-Netzwerk aufgeführt.

In der Tabelle 104 sind Standard-Parameter für das CAN-Netzwerk mit weniger als 64 CAN-Teilnehmern aufgelistet. (Die Tabelle entspricht den Vorgaben der ISO 11898.)

Die Länge der CAN-Busleitung ist abhängig vom Leiterquerschnitt und von der Anzahl der angeschlossenen Busteilnehmer. Die nachfolgende Tabelle enthält Werte für die Buslänge in Abhängigkeit des Querschnitts und der angeschlossenen Busteilnehmer, die eine gesicherte Busverbindung gewährleisten (Tabelle entspricht den Vorgaben der ISO 11898).

Tabelle 103:Kabelquerschnitt, Buslänge und Anzahl der Busteilnehmer entsprechend der ISO 11898

Kabel-Querschnitt [mm]	Maximale Länge [m]		
	n = 32	n = 64	n = 100
0,25	200	170	150
0,5	360	310	270
0,75	550	470	410

n = Anzahl der angeschlossenen Busteilnehmer

Ist die Buslänge größer als 250 m und/oder sind mehr als 64 Teilnehmer angeschlossen, fordert die ISO 11898 eine Restwelligkeit der Versorgungsspannung von  $\leq 5$  Prozent.

Da die Busleitung direkt am COMBICON-Stecker der CPU angeschlossen wird, sind zusätzliche Angaben für Stichleitungen nicht erforderlich. Die Busteilnehmer werden in der Programmiersoftware im Fenster „Steuerungskonfiguration“ der CPU konfiguriert.

Kabelempfehlung: LAPP-Kabel, UNITRONIC-BUS LD

Tabelle 104:Standardparameter für CAN-Netzwerkkel gemäß ISO 11898

Buslänge [m]	Schleifenwiderstand [m $\Omega$ /m]	Aderquerschnitt [mm <sup>2</sup> ]	Busabschlusswiderstand [ $\Omega$ ]	Übertragungsrate bei Leitungslänge [kBit/s]
0 – 40	70	0,25 – 0,34	124	1000 bei 40 m
40 – 300	< 60	0,34 – 0,6	150 – 300	> 500 bei 100 m
300 – 600	< 40	0,5 – 0,6	150 – 300	> 100 bei 500 m
600 – 1000	< 26	0,75 – 0,8	150 – 300	> 50 bei 1000 m

### 13.3 Transparentmodus: Textausgabe über RS232 (Beispiel)

Das Beispiel zeigt eine Textausgabe über die RS232-Schnittstelle der CPU im Transparent-Modus.

```

PROGRAM PLC_PRG
VAR
  BREMSE:TON;
  STEP:UINT;
  dwSioHandle: DWORD;
  WriteBuffer:STRING(26);
  nWriteLength: DWORD;
  typComSettings:COMSETTINGS;
  typComSetSettings:BOOL;
  out AT %QB0:BYTE;
  INP AT %IX0.0:BOOL;
  STEPERR: UINT;
  Closeresult: BOOL;
  Coun: DWORD;
  RESET: BOOL;
END_VAR

(*Zykluszeit/Cycletime: 50ms!*)
CASE STEP OF
0: IF INP =1 THEN (*Start: IX0.0 = TRUE*)
  STEP:=1;
  END_IF
1: (*Öffnen/Open*)
  IF dwSioHandle=0 THEN
    dwSioHandle:=xSysComOpen(Port:=Com1);
    IF (dwSioHandle>0) THEN
      typComSettings.typBaudRate           :=Baud_9600;
      typComSettings.typDataLength        :=Data_8Bit;
      typComSettings.typParity            :=NO_PARITY;
      typComSettings.typPort              :=COM1;
      typComSettings.typStopBits          :=ONE_STOPBIT;
      xSysComSetSettings(dwHandle:=dwSioHandle,
        ComSettings:=ADR(typComSettings));
      STEP:=2;
      RESET:=TRUE;
    ELS
    E
      STEPERR:=STEP;
      STEP:=99;
    END_IF
    WriteBuffer:='Das ist der Sendetext';
  END_IF
END_IF

