

Modular PLC XC-CPU101...(-XV)



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

Störfallservice

Bitte rufen Sie Ihre lokale Vertretung an:

<http://www.eaton.com/moeller/aftersales>

oder

Hotline After Sales Service:

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

AfterSalesEGBonn@eaton.com

Originalbetriebsanleitung

Die deutsche Ausführung dieses Dokuments ist die Originalbetriebsanleitung.

Übersetzung der Originalbetriebsanleitung

Alle nicht deutschen Sprachausgaben dieses Dokuments sind Übersetzungen der Originalbetriebsanleitung.

1. Auflage 2002, Redaktionsdatum 06/02
 2. Auflage 2002, Redaktionsdatum 10/02
 3. Auflage 2003, Redaktionsdatum 04/03
 4. Auflage 2003, Redaktionsdatum 08/03
 5. Auflage 2003, Redaktionsdatum 11/03
 6. Auflage 2003, Redaktionsdatum 12/03
 7. Auflage 2004, Redaktionsdatum 04/04
 8. Auflage 2004, Redaktionsdatum 06/04
 9. Auflage 2004, Redaktionsdatum 11/04
 10. Auflage 2005, Redaktionsdatum 03/05
 11. Auflage 2008, Redaktionsdatum 01/08
 12. Auflage 2010, Redaktionsdatum 10/10
- Änderungen siehe Abschnitt „Änderungsprotokoll“

© Eaton Industries GmbH, 53105 Bonn

Autor: Werner Albrecht, Peter Roersch

Redaktion: Thomas Kracht, Barbara Petrick

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) sind zu beachten.
- Nur entsprechend qualifiziertes Personal gemäß EN 50 110-1/-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 Schutzerde (PE) oder den Potentialausgleich 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 60 364-4-41 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 60 204-1 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. Ggf. 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.).

Inhalt

Zu diesem Handbuch	Änderungsprotokoll	5
	Abkürzungen und Symbole	6
	Weiterführende Dokumentationen	6
1 Aufbau der XC100		7
	CPU mit Netzteil und lokalen Ein-/Ausgängen	7
	24-V-Netzteil mit lokalen Ein-/Ausgängen	7
	– Aufgabe	7
	– Aufbau	8
	– Interrupt-Eingänge anschließen	9
	– Lokale Buserweiterung mit XIOC-BP-EXT	10
	CPU	10
	– Aufgabe	10
	– Einsatzmöglichkeiten der CPU-Typen	10
	– Aufbau	11
	– LED-Zustandsanzeige	11
	– Betriebsarten-Vorwahlschalter	11
	– Multi Media Card (MMC)/Memory-Card	11
	– Programmiergeräte-Schnittstelle	12
	– CANopen-Schnittstelle	14
	– Echtzeituhr	15
	– XC-CPU101-...-XV	15
	– Batterie	15
	Montage der CPU	16
	Demontage der CPU	16
2 Projektierung		17
	Schaltschrankaufbau	17
	– Belüftung	17
	– Geräteanordnung	17
	Störungen vermeiden	17
	– Schutzbeschaltung von Störquellen	17
	– Schirmung	17
	Blitzschutzmaßnahmen	18
	Verdrahtungsbeispiele	18
	– Netzteil	18
	– Spannungsversorgung der Digital-Ein-/Ausgänge	18
3 CPU betreiben		19
	Einschaltverhalten	19
	Ausschaltverhalten	20
	Start-Verhalten	20
	Stopp-Verhalten	20
	Kaltstart (Coldstart)	20
	Warmstart	20
	Test und Inbetriebnahme	20
	– Breakpoint/Einzelschritt-Betrieb	20
	– Einzelzyklus-Betrieb	20
	– Forcen	20
	– Zustandsanzeige der easySoft-CoDeSys	20

Programm-Reset	21
– Reset-Warm	21
– Reset-Kalt	21
– Reset-Ursprung	21
Programm-Parametrierung	21
– Maximale Zykluszeit des Programms	21
– Startverhalten nach PowerOn	21
Bootprojekt erzeugen und transferieren	21
Bootprojekt erzeugen nach Online-Änderung	21
Betriebssystem (BTS) aktualisieren	22
– Betriebssystem vom PC in die Steuerung übertragen	22
– Betriebssystem vom PC auf die MMC übertragen	23
– Betriebssystem von der MMC in die Steuerung übertragen	23
– Update von weiteren XC100-Steuerungen	23

4 Programmbearbeitung und Systemzeit

	25
Zykluszeitüberwachung	25
Systembibliotheken, Funktionsbausteine und Funktionen	25
– Bibliotheksverwalter	25
Zielsystemspezifische Bibliotheken	26
– Lib_Common	26
– Bibliotheken der „Lib_CPU101“	27
Direkter Peripheriezugriff	29
– Funktionen	30
– Fehlercode bei „Direkter Peripheriezugriff“	33
Interrupt-Verarbeitung	34
Interrupt-Priorisierung	34
– Timer-Interrupt	35
– DisableInterrupt	36
– EnableInterrupt	36
– Interrupt-Funktion erstellen und einbinden	37
Systemereignisse	39
Browserbefehle	40
– Browserbefehl „canload“	40
Datenremanenz	41
Programmtransfer	41
Betriebszustände	41
Grenzwerte für die Speichernutzung	42
Ein-/Ausgänge und Merker adressieren	43
– „Adressen automatisch“ aktivieren	43
– „Adressüberschneidungen prüfen“ aktivieren	43
– Ungerade Wort-Adressen	43
– Adressbereich	43
– Adressen von Eingabe-/Ausgabe-Module und Diagnoseadresse frei vergeben oder verändern	44
– „Adressen berechnen“ durchführen	44
Diagnose	44

5 Verbindungsaufbau PC – XC100

	45
Verbindungsaufbau über RS232-Schnittstelle (XC100)	45
– Programmierkabel	45
– Software easySoft-CoDeSys	45

6 Beispielprojekt erstellen		47
	Aufgabe	47
	Vorgehensweise	47
	– Zielsystem einstellen	47
	– XC100-Steuerung konfigurieren	50
	– Programm erstellen	54
7 Programmieren über CANopen-Netzwerk (Routing)		55
	Voraussetzungen	55
	Hinweise	56
	Adressierung	56
	Kommunikation mit der Zielsteuerung	57
	Steuerungskombinationen zum Routing	58
	Anzahl der Kommunikationskanäle	58
8 RS232-Schnittstelle im Transparent-Modus (COM 1/2/3)		59
	Anforderungen an die Funktionalität des Transparent-Modus	60
	– Funktion „SysComOpen“	60
	– Funktion „SysComClose“	63
	– Funktion „SysComRead“	64
	– Funktion „SysComWrite“	65
	– Funktion „SysComSetSettings“	66
	– Funktion „SysComReadControl“	68
	– Funktion „SysComWriteControl“	69
	– Automatisches Schließen der Schnittstelle	69
Anhang		71
	Kompatibilität	71
	Abmessungen	72
	– XC-CPU101...	72
	– Netzfilter XT-FIL-1	72
	– Baugruppenträger	72
	Technische Daten	73
Stichwortverzeichnis		77

Zu diesem Handbuch

Änderungsprotokoll

Redaktionsdatum	Seite	Stichwort	Neu	Änderung	Entfällt
10/02	70	„Externer Filter: Falls erforderlich“			✓
04/03	20	„Warmstart“		✓	
	29	„Direkter Peripheriezugriff“	✓		
	22	„Betriebssystem (BTS) aktualisieren“	✓		
	34	„Interrupt-Verarbeitung“	✓		
	11	„Datenzugriff auf die Multi Media Card“	✓		
	40	„Browserbefehle“	✓		
	51	„Routing“	✓		
08/03	allg.	Baudrate von 57600 in 38400 geändert		✓	
	20	„Zustandsanzeige der easySoft-CoDeSys“	✓		
	32	Meldung „Kommunikation abgebrochen“	✓		
	39	„Systemereignisse“	✓		
	45	„Kommunikationsfehler(#0): Es wird ausgeloggt“	✓		
	69	Batterie-Lebensdauer		✓	
08/03 (Nachdruck)	10, 69	XC-CPU-101-C256k-8DI-6DO (-XV)	✓		
12/03		komplette Überarbeitung			
12/03 (Nachdruck)	41	„Datenremanenz“, 1. Absatz		✓	
04/04	42	„Grenzwerte für die Speichernutzung“	✓		
06/04	18, 68, 72	„Externer 24-V-DC-Netzfilter für Spannungsversorgung der XC100“	✓	✓	
11/04	11	MMC		✓	
	19	„Einschaltverhalten“		✓	
	21	„Reset-Ursprung“		✓	
	21	„Bootprojekt erzeugen und transferieren“		✓	
	22	„Betriebssystem (BTS) aktualisieren“		✓	
03/05	42	„Segmentgröße der XC-CPU101-C256k“	✓		
	43	„Ein-/Ausgänge und Merker adressieren“	✓		
	44	„Diagnose“	✓		
	55	„Programmieren über CANopen-Netzwerk (Routing)“		✓	
01/08	21	„Bootprojekt erzeugen nach Online-Änderung“	✓		
	41	„Datenremanenz“		✓	
	55	„Programmieren über CANopen-Netzwerk (Routing)“		✓	
10/10	allg.	Umstellung auf Eaton-Bezeichnungen	✓		

Abkürzungen und Symbole

In diesem Handbuch werden Abkürzungen und Symbole eingesetzt, die folgende Bedeutung haben:

MWS	Menüwahlschalter
BAS	Betriebsartenschalter
CPU	Zentraleinheit
CRC	Cyclic Redundancy Check
MMC	Multi Media Card
E/A	Ein-/Ausgänge

► zeigt Handlungsanweisungen an

Wählen Sie «Datei → Neu» bedeutet: Aktivieren Sie den Befehl „Neu“ im Menü „Datei“.



Achtung!

warnet vor leichten Sachschäden.



Vorsicht!

warnet vor schweren Sachschäden und leichten Verletzungen.



Warnung!

warnet vor schweren Sachschäden und schweren Verletzungen oder Tod.

Für eine gute Übersichtlichkeit finden Sie auf den linken Seiten im Kopf die Kapitelüberschrift und auf den rechten Seiten den aktuellen Abschnitt, Ausnahmen sind Kapitelanfangsseiten und leere Seiten am Kapitelende.

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. AWB2700-1437D) angegeben.

Alle Handbücher stehen als PDF-Datei zur Verfügung. Sofern sie nicht auf der Produkt-CD mitgeliefert wurden, stehen sie im Internet als PDF-Datei zum Download zur Verfügung. Für ein schnelles Auffinden geben Sie unter der Adresse <http://www.eaton.com/moeller> → **Support** als Suchbegriff die Dokumentationsnummer ein.

1 Aufbau der XC100

Die Steuerungen XC-CPU101-... – im Folgenden kurz XC100 genannt – sind für den Einsatz in Maschinen und Anlagensteuerungen konzipiert. Mit ihren Schnittstellen zum Anschluss eines Programmiergerätes (RS232) und zur Ankopplung von dezentralen CANopen-Erweiterungen bilden diese Steuerungen die Basis zum Aufbau eines umfassenden Automatisierungssystems.

Die Steuerung XC100 ist kompakt aufgebaut und kann zentral und/oder dezentral erweitert werden. Das Grundgerät besteht aus:

- Baugruppenträger,
- Steuerungs- oder Visualisierungs-CPU mit integriertem Netzteil und lokalen Ein-/Ausgängen,
- XIOC-Signalmodule.

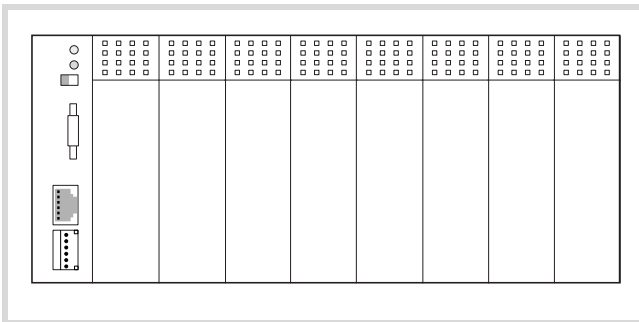


Abbildung 1: Aufbau der XC-CPU101 mit XIOC-Modulen

→ Nähere Einzelheiten zur CPU finden Sie im anschließenden Abschnitt.

Ausführliche Hinweise zu den Baugruppenträgern und XIOC-Modulen finden Sie im Handbuch „Hardware und Projektierung der XIOC-Signalmodule“. Dieses Handbuch ist auf der CD als PDF-Datei verfügbar (h1452d.pdf).

Die jeweils aktuelle Ausgabe des Handbuchs finden Sie unter <http://www.eaton.com/moeller> → **Support**:

Suchbegriff: AWB2725-1452D

CPU mit Netzteil und lokalen Ein-/Ausgängen

Das kompakt aufgebaute CPU-Modul der XC100 ist in zwei Funktionsbereiche gegliedert:

- Prozessoreinheit mit Schnittstellen
- 24-V-Netzteil mit integrierten Digital-Eingängen (acht) und -Ausgängen (sechs).



Abbildung 2: Aufbau CPU-Modul XC-CPU101

- ① Prozessoreinheit
- ② 24-V-Netzteil mit lokalen Ein-/Ausgängen

24-V-Netzteil mit lokalen Ein-/Ausgängen

Das Netzteil stellt für die Prozessoreinheit und die Ein-/Ausgänge (lokale und zentrale) die notwendigen Spannungen zur Verfügung.

Aufgabe

Das Netzteil wandelt die 24-V-DC-Versorgungsspannung in die für das System erforderlichen Spannungen. Diese Spannungen werden auf den Bus des Basis- und – sofern vorhanden – Erweiterungsbaugruppenträgers gelegt.

Die Besonderheit beim Anschluss der 24-V-Versorgungsspannung ist, dass die Versorgung der Prozessoreinheit und der lokalen Ein-/Ausgänge getrennt vorgenommen wird. 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).

Aufbau

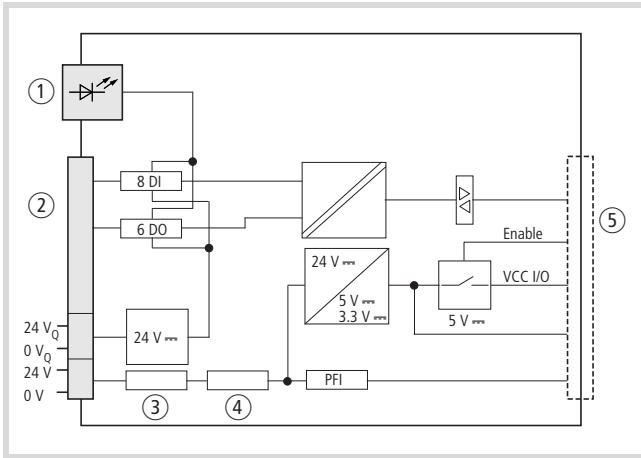


Abbildung 3: Blockschaltbild Spannungsversorgung

- ① Zustandsanzeige E/As
 - ② Front-Anschlussklemmen
 - ③ interner Filter
 - ④ Puffer
 - ⑤ XIOC I/O-Bus, Baugruppenträger
- PFI = Power Fail Interrupt

Der Spannungsanschluss 0V_Q/24V_Q dient alleine der Versorgungsspannung der integrierten lokalen 8 Ein- und 6 Ausgänge und ist zum Bus hin potentialgetrennt.

Der Spannungsanschluss 0V/24V wird intern gefiltert und gepuffert und einem Spannungswandler zugeführt, der die erforderlichen Systemspannungen erzeugt. Das interne Netzteil für die 5-V-Systemspannung ist so konzipiert, dass die Prozessoreinheit mit dem erforderlichen Strom versorgt werden.

⚠ Vorsicht!
 Beim Einsatz der XC100-CPU und der XIOC-Signal-Module aus ABS-Gehäuse-Material gelten die Einschränkungen aus Tabelle 1. ABS-Gehäuse sind auf der Grundfläche, die dem Rückwandbus zugewandt ist, mit „ABS“ gekennzeichnet.

Tabelle 1: Einschränkungen beim Einsatz der XC100-CPU und der XIOC-Signal-Module aus ABS-Gehäuse-Material

eingebaut in:	Innentemperatur Einbaulort:	Strombelastbarkeit der 5-V-Systemspannung des E/A-Busses
CI-Gehäuse	> 40 °C	Einsatz der XC100 nicht zulässig
	0 bis 40 °C	max. 1,5 A ¹
Sicherungsverteiler	0 bis 55 °C	max. 1,5 A ¹
Schaltschrank	> 40 °C	max. 1,5 A ¹
	0 bis 40 °C	max. 3,2 A

1) bei den Ausgängen der CPU aus ABS-Gehäuse-Material gilt ein Gleichzeitigkeits-Faktor g von 0,5

➔ Leistungseinschränkungen für die digitalen E/A-Module aus ABS-Gehäuse-Material sind in der Dokumentation der XIOC-Signalmodule (MN05002002Z-DE; frühere Bezeichnung AWB2725-1452D) beschrieben.

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. Dieser Ablauf wird initiiert durch das PFI-Signal und erwirkt über die CPU die Abschaltung.

Lokale Digital-Eingänge

In der rechten Hälfte der CPU befindet sich hinter der Frontabdeckung der 18-polige Klemmenblock für die Spannungsversorgung der CPU und der lokalen E/As und der physikalische Anschluss der lokalen Ein-/Ausgänge.

Die acht Digital-Eingänge und sechs Halbleiter-Ausgänge sind für 24-V-Signale ausgelegt und haben eine gemeinsame Spannungsversorgung 0V_Q/24V_Q, die zum Bus hin potentialgetrennt ist.

Lokale Digital-Ausgänge

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

⚠ Achtung!
 Bitte beachten Sie die Leistungseinschränkungen der Ausgänge bei ABS-Gehäuse-Material, ➔ Tabelle 1.

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

Klemmenbelegung

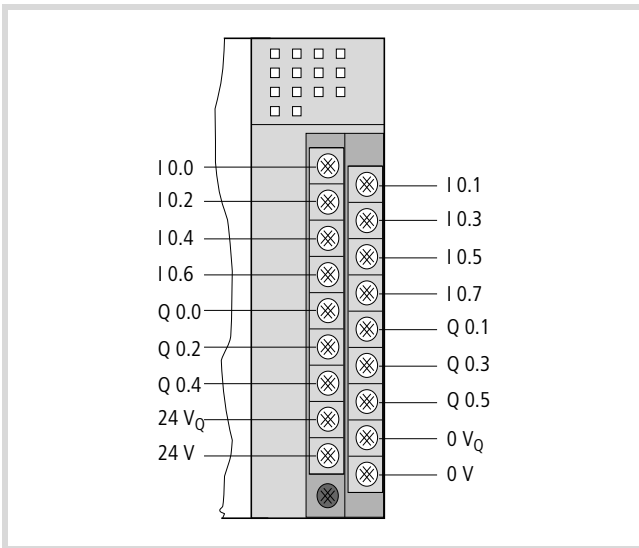


Abbildung 4: Anschlüsse Netzteil und lokales I/O

- I0.0 bis I0.7: lokale Digital-Eingänge
- Q0.0 bis Q0.5: lokale Digital-Ausgänge
- 0V_Q/+24V_Q: Versorgungsspannung der lokalen Ein-/Ausgänge
- 0V/+24V: Versorgungsspannung der Prozessoreinheit

LED-Anzeigen

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

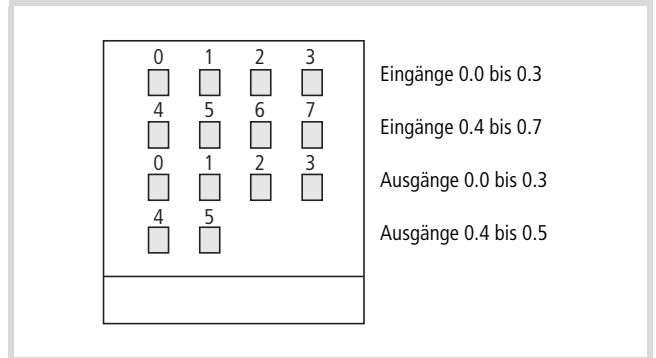


Abbildung 5: LEDs der integrierten Ein-/Ausgänge

Die oberen beiden LED-Zeilen zeigen den Signalzustand der acht Digital-Eingänge des CPU-Moduls (I0.0 bis I0.7), die unteren beiden LED-Zeilen zeigen den Signalzustand der sechs Digital-Ausgänge (Q0.0 bis Q0.5) an.

Interrupt-Eingänge anschließen

Die Eingänge I0.0, I0.1, I0.2, I0.3 können Sie als Interrupt-Eingänge nutzen.

Ausgewertet wird die L/H-Flanke. Die Interrupt-Eingänge wirken unmittelbar und unabhängig von der Zykluszeit der Applikation und starten die programmierte Interrupt-Routinen. Der bis zum Eintreffen des Interrupt-Signals bearbeitete Programmteil wird unverzüglich unterbrochen. Alle weiteren Interrupt-Abläufe sind applikationsbezogen zu programmieren.

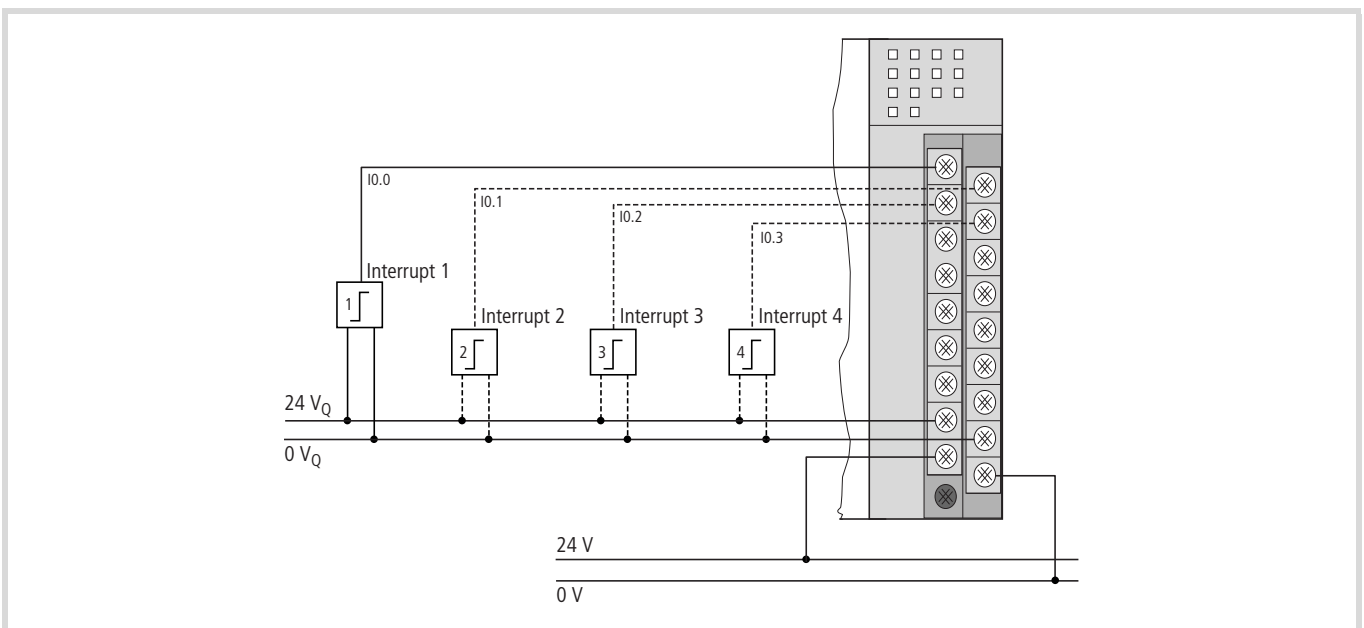


Abbildung 6: Anschluss Interrupt-Eingänge

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

Lokale Buserweiterung mit XIOC-BP-EXT

Der Baugruppenträger XIOC-BP-EXT ermöglicht die Erweiterung des lokalen Systembusses von max. 7 auf max. 15 Steckplätze.

Die intelligenten Module, wie z. B. Netzwerk- und Gateway-Module, können nur auf den E/A-Steckplätze 1 bis 3 gesteckt werden. Der Steckplatz der übrigen Module ist beliebig.

Die mögliche Anordnung der Baugruppenträger ist in der Dokumentation der XIOC-Signalmodule (MN05002002Z-DE) beschrieben. Bitte beachten Sie die Strombilanz zwischen dem Strom, der durch das Netzteil zur Verfügung gestellt wird, und der Stromentnahme durch die Signalmodule.

Weitere Informationen finden Sie in der Dokumentation „XIOC-Signalmodule“ (MN05002002Z-DE). Wie Sie die Buserweiterung über die Software einbinden, ist im Abschnitt „Erweiterung des XIOC-Bus“ beschrieben.

CPU

Die CPUs XC-CPU101...(-XV) basieren auf einem Prozessor mit integrierter CAN-Schnittstelle und verfügen über einen Flash- und batteriegepufferten SRAM-Speicher. Die CAN-Feldbusanschaltung ist potentialgetrennt. Zur Datensicherung ist der Einsatz einer Batterie notwendig.

Die Überwachung der Systemspannung stellt sicher, dass bei einem Unterschreiten eines fest vorgewählten Spannungslevels die Datensicherungsroutine gestartet wird. Um die erforderliche gespeicherte Energie für die Datensicherungsroutine nicht durch Aktivitäten auf der E/A-Ebene zusätzlich zu belasten, wird die 5-V-Systemspannung für die E/A-Module abgeschaltet.

Die interne Echtzeituhr erlaubt zeit- und datumsabhängige Steuerungsfunktionen.

Als Bedien- und Schnittstellenelemente stehen zur Verfügung:

- LED-Anzeige für RUN/Stop und Summenfehler
- Betriebsarten-Vorwahlschalter RUN/Stop
- RS232-Schnittstelle, z. B. für Programmiergeräteanschaltung
- CANopen-Schnittstelle als Feldbusschnittstelle
- Schnittstelle für den Einsatz eines Multi-Media-Speichermoduls (MMC).

Die CPUs der Steuerungen XC100 stehen in unterschiedlichen Versionen zur Verfügung:

- XC-CPU101-C64K-8DI-6DO (-XV)
- XC-CPU101-C128K-8DI-6DO (-XV)
- XC-CPU101-C256K-8DI-6DO (-XV).

C64K, C128K und C256K sind ein Maß für die Größe des Anwenderspeichers.

„XV“ kennzeichnet eine Visualisierungs-CPU und erlaubt die Ansteuerung und die direkte Ankopplung an ein Text-Display (XV-101).

Für die Größe des Applikationsprogramms gelten folgende Speicherwerte:

	XC-CPU101...(-XV)		
	C64K-8DI-6DO	C128K-8DI-6DO	C256K-8DI-6DO
Programmcode	64 kByte	128 kByte	256 kByte
Programmdaten, davon:	64 kByte	128 kByte	256 kByte
Merker	4 kByte	8 kByte	16 kByte
Retaindaten	4 kByte	8 kByte	16 kByte

Die XC-CPU...-XV haben zusätzlich 64 kByte Flashspeicher für Texte

Aufgabe

Die Aufgabe der CPU ist es, aus den eingehenden lokalen und zentralen/dezentralen Signalen entsprechend dem Anwenderprogramm die Ausgangssignale zu generieren.

Ein-/Ausgangssignale können z. B. sein:

- digitale oder analoge Signale
- Befehle aus dem Text-Display¹⁾
- Ausgaben über das Text-Display¹⁾
- Verbindung zum Programmiersystem
- Verbindung zur CANopen-Busschnittstelle
- Verbindung zu Feldbusmodulen, soweit vorhanden
- Verbindung zu intelligenten Signalmodulen, soweit vorhanden.

1) Nur bei XC-CPU...-XV

Einsatzmöglichkeiten der CPU-Typen

CPU-Typen	XC100	Text-Display XV-101-...	
		K42	K84
XC-CPU101...	✓	–	–
XC-CPU101...(-XV)	✓	✓	✓

Aufbau

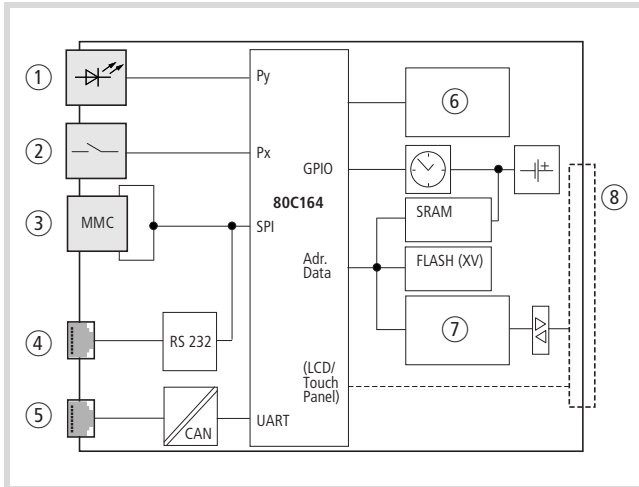


Abbildung 7: Blockschaltbild der XC-CPU101

- ① Zustandsanzeige RUN, Stop, SF
- ② Betriebsarten-Vorwahlschalter
- ③ Multi Media Card
- ④ Programmiergeräte-Schnittstelle: RS232 bei XC-CPU101
- ⑤ CANopen-Feldbusschnittstelle
- ⑥ Spannungsüberwachung
- ⑦ I/O-Businterface
- ⑧ XIOC-I/O-Bus (Baugruppenträger)

LED-Zustandsanzeige

→ Kapitel „Betriebszustände“ auf Seite 41.

Betriebsarten-Vorwahlschalter

Die Betriebsarten „Stop“ und „Run“ wählen Sie über einen Kippschalter in der Front der CPU-Baugruppe. Beachten Sie, dass die Stellung des Betriebsarten-Vorwahlschalters das Verhalten der CPU festlegt. Die Wirksamkeit der Software-Voreinstellungen ist von der Stellung des Betriebsarten-Vorwahlschalters abhängig. Wird im laufenden Betriebszustand „Run“ der Vorwahlschalter in die Stellung „Stop“ gebracht, wechselt die CPU am Ende des laufenden Zyklus vom Betriebszustand „Run“ in den Zustand „Stop“. Generell wird die Schalterstellung des Betriebsarten-Vorwahlschalters zum Ende eines jeden Zyklus abgefragt und in den vorgewählten Zustand gewechselt, → Kapitel „CPU betreiben“.