```

## 13 Anhang

### 13.3 Transparentmodus: Textausgabe über RS232 (Beispiel)

```
2: (*Ausgabe/Output*)
  IF (dwSioHandle>0) THEN
    nWriteLength:=xSysComWrite(dwHandle:=dwSioHandle,
    dwBufferAddress:=ADR(WriteBuffer),
    dwBytesToWrite:=LEN(WriteBuffer)+1,dwTimeOut:=0);
  END_IF
  IF nWriteLength = LEN(WriteBuffer)+1 THEN
    STEP:=3;
    Coun:=coun+1;
  END_IF
3: (*Schliessen/Shut*)
  Closeresult:=xSysComClose(dwHandle:=dwSioHandle);
  IF (Closeresult = TRUE) THEN
    dwSioHandle:=0;
    STEP:=4;
  ELSE
    STEPERR:=STEP;
    STEP:=99;
  END_IF
4: (*Verzögerung/Delay*)
  BREMSE(IN:=1, PT:=T#2s);
  IF BREMSE.Q = 1 THEN
    STEP :=5;
    BREMSE(IN:=0, PT:=T#2s);
  END_IF
5: (*Ende*)
  STEP:=0;
99: (*Fehler/Error*)
  STEPERR:=STEPERR;
END_CASE
```

## 13.4 Zugriff auf die CPU-Laufwerke/Speicherkarte

### 13.4.1 Bibliothek SysLibFile.lib

Die Bibliothek SysLibFile ermöglicht Ihnen den Zugang zum Dateisystem der XC200, der MMC/SD und dem des USB-Sticks.

Die Bibliothek beinhaltet die folgenden Funktionen:

- SysFileClose
- SysFileCopy
- SysFileDelete
- SysFileEOF
- SysFileGetPos
- SysFileGetSize
- SysFileGetTime
- SysFileOpen
- SysFileRead
- SysFileRename
- SysFileSetPos
- SysFileWrite



Informationen zu diesen Funktionen finden Sie in der Online-Dokumentation des Programmiersystems CODESYS unter dem Stichwort „SysFile<Funktion>“

#### **ACHTUNG**

- Die Steuerung darf bei geöffneten Dateien von der MMC/SD oder dem USB-Stick aus nicht ausgeschaltet werden.
- Ein Spannungsausfall kann bei geöffneter Datei zur Zerstörung der Speicherkarte führen.
- Vor dem Ausschalten müssen die geöffneten Dateien geschlossen werden.

## 13.4.2 Modi zum Öffnen einer Datei

### 13.4.2.1 Modus „r“

Der Modus „r“ öffnet die Datei zum Lesen. Das Datei-Handle, das die Funktion SysFileOpen zurückgibt ist ungültig, falls die Datei nicht existiert. Es wird in diesem Fall der Wert „-1“ oder „16#FFFFFFFF“ angezeigt.

Die Datei wird zum sequenziellen Lesen geöffnet und mit jedem weiteren Lesezugriff die Leseposition um die Anzahl der gelesenen Bytes weiter geschoben.

#### 13.4.2.2 Modus „w“

Der Modus „w“ öffnet die Datei im Schreibmodus. Eine vorhandene Datei mit diesem Namen wird dabei überschrieben.

##### **ACHTUNG**

Wenn Sie eine Datei im Modus „w“ öffnen und wieder schließen, wird dabei diese Datei überschrieben und eine Dateilänge von 0 Byte erzeugt.

#### 13.4.2.3 Modus „a“

Der Modus „a“ (append) öffnet die Datei wie im Modus „w“. Beim Schreiben auf diese Datei wird dann der neue Text am Dateiende angehängt.

Den Funktionen `SysFileRead` und `SysFileWrite` wird jeweils ein Buffer- und ein Dateihandle-Rückgabewert aus der Funktion `SysFileOpen` übergeben.

Um eine Datei wieder zu schließen, wird die Funktion `SysFileClose` mit dem Rückgabewert aus der Funktion `SysFileOpen` aufgerufen.

#### 13.4.3 Beispiele zu den Funktionen „SysFile...“

Um eine Datei zu öffnen, wird die Funktion `SysFileOpen` benutzt. Die Funktion bekommt den Dateinamen – mit vollständigem Dateipfad – übergeben. Weiter erhält die Funktion den Modus, in dem die Datei geöffnet werden soll.

##### **Öffnen im Modus „r“**

```
OpenFile1 := SysFileOpen('\disk_sys\project\File1', 'r');
```

##### **Öffnen im Modus „w“**

```
OpenFile2 := SysFileOpen('\disk_mmc\MOELLER\XC-CPU201-EC512k-8DI-6DO\Project\File2', 'w');
```

##### **Öffnen im Modus „a“**

```
OpenFile3 := SysFileOpen('\disk_usb\MOELLER\XC-CPU201-EC512k-8DI-6DO\Project\File3', 'a');
```

##### **Schließen einer Datei**

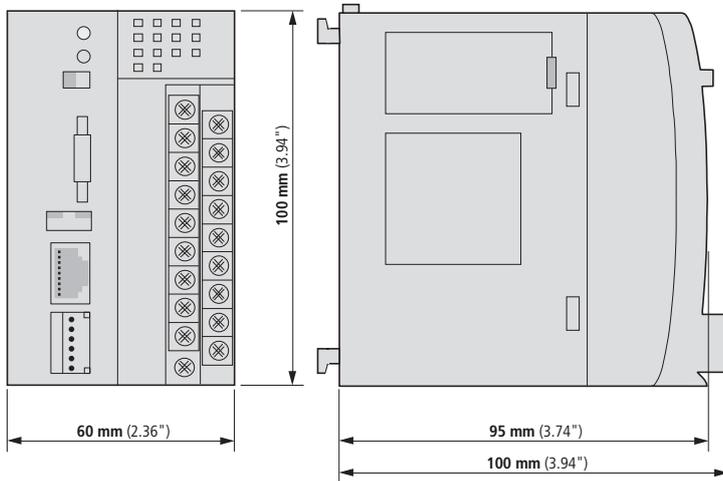
```
CloseFile:=SysFileClose(OpenFile2);  
CloseFile:=SysFileClose(OpenFile3);
```



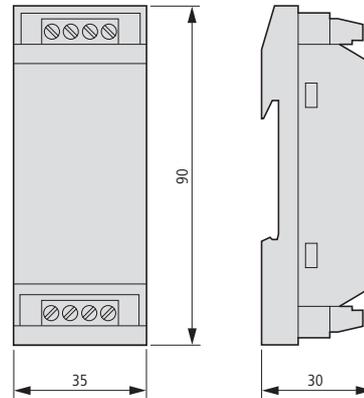
Bei der XC-CPU202 heißt das Verzeichnis `\CONTROL` anstelle von `\MOELLER`.