Multi Media Card (MMC)/Memory-Card

Die Multi Media Card dient als optionales Backup-Medium für das (Boot-)Projekt und zur Ablage von Rezepturdaten. Das Betriebssystem unterstützt Speichergrößen bis max. 128 MByte. Eaton bietet zurzeit die MMCs in den Speichergrößen 16 und 32 MByte mit den Typenbezeichnungen XT-MEM-MM16M und XT-MEM-MM32M an. Zum Beschreiben der Multi Media Card

stecken Sie diese in den Schacht MEM CARD in der CPU. Mit dem Kommando „Bootprojekt erzeugen“ können Sie das Projekt auf die MMC transferieren.

→ Ab der Betriebssystem(BTS)-Version 03.03 besteht die Möglichkeit, das BTS auf die Speicherkarte zu transferieren, und es von dort in andere Steuerungen zu übertragen, → Abschnitt „Betriebssystem (BTS) aktualisieren“ auf Seite 22.

Löschfunktionen

Um den gesamten Inhalt der MMC zu löschen, benutzen Sie den Browserbefehl „Format“. Mit dem Befehl „Reset (Ursprung)“ löschen Sie das Bootprojekt und das Betriebssystem auf der MMC.

Datenzugriff auf die Multi Media Card

Die Library „XC100_File“ ist in der „Lib_CPU101“ enthalten. Sie stellt die Bausteine für den Zugriff auf die MMC zur Verfügung. Hierzu ist es erforderlich, die entsprechende Library in den „Bibliotheksverwalter“ hinzuzufügen:

- ▶ Wechseln Sie in den Bibliotheksverwalter und positionieren Sie den Mauszeiger im Feld für die Libraries. Drücken Sie dann die rechte Maustaste.
- ▶ In dem neu geöffnetem Informationsfenster wählen Sie den Befehl „Weitere Bibliothek einbinden“.
- ▶ Wählen Sie die Library „Lib_CPU101“ und anschließend die Datei „XC100_File“ aus. Öffnen Sie diese Datei.

Mit dem Befehl „Öffnen“ wird der Baustein in den Bibliotheksverwalter eingebunden. Nachfolgende Funktionen stehen Ihnen jetzt zur Verfügung:

- FileClose
- FileDelete
- FileGetSize
- FileOpen
- FileRead
- FileRename
- FileSetPos
- FileWrite.

Weitere Informationen zu diesem Baustein finden Sie im Abschnitt „Bibliotheken der XC100_File.lib“ und in der Dokumentation „Funktionsbausteine zur easySoft-CoDeSys“ (MN05010002Z-DE).

⚠ Achtung!

- Das „FAT16-Filesystem“ ist nicht transaktionssicher.
- Die Steuerspannung/Steuerung darf bei einem geöffnetem File-Dienst nicht ausgeschaltet werden.
- Ein Spannungsausfall oder ein Abschalten der Versorgungsspannung bei geöffnetem File-Dienst kann zur Zerstörung der Multi Media Card führen.

Programmiergeräte-Schnittstelle

Die CPU ist mit einer RS232-Schnittstelle ausgerüstet. Diese serielle Schnittstelle erlaubt eine Punkt-zu-Punkt-Verbindung. Die Handshake-Leitungen stehen nicht zur Verfügung. Über die RS232-Schnittstelle erfolgt die Kommunikation zwischen der Steuerung und dem Programmiergerät. Die Schnittstelle ist physikalisch eine RJ45-Buchse. Verwenden Sie zur Verbindung der XC100 mit dem PC das Programmierkabel XT-SUB-D/RJ45. Die Schnittstelle ist nicht galvanisch getrennt.

Programmerschnittstelle

RJ45-Buchse		RS232
8	8	RxD
7	7	GND
6	6	–
5	5	TxD
4	4	GND
3	3	–
2	2	–
1	1	–

Datenübertragungsrate ändern

- ▶ Öffnen Sie das Dialogfeld <Ressourcen → Steuerungskonfiguration>.
- ▶ Aktivieren Sie die Registerkarte „Weitere Parameter“.
- ▶ Wählen Sie im Listenfeld „Baudrate“ die gewünschte Datenübertragungsrate aus. Im Beispiel ist dies 38400 Bit/s.

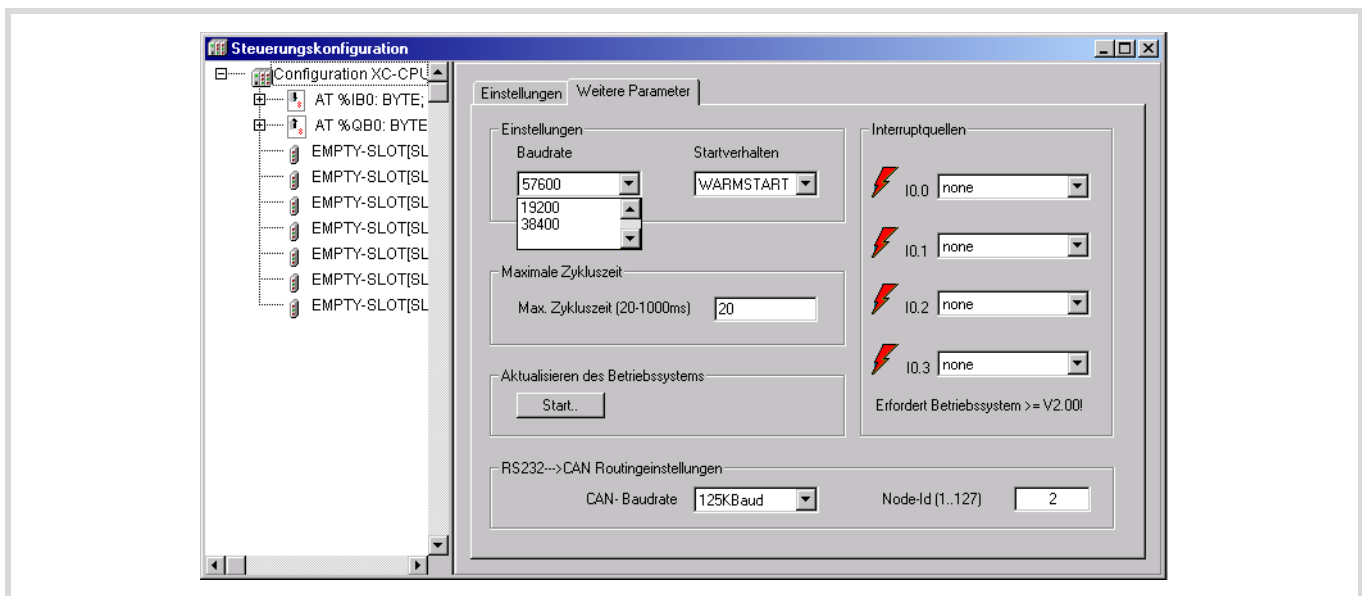


Abbildung 8: Steuerungskonfiguration – „Weitere Parameter“

- ▶ Schließen Sie das Fenster „Weitere Parameter“.
- ▶ Wählen Sie das Menü <Online → Einloggen>.

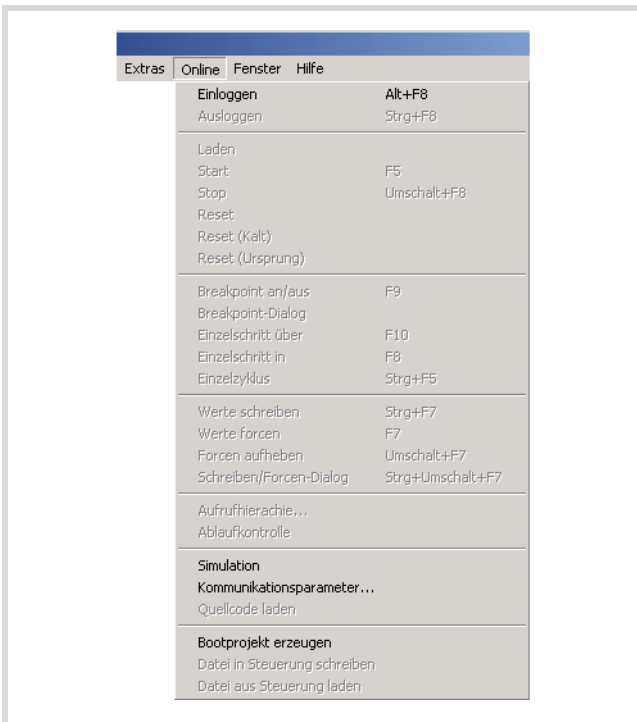


Abbildung 9: Menü „Online“

Sie erhalten folgende Abfrage (→ Abbildung 10):

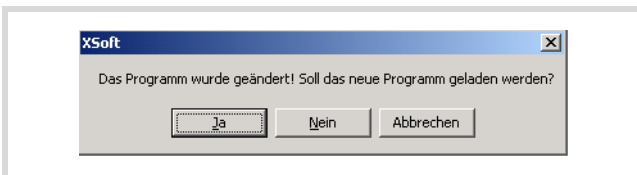


Abbildung 10: Abfrage nach Programmänderung

- ▶ Wählen Sie das Feld mit der Voreinstellung der Baudrate durch einen Doppelklick an.

Das Feld wird grau unterlegt.

- ▶ Mit weiteren Doppelklicks in diesem Feld wählen Sie die gewünschte Baudrate, z. B. 38400 Bit/s, aus. Bestätigen Sie diese mit „OK“.
- ▶ Wählen Sie erneut das Menü <Online → Einloggen> aus.

Sie erhalten wieder den folgenden Hinweis:

- ▶ Beantworten Sie diese Abfrage mit „Ja“ und Sie erhalten die Fehlermeldung für Kommunikationsfehler, da die Baudrate zwischen XC100 und easySoft-CoDeSys nicht übereinstimmt. Die folgenden Schritte beschreiben, wie Sie die Baudrate einstellen.

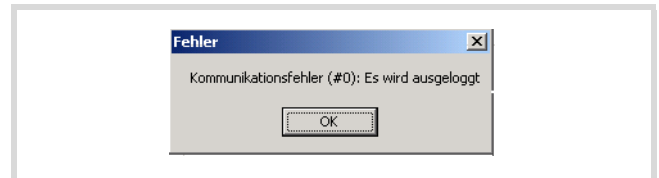


Abbildung 11: Kommunikationsfehler

- ▶ Beantworten Sie diese Fehlermeldung mit „OK“.
- ▶ Wählen Sie das Menü <Online → Kommunikationsparameter> aus (→ Abbildung 9).

Sie erhalten das „Kommunikationsfenster“ wie in der folgenden Abbildung dargestellt.

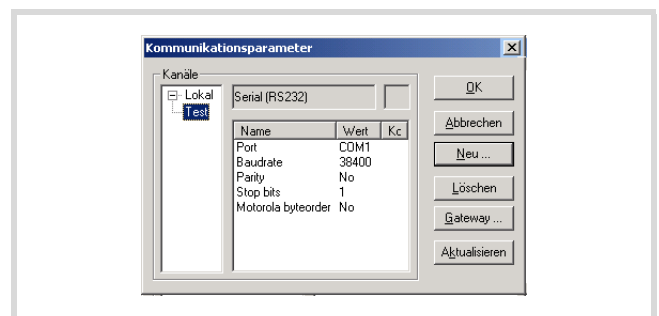


Abbildung 12: Kommunikationsparameter

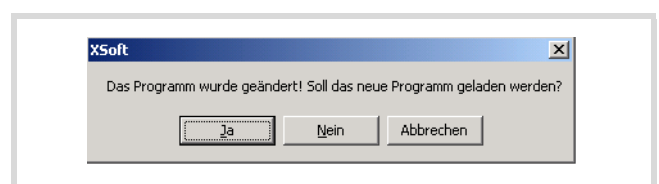


Abbildung 13: Abfrage nach Programmänderung

- ▶ Antworten Sie auch hier wieder mit „Ja“.
- ▶ Wählen Sie das Menü <Online → Start> aus (→ Abbildung 9). Sie bringen damit die Steuerung in den RUN-Betrieb.

Die weitere Kommunikation zwischen der XC100 und dem PC als Programmiergerät erfolgt mit der ausgewählten Datenübertragungsrate.

CANopen-Schnittstelle

Die CPUs können Sie über die potentialgetrennte ISO-11898-Schnittstelle an den CANopen-Bus anschließen.

Die Steckerbelegung ist wie folgt:

Klemme	Signal
6	GND
5	CAN_L
4	CAN_H
3	GND
2	CAN_L
1	CAN_H

Steckertyp: 6-poliger, steckbarer Federzugklemmenblock
 Leiteranschlüsse: bis 0,5 mm²

Die CPUs können am CAN-Bus sowohl als Netzwerk-(NMT-) Master als auch als NMT-Slave betrieben werden.

Die CPU kann dazu verwendet werden, CAN-Telegramme direkt aus dem Anwenderprogramm zu senden und zu empfangen. Eine Unterbrechung des CAN-Bus wird nur erkannt, wenn die entsprechenden CAN-Teilnehmer von der Steuerung her überwacht werden (Nodeguarding-Funktion).

Spannungsversorgung

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

Start/Stop-Verhalten

Wenn Sie den Betriebsarten-Vorwahlschalter (BAS) in Stellung „Stop“ setzen, werden alle Ausgänge der dezentralen Geräte auf Signal „0“ gesetzt.

Busabschlusswiderstände

An den Netzwerkenden müssen Busabschlusswiderstände von 120 Ω eingesetzt werden.

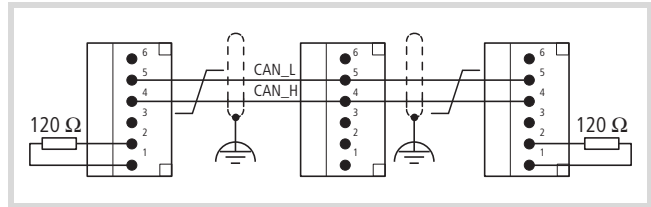


Abbildung 14: Möglicher Aufbau eines CANopen-Busses mit Busabschlusswiderständen

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

Eigenschaften des CANopen-Kabels

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

- Wellenwiderstand 100 bis 120 Ω
- 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 CANopen-Netzwerk aufgeführt.

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

Tabelle 2: Standardparameter für CANopen-Netzwerkkabel entsprechend der ISO 11898

Buslänge	Schleifenwiderstand	Aderquerschnitt	Busabschlusswiderstand	Übertragungsrate bei Leitungslänge
[m]	[mΩ/m]	[mm ²]	[Ω]	[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

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

Tabelle 3: 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 $\leq 5\%$.

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

Echtzeituhr

Die XC100 besitzt eine Echtzeituhr, die Sie im Anwenderprogramm über Funktionen aus der Bibliothek „SysLibRTC“ 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“.

→ Für die XC100 können Sie auch weiterhin die Funktionsbausteine „GetRealTimeClock“ (Auswerten der Echtzeituhr) und „SetRealTimeClock“ (Setzen der Echtzeituhr) verwenden. Diese werden aber von den Steuerungen XC200 und XN-PLC-CANopen nicht unterstützt.

Mehr Informationen zu den Funktionsbausteinen finden Sie im separaten Handbuch „Funktionsbausteine zur easySoft-CoDeSys“ (MN05010002Z-DE; frühere Bezeichnung AWB2786-1456D).

XC-CPU101-...-XV

Die Geräte XC-CPU101-...-8DI-6DO- XV sind mit einem erweiterten Betriebssystem ausgestattet. Diese Funktionalität erlaubt es, diese CPUs an Text-Displays der Typenreihe XV-101-.. zu betreiben.

→ Die Text-Displays werden in dem separaten Handbuch „Hardware und Projektierung“ (AWB2726-1461D) beschrieben.

Batterie

Zur Datensicherung wird eine Lithium-Batterie vom Typ 1/2 AA (3,6 V) verwendet. Der Batterie-Schacht befindet sich an der linken Außenseite der CPU hinter einer Abdeckplatte. Der Ladezustand der Batterie wird überwacht. Unterschreitet die Batteriespannung einen fest vorgewählten Level, wird eine Sammelfehlermeldung ausgegeben.

Die Pufferzeiten sind:

- Worst Case: 3 Jahre bei Dauerpufferung
- Typisch: 5 Jahre bei Dauerpufferung



Achtung!

Um einen Datenverlust zu vermeiden, ist der Batteriewechsel bei eingeschalteter Versorgungsspannung vorzunehmen.

Bestellbezeichnung der Batterie: XT-CPU-BAT-1

Montage der CPU

→ Ausführliche Hinweise zur Montage des Baugruppenträgers und der XIOC-Module finden Sie im Handbuch „Hardware und Projektierung der XIOC-Signalmodule“ (MN05002002Z-DE). Hier finden Sie auch weitere Informationen über die verschiedenen Typen von Baugruppenträgern und über die jeweilige Steckplatzbelegung der CPU und der XIOC-Signalmodule.

- ▶ 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**.

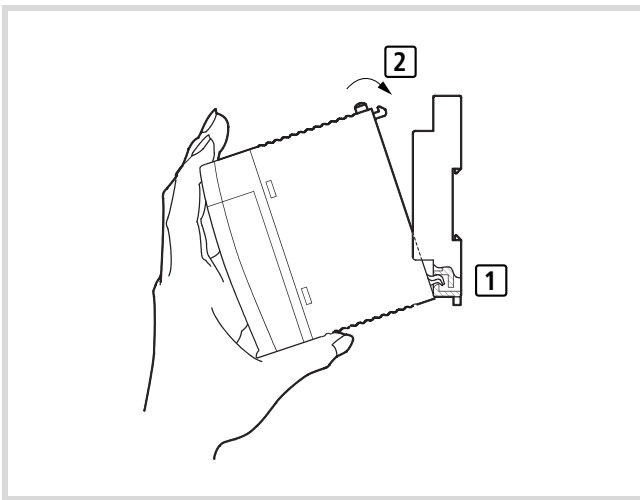


Abbildung 15: Montage der CPU

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**.

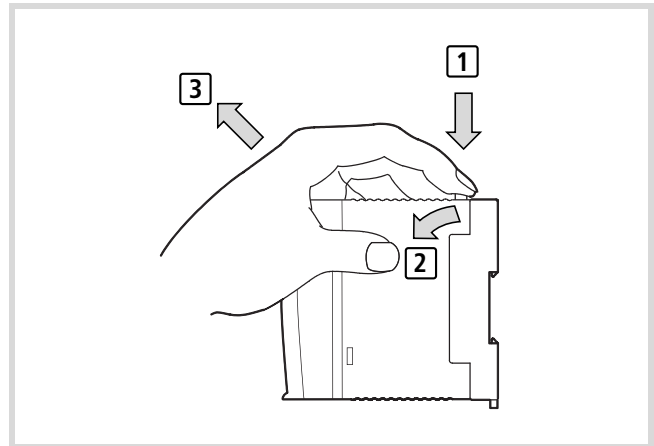


Abbildung 16: Demontage der Module

2 Projektierung

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 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 unterschiedlichen Leistungs- und Störniveaus vorzunehmen. Bei kleinen Schaltschränken genügt oft schon eine Abschottung durch Trennbleche, um Störeinflüsse zu reduzieren.

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.

Geräteanordnung

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

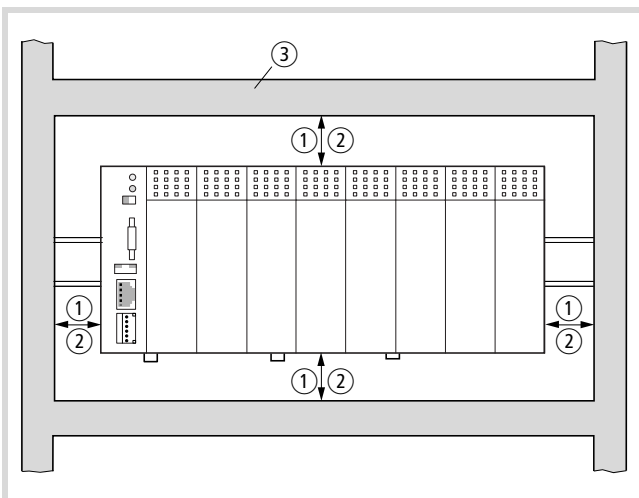


Abbildung 17: Schaltschrankaufbau

- ① Abstand > 50 mm
- ② Abstand > 75 mm zu aktiven Elementen
- ③ Kabelkanal

Störungen vermeiden

Leitungsführung und Verdrahtung

Es gibt 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. Damit vermeiden Sie kapazitive und induktive Einkopplungen. Ist eine getrennte Leitungsführung nicht möglich, ist in erster Linie die Störleitung zu schirmen.

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

- ▶ Vermeiden Sie parallel geführte Abschnitte von Leitungen unterschiedlichen Leistungsniveaus.
- ▶ Trennen Sie grundsätzlich Wechselspannungsleitungen von Gleichspannungsleitungen.
- ▶ 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 gegensinnigen Stromfluss wird die Summe aller Ströme bei diesem Leitungspaar null. Die entstehenden Felder werden kompensiert.

Schutzbeschaltung von Störquellen

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

→ Geschaltete Induktivitäten sollen grundsätzlich geschützt werden.

Schirmung

- ▶ Verwenden Sie zum Anschluss an die Datenschnittstellen Kabel mit Schirm. Generell gilt: Je kleiner die Kopplungsimpedanz, desto besser die Schirmwirkung.

Blitzschutzmaßnahmen

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.

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.

Zu Fragen der Leitungsverlegung und der Schirmungsmaßnahmen beachten Sie bitte folgende Handbücher:

- AWB27-1287-D „EMV-Projektierungsrichtlinie für Automatisierungssysteme“
- TB27-001-D „Elektromagnetische Verträglichkeit (EMV) von Automatisierungsanlagen“
- TB02-022-D „Elektromagnetische Verträglichkeit (EMV) von Maschinen und Anlagen“

Verdrahtungsbeispiele

➔ Verdrahtungsbeispiele zu den XIOC-Modulen finden Sie im Handbuch „Hardware und Projektierung der XIOC-Signalmodule“ (MN05002002Z-DE; frühere Bezeichnung AWB2725-1452D).

Netzteil

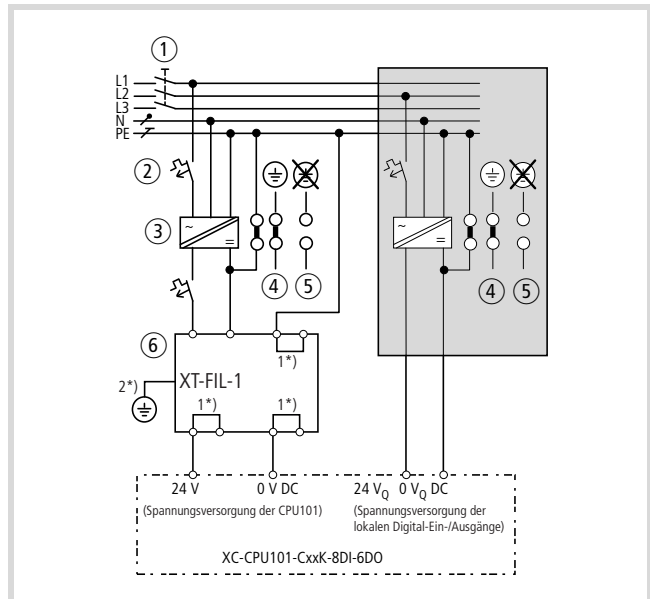


Abbildung 18: Verdrahtungsbeispiel für Netzteil

- ① Hauptschalter
- ② Leitungsschutzorgan
- ③ 24-V-DC-Versorgungsspannung
- ④ Geerdeter Betrieb
- ⑤ Bei ungeerdetem Betrieb muss eine Isolationsüberwachung eingesetzt werden (IEC 204-1, EN 60204-1, DIN EN 60204-1).
- ⑥ 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. Der Filter ist nicht Bestandteil der CPU und muss deshalb getrennt bestellt werden:
 Typ: XT-FIL-1, Best.-Nr.: 285316
 (Lieferant: Eaton Industries GmbH)
 ➔ „Abmessungen“ auf Seite 72
 ➔ „Technische Daten“ auf Seite 77

1*) intern gebrückt

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

Spannungsversorgung der Digital-Ein-/Ausgänge

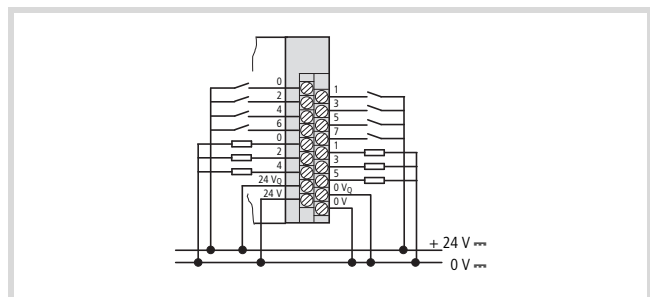


Abbildung 19: Beispiel der Verdrahtung des Klemmblocks

Das Verdrahtungsbeispiel zeigt die Verdrahtung bei separater Spannungsversorgung der Ein-/Ausgänge.

3 CPU betreiben

Einschaltverhalten

Nach dem Einschalten der Versorgungsspannung führt die CPU einen Selbsttest und mehrere CRC-Prüfungen durch. Stellt Sie einen Fehler fest, verharrt sie im Zustand „Einschalten nicht ok“, → Kapitel „Betriebszustände“ auf Seite 41.

Nach erfolgreicher Beendigung der Tests übernimmt das Betriebssystem (BTS) die Kommunikation mit dem Programmiersystem, die Ausführung und das Debugging des Applikationsprogrammes. Es unterstützt nur ein Applikationsprogramm.

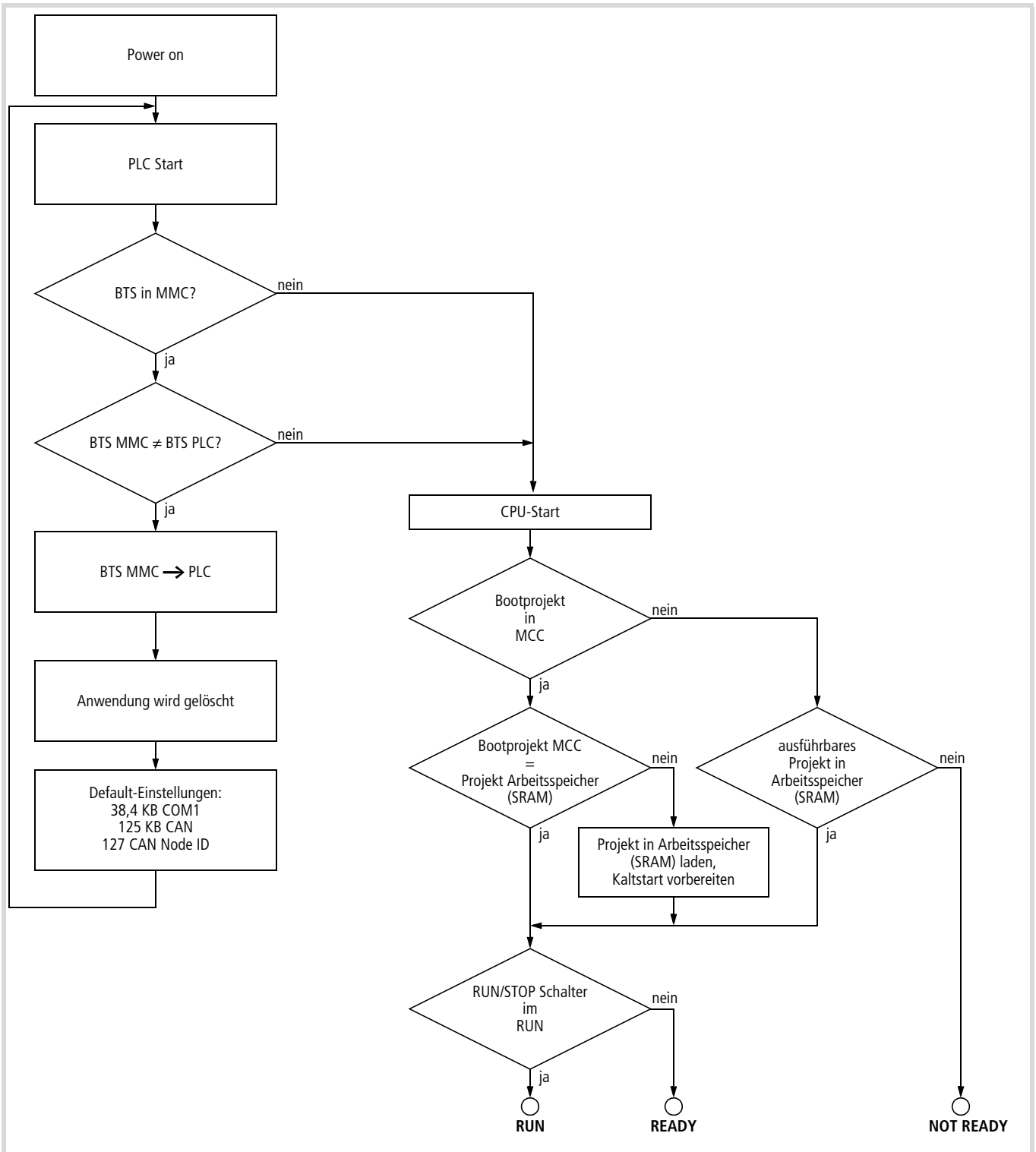


Abbildung 20: Einschaltverhalten

Ausschaltverhalten

Das Ausschalten (Betriebsarten-Vorwahlschalter: Run → Stop) führt zu einer Unterbrechung der Programmbearbeitung am Zyklusende. Bei einem Spannungseinbruch (Erkennung über das PFI-Signal) wird die Programmbearbeitung sofort beendet. Gleichzeitig werden die Ausgänge abgeschaltet. Nach Spannungswiederkehr führt die Steuerung einen Neustart durch (siehe Abschnitt „Einschaltverhalten“).

Start-Verhalten

Das Startverhalten der Steuerung ist abhängig von:

- der Stellung des lokalen Betriebsarten-Vorwahlschalters
- dem im Programmiersystem parametriertem Startverhalten.

(Bei der CPU-Variante „XV“ (Visualisierungs-CPU) ist zusätzlich noch eine Bedienung über das System-Menü des Displays möglich).

Für einen Wechsel des Betriebszustandes von „Stop“ in „Run“ ist die Stellung des Betriebsartenschalters maßgebend und kann nicht durch entsprechende Vorwahl im Programmiersystem erzwungen werden.