### 13.5 Abmessungen

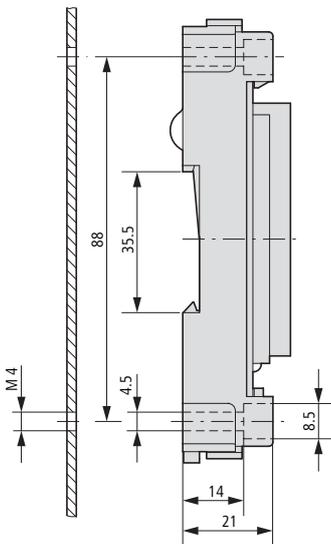
**XC-CPU200**



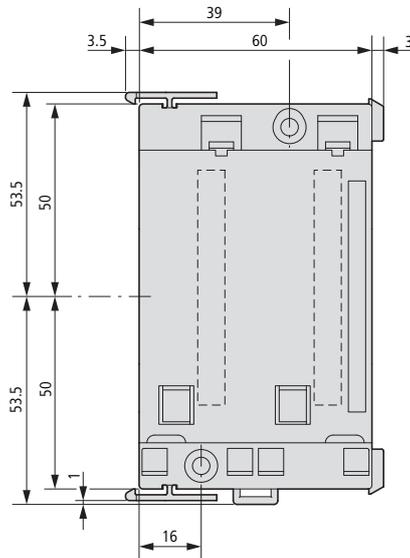
**Netzfilter XT-FIL-1**



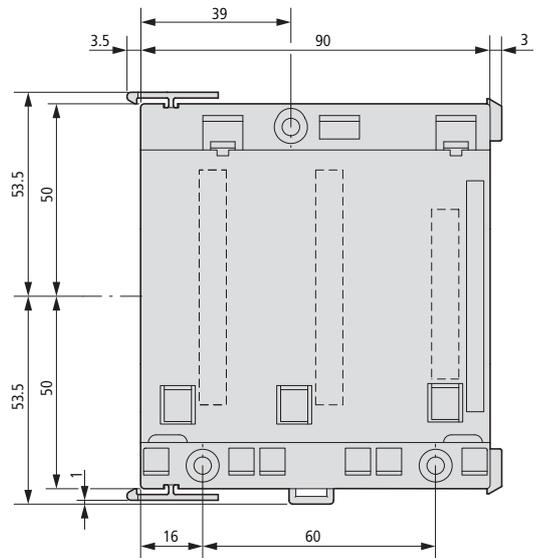
**Baugruppenträger**



**XIOC-BP-XC**



**XIOC-BP-XC1**



## 13 Anhang

### 13.6 Technische Daten

#### 13.6 Technische Daten

			XC-CPU201-EC256-8DI-6DO(-XV), XC-CPU201-EC512-8DI-6DO(-XV) XC-CPU202-EC4M-8DI-6DO-XV
<b>Allgemeines</b>			
Normen und Bestimmungen			IEC/EN 61131-2 EN 50178
Umgebungstemperatur		°C	0 - +55
Lagertemperatur		°C	-25 - +70
Einbaulage			waagrecht
relative Luftfeuchte, keine Betauung (IEC/EN 60068-2-30)		%	10 - 95
Luftdruck (Betrieb)		hPa	795 - 1080
Schwingfestigkeit			5 - 8,4 Hz $\pm$ 3,5 mm 8,4 - 150 Hz $\pm$ 1,0 g
Schockfestigkeit			15 g/11 ms
Überspannungskategorie			II
Verschmutzungsgrad			2
Schutzart			IP20
Bemessungsisolationsspannung		V	500
Störaussendung			EN 61000-6-4, Klasse A
Störfestigkeit			EN 61000-6-2
Batterie (Lebensdauer)			Worst case 3 Jahre, typisch 5 Jahre
Gewicht		kg	0,23
Maße (B x H x T)		mm	90 x 100 x 100
Anschlussklemmen			steckbarer Klemmenblock
Anschlussquerschnitte			
Schraubklemmen			
	feindrätig mit Aderendhülse	mm <sup>2</sup>	0,5 - 1,5
	eindrätig	mm <sup>2</sup>	0,5 - 2,5
Federzugklemmen			
	feindrätig	mm <sup>2</sup>	0,34 - 1,0
	eindrätig	mm <sup>2</sup>	0,14 - 1,0
Elektromagnetische Verträglichkeit (EMV)			→ Seite 148

				XC-CPU201-EC256-8DI-6DO(-XV), XC-CPU201-EC512-8DI-6DO(-XV) XC-CPU202-EC4M-8DI-6DO-XV
<b>Spannungsversorgung der CPU (24 V/0 V)</b>				
Netzausfallüberbrückung				
	Dauer des Einbruchs	ms		10
	Wiederholrate	s		1
Eingangsnennspannung				V DC 24
zulässiger Bereich				V DC 20,4 - 28,8
Stromaufnahme				A typisch 0,85
Restwelligkeit				% ≤ 5
	maximale Verlustleistung (ohne lokale I/O)	P <sub>V</sub> W		6
Überspannungsschutz				ja
Verpolungsschutz				ja
Netzfilter, extern				Typ: XT-FIL-1
Netzfilter, intern				ja
Einschaltstrom				x I <sub>n</sub> keine Begrenzung (Begrenzung nur durch vorgeschaltetes 24-V-DC-Netzteil)
Ausgangsspannung für die Signalmodule				
	Ausgangsnennspannung	V DC		5
	Ausgangsstrom	A		3,2
Leerlauffestigkeit				ja
Kurzschlussfestigkeit				ja
galvanische Trennung zur Versorgungsspannung				nein
<b>CPU</b>				
Mikroprozessor				XC-CPU201...: Risc-Prozessor XC-CPU202...: ARM 532MHz
Speicher				
	Programmcode	kByte		XC-CPU201-EC256...: 512 ab Betriebssystemversion V1.04.01 XC-CPU201-EC512...: 2048 ab Betriebssystemversion V1.04.01 XC-CPU202-EC4M...: 4096
	Programmdaten	kByte		XC-CPU201-EC256...: 256 XC-CPU201-EC512...: 512 XC-CPU202-EC4M...: 512
	Merker (EC256K/EC512K/EC4M)	kByte		16/16716
	Retain-Daten (EC256K/EC512K/EC4M)	kByte		32/32/64
	Persistent-Daten (EC256K/EC512K/EC4M)	kByte		32/32/64
Watchdog				ja
RTC (Real Time Clock)				ja
Schnittstellen				
Multi Media Card oder Secure Digital Card				ja, optional, separat bestellen

## 13 Anhang

### 13.6 Technische Daten

			XC-CPU201-EC256-8DI-6DO(-XV), XC-CPU201-EC512-8DI-6DO(-XV) XC-CPU202-EC4M-8DI-6DO-XV
Ethernet-Schnittstelle			
Datenübertragungsrate	MBit/s		10/100
Anschlussstechnik			RJ45
RS232-Schnittstelle (ohne Handshake-Leitungen)			
Datenübertragungsrate	Bit/s		4800, 9600, 19200, 38400, 57600, 115200
Anschlussstechnik			RJ45
galvanische Trennung			nein
in der Betriebsart „Transparentmodus“			
Datenübertragungsrate	Bit/s		300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200
Zeichenformate			8E1, 8O1, 8N1, 8N2
CAN(open)/easyNet			
Datenübertragungsrate	kBits/s		20/50/100/125/250/500/800/1000
Potentialtrennung			ja
Geräteprofil			nach DS301V4
29 Bit Identifier			XC-CPU201: nein XC-CPU202: ja
PDO-Art			asyn., cyc., acyc.
Anschluss			steckbarer Federzug-Klemmenblock, 6-polig
Busabschlusswiderstände			bei XC-CPU201...: extern; bei XC-CPU202...: schaltbar
Teilnehmer		Anzahl	maximal 126
USB-Schnittstelle, V1.1 bei XC-CPU201..., V2.0 bei XC-CPU202...			
Datenübertragungsrate (Autochanging)	MBit/s		1,5/12
Potentialtrennung			nein
Stromversorgung für angeschlossene Geräte:			
Nennspannung	V DC		5
max. Strom	A		0,5
Anschlussstechnik			Down-Stream-Stecker
Spannungsversorgung der lokalen Ein-/Ausgänge (24 V <sub>Q</sub> /0 V <sub>Q</sub> )			
Nennspannung	V DC		24
Spannungsbereich	V DC		20,4 - 28,8
Stromaufnahme	A		maximal 3 (lastabhängig)
Potentialtrennung			
Spannungsversorgung gegen CPU-Spannung			ja
Überspannungsschutz			ja
Verpolungsschutz			ja