Beim Programmstart wird geprüft, ob die konfigurierten Ein-/Ausgänge mit den physikalisch vorhandenen übereinstimmen. Weiter wird geprüft, ob das parametrierte Modul oder ob physikalisch ein anderer Modultyp vorhanden ist. Ein nicht vorhandenes Modul hat keinen Einfluss auf den Start des Applikationsprogramms, ein anderer Modultyp verhindert dies.

Beim Starten des Applikationsprogramms wird differenziert zwischen (siehe auch entsprechende Folge-Abschnitte):

- Kaltstart
- Warmstart

Stopp-Verhalten

Die Bearbeitung des Applikationsprogramms wird immer am Ende des Programmzyklus angehalten.

Kaltstart (Coldstart)

Ein Kaltstart wird beim ersten Start, nach dem Laden des Programms auf die Steuerung, und nach jedem „Reset-Kalt“ ausgeführt. Hierbei werden alle Variablen des Programms mit ihren Initialisierungswerten initialisiert und das Programm gestartet.

Warmstart

Alle weiteren Starts des geladenen Programms sowie nach „Reset-Warm“ sind Warmstarts. Die mit RETAIN deklarierten Variablen behalten ihre Werte, die übrigen Variablen werden mit ihren Initialisierungswerten initialisiert und das Programm gestartet.

Test und Inbetriebnahme

Die Steuerung unterstützt folgende Test- und Inbetriebnahme-Möglichkeiten:

- Breakpoint/Einzelschritt-Betrieb
- Einzelzyklus-Betrieb
- Forcen
- Online-Änderung
- Zustandsanzeige (Power Flow).

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 Einzelschritt-Betrieb ausgeführt werden. Die Zykluszeitüberwachung ist deaktiviert.



Vorsicht!

Die zu diesem Zeitpunkt gesetzten Ausgänge bleiben gesetzt!

Einzelzyklus-Betrieb

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

Forcen

Alle Variablen eines Applikationsprogramms können zwangsgesetzt werden. Werden Variablen von physikalischen Ausgängen des lokalen I/Os zwangsgesetzt, werden diese nur im Zustand „Run“ zur Peripherie durchgeschaltet.

Zustandsanzeige der easySoft-CoDeSys

- Der Signalzustand der physikalischen, booleschen Eingänge wird im „Stop“- und „Start“-Modus angezeigt.
- Der Signalzustand der physikalischen, booleschen Ausgänge wird nur im „Start“ (RUN)-Modus angezeigt.
- Die Anzeige für ein Low-Signal wird mit „FALSE“ angezeigt und ist schwarz hinterlegt.
- Die Anzeige für ein High-Signal wird mit „TRUE“ angezeigt und ist blau hinterlegt.
- Alle anderen Variablen werden nur im „Start“-Modus mit dem jeweils aktuellen Variablen-Wert dargestellt.

Programm-Reset

Das Applikationsprogramm kann in folgenden Stufen zurückgesetzt werden:

- Reset-Warm
- Reset-Kalt
- Reset-Ursprung

Reset-Warm

Entspricht der Initialisierung bei einem Warmstart, siehe Abschnitt „Warmstart“ auf Seite 20.

Reset-Kalt

Entspricht der Initialisierung bei einem Kaltstart, siehe Abschnitt „Kaltstart (Coldstart)“ auf Seite 20.

Reset-Ursprung

Das Applikationsprogramm der Steuerung wird komplett gelöscht. Die Steuerung ist anschließend im Zustand „NOT READY“. Das Boot-Projekt und das Betriebssystem auf der MMC werden ebenfalls gelöscht.

Programm-Parametrierung

Ein Applikationsprogramm besitzt verschiedene Parameter, die im Programmiersystem eingestellt werden können:

- Maximale Zykluszeit des Programms
- Startverhalten nach PowerOn
- Parameter für CAN-Routing.

Maximale Zykluszeit des Programms

Die maximale Zykluszeit des Anwenderprogramms kann im Bereich von 20 bis 1000 ms eingestellt werden. Der Defaultwert beträgt 20 ms.

Startverhalten nach PowerOn

Diese Einstellung legt fest, wie sich die Steuerung nach dem Einschalten verhalten soll, wenn ein Applikationsprogramm vorhanden ist und der Betriebsarten-Vorwahlschalter in der Position „Run“ steht.

Folgende Einstellungen sind möglich:

- WARMSTART (Default-Einstellung),
- COLDSTART,
- HALT.

Bootprojekt erzeugen und transferieren

Ein Bootprojekt wird von einem geladenen Anwenderprogramm erzeugt und auf der MMC abgespeichert.

Um ein Bootprojekt zu erstellen, sind folgende Schritte notwendig:

- ▶ Stecken Sie eine MMC in den „MEM Card“-Schacht der CPU.
- ▶ Wechseln Sie in den Ordner „Online“.
- ▶ Wählen Sie den Befehl „Einloggen“.
- ▶ Wählen Sie den Befehl „Bootprojekt erzeugen“.

➔ Das Bootprojekt ist an die BTS-Version gebunden, mit der es erzeugt wurde. Wollen Sie das Bootprojekt einer Steuerung in eine weitere Steuerung übertragen, entnehmen Sie die MMC (mit Bootprojekt) der Quellsteuerung und stecken Sie in die Zielsteuerung. Schalten Sie die Steuerung anschließend aus und ein.

Die Zielsteuerung kann nur mit dem neuen Bootprojekt arbeiten, wenn sie das gleiche BTS wie die Quellsteuerung hat, mit der das Bootprojekt erzeugt wurde, ➔ Abschnitt „Betriebssystem (BTS) aktualisieren“ auf Seite 22.

Bootprojekt erzeugen nach Online-Änderung

Nach einer Online-Änderung können Sie ein neues Bootprojekt erzeugen.

Beachten Sie folgenden angezeigten Hinweis:

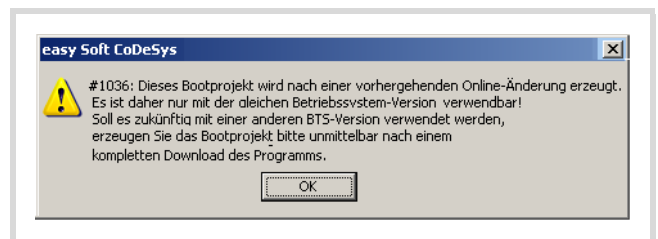


Abbildung 21: Download: Auswahl und Information

Betriebssystem (BTS) aktualisieren

Bei der XC100 haben Sie die Möglichkeit, das mit der Steuerung ausgelieferte Betriebssystem (BTS) durch ein aktuelles zu ersetzen. Eaton bietet die jeweils aktuelle BTS-Version im Internet als Download an.

→ Nicht alle Funktionen des neuen Betriebssystems werden von älteren XC100-Versionen unterstützt.

Sie haben zwei Möglichkeiten, das BTS zu übertragen:

- Direkt vom PC in die Steuerung,
- Vom PC in die MMC.

Eine Übertragung des BTS vom PC in die MMC der Steuerung ist möglich, wenn die Steuerung ein BTS ab Version 03.03 enthält.

Betriebssystem vom PC in die Steuerung übertragen

Wird ein BTS in die Steuerung geladen, wird das bestehende BTS sowie das Anwenderprogramm gelöscht.

Vorgehensweise:

- ▶ Stecken Sie eine MMC in den „MEM“-Schacht der CPU.
- ▶ Stellen Sie eine serielle Verbindung über die RS232-Schnittstelle des PCs mit der XC100 her, → Seite 45.
- ▶ Aktivieren Sie im Fenster „Steuerungskonfiguration“ die Registerkarte „Weitere Parameter“.

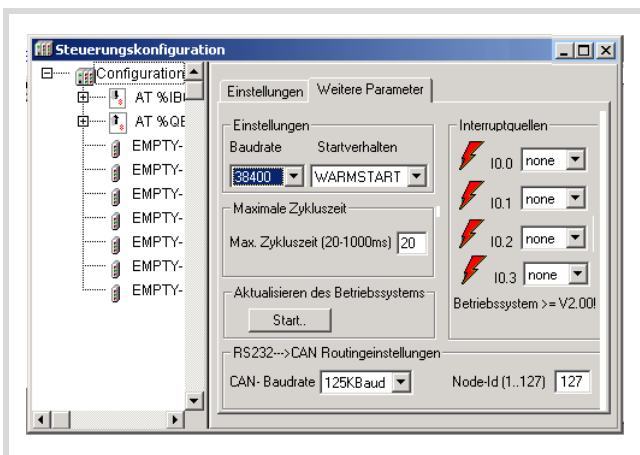


Abbildung 22: Download des BTS XC100 starten

- ▶ Klicken Sie auf die Schaltfläche „Start“.

Das „Download“-Fenster öffnet sich.

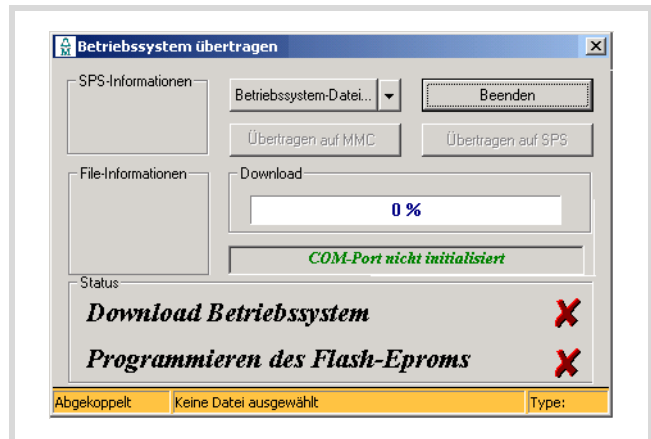


Abbildung 23: Download: Auswahl und Information

- ▶ Betätigen Sie in die Schaltfläche „Betriebssystem-Datei“ und wählen Sie die gewünschte Betriebssystem-Datei (*.hex) aus.

→ Zuletzt geöffnete Dateien können Sie über das Drop-Down-Menü auswählen.

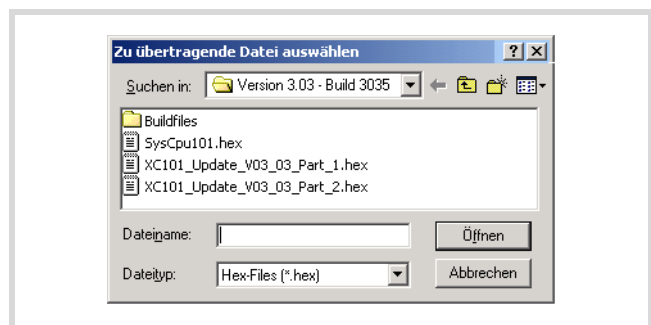


Abbildung 24: Betriebssystem-Datei auswählen

Nach der Auswahl der BTS-Datei (ab Version V03.03) erhalten Sie eine Information über den Zieltyp und die File-Version.

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

Die Übertragung beginnt.

→ Erscheint ein Warnzeichen im Feld „Programmieren des Flash-Eproms“, darf die Spannung nicht abgeschaltet werden!

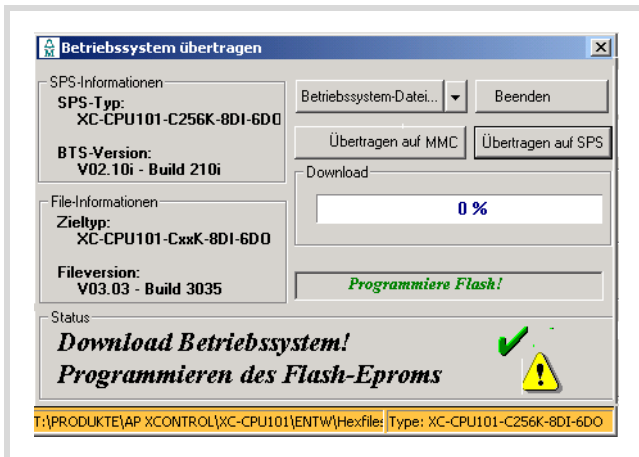


Abbildung 25: Warnung beim Download

Warten Sie die folgende Anzeige ab.

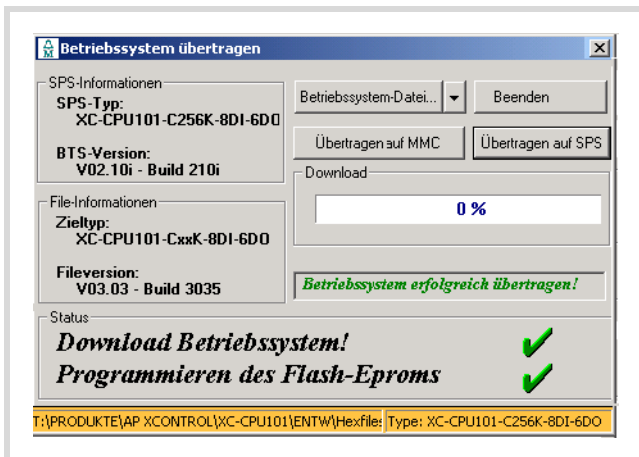


Abbildung 26: Betriebssystem auf SPS übertragen

► Klicken Sie in diesem Fenster auf die Schaltfläche „Beenden“.

Nach einem Download des BTS in die Steuerung kann die Meldung „Kommunikation abgebrochen“ erscheinen, da die Steuerung nach jedem Download des BTS neu gebootet wird. Nach jedem Booten ist ein erneutes Einloggen erforderlich.

Wenn Sie sich in die Steuerung einloggen erscheint die Meldung: „Kein Programm auf der Steuerung! Soll das neue Programm geladen werden?“

Nachdem Sie die Frage mit „Ja“ bestätigt haben, wird das Programm geladen. Dann können Sie das Bootprojekt erstellen.

Betriebssystem vom PC auf die MMC übertragen

Wird ein BTS in die MMC geladen, wird das BTS und das Bootprojekt auf der MMC und das Anwenderprogramm in der Steuerung gelöscht. Die Vorgehensweise ist analog zur Beschreibung im Abschnitt „Betriebssystem vom PC in die Steuerung übertragen“ auf Seite 22.

Betriebssystem von der MMC in die Steuerung übertragen

Schalten Sie die Steuerung aus und ein. Die Übertragung kann über 30 Sekunden dauern, da die CPU mehrfach gebootet wird.

Unterbrechen Sie den Vorgang nicht, indem Sie beispielsweise die Versorgungsspannung ausschalten!

Update von weiteren XC100-Steuerungen

Wenn Sie das neue Betriebssystem auf weitere Steuerung übertragen wollen, stecken Sie die MMC in die Steuerung, die ebenfalls über ein BTS ab V03.03 verfügt. Beim Einschaltvorgang wird das BTS der Steuerung aktualisiert und ein Bootprojekt in die Steuerung geladen.

4 Programmbearbeitung und Systemzeit

Das Applikationsprogramm wird zyklisch abgearbeitet. Vor jedem Programmzyklus wird das Abbild der Eingänge eingelesen und am Ende des Zyklus wird das Ausgangsabbild auf die physikalischen Ausgänge geschrieben. Zusätzlich werden alle anfallenden Systemaktivitäten vor bzw. nach dem Zyklus bearbeitet.

Hierunter fallen z. B.:

- Kommunikation mit der easySoft-CoDeSys
- Online-Änderungen
- Bearbeiten des CANopen-Protokollstacks usw.

Aufgrund der Softwarearchitektur des Laufzeitsystems kann es zu zeitlichen Jittern zwischen den einzelnen Abarbeitungszyklen kommen.

Zykluszeitüberwachung

Die Zykluszeitüberwachung überwacht die zyklische Task des Applikationsprogramms mittels eines Hardware-Timers. Beim Überschreiten der parametrisierten Zeit werden die Ausgänge der Steuerung abgeschaltet und die XC100 in den Zustand „Stop“ gesetzt.

Systembibliotheken, Funktionsbausteine und Funktionen

Für Ihre Applikation können Sie verschiedene Systembibliotheken mit den dazugehörigen Funktionen und Funktionsbausteinen nutzen.

Grundsätzlich stehen nach der Zielsystemauswahl folgende Bibliotheken zur Verfügung:

- Standard.lib
- RTCLib.lib
- Counter.lib
- SYSLIBCALLBACK.LIB

In diesen Librarys sind generelle IEC-Bausteine und Funktionen für XC-Steuerungen enthalten. Sie finden die oben genannten Bibliotheken in der easySoft-CoDeSys unter <Ressourcen → Bibliotheksverwalter>. Weitere Bibliotheken, die Sie nachträglich installieren, werden ebenfalls im Bibliotheksverwalter angezeigt.

Die Beschreibung der Funktionsbausteine und Funktionen finden Sie in der Library/Online-Hilfe des Programmiersystems:

- ▶ Starten Sie die easySoft-CoDeSys, klicken Sie auf die Schaltfläche „Hilfe“ und wählen Sie „Inhalt“. In diesem Fenster können Sie wählen zwischen „Inhalt“, „Index“ oder „Suchen“.

Bibliotheksverwalter

Im Bibliotheksverwalter sind alle installierten Bibliotheken abgelegt.

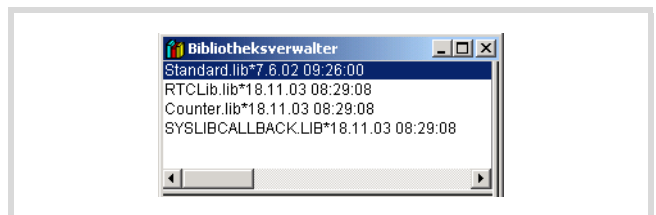


Abbildung 27: Standardbibliotheken im Bibliotheksverwalter

Zielsystemspezifische Bibliotheken

Für das Zielsystem XC100 sind die Librarys „Lib_Common“ und „Lib_CPU101“ erforderlich.

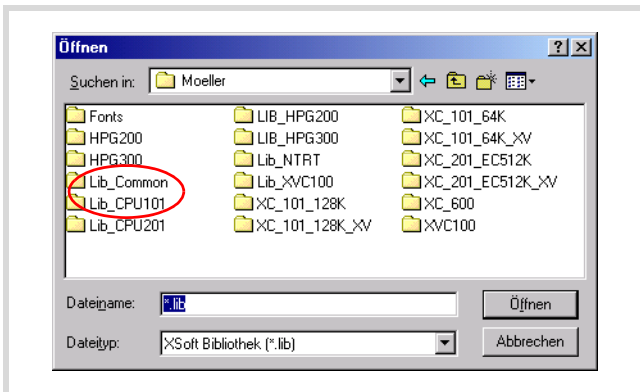


Abbildung 28: Zielsystemspezifische Bibliotheken

Lib_Common

In der „Lib_Common“ sind die Grundfunktionen enthalten, die für die gesamten XC-Zielsysteme erforderlich sind. Zudem enthält diese Bibliothek die S40-typischen Funktionsbausteine, die aus dem PS40-System in das easySoft-CoDeSys-System übernommen wurden.

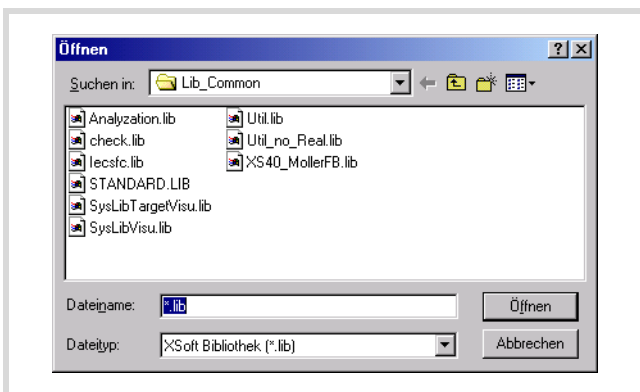


Abbildung 29: Dateien der „Lib_Common“

Weitere Informationen zu den Librarys „STANDARD.lib“ und „XS40_MoellerFB.lib“ finden Sie gleich anschließend. Auf die übrigen Librarys wird hier nicht näher eingegangen; sie sind im Handbuch zur Programmiersoftware (MN05010003Z-DE; frühere Bezeichnung AWB2700-1437D) und/oder in der Library/Online-Hilfe des Programmiersystems beschrieben.

Standard.lib

In der Standard.lib sind die IEC-Funktionsbausteine und Standard-Funktionen enthalten.

Die Beschreibung der Funktionen und Funktionsbausteine finden Sie im Handbuch zur Programmiersoftware (MN05010003Z-DE; frühere Bezeichnung AWB2700-1437D) und in der Library/Online-Hilfe des Programmiersystems.

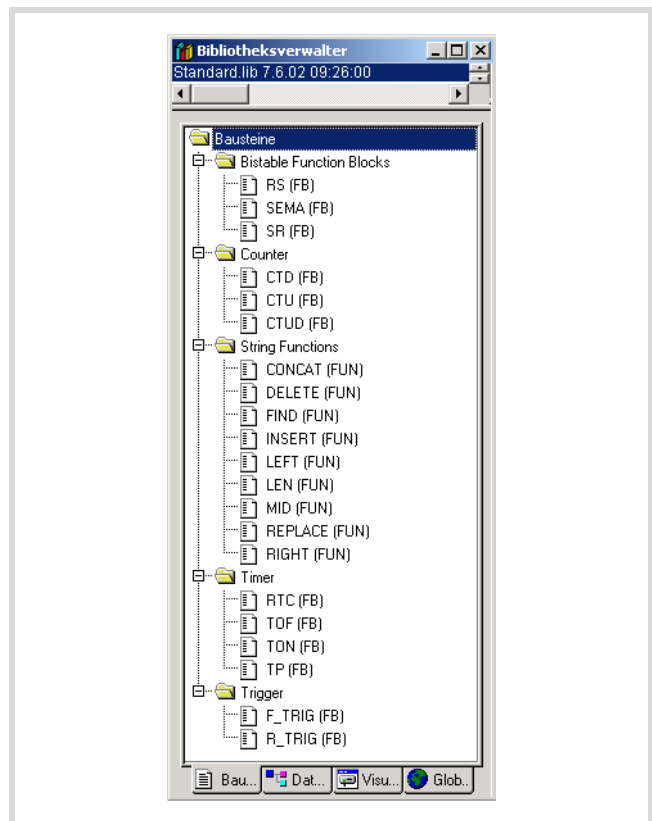


Abbildung 30: Funktionsbausteine und Funktionen der „Standard.lib“

XS40_MoellerFB.lib

Die Beschreibung der Library „XS40_MoellerFB“ finden Sie im Handbuch „Funktionsbausteine zur easySoft-CoDeSys“ (MN05010002Z-DE; frühere Bezeichnung AWB2786-1456D).

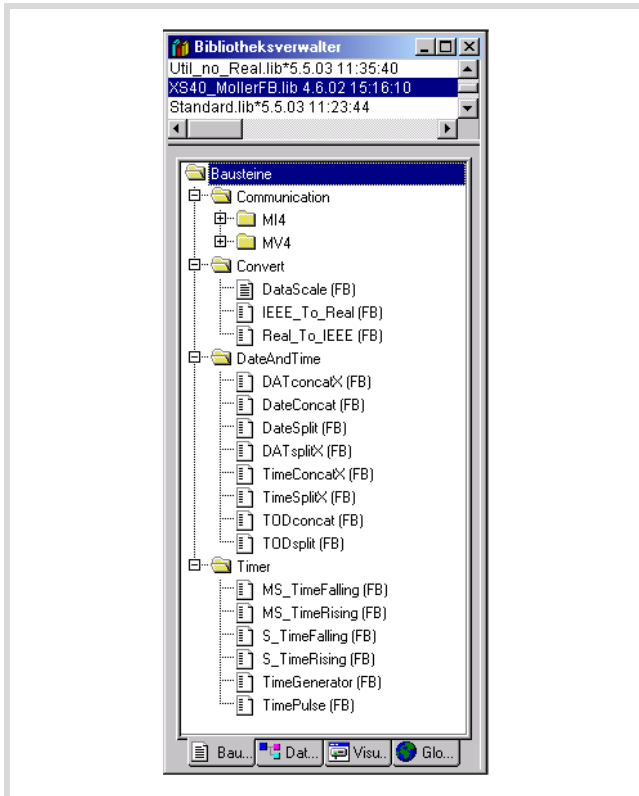


Abbildung 31: Funktionsbausteine der „XS40_MoellerFB.lib“

Bibliotheken der „Lib_CPU101“

Libraries, die für das XC100-Zielsystem erforderlich sind, sind in der Bibliothek „Lib_CPU101“ zusammengefasst.

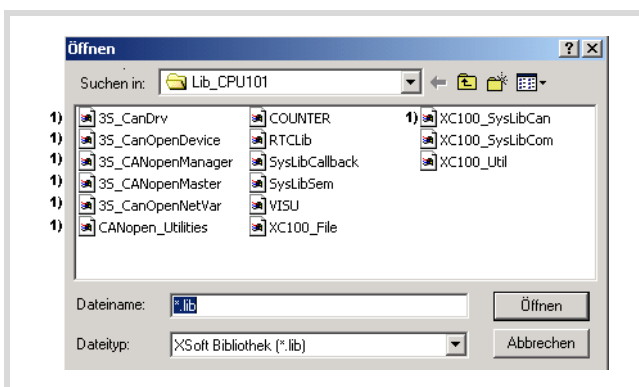


Abbildung 32: Funktionsbausteine der „Lib_CPU101“

Die mit „1)“ gekennzeichneten Libraries beinhalten Funktionen für den CANopen-Feldbus. Die CANopen-Bus-Erweiterung ist in den Anwenderhinweisen zum CANopen-Bus“ beschrieben. Die Anwenderhinweise informieren über folgende Themen:

- XC-...-XION: Einbindung der XION-Produkte (AN2700K18D)
- XC-...-XC: Netzwerkvariable (AN2700K19D)
- Kopplung mehrerer autarker Steuerungen (CANDevice) über CANopen (AN2700K20D)
- Projektierung von CAN-Teilnehmern (AN2700K27D).

Bibliothek „Counter.lib“:

Die Library COUNTER stellt Funktionen zur Verfügung, um die Zählermodule XIOC-1CNT-100KHZ und XIOC-2CNT-100KHZ in das XC100-System einzubinden.

Die Beschreibung der Library COUNTER finden Sie im Handbuch „Funktionsbausteine zur easySoft-CoDeSys“ (MN05010002Z-DE; frühere Bezeichnung AWB2786-1456D) im Abschnitt „Zählerbausteine: counter.lib“.

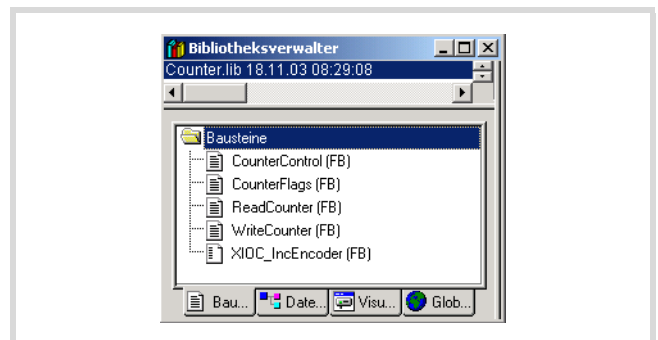


Abbildung 33: Funktionsbausteine der „Counter.lib“

Bibliothek „RTCLib“

Die Library „RTCLib“ stellt Lese- und Schreib-Funktionen für den Zugriff auf die Echtzeituhr zur Verfügung.

Die Beschreibung der Library „RTCLib“ finden Sie im Handbuch „Funktionsbausteine zur easySoft-CoDeSys“ (MN05010002Z-DE; frühere Bezeichnung AWB2786-1456D) im Abschnitt „Uhrenbausteine: RTCLib.lib“.

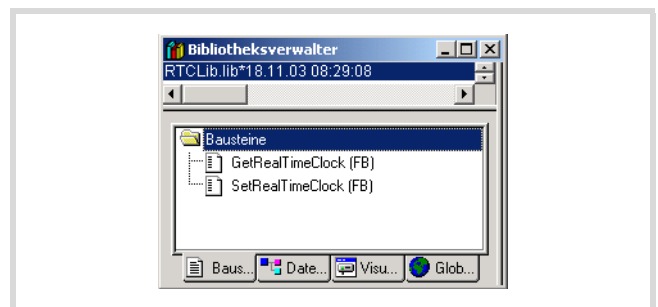


Abbildung 34: Funktionsbausteine der „RTCLib.lib“

Bibliothek SYSLIBCALLBACK.LIB

Die SYSLIBCALLBACK.LIB dient dazu, definierte Callback-Funktionen für Laufzeitergebnisse zu aktivieren (z. B. An- und Abmelden der Eventfunktionen).

Die Beschreibung der Library finden Sie im Handbuch zur Programmiersoftware (MN05010003Z-DE; frühere Bezeichnung AWB2700-1437D) und in der Library/Online-Hilfe des Programmiersystems.

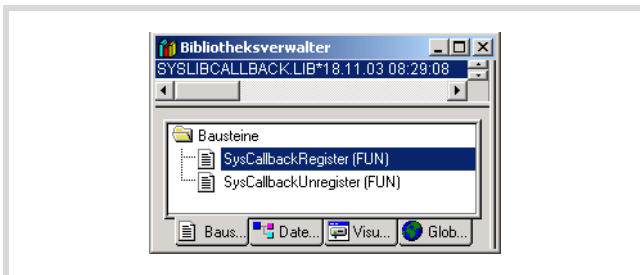


Abbildung 35: Funktionen der „SysLibCallBack.lib“

Bibliothek „SysLibSem“

Die Library „SysLibSem“ stellt Funktionen zur Verfügung, um Semaphore (in sich abgeschlossen Informationsträger, die nicht abgebrochen werden können) für die Synchronisation von Tasks zu erzeugen und zu benutzen. Semaphore dienen dazu, den gleichzeitigen Zugriff auf kritische Daten zu verhindern, die von mehreren Tasks verwendet werden. Greift eine Task auf einen bestimmten Bereich zu, verhindert der Mechanismus, dass andere Tasks zeitgleich zugreifen können. Erst wenn die zugriffsberechtigte Task den Zugriff beendet hat, kann eine andere Task auf diesen Bereich zugreifen.

Die Beschreibung der Funktionen finden Sie im Handbuch zur Programmiersoftware (MN05010003Z-DE; frühere Bezeichnung AWB2700-1437D) und in der Library/Online-Hilfe des Programmiersystems.

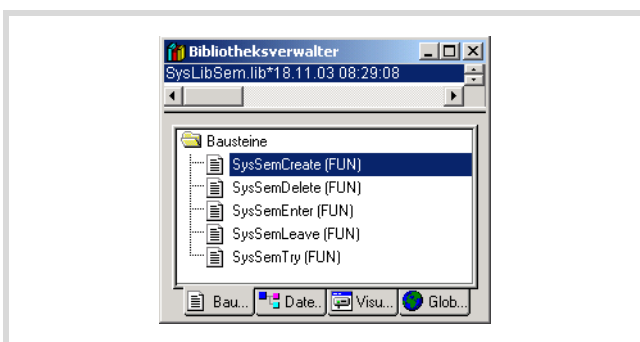


Abbildung 36: Funktionen der „SysLibSem.lib“

Bibliothek „VISU“

Die Library „VISU“ stellt Funktionen zur Verfügung, mit denen in der easySoft-CoDeSys eine Visualisierung realisiert werden kann.

Die Beschreibung der Library „VISU“ finden Sie im Handbuch „Funktionsbausteine zur easySoft-CoDeSys“ (MN05010002Z-DE; frühere Bezeichnung AWB2786-1456D) im Abschnitt „Visualisierungsbausteine: VISU.lib“.

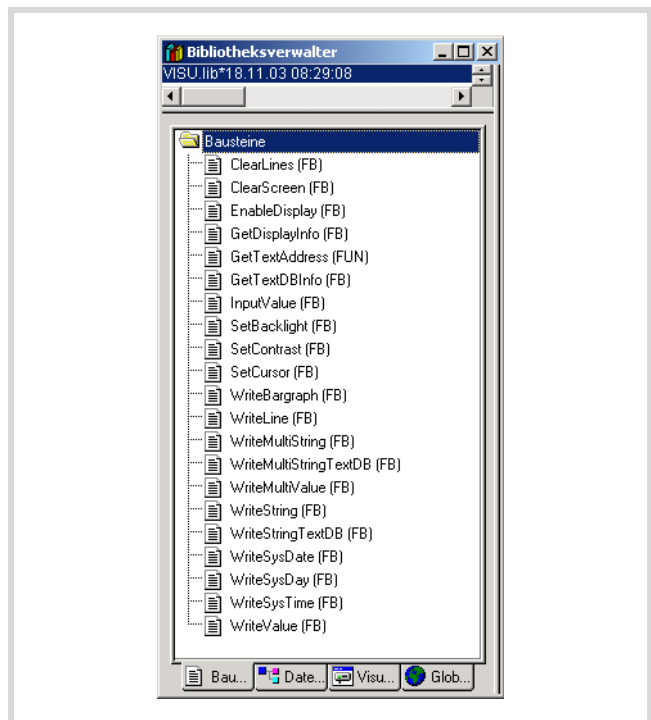


Abbildung 37: Funktionen der „VISU.lib“

Bibliothek „XC100_File.lib“

Die Library „XC100_File.lib“ stellt Funktionen zur Verfügung, um mit der easySoft-CoDeSys auf das Datei-System der Multi Media Card „MMC“ zugreifen zu können.

Die Beschreibung der Library „XC100_File.lib“ finden Sie im Handbuch „Funktionsbausteine zur easySoft-CoDeSys“ (MN05010002Z-DE; frühere Bezeichnung AWB2786-1456D) im Abschnitt „Dateizugriffsbausteine: XC100_File.lib“.

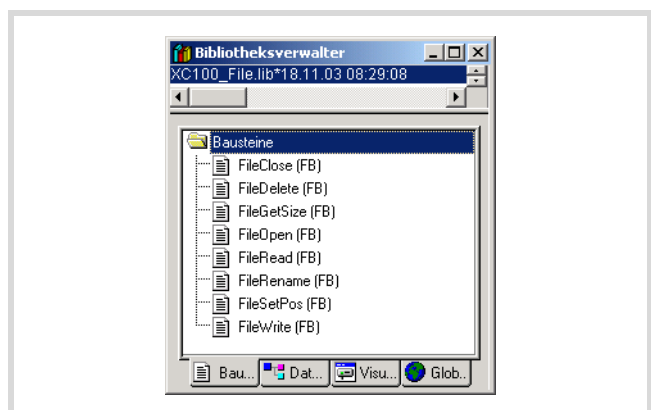


Abbildung 38: Funktionen der „XC100_File.lib“

Bibliothek „XC100_SysLibCom.lib“

Die Library „XC100_SysLibCom.lib“ stellt Funktionen zur Verfügung um die RS232-Schnittstelle im Transparentmode betreiben zu können.

Die Beschreibung der Library „XC100_SysLibCom.lib“ finden Sie in diesem Handbuch im Abschnitt „RS232-Schnittstelle im Transparentmode“.

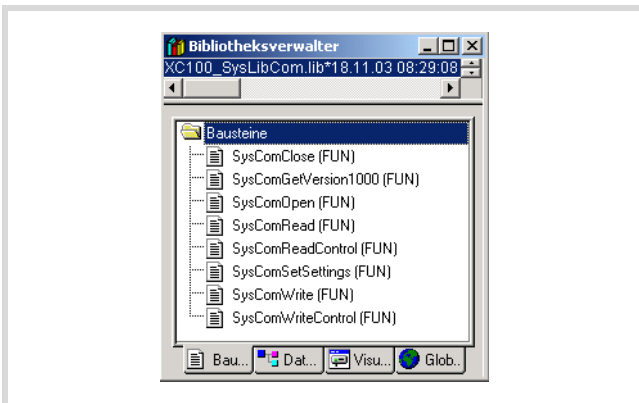


Abbildung 39: Funktionen der „XC100_SysLibCom.lib“

Bibliothek „XC100_Util.lib“

Die Library „XC100_Util.lib“ stellt zusätzliche, verschiedene Bausteine zur Verfügung, um z. B. folgende Funktionalitäten in die Applikation einbinden zu können:

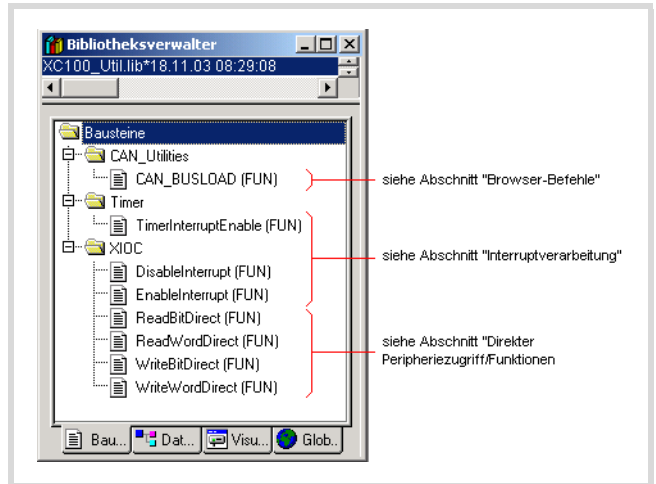


Abbildung 40: Funktionen der „XC100_Util.lib“

Die Beschreibung der Funktionsbausteine finden Sie im Handbuch „Funktionsbausteine zur easySoft-CoDeSys“ (MN05010002Z-DE; frühere Bezeichnung AWB2786-1456D).

Direkter Peripheriezugriff

Die Funktion „Direkter Pripheriezugriff“ ermöglicht es, direkt auf die lokalen und zentralen Ein- und Ausgangs-Signale der Steuerung zuzugreifen. Der E/A-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-100-Steuerung. XIOC-Signalbaugruppen, die über ein Bussystem eingebunden werden können, sind über den „Direkter Pripheriezugriff“ 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 die Tabelle 4.

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

Tabelle 4: Übersicht „Direkter Peripheriezugriff“

Baugruppe	E/A-Bit-Zugriff			E/A-Word-Zugriff			E/A-Slot Param.
	Read	Write	Param./Baugr.	Read	Write	Param./Baugr.	
XC-CPU101-C256K-8DI-6DO	✓	✓	DI: 0 bis 7; DO: 0 bis 5	✓	✓	0	0
XC-CPU101-C256K-8DI-6DO-XV	✓	✓	DI: 0 bis 7; DO: 0 bis 5	✓	✓	0	0
XC-CPU101-C128K-8DI-6DO	✓	✓	DI: 0 bis 7, DO: 0 bis 5	✓	✓	0	0
XC-CPU101-C128K-8DI-6DO-XV	✓	✓	DI: 0 bis 7, DO: 0 bis 5	✓	✓	0	0
XC-CPU101-C64K-8DI-6DO	✓	✓	DI: 0 bis 7, DO: 0 bis 5	✓	✓	0	0
XC-CPU101-C64K-8DI-6DO-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-16DI-AC	✓	–	0 bis 15	✓	–	0	1 bis 15

Baugruppe	E/A-Bit-Zugriff			E/A-Word-Zugriff			E/A-Slot
	Read	Write	Param./Baugr.	Read	Write	Param./Baugr.	Param.
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-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-SER	–	–	–	–	–	COM2, COM3	1 bis 15
XIOC-NET-DP-M	–	–	–	–	–	–	1 bis 3

Funktionen

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-Variablen wird nicht verändert, wenn bei der Bearbeitung ein Fehler auftritt.

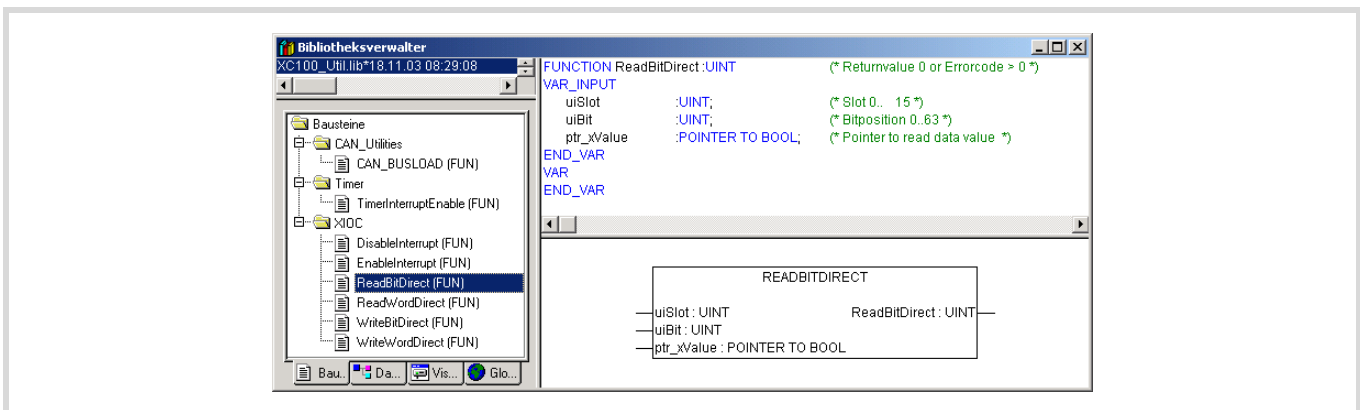


Abbildung 41: Funktion ReadBitDirect

Tabelle 5: Parameter der Funktion „ReadBitDirect“

uiSlot:	Steckplatznummer des Signalmoduls. Mögliche Parameter siehe Tabelle 4 auf Seite 29
uiBit:	Bitposition innerhalb des Eingangswertes des Signalmoduls. Mögliche Parameter siehe Tabelle 4 auf Seite 29
ptr_xValue:	Zeiger auf den Variablenwert
ReadBitDirect:	Anzeige des Fehlcodes, siehe Tabelle 9 auf Seite 33

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-Variablen wird nicht verändert, wenn bei der Bearbeitung ein Fehler auftritt.

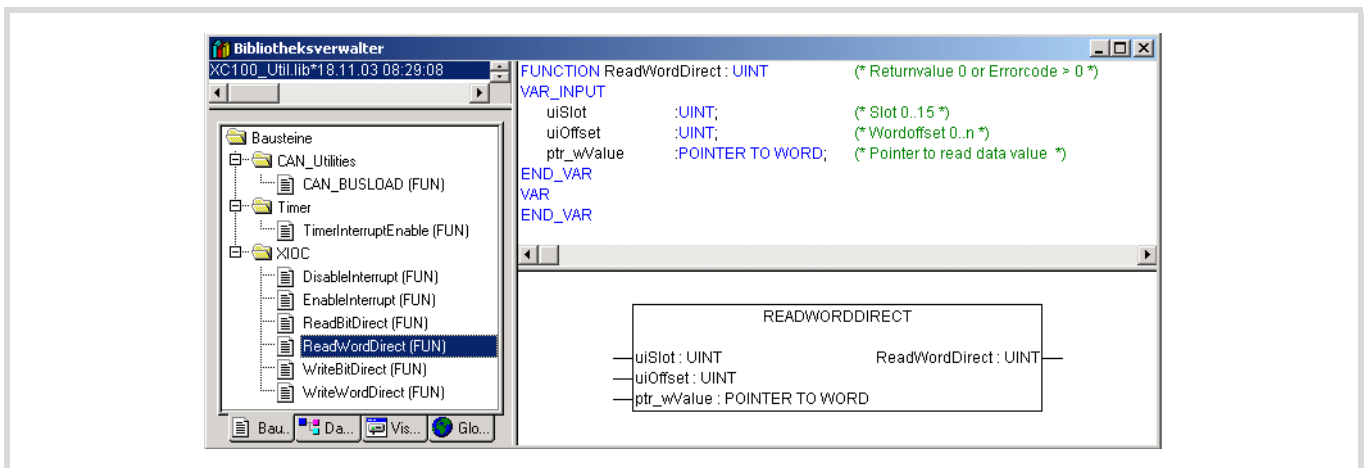


Abbildung 42: Funktion ReadWordDirect

Tabelle 6: Parameter der Funktion „ReadWordDirect“

uiSlot:	Steckplatznummer des Signalmoduls. Mögliche Parameter siehe Tabelle 4 auf Seite 29.
uiOffset:	Wordoffset innerhalb des Signalmoduls. Mögliche Parameter siehe Tabelle 4 auf Seite 29
ptr_wValue:	Zeiger auf den Variablenwert
ReadWordDirect:	Anzeige des Fehlcodes, siehe Tabelle 9 auf Seite 33

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 6 Ausgänge der XC100-CPU mit dem Steckplatz „0“ möglich.

Für die Steckplätze „1 und höher“ gilt folgende Einschränkung:

Ein einzelner Ausgang kann beim „direkten Peripheriezugriff“ nicht beschrieben werden. Es erfolgt immer ein Beschreiben des gesamten Ausgangswortes. Dies führt dazu, dass zum Zeitpunkt des „direkten Zugriffs“ das gerade aktuelle Ausgangsabbild mit dem modifizierten Ausgangsbit ausgegeben wird. Somit erfolgt eine Ausgabe von anderen Ausgängen innerhalb des Ausgangswortes bereits zum Zeitpunkt des Zugriffs und nicht erst am Zyklusende.

Aus diesem Grund sollten Sie bei einem „direkten Peripheriezugriff“ auf ein Ausgangsbit innerhalb eines Programmzyklus nicht die restlichen Ausgänge des Ausgangswortes verwenden und auswerten.

Eine weitere Aktualisierung des Ausgangswortes erfolgt am Zyklusende.

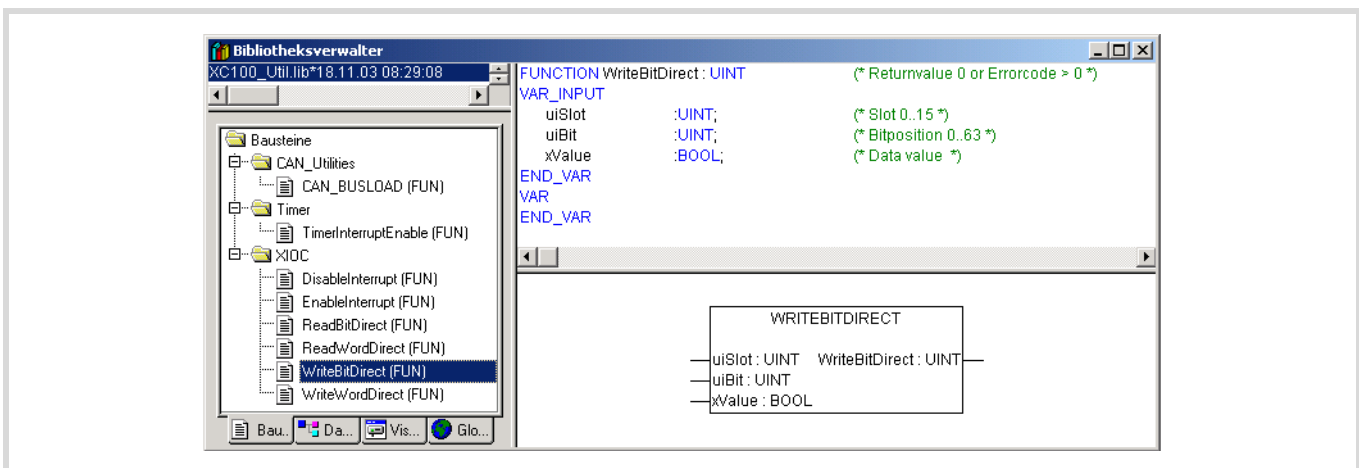


Abbildung 43: Funktion WriteBitDirect

Tabelle 7: Parameter der Funktion „WriteBitDirect“

uiSlot:	Steckplatznummer des Signalmoduls. Mögliche Parameter siehe Tabelle 4 auf Seite 29
uiBit:	Ausgangsbit innerhalb des Signalmoduls. Mögliche Parameter siehe Tabelle 4 auf Seite 29
xValue:	Eingangsparameter vom Typ „Bit“
WriteBitDirect:	Anzeige des Fehlercodes, siehe Tabelle 9 auf Seite 33

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 Ausgangswortes erfolgt am Zyklusende.

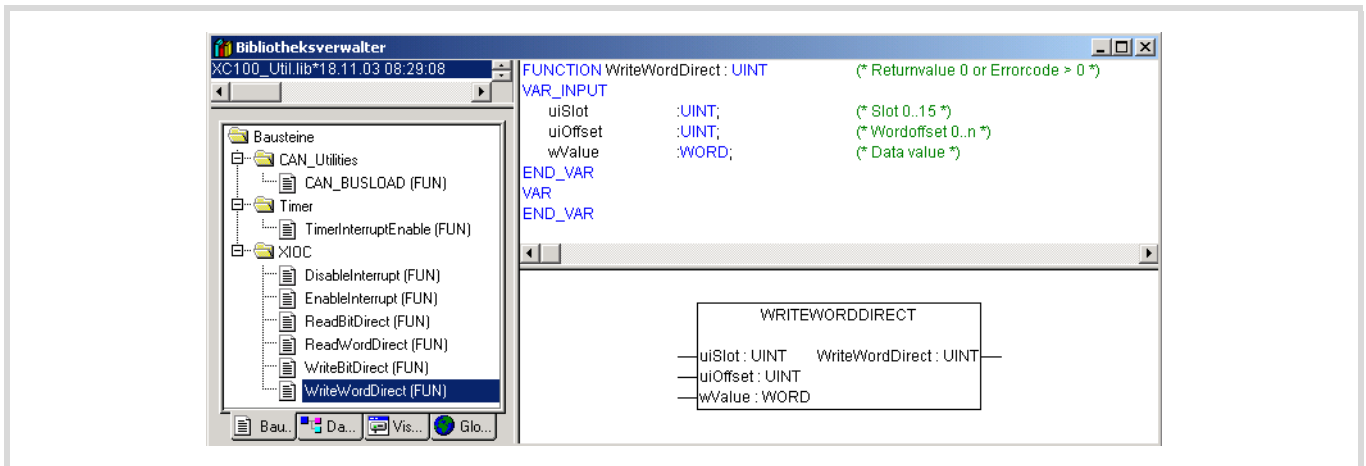


Abbildung 44: Funktion WriteWordDirect

Tabelle 8: Parameter der Funktion „WriteWordDirect“

uiSlot:	Steckplatznummer des Signalmoduls. Mögliche Parameter siehe Tabelle 4 auf Seite 29
uiOffset:	Ausgangswort innerhalb des Signalmoduls. Mögliche Parameter siehe Tabelle 4 auf Seite 29
wValue:	Eingangsparameter vom Typ „Word“
WriteWordDirect:	Anzeige des Fehlercodes, siehe Tabelle 9 auf Seite 33

Fehlercode bei „Direkter 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 (→ Tabelle 9) ausgegeben. Die Datenfelder für die Wertübergabe bleiben unverändert.

Tabelle 9: Fehlercodes beim direkten Peripherie-Zugriff im Datentyp IO_ACCESS_NO_ERROR

IO_ACCESS_NO_ERROR:	0: kein Fehler
IO_ACCESS_INVALID_SLOTNUMBER	1: Steckplatz = 0 oder größer 7
IO_ACCESS_INVALID_OFFSET	2: Bit-/Word-Offset ist zu groß
IO_ACCESS_DENIED	3: ungültiger Zugriff, z. B. Schreibzugriff auf Eingangsmodul, Lesezugriff auf Ausgangsmodul oder Zugriff auf nicht vorhandenen Adressbereich (Offset zu groß)
IO_ACCESS_NO_MODULE	4: Kein Modul auf parametrisiertem Steckplatz vorhanden
IO_ACCESS_INVALID_VALUE	4: Ergebnis ist nicht „0“ oder „1“ bei „WriteBitDirect“
IO_ACCESS_INVALID_BUFFER	5: Kein oder falscher Zeiger auf die Ausgangsvariable

Interrupt-Verarbeitung

In der XC100 können Sie bis zu sieben Interrupt-Ereignisse programmieren und parametrieren. Interrupts können ausgelöst werden durch:

- physikalischen Eingänge I0.0 bis I0.3 der XC-CPU101,
- interruptfähige XIOC-Signalmodule,
- TIMER_Interrupt.

Die in der Abbildung 45 aufgeführten Interrupt-Events stehen Ihnen zur Verfügung:

Beim Auftreten eines Interrupts führt das Laufzeitsystem die mit der Interruptquelle verknüpfte Programm-Organisations-Unit (POU) aus. Die Ausführung der POU wird zeitlich überwacht. Hierfür wird die parametrisierte Zykluszeit verwendet. Die Interrupts werden beim Zustandwechsel nach RUN 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 Zyklus kann zum Überschreiten der programmierten Zykluszeit führen und einen Reset durch den Watchdog auslösen.

Anwender-Interrupts können aus dem Programm heraus gesperrt und wieder freigegeben werden. Hierzu gibt es die Funktionen „DisableInterrupt“ und „EnableInterrupt“. Ein Aufrufparameter in der easySoft-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 „Enable Interrupt“ sind Bestandteil der Library „XC100_Util.lib“. Diese Library müssen Sie – falls noch nicht vorhanden – in den Bibliotheksverwalter der easySoft-CoDeSys einbinden.

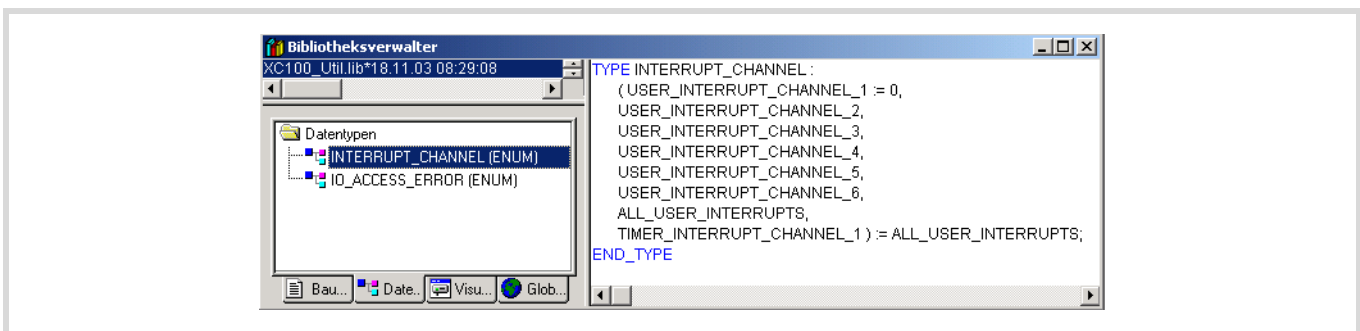


Abbildung 45: Anwender-Events

Interrupt-Priorisierung

Einem physikalischen Interrupt-Eingang I0.0, I0.1, I0.2 oder I0.3 weisen Sie bei der Taskkonfiguration einen Interrupt-Kanal zu. Es stehen 6 Kanäle zur Verfügung. Die Kanäle 1 bis 3 haben eine hohe Priorität, die Kanäle 4 bis 6 haben eine niedrige Priorität.

Niederpriorige Interrupts können von höherpriorigen unterbrochen werden.

Timer-Interrupt

Zusätzlich zu den sechs verschiedenen Interrupt-Kanälen für interruptfähige Eingänge steht ein Interrupt-Kanal für ein Timer-Interrupt zur Verfügung. Die Startbedingung und die Sollwertvorgabe erfolgt in der easySoft-CoDeSys applikationsbezogen.

Die nachfolgende Abbildung zeigt die Funktion und die Parametrierung für den „Timer-Interrupt“. Diese Funktion ist in der „XC101_Util.lib“ enthalten.

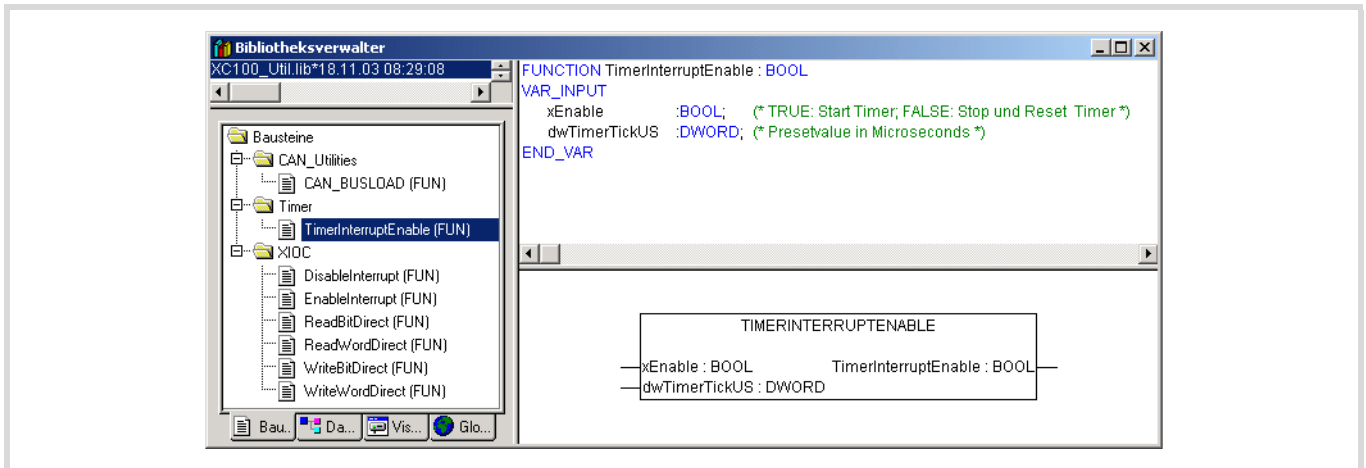


Abbildung 46: Funktion „Timer-Interrupt“

Die Sollwertvorgabe erfolgt am Eingang „dwTimerTickUS“. Der Minimalwert beträgt 500, der Maximalwert 2500000 Mikrosekunden. Wird am Bausteineingang „dwTimerTickUS“ der untere Grenzwert unter- bzw. der obere Grenzwert überschritten, liefert der Funktionsaufruf als Rückgabewert FALSE und der Timer wird nicht gestartet. Der Sollwert wird beim Start des Timers übernommen und kann zur Laufzeit nicht geändert werden.

Die Verknüpfung des Timer-Interrupts mit dem dazu gehörendem IEC-Programm erfolgt wie bei den IO-Interrupts über <Taskkonfiguration → System-Ereignisse>.

Die eingetragene Interrupt-Funktion wird unmittelbar bei Auftreten des Interrupts ausgeführt. Der laufende IEC-Programmzyklus wird an beliebiger Stelle unterbrochen.

Der Timer-Interrupt kann aber selbst durch höher priorie System-Interrupts wie z. B. CAN unterbrochen werden. Während der Ausführung des Timer-Interrupts ist die Zykluszeitüberwachung aktiv. Die Zykluszeitüberwachung orientiert sich ausschließlich an dem eingetragenen Wert für die maximale Programm-Zykluszeit. Treten zu häufige Timer- und IO-Interrupts auf, kann dies zur Überschreitung der Programm-Zykluszeit führen. Wird die Programm-Zykluszeit überschritten, wechselt die XC-CPU101 vom Betriebszustand RUN in STOP.

Den Timer-Interrupt können Sie aus dem Anwenderprogramm heraus sperren und freigeben. Verwenden Sie hierfür die Funktionen „DisableInterrupt“ und „EnableInterrupt“.