		XC-CPU201-EC256-8DI-6DO(-XV), XC-CPU201-EC512-8DI-6DO(-XV) XC-CPU202-EC4M-8DI-6DO-XV
<b>Digital-Eingänge</b>		
Eingangsnennspannung	V DC	24, Polarität beachten
Spannungsbereich	V DC	19,2 - 30
Eingangsstrom pro Kanal bei Nennspannung		
Funktionalität: Normaler Digital-Eingang	mA	typisch 3,5
Funktionalität: Schneller Digital-Eingang	mA	typisch 7
Verlustleistung pro Kanal		
Funktionalität: Normaler Digital-Eingang	mW	typisch 85
Funktionalität: Schneller Digital-Eingang	mW	typisch 168
Schaltpegel nach EN 61131-2		
Grenzwerte-Typ „1“	V DC	low < 5, high > 15
Eingangsverzögerung		
Funktionalität: Normaler Digital-Eingang		
Aus r Ein	ms	typisch 0,1
Ein r Aus	ms	typisch 0,1
Funktionalität: Schneller Digital-Eingang		
Aus r Ein	ns	typisch 7
Ein r Aus	ns	typisch 1
Eingänge	Anzahl	8
Kanäle mit gleichem Bezugspotential	Anzahl	8
davon nutzbar als		
Interrupt-Eingänge	Anzahl	2
Zählereingang 32 Bit oder	Anzahl	1
Zählereingang 16 Bit oder	Anzahl	2
Inkrementalgeberingang (Spur A, B, C)	Anzahl	1
maximale Eingangsfrequenz	kHz	50
Zustandsanzeige		LED
<b>Digital-Ausgänge</b>		
Verlustleistung pro Kanal		
QX0.0 bis QX0.5	W	0,08
Laststromkreis		
QX0.0 bis QX0.5	A	0,5
Ausgangsverzögerung		
Aus r Ein		typisch 0,1 ms
Ein r Aus		typisch 0,1 ms
Kanäle	Anzahl	6
Kanäle mit gleichem Bezugspotential	Anzahl	6

## 13 Anhang

### 13.6 Technische Daten

			<b>XC-CPU201-EC256-8DI-6DO(-XV), XC-CPU201-EC512-8DI-6DO(-XV) XC-CPU202-EC4M-8DI-6DO-XV</b>
Zustandsanzeige			LED
Einschaltdauer		% ED	100
Gleichzeitigkeitsfaktor		g	1

<b>Elektromagnetische Verträglichkeit</b>			
<b>Störfestigkeit</b>			
ESD (IEC/EN 61000-4-2)	Kontaktentladung		4 kV
	Luftentladung		8 kV
RFI (IEC/EN 61000-4-3)	AM (80 %)	80 – 1000 MHz	10 V/m
Handy (IEC/EN 61000-4-3)	PM	800 – 960 MHz	10 V/m
Burst (IEC/EN 61000-4-4)	Netz/Digital-I/O (direkt)		2 kV
	Analog-I/O, Feldbus (kapazitive Kopplung)		1 kV
Surge (IEC/EN 61000-4-5)	Digital-I/O, unsymmetrisch		0,5 kV
	Analog-I/O, unsymmetrisch, Kopplung auf den Schirm		1 kV
	Netz DC unsymmetrisch		1 kV
	Netz DC symmetrisch		0,5 kV
Leitungsgeführte Störgröße, induziert durch hochfrequente Felder (früher: Einströmung) (IEC/EN 61000-4-6)			3 V

### 13.7 Technische Daten – Filter

		24-V-DC-Filter XT-FIL-1
<b>Allgemeines</b>		
Normen und Bestimmungen		IEC/EN 61131-2 EN 50178
Umgebungstemperatur	°C	0 - +55
Lagerung	°C	-25 - +70
Einbaulage		waagrecht/senkrecht
relative Luftfeuchte, keine Betauung (IEC/EN 60068-2-30)	%	10 bis 95
Luftdruck (Betrieb)	hPa	795 - 1080
Schwingfestigkeit		5 - 8,4 Hz ±3,5 mm 8,4 - 150 Hz ±1,0 g
Schockfestigkeit		15 g/11 ms
Schlagfestigkeit		500 g/∅ 50 mm ±25 g
Überspannungskategorie		II
Verschmutzungsgrad		2
Schutzart		IP20
Bemessungsstoßspannung	V	850
Störaussendung		EN 61000-6-4, Klasse A
Störfestigkeit		EN 61000-6-2
Gewicht	g	95
Maße (B x H x T)	mm	35 x 90 x 30
Anschlussklemmen		Schraubklemme
Anschlussquerschnitte		
Schraubklemmen		
feindrätig mit Aderendhülse	mm <sup>2</sup>	0,2 - 2,5 (AWG22-12)
eindrätig	mm <sup>2</sup>	0,2 - 2,5 (AWG22-12)
<b>Spannungsversorgung</b>		
Eingangsspannung	V DC	24
zulässiger Bereich	V DC	20,4 - 28,8
Restwelligkeit	%	≤ 5
Überspannungsschutz		ja
Potentialtrennung		
Eingangsspannung gegen PE		ja
Eingangsspannung gegen Ausgangsspannung		nein
Ausgangsspannung gegen PE		ja
Ausgangsspannung	V DC	24
Ausgangsstrom	A	2,2