DisableInterrupt

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

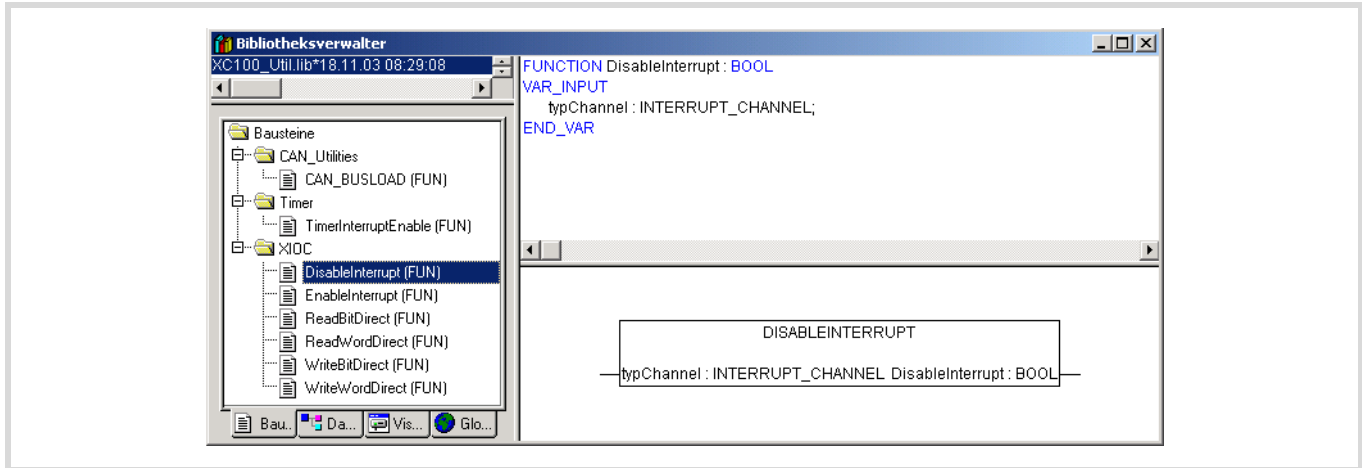


Abbildung 47: Funktion DisableInterrupt

EnableInterrupt

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

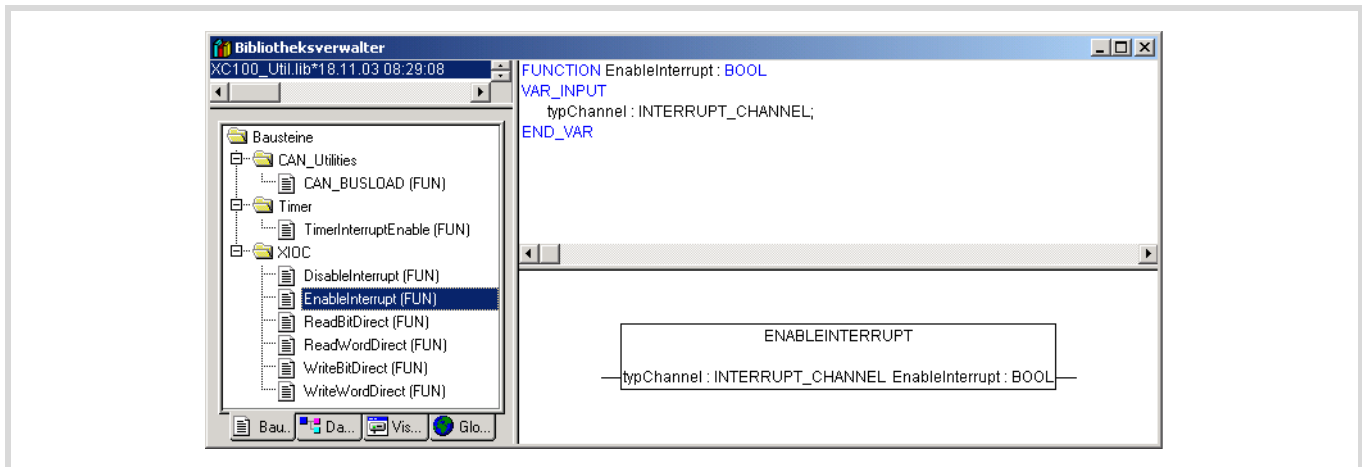


Abbildung 48: Funktion EnableInterrupt

Interrupt-Funktion erstellen und einbinden

Der formale Ablauf für die Erstellung und Einbindung einer Interruptfunktion ist nachfolgend in einzelnen Schritten beschrieben.

Im Beispiel soll ein H-Signal am Eingang I0.0 in einen Interrupt-Baustein verzweigen und diesen abarbeiten.

- Hierzu erstellen Sie einen Programmbaustein für die normale Applikation („PLC-PRG“) und einen weiteren Baustein mit der Interrupt-Funktionalität „Interrupt_1“.

Die folgende Abbildung zeigt beide Bausteine:

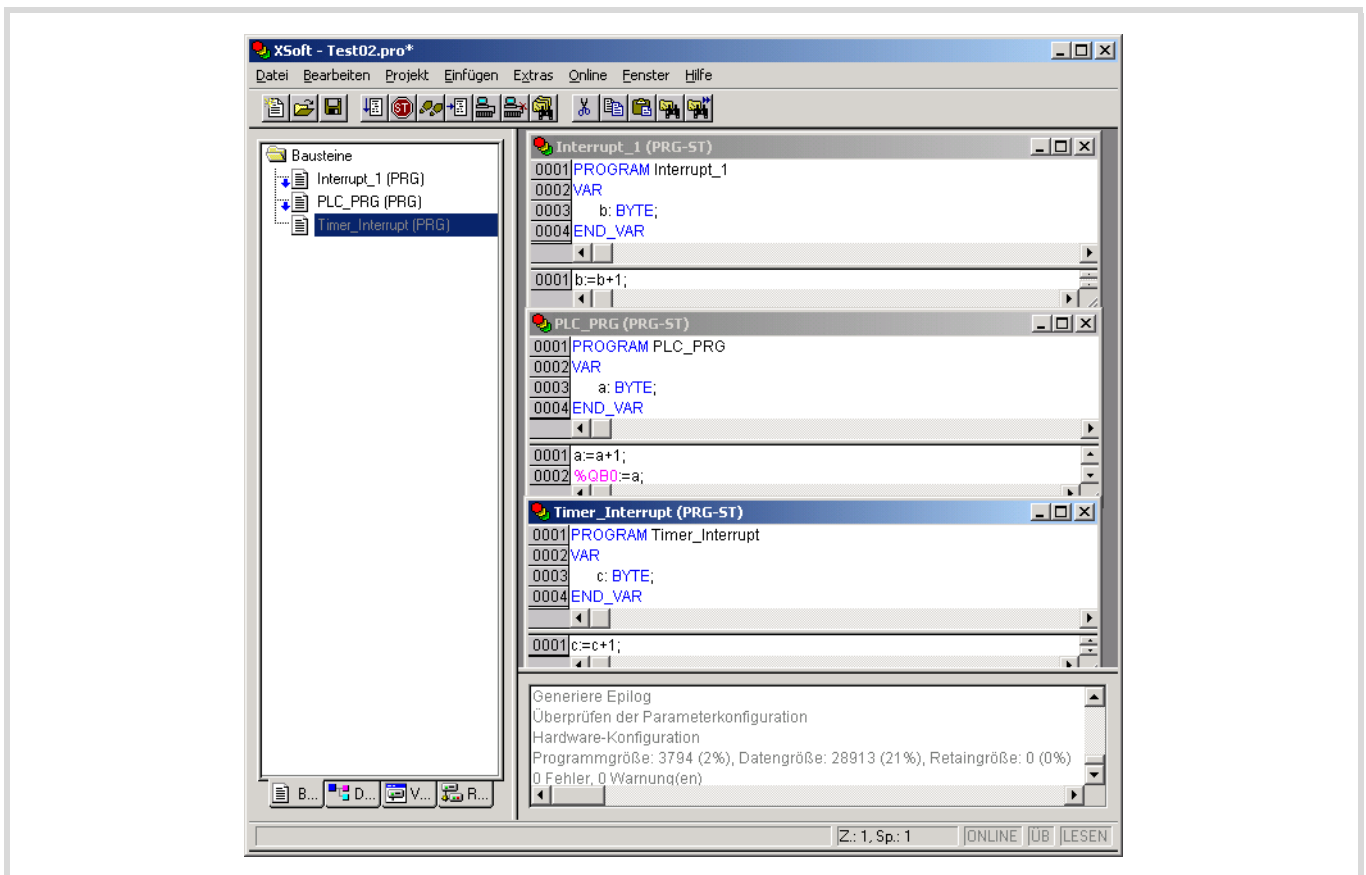


Abbildung 49: Bausteine „PLC_PRG“ und „Interrupt_1“

- Wechseln Sie dann in die Steuerungskonfiguration und weisen Sie dem Eingang I0.0 den Interrupt_1 aus dem Listenfeld zu.

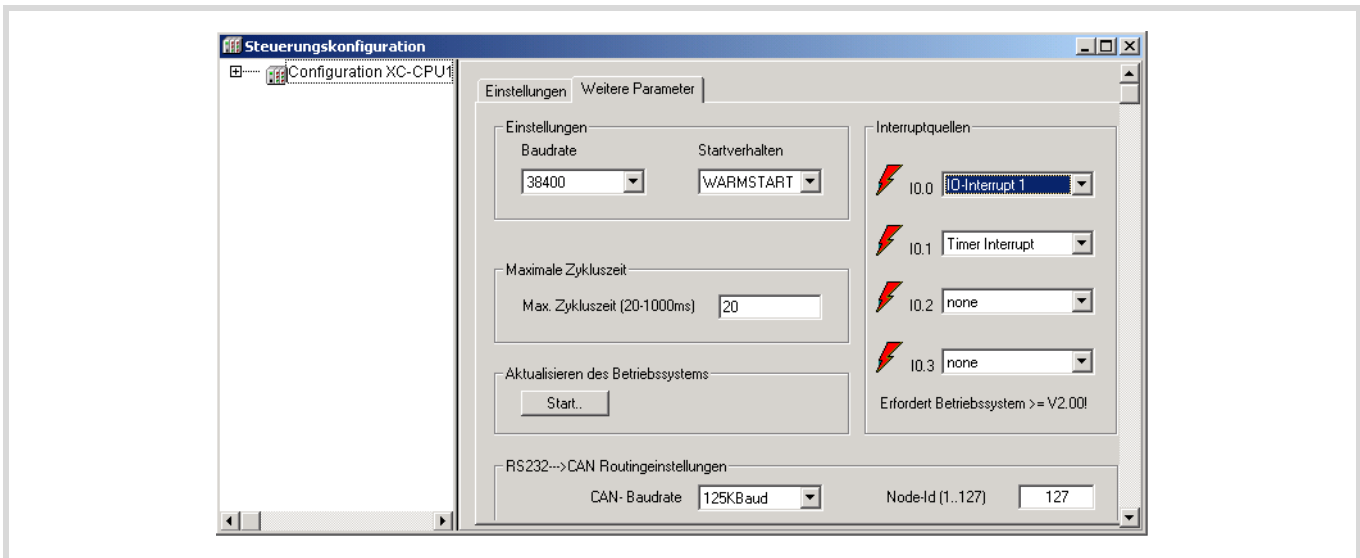


Abbildung 50: Eingang IO.0 den Interrupt 4 zuweisen

- ▶ Wechseln Sie in die Taskkonfiguration und markieren Sie im Eingabefeld „System-Ereignisse“ das Kontrollkästchen „IO-Interrupt1“.
- ▶ Gehen Sie jetzt in der gleichen Zeile bis zur Spalte „aufgerufene POU“, markieren dieses Feld mit der linken Maustaste und drücken die Funktionstaste „F2“.

Das Fenster „Eingabehilfe“ wird geöffnet, in dem alle vordefinierten Programme gelistet sind:

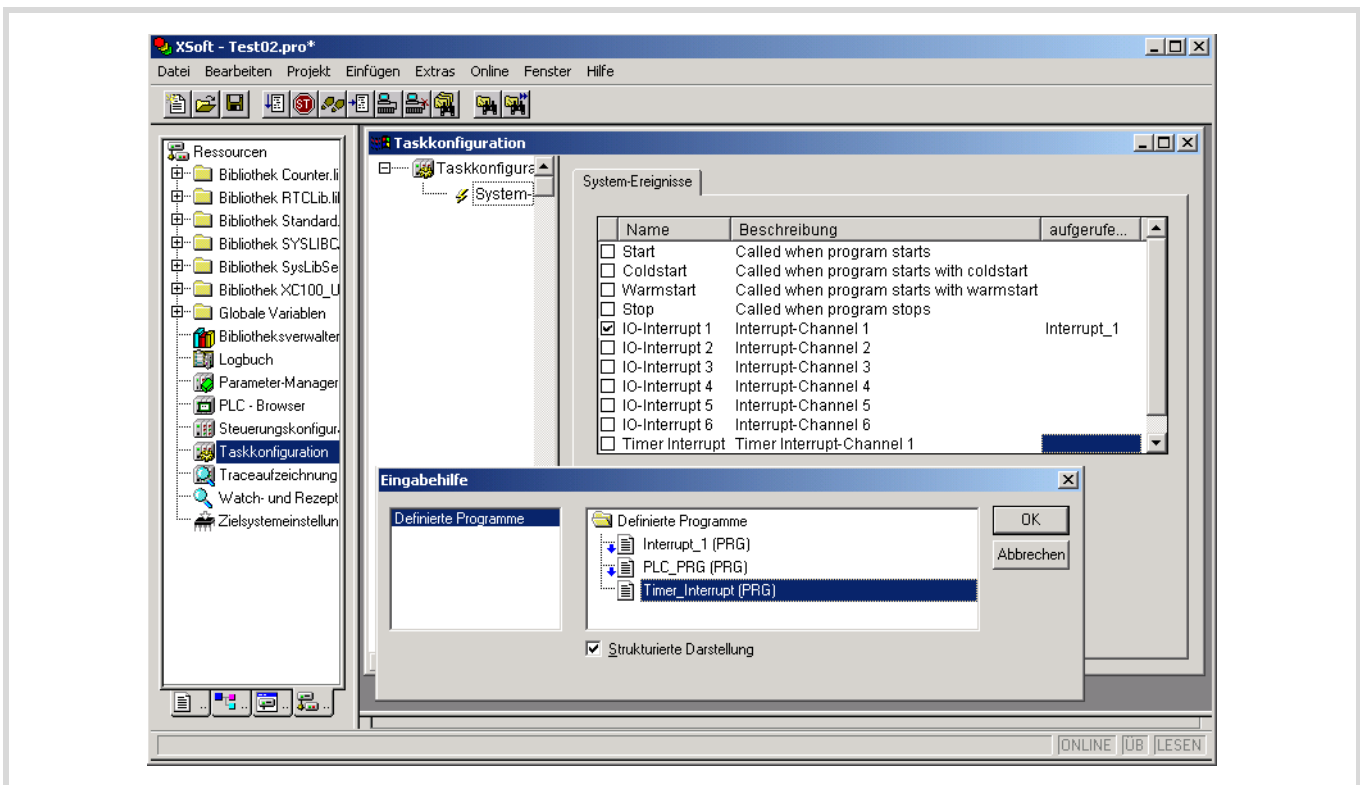


Abbildung 51: Interruptquellen parametrieren

- Wählen Sie die POU „Interrupt_1 (PRG)“ mit einem Doppelklick aus und anschließend die POU „Timer_Interrupt (PRG)“.

Sie erhalten das nachfolgende Fenster:

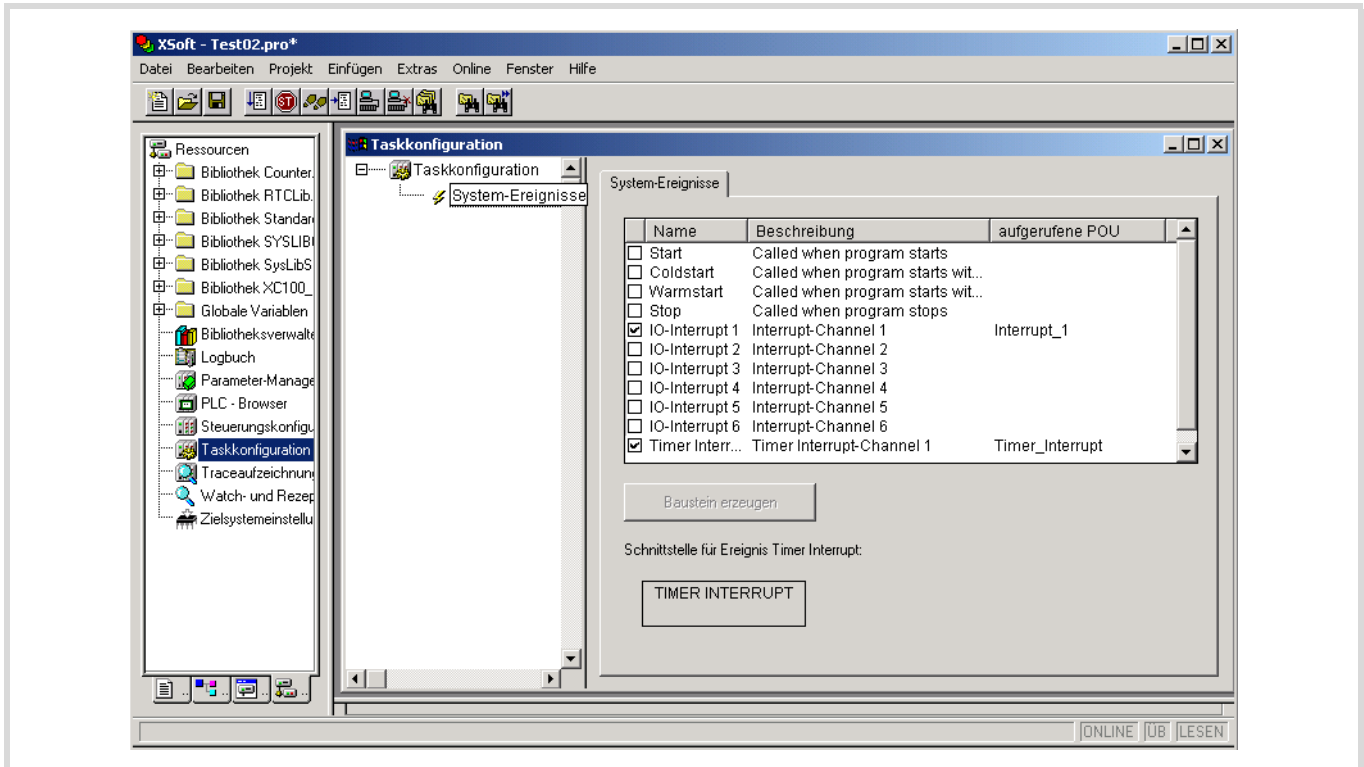


Abbildung 52: Fertige Taskkonfiguration Interrupt-Baustein

- Speichern Sie das erstellte Programm, übersetzen dieses, loggen sich in die Steuerung ein und testen den von Ihnen erstellten Programmbaustein auf seine Funktion.

Systemereignisse

Anstelle einer Task kann auch ein Systemereignis (Event) einen Projektbaustein zur Abarbeitung aufrufen. Die dazu verwendbaren Systemereignisse sind zielsystemabhängig. Sie setzen sich zusammen aus den Standardsystem- und den zielsystemabhängigen Ereignissen. Mögliche Ereignisse sind zum Beispiel:

- Stop
- Start
- Coldstart
- Warmstart
- IO-Interrupt
- Timer-Interrupt
- Online-Änderung.

siehe auch Abbildung 52.

→ Der Einzelschritt-Betrieb ist bei den Programmbausteinen der Systemereignisse nicht möglich.

Browserbefehle

Mit den Browser-Befehlen können Sie direkt auf Zustände/Ereignisse in der XC100 zugreifen. Die Online-Beschreibung in der easy-Soft-CoDeSys finden Sie unter <Ressourcen → PLC-Browser>.

Befehl	Beschreibung
?	Liste der implementierten Kommandos holen.
reflect	Aktuelle Kommandozeile spiegeln, zu Testzwecken
mem	memory-dump, Syntax: mem <start-addr> <end-addr>
memc	wie mem, Adressen werden zur Startadresse des Code-Bereichs addiert
memd	wie mem, Adressen werden zur Startadresse des Datenbereichs addiert.
pinf	Projektinformationen ausgeben
ppt	Bausteinzeiger-Tabelle ausgeben
dpt	Datenzeiger-Tabelle ausgeben
pid	Projekt-ID ausgeben
cycle	Zykluszeit ausgeben

Befehl	Beschreibung
GetNodeId	CANopen Node ID ausgeben
SetNodeId	CANopen Node ID setzen
metrics	SPS-Informationen ausgeben
reload	Bootprojekt von der MMC auf die SPS laden
remove	Bootprojekt von der MMC löschen
format	Formatieren der MMC
getswitchpos	Schalterstellung ausgeben
getbattery	Batteriezustand ausgeben
getrtc	Echtzeituhr auslesen [HH:MM:SS]
setrtc	Echtzeituhr setzen [HH:MM:SS]
canload	Zeigt die Auslastung des CANopen-Feldbusses an.

Browserbefehl „canload“

Die Funktion „canload“ ist in der „XC100_Util.lib“ enthalten. Sie können Sie als Browserbefehl aufrufen. Bei diesem Befehl wird die momentane Belastung des lokalen CANopen-Bus ermittelt. Über eine Integrationszeit wird das Datenaufkommen des Busses ermittelt und in Relation zur CAN-Baudrate die Busbelastung ermittelt.

Nach Aufruf des Browserbefehls erhalten Sie beispielsweise folgende Information:

CAN-Busload = 51 Percent

Baudrate: 125 Kbaud

Integration-Time: 504 ms.

Bei einer Busbelastung von 75 Prozent oder mehr erscheint zusätzlich noch der Warnhinweis ATTENTION: HIGH BUSLOAD.



Achtung!

Die Überlastung des lokalen CAN-Busses in Zusammenhang mit weiteren kurzzeitigen Lastspitzen kann zu CAN-Datenverlust führen.

Zusätzlich zum Browserbefehl steht ein Funktionsaufruf für den Funktionsbaustein CAN_BUSLOAD zur Ermittlung der CAN-Busauslastung aus dem Anwenderprogramm heraus zur Verfügung.

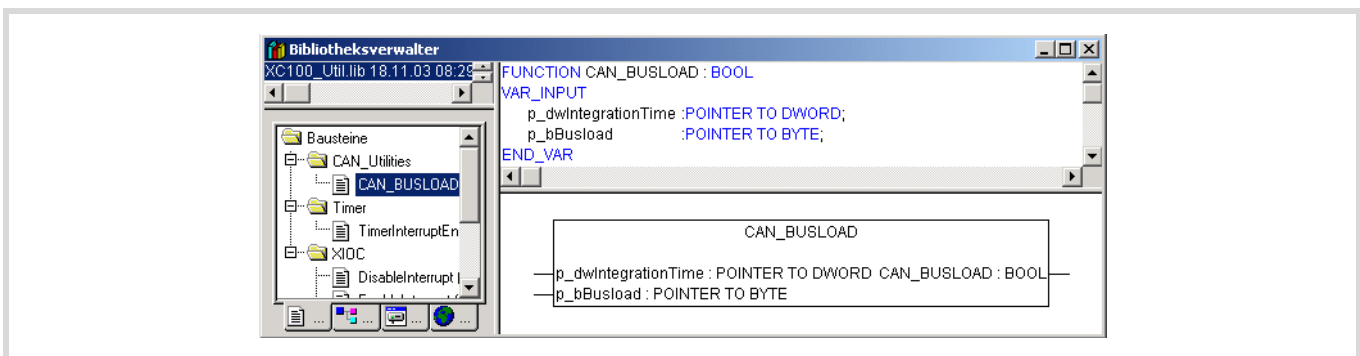


Abbildung 53: Funktion „Can_Busload“

Datenremanenz

Die Steuerung besitzt einen Speicherbereich für remanente Daten → Seite 10. Die mit „VAR RETAIN“ deklarierten Variablen werden in diesem Bereich abgelegt und sind somit beim Warmstart des Applikationsprogramms remanent (Achtung: Gilt nicht für I-, Q- und M-Variable!). Kaltstart-remanente Daten „VAR RETAIN Persistent“ werden nicht unterstützt. Beim Abschalten der Steuerung ist bei gesteckter Batterie die Datenremanenz gewährleistet.

Kann bei einem Spannungsaus/-abfall ein laufender Zyklus nicht beendet werden, sind die Daten nicht konsistent, da der laufende Zyklus an beliebiger Stelle abgebrochen wird. Bei Spannungswiederkehr wird der Restzyklus nicht beendet. Die Steuerung startet entsprechend dem eingestellten Startverhalten.

Soll dies ausgeschlossen werden, sind entsprechende Maßnahmen zu projektieren. Hierfür eignet sich z. B. eine unterbrechungsfreie Stromversorgung mit zusätzlicher Akku-Pufferung.

Programmtransfer

Der Transfer eines Applikationsprogramms erfolgt immer in den batteriegepufferten SRAM-Bereich der Steuerung. Anschließend kann mit dem Kommando „Bootprojekt erzeugen“ ein Backup auf der Multi Media Card erzeugt werden. Das Anlegen eines Programm-Backup ist nur im Zustand „HALT“ möglich. Der interne FLASH-Speicher der CPU kann nicht als Speicher für ein Programm-Backup verwendet werden.

Betriebszustände

In der folgenden Übersicht finden Sie die Zustandsdefinitionen der XC100. Die LED-Anzeigen der jeweiligen Zustände sind mit aufgeführt.

Zustand	Anzeige		Definition
	RUN/STOP	SF	
System-Test	aus	aus	System-Test wird durchgeführt
System-Update	ein	ein	System-Update wird durchgeführt
Einschalten OK	aus	aus	System-Test ohne Fehler abgeschlossen
Einschalten nicht OK	blinken	blinken	System-Test ergab einen Fehler
NOT READY	aus	ein	kein Applikationsprogramm vorhanden
STOP	blinken	aus	Applikationsprogramm geladen, SPS im Zustand „STOP“
RUN	ein	aus	Applikationsprogramm geladen, SPS im Zustand „RUN“
STOP/RUN mit Sammelfehler/ Diagnosemeldung	ein	ein	Sammel-Fehler/Diagnosemeldung steht an

Grenzwerte für die Speichernutzung

Der Datenspeicher der XC100 ist in Speichersegmente gegliedert. Die Speichergröße der einzelnen Segmente geht aus Abbildung 54 hervor. Die globalen Daten verfügen über mehrere Segmente. Die gewünschte Anzahl können Sie in Abhängigkeit von der Programmgröße festlegen.

Die Segmentgröße für die verschiedenen Steuerungstypen erkennen Sie aus den <Zielsystemeinstellungen → Speicher-aufteilung>:

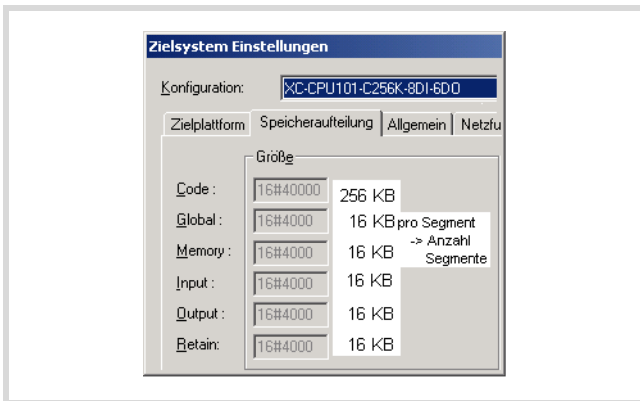


Abbildung 54: Segmentgröße der XC-CPU101-C256k

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-CPU101-C64K-8DI-6DO	16
XC-CPU101-C128K-8DI-6DO	12
XC-CPU101-C256K-8DI-6DO	14

Defaultmäßig steht die Anzahl der Segmente auf 1.

Die Anzahl der Segmente verändern Sie wie folgt:

- Wählen Sie im Verzeichnis <Projekt → Optionen → Übersetzungsoptionen> das Feld Datensegmente an und tragen Sie entsprechend dem ausgewählten Steuerungstyp die oben angegebene Anzahl der Segmente ein.

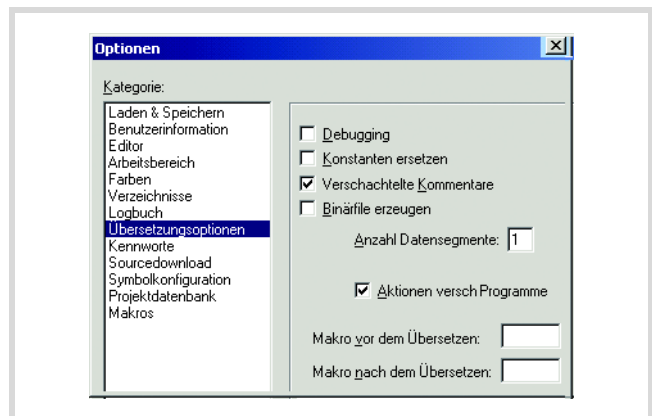


Abbildung 55: Speicherverwaltung: Anzahl der Datensegmente ändern

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.

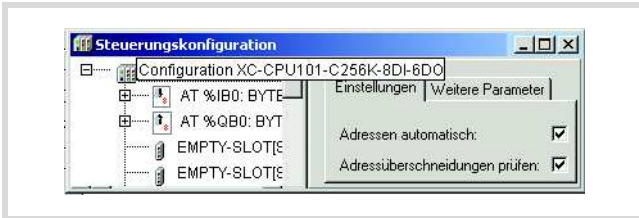


Abbildung 56: Defaulteinstellung zur Adressierung

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.

„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-Slave oder CAN-Teilnehmer 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“, leiben die Adressen bei Änderungen/Erweiterungen bestehen.

„Adressüberschneidungen prüfen“ aktivieren

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

Ungerade Wort-Adressen

(Unabhängig von der Einstellung „Adressüberschneidung prüfen“)

Wenn Sie einem wortadressierbaren Modul im Feld „Eingabeadresse“ eine ungerade Offset-Adresse zuordnen, z. B. IB3, erscheint im Steuerungskonfigurator automatisch die folgende gerade Word-Adresse (IW4).

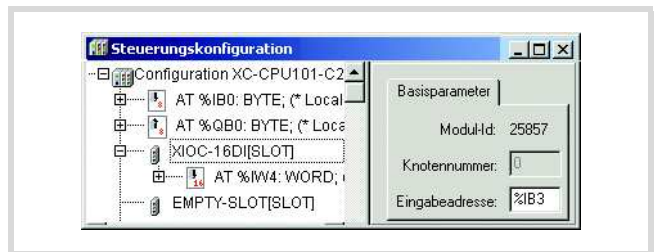


Abbildung 57: Ungerade Adressen

Adressbereich

Adressen können Sie nur innerhalb der gültigen Bereiche vergeben. Die Bereichsangaben finden Sie unter <Zielsystem-einstellungen → Speicheraufteilung → Größe>.

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

Tabelle 10: Adressbereiche

Steuerung	Input			Output			Merker		
	Größe	Max. Byteadr.	Max. Wordadr.	Größe	Max. Byteadr.	Max. Wordadr.	Größe	Max. Byteadr.	Max. Wordadr.
XC100-64k	2k	2047	2046	2k	2047	2046	4k	4095	4094
XC100-128k	4k	4095	4094	4k	4095	4094	8k	8191	8190
XC100-256k	16k	16383	16382	16k	16383	16382	16k	16383	16382
XC200-256k	4k	4095	4094	4k	4095	4094	16k	16383	16382
XC200-512k	4k	4095	4094	4k	4095	4094	16k	16383	16382

Adressen von Eingabe-/Ausgabe-Module und Diagnose-adresse 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.

„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 „Adressen berechnen“ durchführen, 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.

Diagnose

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

Art der Diagnose	Funktionsbaustein	Bibliothek	Dokumentation
Überprüfung der XI/OC-Module: <ul style="list-style-type: none"> • Stimmt die Konfiguration im Konfigurator mit der Hardware überein? • Ist die Modulfunktion ok? Bemerkung: Diese Überprüfungen erfolgen einmalig beim Einschalten oder nach dem Laden/Start des Programms	XDiag_SystemDiag	xSysDiagLib	MN05010002Z-DE; frühere Bezeichnung AWB2786-1456D
Überprüfung des Moduls XIOC-NET-DP-M und der Teilnehmer am DP-Strang	XDiag_SystemDiag XDiag_ModuleDiag	XSysDiagLib	MN05010002Z-DE; frühere Bezeichnung AWB2786-1456D
	DiagGetState	BusDiag	MN05002002Z-DE; frühere Bezeichnung AWB2725-1452D
Überprüfung des Moduls XIOC-NET-DP-S	XDiag_SystemDiag XDiag_ModuleDiag	xSysDiagLib	MN05010002Z-DE; frühere Bezeichnung AWB2786-1456D
DP-Slave stellt dem Master zusätzliche Diagnosedaten zur Verfügung	XDPS_SendDiag	xSysNetDPSDiag	MN05002002Z-DE

5 Verbindungsaufbau PC – XC100

In diesem Abschnitt werden die erforderlichen Maßnahmen beschrieben, um den PC als Programmiergerät (Hard- und Software) mit der XC100 zu verbinden.

Verbindungsaufbau über RS232-Schnittstelle (XC100)

Die Kommunikation erfolgt über die serielle, nicht opto-entkoppelte RS232-Schnittstelle. Als PC-Schnittstelle können Sie die COM1- oder COM2-Schnittstelle verwenden. Zur Herstellung der physikalischen Verbindung verwenden Sie bitte das Programmierkabel XT-SUB-D/RJ45.

Programmierkabel

Der Aufbau des Programmierkabels ist wie folgt:

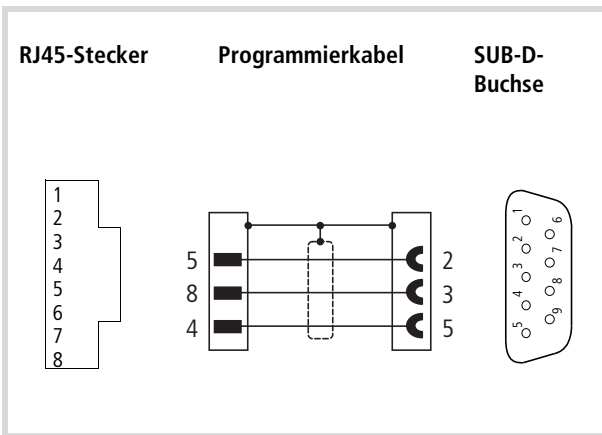


Abbildung 58: Aderbelegung RS232-Programmierkabel

Software easySoft-CoDeSys

Über die easySoft-CoDeSys legen Sie die Kommunikationsparameter fest:

- ▶ Rufen Sie in der easySoft-CoDeSys den Menüpunkt «Online → Kommunikationsparameter» auf und wählen Sie die COM1- oder COM2-Schnittstelle aus.
- ▶ Wählen Sie die in der Abbildung 59 eingetragenen Werte vor.

Die Default-Werte können Sie verändern, indem Sie auf den Werte-Eintrag einen Doppelklick ausführen.

→ Weitere Hinweise zu den Kommunikationsparametern finden Sie im Handbuch zur Programmiersoftware (AWB2700-1437D).

Kommunikationsfehler(#0): Es wird ausgeloggt

Kann zwischen dem Programmier-PC und der XC100 keine Verbindung aufgebaut werden, überprüfen Sie:

- die physikalische Verbindung,
- die Baudrate der Kommunikationsparameter in der easySoft-CoDeSys,
- die Baudrate in der XC100 (defaultmäßig auf 38400 kBit/s eingestellt),
- die Baudrate der CAN-Verbindung, falls der CAN-Feldbus verwendet wird,
- ob die Schnittstellenparameter in der easySoft-CoDeSys und in der XC100 übereinstimmen!

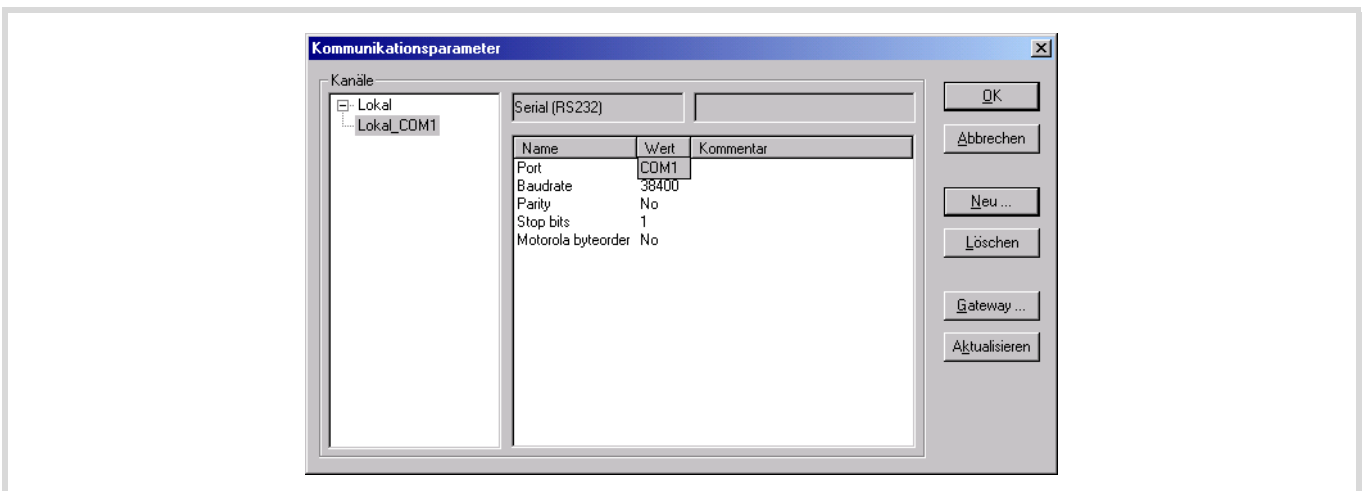


Abbildung 59: Vorwahl Kommunikationsparameter

6 Beispielprojekt erstellen

Im folgendem Beispiel lernen Sie den Umgang mit der Software easySoft-CoDeSys kennen. Sie erstellen ein Projekt, indem Sie eine Konfiguration und ein Programm erstellen. Nach dem Download des Projekts können Sie das Programm testen.

Die Konfiguration wird mit dem Konfigurationseditor der easySoft-CoDeSys erstellt. Es wird zwischen lokalen und zentralen E/As differenziert:

Die lokalen (digitalen) Ein-/Ausgänge sind Bestandteil des CPU-Moduls und sind auf der Netzteilplatine angeordnet. Sie sind im Konfigurationseditor bereits vorkonfiguriert.

Die zentralen Ein-/Ausgänge befinden sich auf den Signalmodulen, die über die Baugruppenträger eingebunden werden können. Konfigurieren Sie diese entsprechend der Applikation hinzu. Es können die zur Verfügung stehenden XIOC-Signal-Module eingesetzt werden.

An die CANopen-Schnittstelle sind entsprechend geeignete CANopen-Feldbus-Teilnehmer anzuschließen.

Basis für die Konfiguration ist der folgende Hardware-Aufbau.

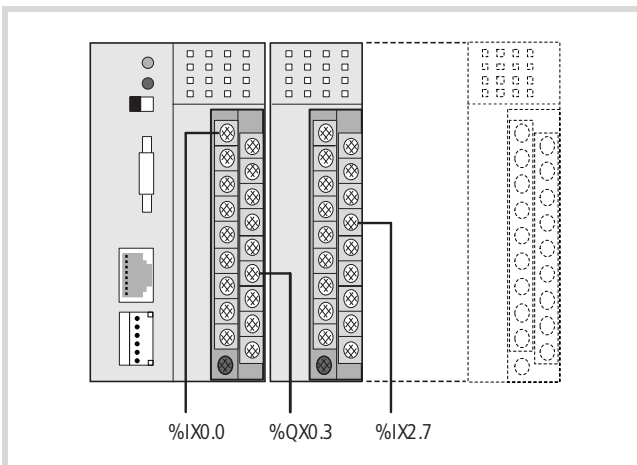


Abbildung 60: Hardware-Aufbau für das Beispielprojekt

Aufgabe

Verknüpfen Sie die Eingänge I0.0 und I2.7 der XC100 mit „AND“. Das Verknüpfungsergebnis soll am Ausgang Q0.3 dargestellt werden.

Im zweiten Schritt werden Ein- und Ausgänge über einen CAN-Master eingelesen bzw. ausgegeben.

Bevor Sie das CANMaster[VAR]-Modul in die Steuerungskonfiguration einbinden, sind entsprechende CAN-Bibliotheken zu aktivieren. Dieser Ablauf erfolgt nicht automatisch, sondern ist konkret durch den Anwender durchzuführen.

Vorgehensweise

Zielsystem einstellen

Nach dem Start der easySoft-CoDeSys erstellen Sie eine neue Datei:

- ▶ Wählen Sie den Menüpunkt «Datei → Neu» an.
- ▶ Beantworten Sie die Frage nach der Speicherung des alten Projekts.

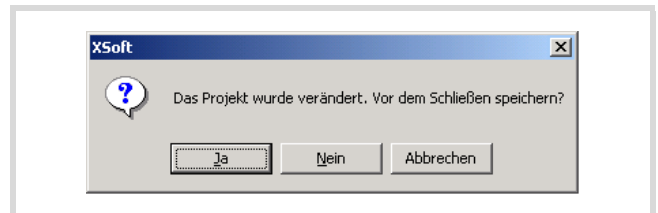


Abbildung 61: Altes Projekt speichern?

Wählen Sie das Zielsystem aus. Im Beispiel wurde das System XC-CPU101-C64K-8DI-6DO selektiert.

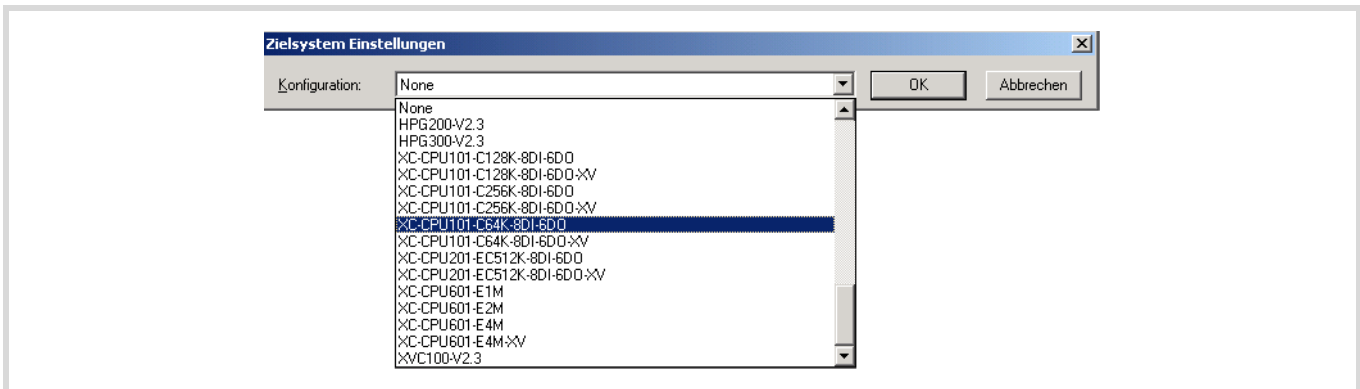


Abbildung 62: Auswahl Zielsystem

Ein Doppelklick auf das Zielsystem führt zu folgender Abbildung. Die Datei-Reiter „Zielplattform“, „Speicheraufteilung“ und „Allgemein“ stellen nur Informationen über das Zielsystem dar. Hier sind keine weiteren Einstellungen möglich.

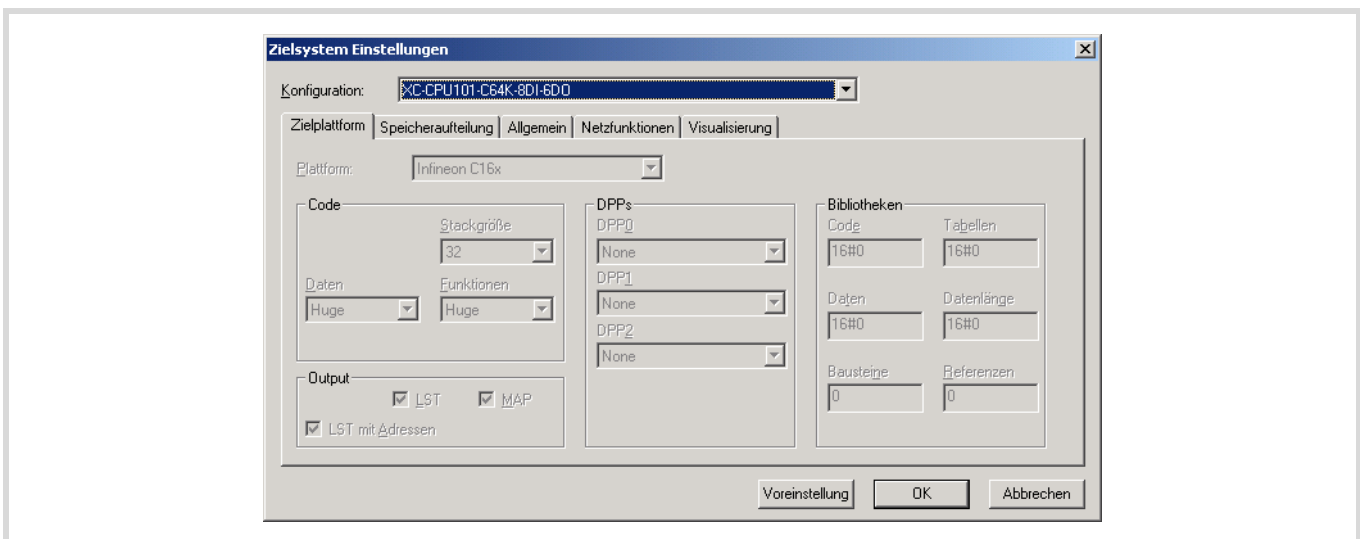


Abbildung 63: Zielsystem Einstellungen – Zielplattform

- Soll ein CANopen-Teilnehmer in die Konfiguration eingebunden werden, markieren Sie bei aktiver Registerkarte „Netzfunktionen“ das Kontrollkästchen „Netzwerkvariable unterstützen“.

Sie erhalten den Hinweis, dass bei diesem Zielsystem das CAN-Netzwerk unterstützt wird.

Das Kontrollkästchen „Netzwerkvariable unterstützen“ müssen Sie nur dann aktivieren, wenn Sie mit Netzwerkvariablen arbeiten wollen. Ein Aktivieren führt dazu, dass automatisch die für den Betrieb der Netzwerkvariablen benötigten Libraries hinzugefügt werden. Wenn Sie CAN-Master/CAN Device verwenden, wird diese Funktion nicht benötigt.

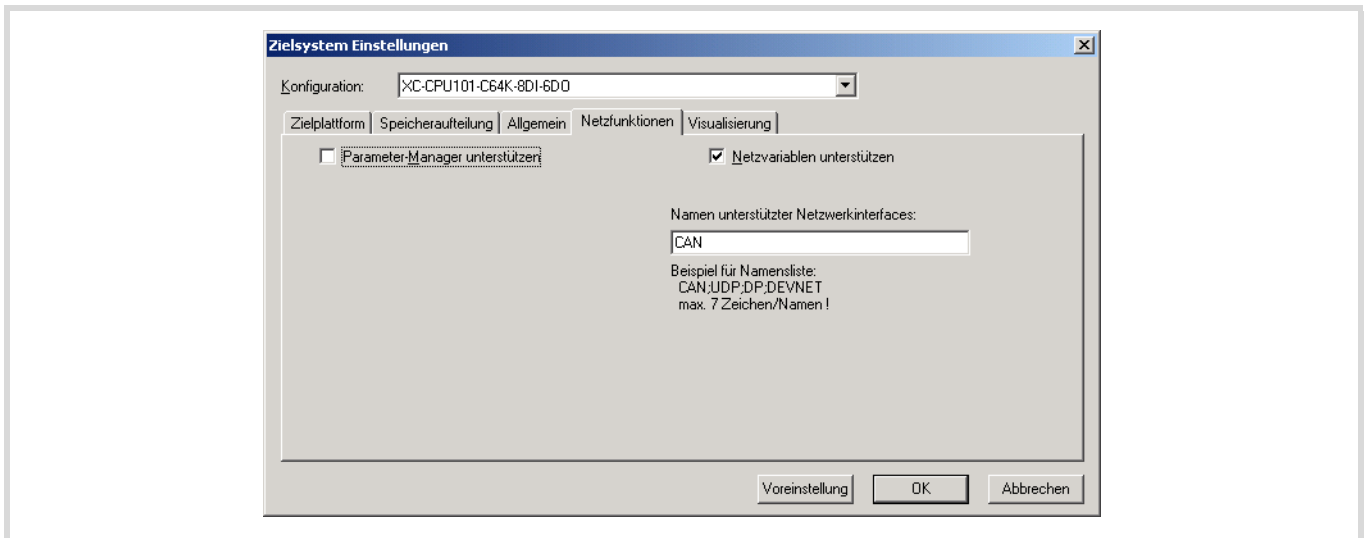


Abbildung 64: Zielsystem Einstellungen – CAN-Netzfunktionen (1)

- ▶ Aktivieren Sie das Kontrollkästchen „Parameter-Manager unterstützen“, um zusätzliche Hinweise zu den Bereichsgrenzen zu sehen.

Der Parameter-Manager wird in Bezug auf CAN nur für CAN Device benötigt. Hierzu sind Standardeinstellungen vorhanden, sodass Sie keine Änderungen vornehmen müssen.

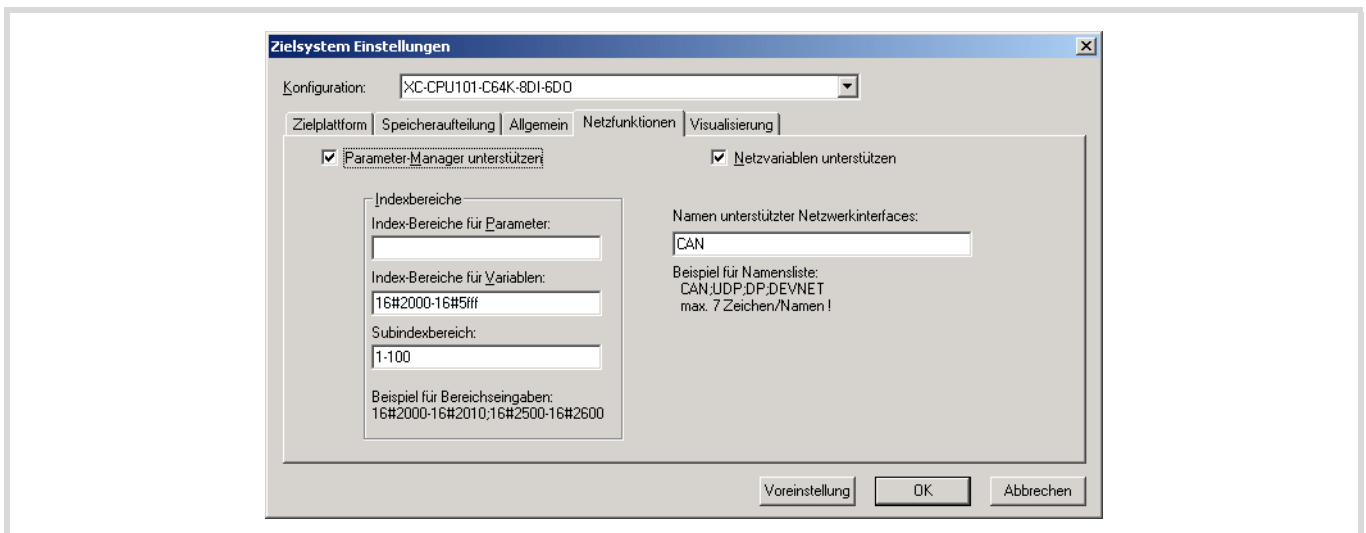


Abbildung 65: Zielsystem Einstellungen – CAN-Netzfunktionen (2)

- ▶ Schließen Sie diese Auswahl mit „OK“ ab.
- ▶ Wählen Sie den Baustein „Programm“ und die Programmiersprache „AWL“ aus:



Abbildung 66: Bausteintyp auswählen

- Bestätigen Sie mit OK und speichern Sie die Datei unter: „beispiel-1“.

Es erscheint ein Fenster, von dem aus Sie nun programmieren oder konfigurieren können:

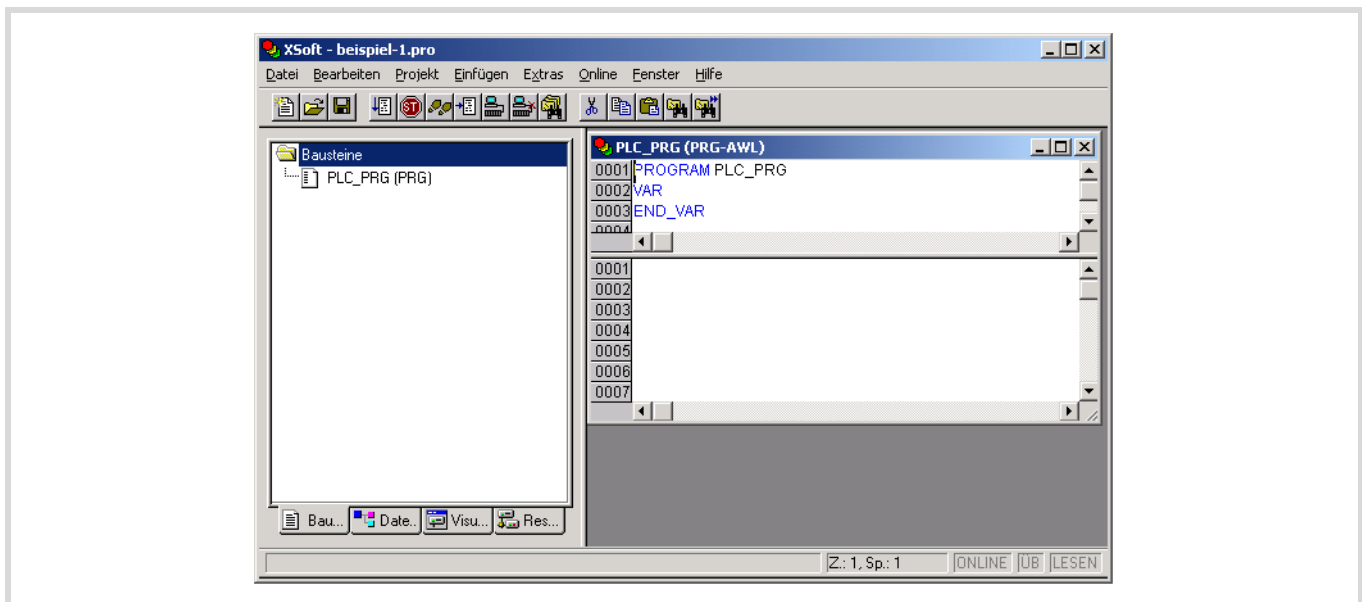


Abbildung 67: Bausteintyp „Programm“ in AWL-Darstellung

XC100-Steuerung konfigurieren

Im Beispiel soll die „XC-CPU101-C64K-8DI-6DO“ eingesetzt werden.

- Wählen Sie das Register „Ressourcen“ (linke Fensterhälfte, unten), um die XC100 mit den lokalen und zentralen Ein-/Ausgängen zu konfigurieren.

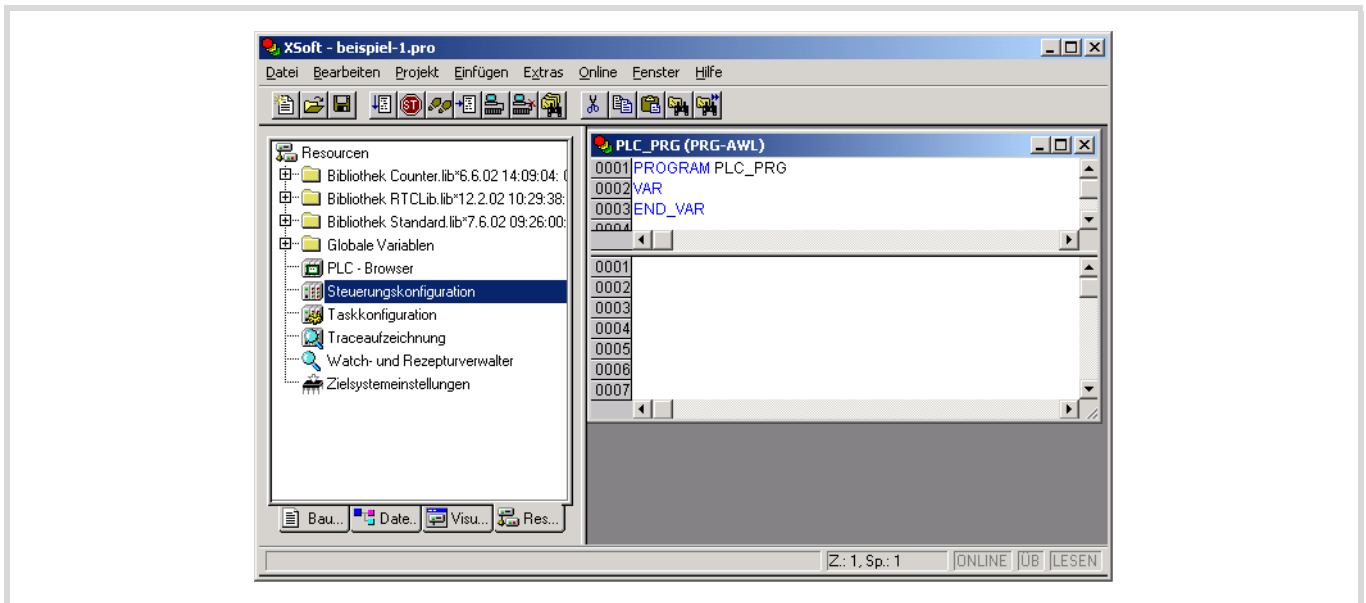


Abbildung 68: XC100-Steuerung konfigurieren

► Klicken Sie doppelt auf den Ordner „Steuerungskonfiguration“.

Es öffnet sich ein weiteres Fenster „Steuerungskonfiguration“.

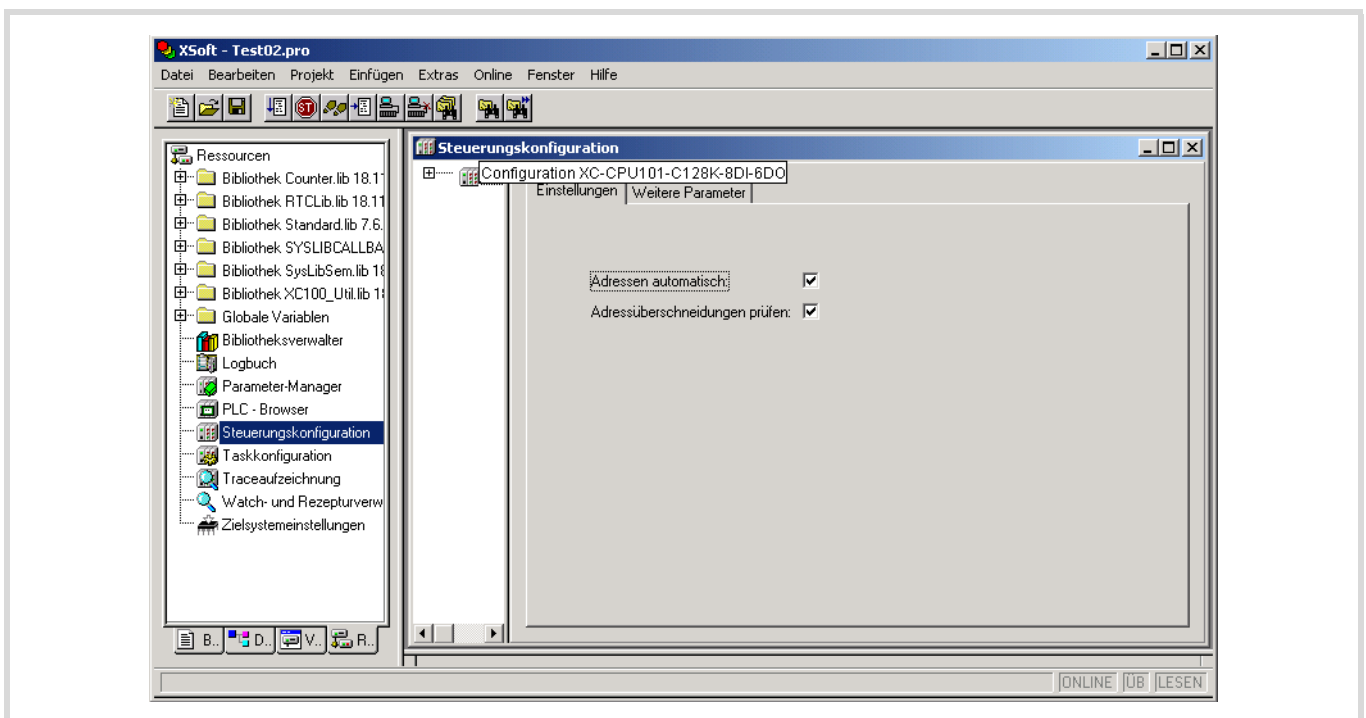


Abbildung 69: Grundkonfiguration der XC100-Steuerung – Einstellungen

► Klicken Sie auf die Registerkarte „Weitere Parameter“.

Es erscheint ein Fenster mit den Default-Werten der „XC-CPU101-C64K-8DI-6DO“.