## 13 Anhang

### 13.7 Technische Daten – Filter

# Stichwortverzeichnis

## A

Abmessungen	143
Abschaltlogik	12
Adressierung	
Ein-/Ausgänge und Merker	68
Steuerung am CAN-Bus	83
Anschluss	
Ein-/Ausgänge	25
Inkrementalgeber	26
Interrupt-Aktoren	27
Netzteil und lokale Ein-/Ausgänge	13
PC	28
Spannungsversorgung	25
Vor-/Rückwärtszähler	27
Anwenderprogramm, Speicherwerte	11
Anwenderspeicher, Größe	11
Arbeitsspeicher	16
Ausgänge	
adressieren	68
konfigurieren und parametrieren	89
Signalzustand	13
Verdrahtungsbeispiel	25

## B

Batterie	14
Baudrate	83
festlegen/ändern	73
Belüftung	22
Betrieb	31
Betriebssystem	
aktualisieren	43
Betriebszustände CPU (LED-Anzeige)	36, 65
Bibliothek	141
installieren	102
Blitzschutzmaßnahmen	24
Blockgröße, für Datenübertragung	81
Bootprojekt	41
löschen, auf SD/MMC	42
löschen, auf USB-Stick	42
Breakpoint	37
Browser-Befehle	130

## C

CAN	
-Deviceparameter	85
-Master-RoutingEinstellungen	83
-Schnittstelle	17
-Schnittstelle, Belegung	30
-Telegramme, aus Anwenderprogramm senden/empfangen	18
CANopen-Kabel, Eigenschaften	138
CAN-Stack	57
CODESYS-Gateway-Server	82
COLDSTART (Startverhalten)	34
ComProgramming	115
Counter	95
CPU	
Anzeige der Betriebszustände	36
Auslastung	133
Funktionsbereiche	11

## D

Datensicherung	14
Datenübertragung, Blockgröße	81
Debugging	36, 130
Dekrementieren	20, 27
Diagnose	71, 130
über CAN	85
Dialogsprache, für Fehler- und Ereignislisten	135
Direkter Peripheriezugriff	60
Fehlercode	65
Dokumentationen, online	9
Download von Programmen	39

## E

easyNet-Schnittstelle	17, 30
Echtzeituhr	14
Einbaulage	23
Eingänge	
adressieren	68
konfigurieren und parametrieren	89
Signalzustand	13
Verdrahtungsbeispiel	25
Einzelschritt-Betrieb	37
Einzelzyklus-Betrieb	37
Elektromagnetische Beeinflussung	22
Ereignisgesteuerte Task	52
Ereignisliste	135
Ethernet-Kabel, Eigenschaften	137