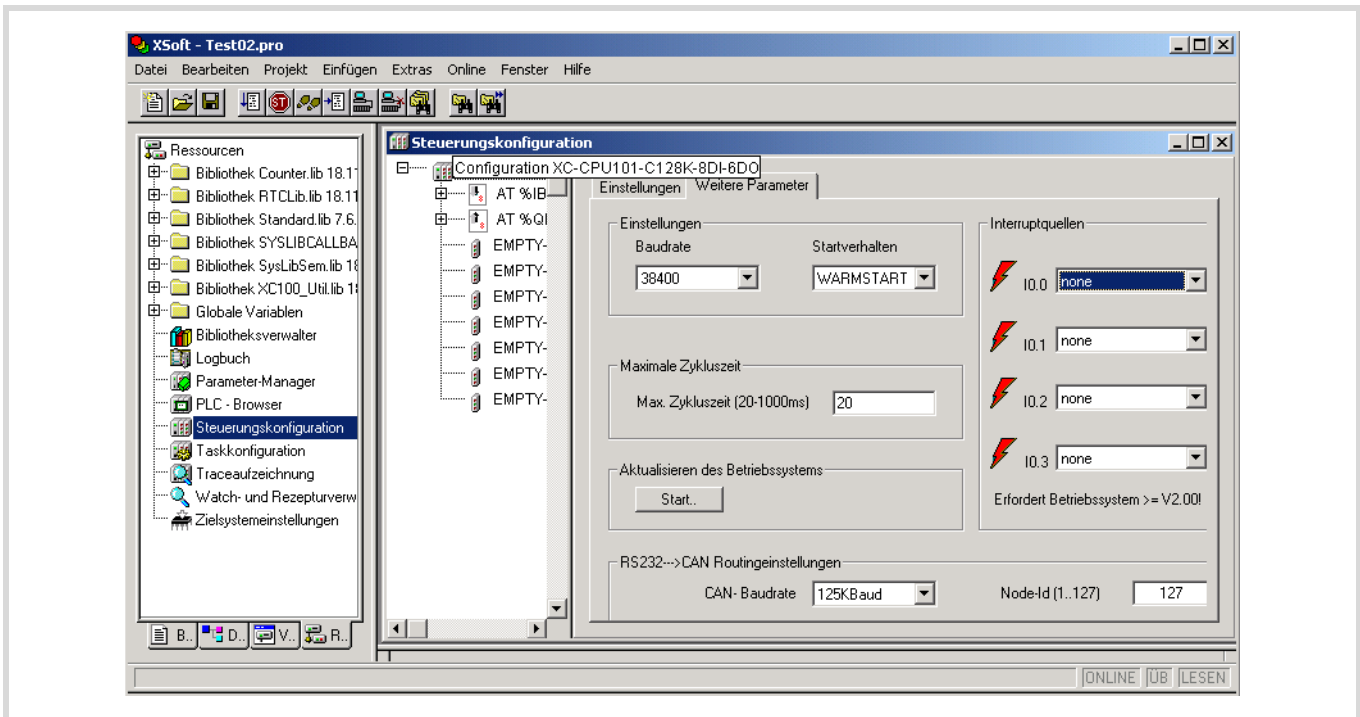


Abbildung 70: Grundkonfiguration der XC100-Steuerung – Weitere Parameter

► Zur Darstellung der E/A-Konfiguration klicken Sie hierfür auf das Pluszeichen vor dem Ordner „XC-CPU101-C64K-8DI-6DO“.

Die lokalen Ein- und Ausgänge (Bestandteil der CPU) sind bereits vorkonfiguriert:

- „AT %IB0;Byte; (* Lokale Eingänge*)“
- „AT %QB0;Byte; (* Lokale Ausgänge*)“

Darüber hinaus können Sie bis zu 15 zentrale Signalmodule parametrieren. Die Steckplätze „EMPTY-SLOT“ sind Platzhalter für die zentrale Erweiterung mit Signalmodulen.

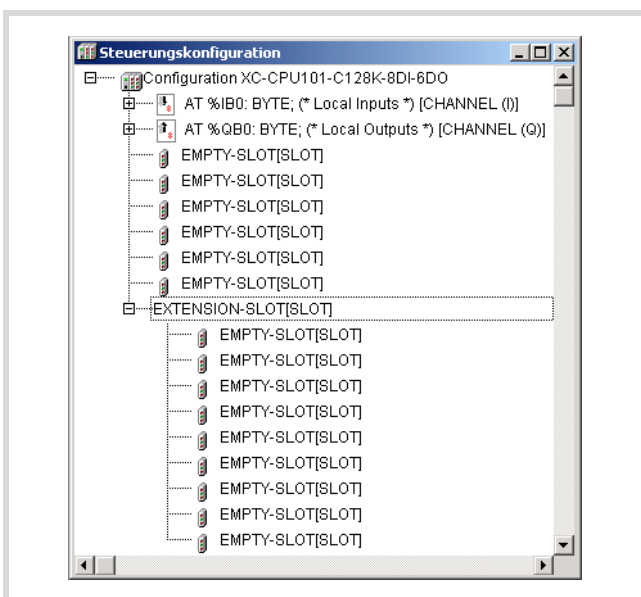


Abbildung 71: Grundkonfiguration der XC100-Steuerung – lokale E/A

Soll ein zentrales, digitales Eingangsmodul mit 16 Eingängen direkt neben der CPU eingebunden werden, sind folgende Schritte durchzuführen:

- Markieren Sie den ersten „EMPTY-SLOT“ und drücken Sie anschließend die rechte Maustaste.

Es öffnet sich ein Fenster.

- Wählen Sie hier das Feld „Element ersetzen“ aus.

Das sich nun öffnende Fenster listet die zur Verfügung stehenden Signalmodule auf.

- Wählen Sie das Modul „XIOC-16DI“.

Die Konfiguration sieht nun wie folgt aus:

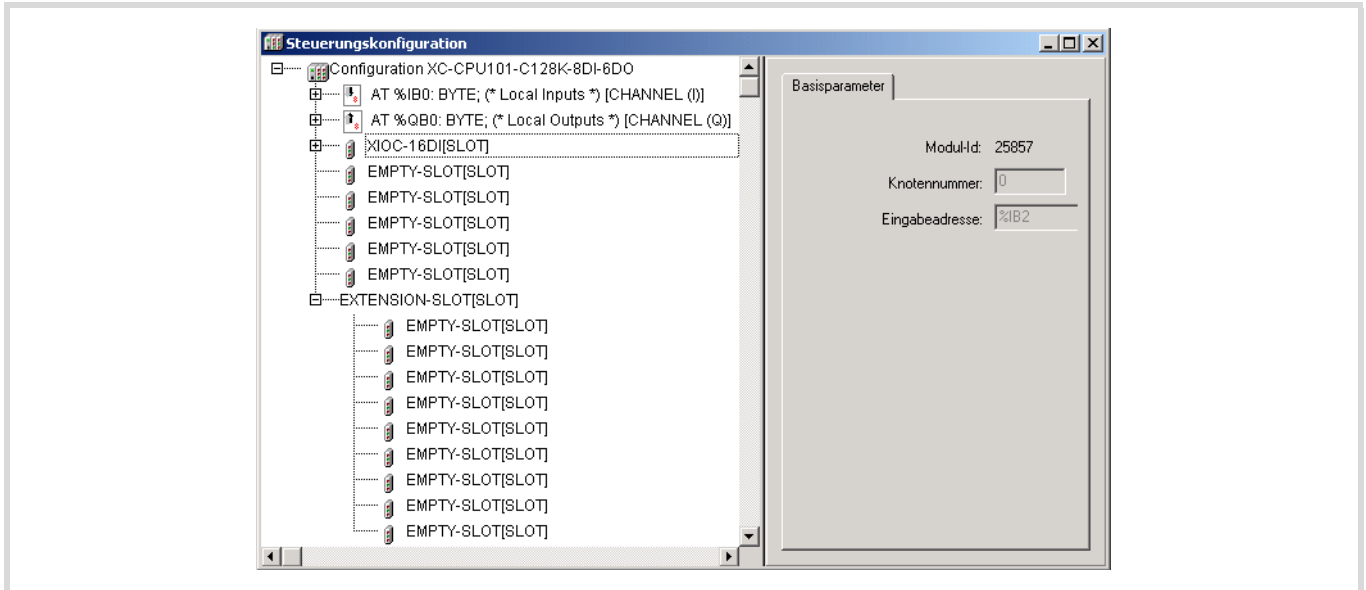


Abbildung 72: Konfiguration XC-CPU101-C64K-8DI-6DO

- ▶ Klicken Sie zusätzlich auf die Pluszeichen vor den Modulen
 - „AT %IB0;Byte; (*Lokale Eingänge*)“
 - „AT %QB0;Byte; (*Lokale Ausgänge*)“
 - XIQC-16DI (SLOT).

Sie erhalten umfassende Informationen mit dem physikalischen Adress-Bereich der Ein- und Ausgänge.

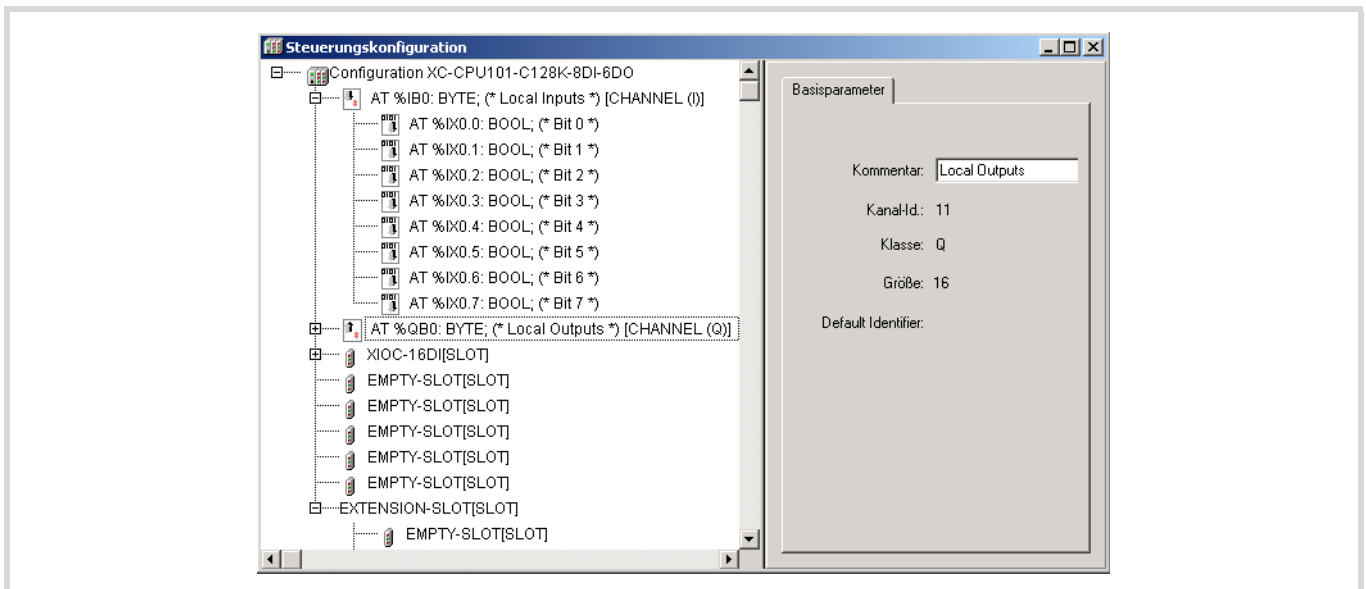
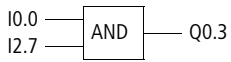


Abbildung 73: Adress-Bereich der Konfiguration

Programm erstellen

Wie in der Aufgabenstellung auf Seite 47 beschrieben, soll der Eingang I0.0 mit dem Eingang I2.7 „AND“ verknüpft werden. Das Verknüpfungsergebnis wird auf dem Ausgang Q0.3 dargestellt.



- Wählen Sie das Register „Bausteine“ und klicken Sie doppelt auf das Element „PLC-PRG“. Die Deklarations- und Programmfenster öffnen sich.

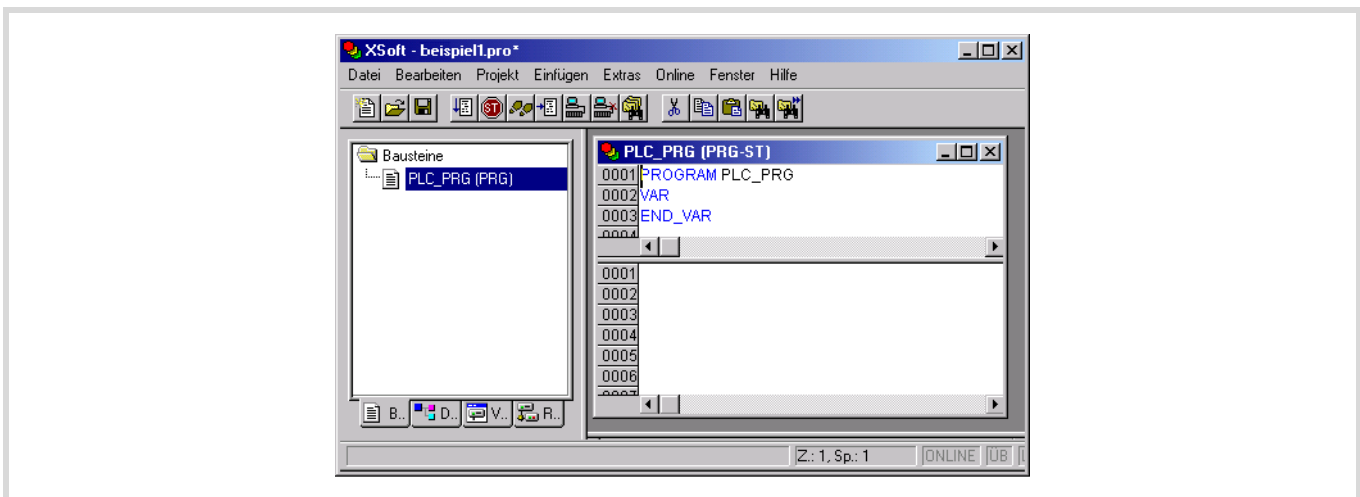


Abbildung 74: Programm- und Deklarationsfenster

- Erstellen Sie die Deklaration und das Programm wie in der folgenden Abbildung beschrieben und übersetzen Sie das Projekt.

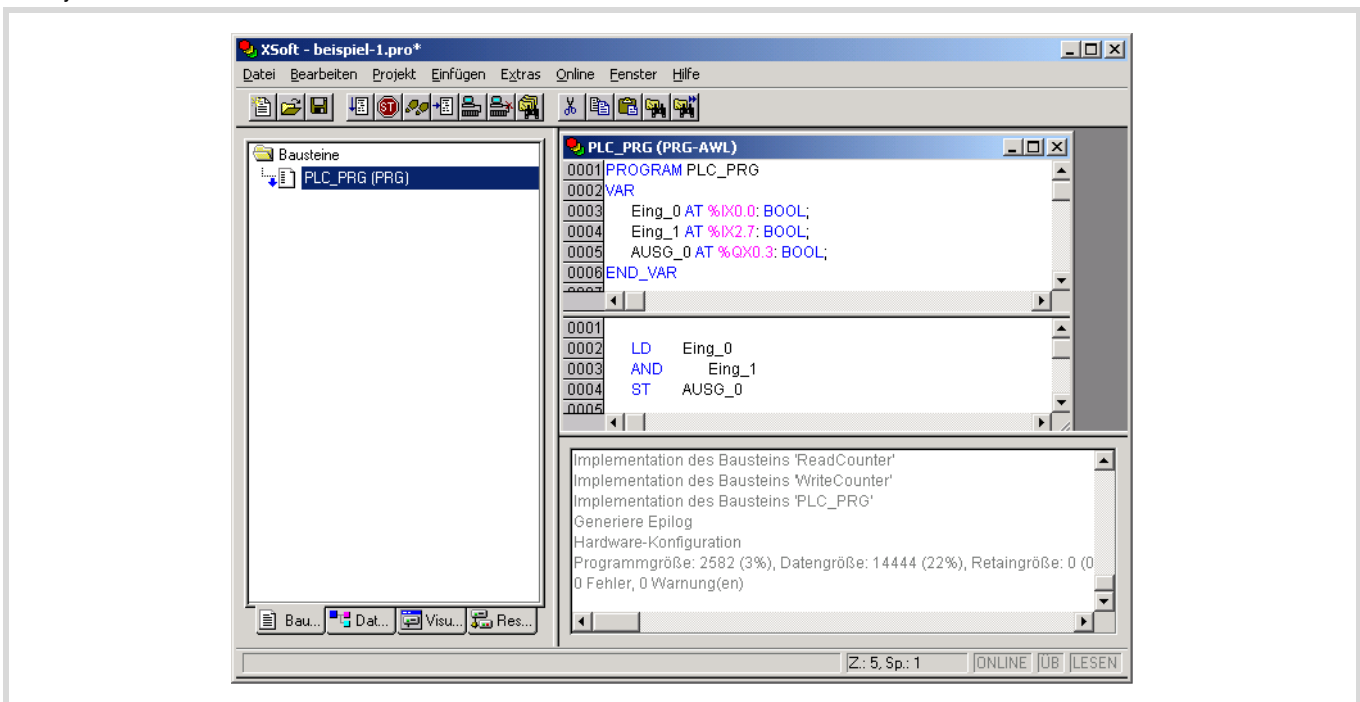


Abbildung 75: Übersetztes Programm

- Laden Sie das Projekt in die Steuerung.

- Testen Sie das Projekt.

7 Programmieren über CANopen-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 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 bietet 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 TCP/IP). Dies macht sich beispielsweise durch langsamere Aktualisierungszeiten von Visualisierungselementen (Variablen) oder langsamere Download-Geschwindigkeiten bemerkbar.

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 die über gleiche aktive CAN-Baudrate verfügen.
- Auf beiden Steuerungen muss eine gültige Routing-Node-ID eingestellt sein.

Routing über XC200

Führen Sie bei einer Verbindung zwischen XC200 und PC über TCP/IP einen Programmtransfer oder ein Routing aus, müssen Sie die Blockgröße der zu übertragenen Daten einstellen. Die Blockgröße (4 kByte oder 128 kByte) ist abhängig von der Art des Transfers (Programmtransfer oder Routing) und des Betriebssystems, → Tabelle 11.

Tabelle 11: Blockgröße für die Datenübertragung

	Programm-/Datei-Transfer		Routing	
	BTS < V1.03.02	BTS ≥ V1.03.02	BTS < V1.03.02	BTS ≥ V1.03.02
Blockgröße Default: 128 kByte	128 kByte	128/4 kByte	Routing nicht möglich	4 kByte



Achtung!

Der Programmdownload mit einer Blockgröße von 4 kByte auf eine Steuerung mit dem Betriebssystem < 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).

Das Routing mit der XC200 ist ab der Betriebssystem-Version V1.03.02 möglich.

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)

Einstellung der Blockgröße:

- ▶ Schließen Sie alle easySoft-CoDeSys-Anwendungen.
- ▶ Schließen Sie den CoDeSys-Gateway-Server.

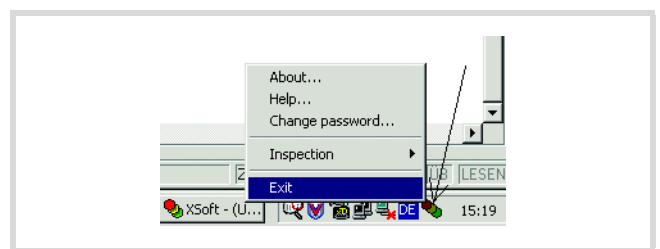


Abbildung 76: CoDeSys-Gateway-Server schließen

- ▶ Ändern Sie die Blockgröße auf den gewünschten Wert.

Zum Eintrag der Blockgröße in die Registry stehen im easySoft-CoDeSys-Installationsverzeichnis folgende *.reg-Dateien zur Verfügung:

BlockSizeDefault.reg	Trägt eine Blockgröße (Defaultwert) von 20000 _{hex} = 128 kByte in die Registry ein.
BlockSizeRout.reg	Trägt eine Blockgröße von 1000 _{hex} = 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_{hex} (=128 kByte), die Blockgröße für das Routing beträgt 1 000_{hex} (= 4 kByte).

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 angekoppelt werden."
- 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.

Adressierung

Steuerungen am CAN-Bus können als Master oder als 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.

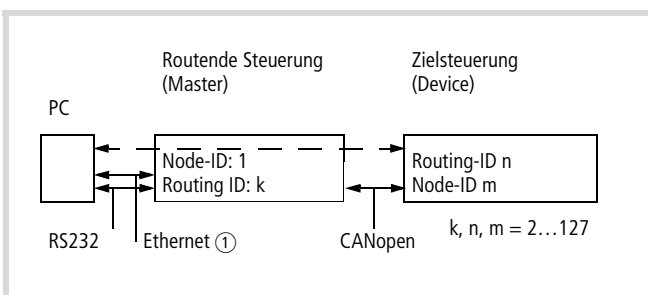


Abbildung 77: Routing über XC..., EC4-200

① Ethernet-Verbindung nur bei XC200 möglich

Tabelle 12: Beispiel zur Einstellung der Node-Id, Baudrate

Steuerung	Funktion	Node-ID	Routing-ID	Baudrate	→ Abb.
Routende Steuerung	Master	1	127	125 kB	79
Zielsteuerung	Device	3	54	125 kB	80

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

Die Routing-ID beim Master (XC100, XC200) stellen Sie in der Steuerungskonfiguration im Register „Weitere Parameter“ ein:

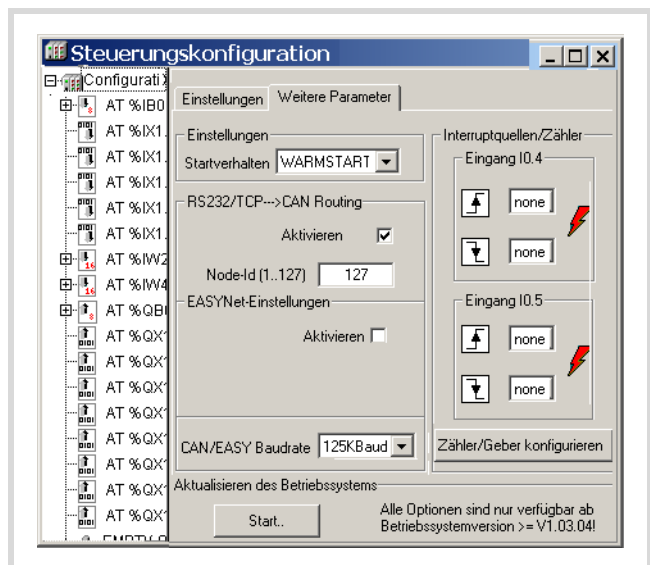


Abbildung 78: CAN-Master-Routing Einstellungen (XC200)

Die ID zur Basiskommunikation legen Sie im Ordner „CanMaster“ im Register „CAN-Parameter“ (Abbildung 79) fest.

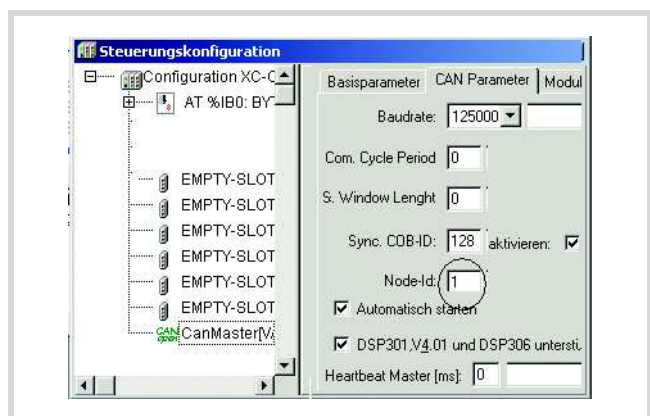


Abbildung 79: CAN-Master: Node-ID zur Basiskommunikation

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 loggen Sie sich ein.

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.

Setzen Sie eine XC100 ein, so tragen Sie für beide IDs die gleiche Nummer ein. Die Routing-ID und die Baudrate der Zielsteuerung (XC100) legen Sie in der Steuerungskonfiguration im Fenster „Weitere Parameter“ fest. Geben Sie im Feld „RS232 → CAN Routingeinstellungen“ die Baudrate und Node-Id ein.

Die CAN-Baudrate hängt von der Betriebssystem-Version ab:

Tabelle 13: Baudraten für CAN-Verbindung

Baudrate	Betriebssystem-Version	
	< V. 2.0	≧ V 2.0
10 000	✓	✓
20 000	✓	✓
50 000	✓	✓
100 000	✓	✓
125 000	✓	✓
250 000	✓	✓
500 000	–	✓
800 000	–	✓
1 000 000	–	✓

➔ Um eine zügige Datenübertragung zu gewährleisten, sollte das Routing nur mit CAN-Baudraten von mindestens 125 kBit/s durchgeführt werden.

Die ID zur Basiskommunikation legen Sie im Ordner „CanDevice“ im Register „CAN-Einstellungen“ fest, ➔ Abbildung 80.

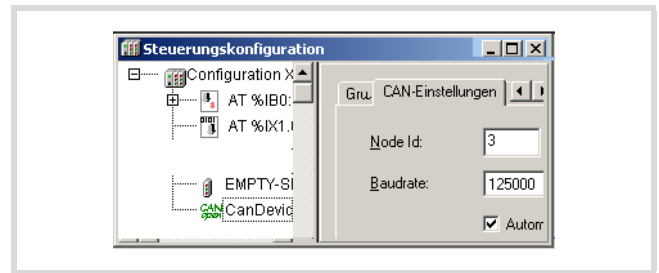


Abbildung 80: CAN-Deviceparameter

ID und Baudrate werden mit dem Projektdownload in die Steuerung übertragen.

Beispiel

Das folgende Beispiel, das auf der Abbildung 81 basiert, zeigt den Zugriff auf ein Steuerungsprogramm.

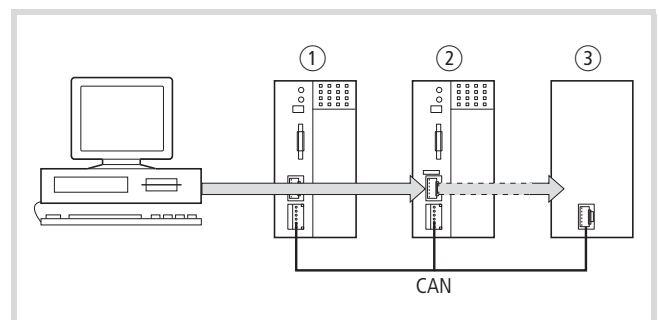


Abbildung 81: Diagnosemöglichkeiten

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

Den PC haben Sie an die Steuerung mit der Node-Id „2“ angeschlossen und Sie wollen auf die Zielsteuerung mit der Routing-Id „3“ 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 im Online-Menü „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 vergeben, z. B. „Rout_232“.
- ▶ Bestätigen Sie mit OK und Sie gelangen zurück zum Ausgangsfenster.

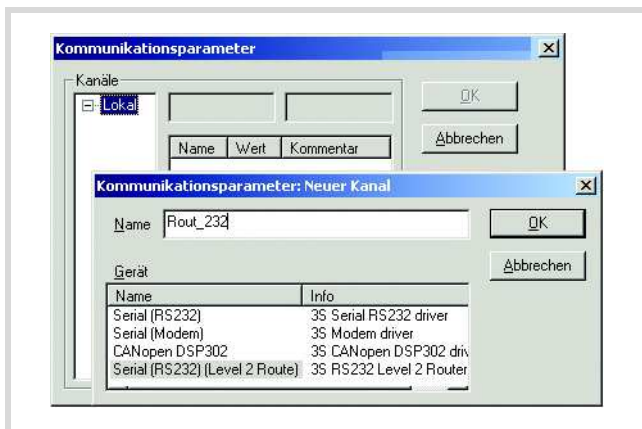


Abbildung 82: Kanal parametrieren

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

- ▶ Rufen Sie die Kommunikationsparameter im „Online“-Menü nochmals auf und wählen Sie die Steuerung aus, die Sie programmieren/testen möchten.
- ▶ Tragen Sie dazu die Target-ID, im Beispiel die Nummer 3, 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 3 ein und bestätigen Sie mit OK.
- ▶ Loggen Sie sich ein und führen Sie die Aktion aus.

Steuerungskombinationen zum Routing

Folgende Steuerungen unterstützen das Routing:

Von →	XC100 XC121	XC200 ¹⁾	EC4-200
Nach ↓			
XC100, XC121	x	x	x
XC200 ¹⁾	x	x	x
EC4-200	x	x	x

1) XC200 ab Version V01.03.01

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 14: Art und Anzahl der Kommunikationskanäle

Kommunikationskanal	Steuerung	Max. Kanalanzahl
TCP/IP Level2Route	XC200	5
Serial RS232 Level2Route	XC.../EC4-200	1

8 RS232-Schnittstelle im Transparent-Modus (COM 1/2/3)

Der Transparent-Modus dient dem Datenaustausch mit Datenendgeräten (z. B. Terminal, Drucker, PC, Messgeräte usw.) über die serielle Schnittstellenverbindung RS232. Die Datenübertragung findet dabei transparent statt; d. h., die Daten werden übertragen, ohne sie weiter zu interpretieren.

Diese Funktionalität wird bei der XC100 durch die Library „XC100_SysLibCom“ erbracht; die Library muss somit in den „Bibliotheksverwalter“ eingebunden sein. Die Library enthält Funktionen zum Öffnen und Schließen der Schnittstelle, zum Senden und Empfangen von Daten und zum Setzen der Schnittstellenparameter und Kontrollleitungen.

Werden in den Funktionsbausteinen Kontrollleitungen der integrierten XC100-Schnittstelle RS232 angesprochen, sind diese außer Funktion, da diese Kontrollleitungen physikalisch nicht vorhanden sind. Aus Kompatibilitätsgründen werden diese „XC100_SysLibCom“-Funktionen jedoch mitgeführt.

Bei der Hardware-Schnittstellenbaugruppe XIOC-SER stehen die Kontrollleitungen zur Verfügung und werden über die Funktionsaufrufe „SysComReadControl“ und „SysComWriteControl“ bedient.

Ist der Transparent-Modus geöffnet, ist keine Kommunikation mit dem Programmiersystem möglich. Der Transparent-Modus muss erst wieder geschlossen werden. Mit Schließen des Transparent-Modus werden wieder die ursprünglichen Kommunikationsparameter initialisiert. Das Schließen des Transparent-Modus erfolgt zwangsweise bei einem SPS-Zustandswechsel in den STOP.