Ethernet-Schnittstelle .....	16	Funktionen .....	102
<b>F</b>		CAN_BUSLOAD .....	103
Fehlercode, bei direktem Peripheriezugriff .....	65	DisableInterrupt .....	98
Fehlerliste .....	135	EnableInterrupt .....	98
Flash .....	16	GetSlotPtr .....	65
Forcen .....	37	IEC_DeleteErrorList .....	104
		IEC_DeleteEventList .....	104
		IEC_GetErrorID .....	105
		IEC_GetEventID .....	105
		IEC_GetNrOfErrors .....	105
		IEC_GetNrOfEvents .....	106
		IEC_WriteError .....	106
		IEC_WriteEvent .....	107
		Read...Direct .....	62
		SysFile... .....	142
		UIP_AddUserIPAddress .....	126
		UIP_DeleteUserIPAddress .....	127
		UIP_GetFirstuserIPAddress .....	127
		UIP_GetNextUserIPAddress .....	128
		UIP_GetUserIPAddressCount .....	128
		UIP_MakeUserIP .....	129
		UT12_CopyProjectToMmc .....	122
		UT12_CopyProjectToUsb .....	122
		UT12_CreateStartupIni .....	123
		UT12_DisableComProgramming .....	115
		UT12_EnableComProgramming .....	115
		UT12_EnableDHCP .....	116
		UT12_GetComConfig .....	115
		UT12_GetIPConfig .....	108, 117
		UT12_GetIPDns .....	117
		UT12_GetIPWins .....	118
		UT12_GetKeepAliveTime .....	118
		UT12_GetMacAddress .....	110
		UT12_GetMacAddress .....	119
		UT12_GetPlcVersionList .....	123
		UT12_IsDHCPEnabled .....	119
		UT12_Reboot .....	112, 124
		UT12_RemoveProjectFromMmc .....	124
		UT12_RemoveProjectFromUsb .....	125
		UT12_RemoveStartupIni .....	125
		UT12_SaveRegistry .....	113, 126
		UT12_SetComConfig .....	116
		UT12_SetIPConfig .....	110, 120
		UT12_SetIPDns .....	120
		UT12_SetIPGateway .....	111, 121
		UT12_SetIPWins .....	121
		UT12_SetKeepAliveTime .....	122
		Write...Direct .....	63
		Funktionsbausteine .....	102
		Funktionsbereiche, CPU .....	11



Secure Digital Card .....	15	Unterbrechung, CAN-Bus .....	18
Segmente .....	66	USB-Schnittstelle, Belegung .....	29
Signalzustand Ein-/Ausgänge .....	13	USB-Stick .....	15, 16
Spannungseinbruch .....	12	USER_IP .....	126
Spannungsversorgung		<b>V</b>	
anschließen .....	25	Verbindungsaufbau .....	72
ausschalten/unterbrechen (Verhalten) .....	36	Verdrahtung .....	23
für Prozessoreinheit u. lokale Ein-/Ausgänge ..	12	Verdrahtungsbeispiel	
Speicher		Ein-/Ausgänge .....	25
Applikationsprogramm .....	11	Netzteil .....	25
-karte .....	141	Vor-/Rückwärtszähler .....	19
-nutzung, Grenzwerte .....	66	Vorwärtszähler .....	95
-systeme .....	16	anschließen .....	27
Sprachumstellung, Fehler- und Ereignisliste		<b>W</b>	
(Browser-Befehle) .....	135	WARMSTART (Startverhalten) .....	34
Startup.ini-Datei .....	77	Watchdog .....	57
Startverhalten .....	31	Webserver .....	11
einstellen in der CoDeSys .....	34	Web-Visualisierung .....	66
Störeinflüsse .....	22	<b>X</b>	
Subnetmask-Adresse .....	108	XIOC-Module .....	10
SysLibRTC.pdf .....	14	<b>Z</b>	
System		Zähler .....	95
-auslastung, CPU .....	133	Zustandsanzeige .....	37
-bibliotheken .....	102	Zyklische Task .....	51
-ereignisse .....	50		
-parameter vorgeben (in Datei Startup.ini) ..	77		
-speicher .....	16		
-zeiten .....	50		
Systemereignisse .....	53		
Systemspeicher			
disk_mmc .....	15		
disk_sys .....	15		
disk_usb .....	15		
<b>T</b>			
Task			
-bedingung .....	50		
-konfiguration .....	50		
-überwachung .....	57		
zyklische .....	51		
TCP/IP-Verbindung (beim Routing) .....	81		
Technische Daten .....	144		
Test und Inbetriebnahme .....	36		
Textausgabe über die RS-232-Schnittstelle ...	139		
Transparent-Modus .....	87, 139		
Typ (Taskbedingung) .....	50		
<b>U</b>			
Überwachungszeit, Task .....	57		