Voraussetzung, um die RS232-Schnittstelle im Transparent-Modus betreiben zu können, ist die Hardware-Version 02 und die Betriebssystem-Version V02.00 oder höher.

Anforderungen an die Funktionalität des Transparent-Modus

Funktion „SysComOpen“

Die Funktion öffnet die RS232-Schnittstelle für den Transparent-Modus. Nach erfolgreichem Öffnen der Schnittstelle gibt die Funktion einen Rückgabewert größer 0 zurück.

- Tragen Sie diesen Wert bei den folgenden Funktionen als „dwHandle“-Parameter ein.

Ist ein Fehler aufgetreten, ist der Rückgabewert gleich „0“. Der Transparent-Modus der Schnittstelle wird dann nicht frei gegeben.

Für die Bedienung der seriellen Schnittstelle stehen nachfolgende Parameter (ENUMERATION-Typen/Aufzählungstypen) zur Verfügung:

Baudrate für COM 1, COM 2, COM 3

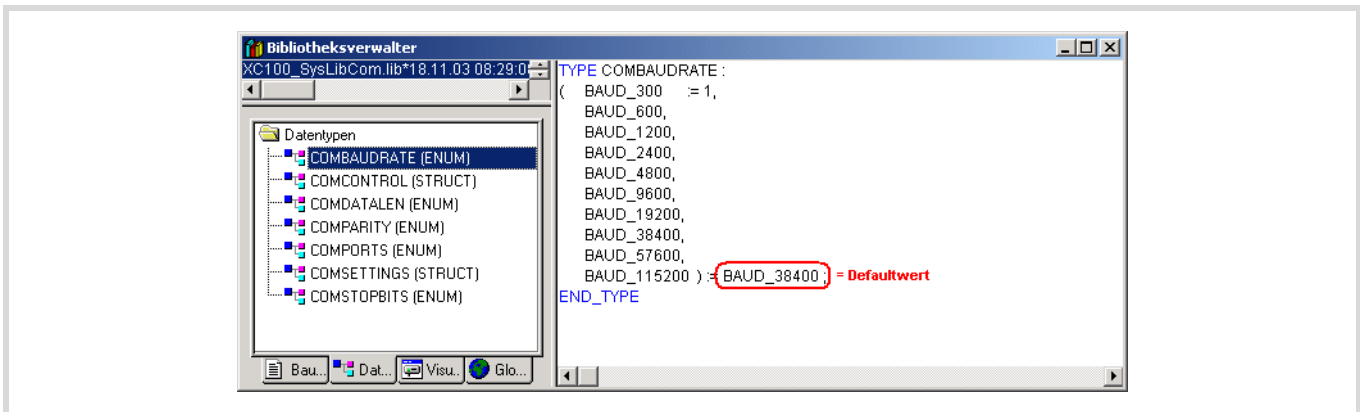


Abbildung 83: Baudrate auswählen

Anzahl der Datenbits

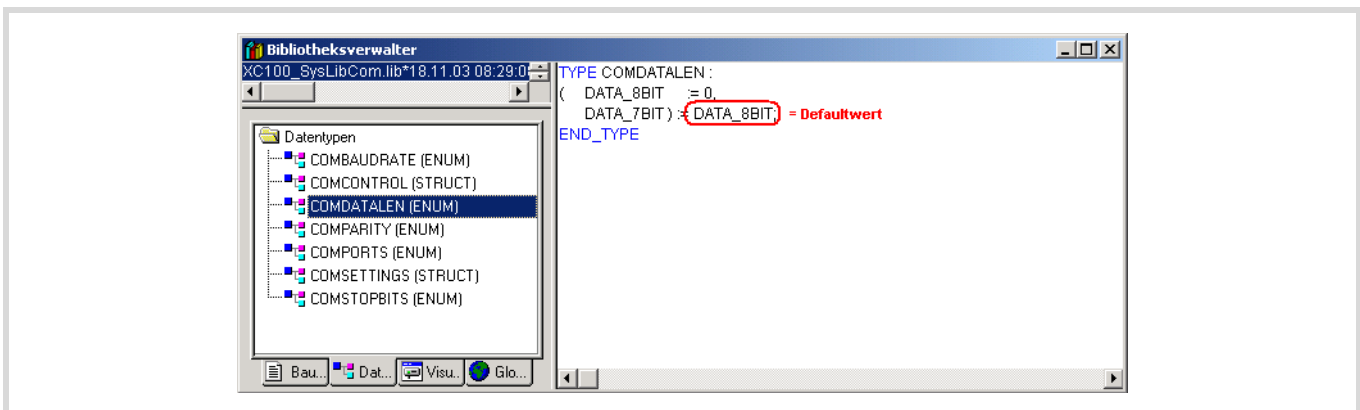


Abbildung 84: Anzahl der Datenbit

Auswahl der Parität

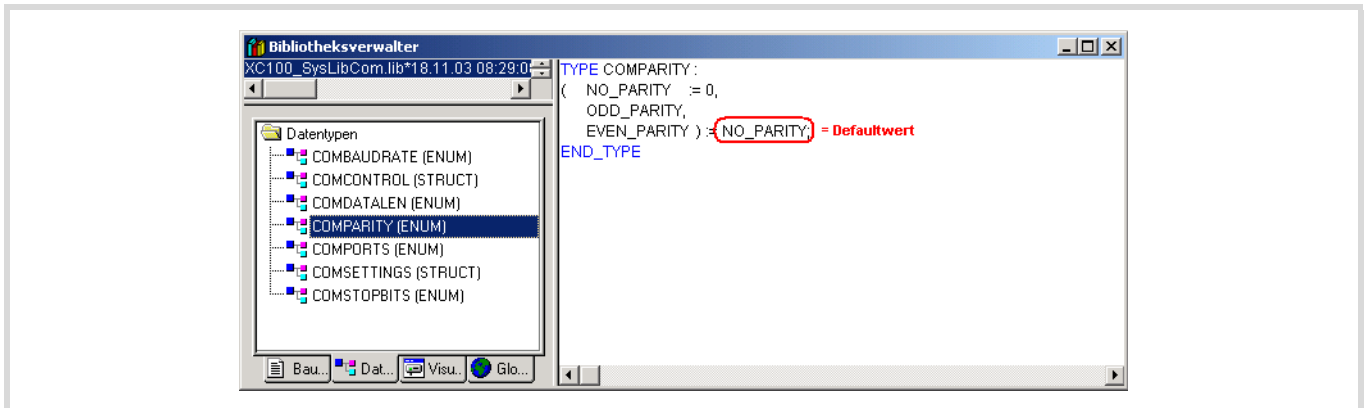


Abbildung 85: gerade/ungerade Parität

Auswahl der COM-Schnittstelle

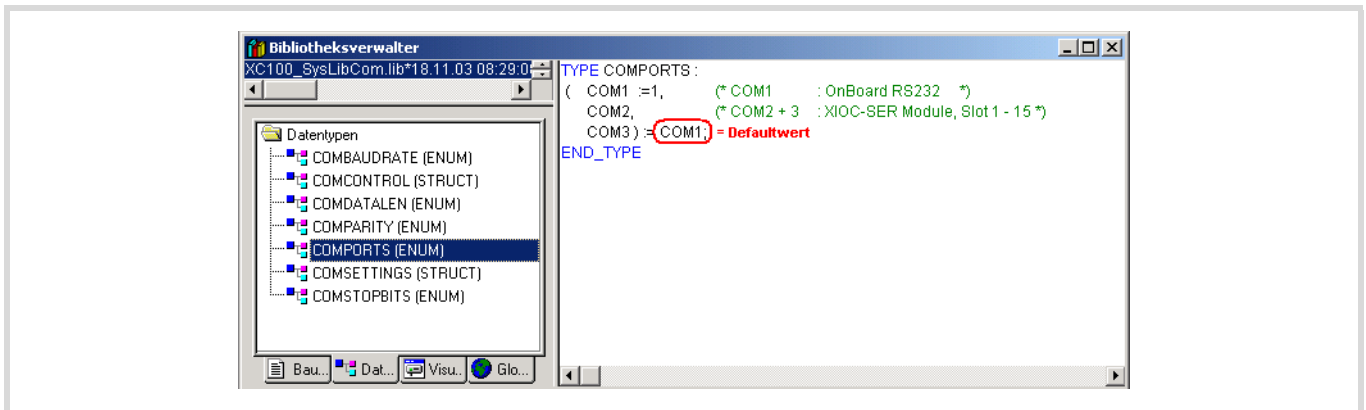


Abbildung 86: Auswahl der COM-Schnittstelle

Anzahl der Stopbits

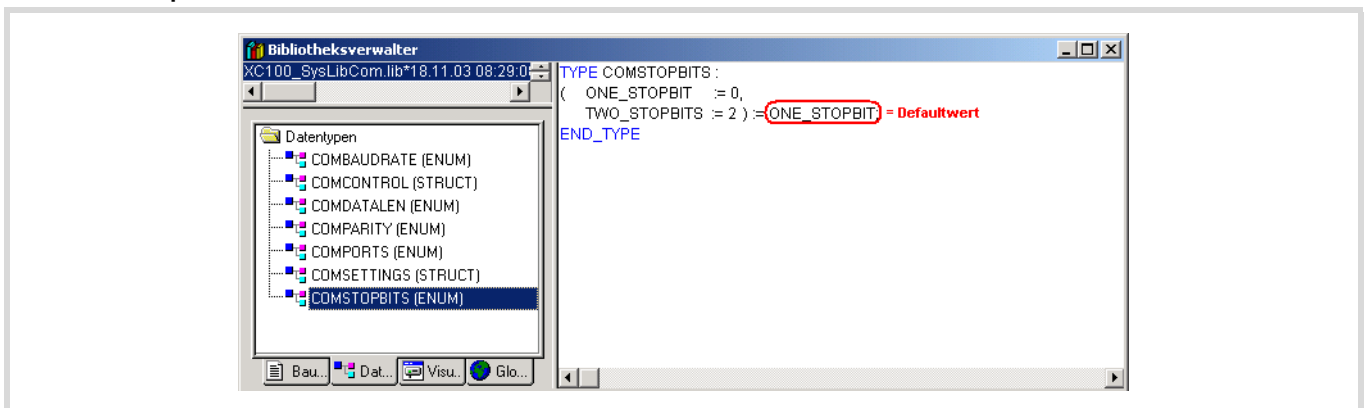


Abbildung 87: Stopbits

Parametrierung der Kontroll-/Steuerleitungen der COM 2/COM 3-Schnittstelle

Die Bedienung der Kontroll-/Steuerleitungen erfolgt mit dem Baustein „ComControl“. Diese Funktion bedient nur die Schnittstellenleitungen DTR, DSR und DCD der Hardware-Schnittstengruppe XIOC-SER. Die auf der Baugruppe vorhandene ERROR-LED wird ebenfalls über diesen Baustein bedient und mit dem Befehl TRUE angesteuert. Werden die entsprechenden Schnittstellenleitungen mit TRUE parametrierung, ist ein Schreib- bzw. Lesezugriff möglich.

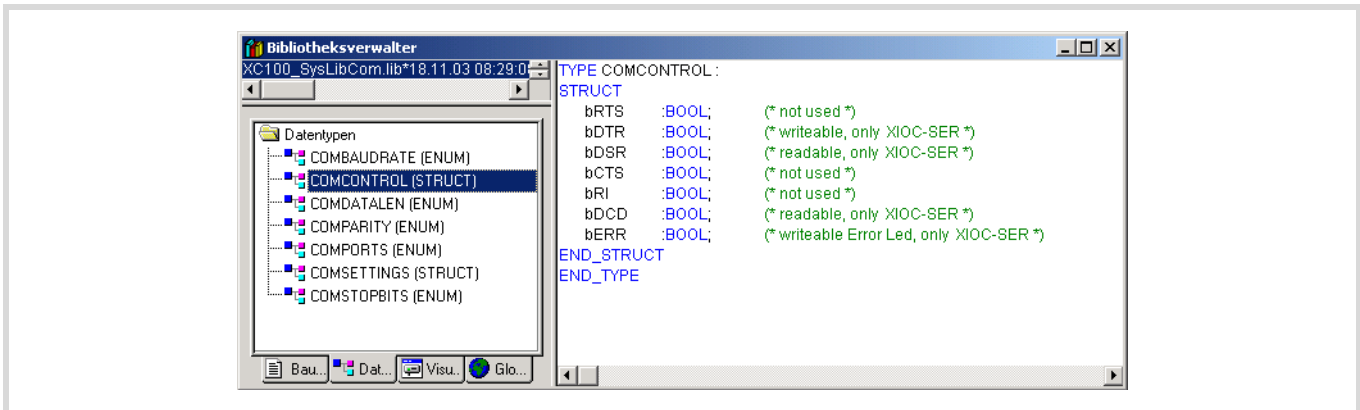


Abbildung 88: Parametrierung der Kontrollleitungen der Schnittstellen COM 2 und COM 3 und der ERR-LED

Parametrierung der Schnittstelle

In dem Baustein „ComSettings“ werden die kompletten Schnittstellenparameter der COM 2- oder COM3-Schnittstelle abgelegt und gespeichert. Diese Parameter werden beim Aufruf des Bausteins aktiviert und der entsprechenden Hardware-Schnittstelle zugeordnet.

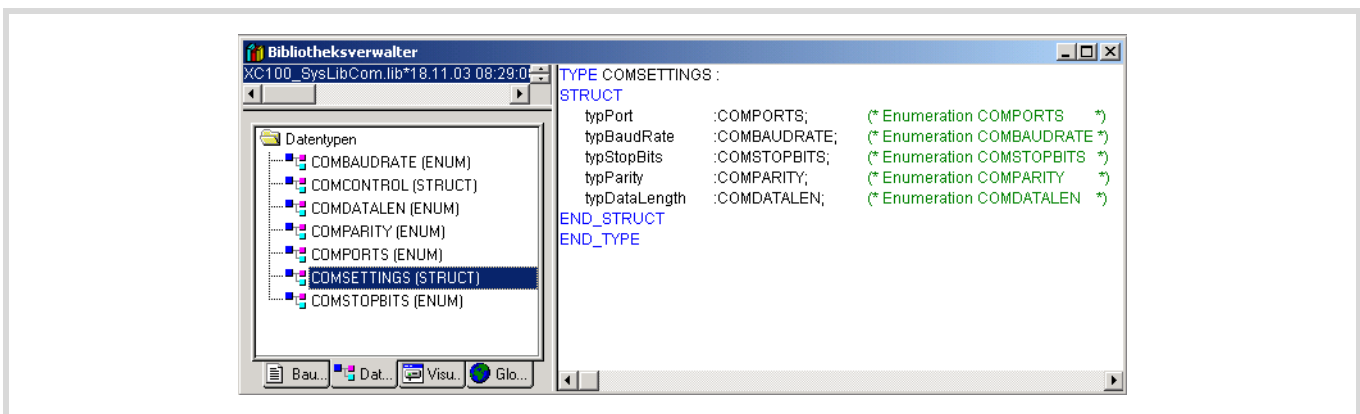


Abbildung 89: Schnittstellenparameter der COM2- oder COM3-Schnittstelle

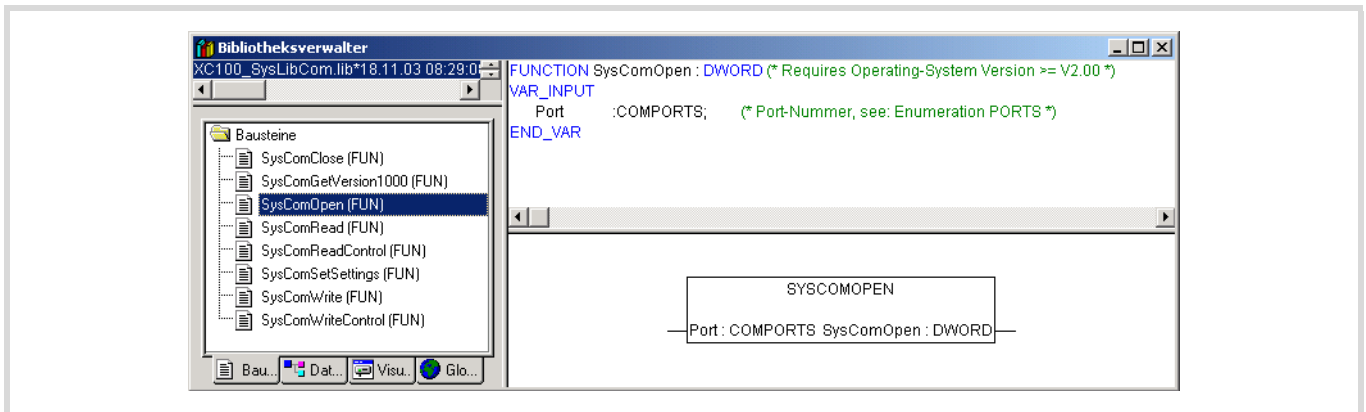


Abbildung 90: Funktion „SysComOpen“

Tabelle 15: Parameter der Funktion „SysComOpen“

Port	Auswahl der Schnittstelle
	Parameter: Spezifiziert die zu öffnende Schnittstelle.
SysComOpen	Rückgabewert 0: Öffnen der RS232-Schnittstelle war nicht erfolgreich. Rückgabewert > 0: Öffnen der RS232-Schnittstelle war erfolgreich.

Funktion „SysComClose“

Die Funktion schließt die im Transparent-Modus geöffnete RS232-Schnittstelle. Beim Schließen werden die zuletzt eingestellten Kommunikationsparameter wiederhergestellt. Die Funktion liefert als Rückgabewert TRUE, wenn die Aktion erfolgreich abgeschlossen wurde.

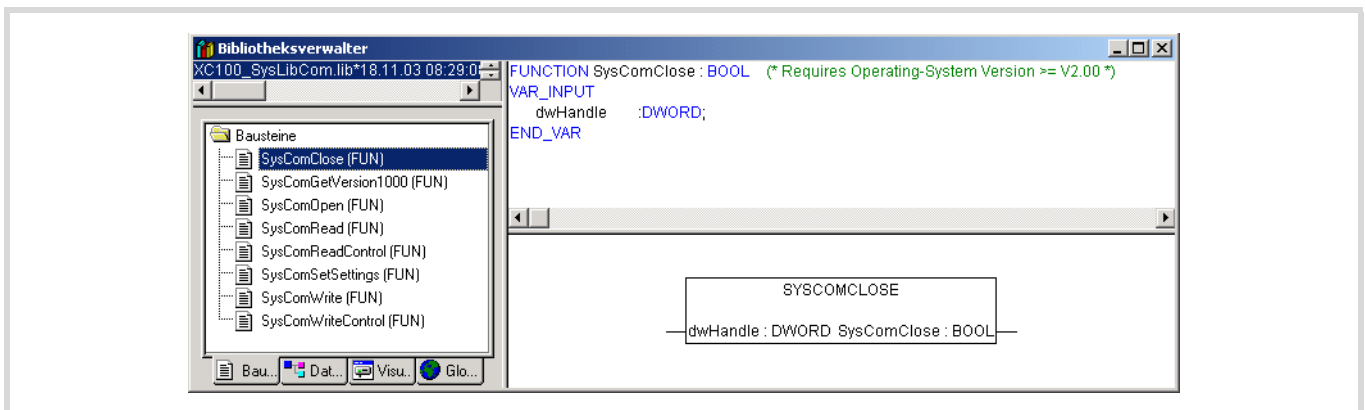


Abbildung 91: Funktion SysComClose

Tabelle 16: Parameter der Funktion „SysComClose“

dwHandle	Rückgabewert aus der Funktion „SysComOpen“
SysComClose	Rückgabewert TRUE: Schließen der RS232-Schnittstelle war erfolgreich

Funktion „SysComRead“

Mit dieser Funktion können über die RS232-Schnittstelle im Trans-
parent-Modus empfangene Daten gelesen werden.

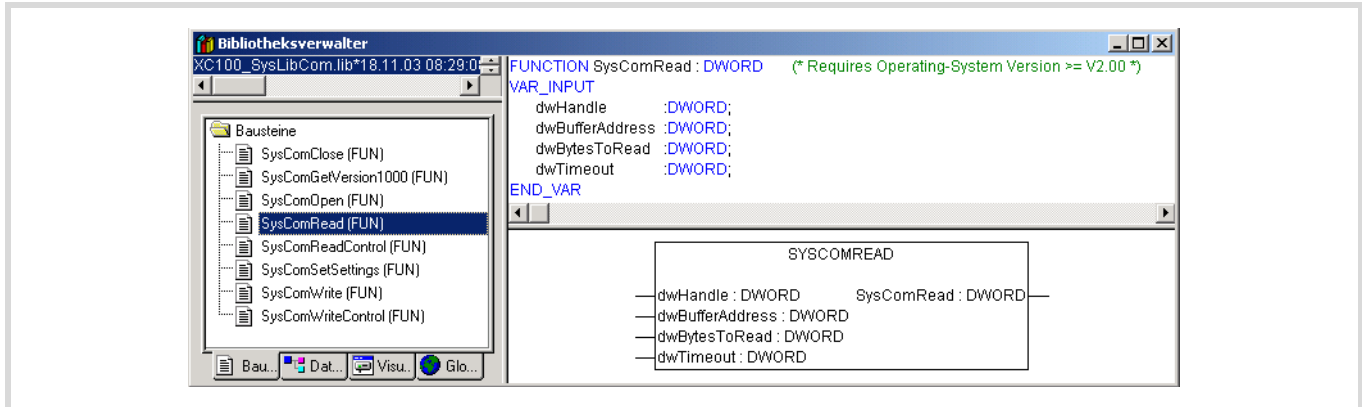


Abbildung 92: Funktion SysComRead

Tabelle 17: Parameter der Funktion „SysComRead“

dwHandle	Rückgabewert aus der Funktion „SysComOpen“
SysComClose	Rückgabewert TRUE: Schließen der RS232-Schnittstelle war erfolgreich.
dwBufferAddress	Adresse, unter der die eingelesenen Daten abgelegt werden
dwBytesToRead	Begrenzung der max. Anzahl der Datenbytes (COM 1: max. 190 Byte; COM 2 + 3: max. 250 Byte)
dwTimeout	Parameter ohne Bedeutung
SysComRead	Rückgabewert: Informiert über die Anzahl der gelesenen Datenbytes.



Achtung!

Eine Prüfung der Bufferadresse bzw. der Buffergröße findet nicht statt!

Funktion „SysComWrite“

Diese Funktion erlaubt die Ausgabe von Daten über die RS232-Schnittstelle.

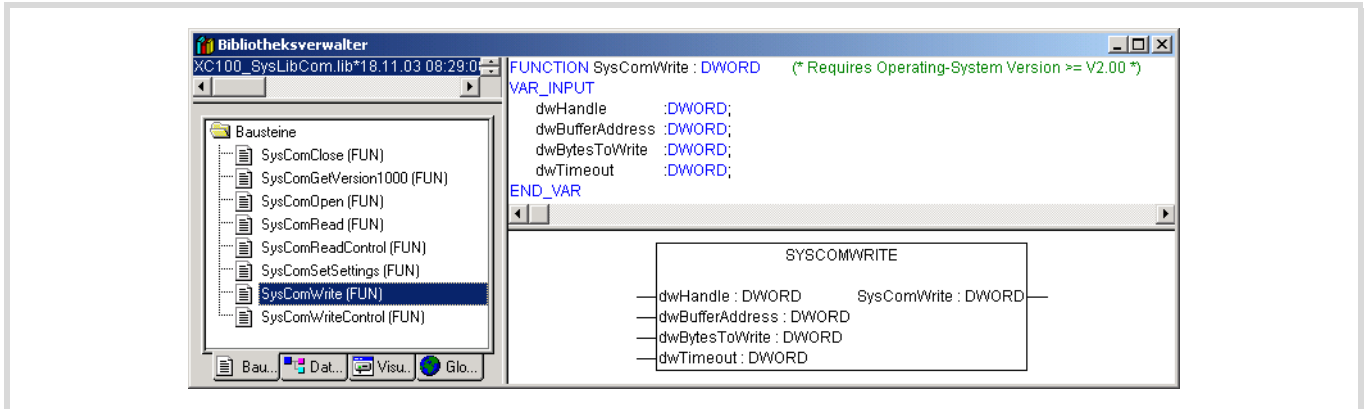


Abbildung 93: Funktion SysComWrite

Tabelle 18: Parameter der Funktion „SysComWrite“

dwHandle	Rückgabewert aus der Funktion „SysComOpen“
dwBufferAddress	Adresse, unter der die auszugebenden Daten abgelegt sind
dwBytesToWrite	Anzahl der Datenbytes, die gesendet werden (COM 1: max. 190 Byte; COM 2, 3: max. 250 Byte)
dwTimeout	Parameter ohne Bedeutung
SysComWrite	Rückgabewert: Informiert über die Anzahl der gesendeten Daten.



Achtung!

Eine Prüfung der Bufferadresse bzw. der Buffergröße findet nicht statt!

Funktion „SysComSetSettings“

Mit dieser Funktion können die Schnittstellenparameter der RS232-Schnittstelle für den Transparent-Modus eingestellt werden.

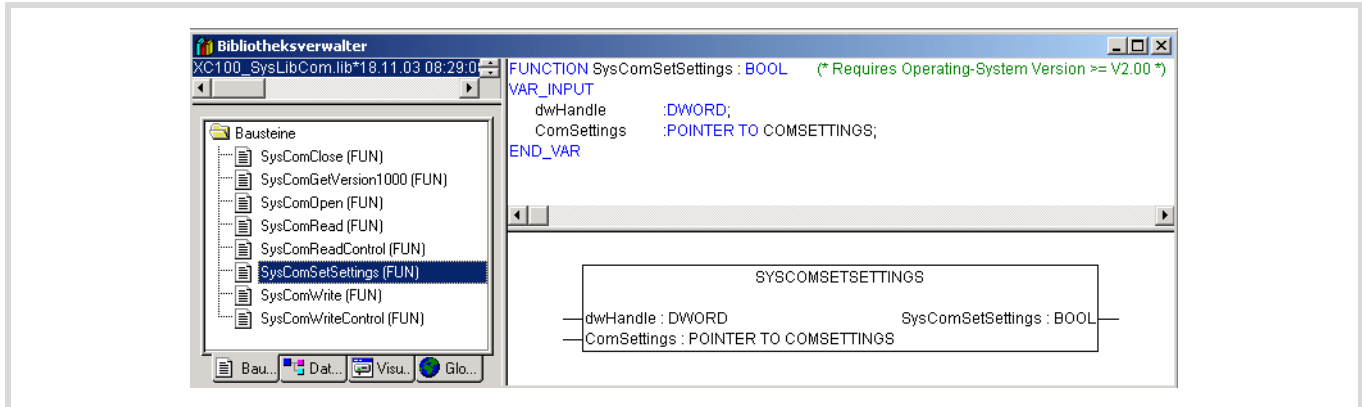


Abbildung 94: Funktion SysComSettings

Tabelle 19: Parameter der Funktion „SysComSettings“

dwHandle	Rückgabewert aus der Funktion „SysComOpen“
ComSettings	Zeiger, der auf den Speicherbereich zeigt, in dem die Schnittstellenparameter abgelegt sind
SysComSetSettings	Rückgabewert TRUE, wenn die Schnittstelle erfolgreich parametrierung wurde; sonst FALSE

Beispiel

Das Beispiel zeigt das Öffnen, eine Textausgabe und das Schließen der RS232-Schnittstelle bei der XC100.

```

PLC_PRG (PRG-ST)
0001 PROGRAM PLC_PRG
0002 VAR
0003   dwSioHandle      :DWORD;
0004   OpenTrigger     :F_TRIG;
0005   CloseTrigger    :F_TRIG;
0006   WriteBuffer     :STRING(26);
0007   nWriteLength   :DWORD;
0008   typComSettings  :COMSETTINGS;
0009   Eingang0        :BOOL;
0010   Eingang1        :BOOL;
0011   typComSetSettings :BOOL;
0012 END_VAR
0001 (* Öffnen der Schnittstelle für den Transparentmode und Einstellen der Schnittstellenparameter *)
0002 OpenTrigger(CLK=Eingang0);
0003 IF (OpenTrigger.Q=TRUE) THEN
0004   dwSioHandle:=SysComOpen(Port=COM1);
0005   IF (dwSioHandle > 0) THEN
0006     typComSettings.typBaudRate :=Baud_19200; (* 4800; 9600; 19200; ... *)
0007                                           (* ... 38400; 57600 BIts *)
0008     typComSettings.typDataLength :=Data_8BIT; (* 7; 8 Bit *)
0009     typComSettings.typPARITY :=NO_PARITY; (* NO; ODD; EVEN - Parity *)
0010     typComSettings.typPORT :=COM1; (* COM1 (XC100) *)
0011     typComSettings.typStopBits :=ONE_STOPBIT; (* 1; 2 StopBit *)
0012     SysComSetSettings(dwHandle :=dwSioHandle,
0013                     ComSettings :=ADR(typComSettings));
0014   END_IF
0015   WriteBuffer:= 'Das ist der Sendetext';
0016 END_IF
0017 (* Schliessen der Schnittstelle *)
0018 CloseTrigger(CLK=Eingang1);
0019 IF (CloseTrigger.Q=TRUE) THEN
0020   SysComClose(dwHandle:= dwSioHandle); dwSioHandle:=0;
0021 END_IF
0022 (* Ausgabe eines Textes über den Transparentmode der XC100 - RS232-Schnittstelle *)
0023 IF ( (* Timer.Q=TRUE AND *) dwSioHandle > 0) THEN
0024   nWriteLength:=SysComWrite(dwHandle:=dwSioHandle, dwBufferAddress:=ADR(WriteBuffer),
0025                             dwBytesToWrite:=LEN(WriteBuffer), dwTimeOut:=0);
0026 END_IF

```

Abbildung 95: Programmbeispiel Transparent-Modus

Funktion „SysComReadControl“

Bei der OnBoard-RS232-Schnittstelle der CPU101 stehen die Kontroll-/Steuerleitungen nicht zur Verfügung. Daher ist der Baustein „SysComReadControl“ für diese Schnittstelle nicht anzuwenden.

Die Hardware-Schnittstellenbaugruppe XIOC-SER verfügt über Kontroll-/Schnittstellenleitungen. Somit ermöglicht der Baustein „SysComReadControl“ den Lesezugriff auf die Kontroll-/Schnittstellenleitungen der COM2- und COM3-Schnittstelle.

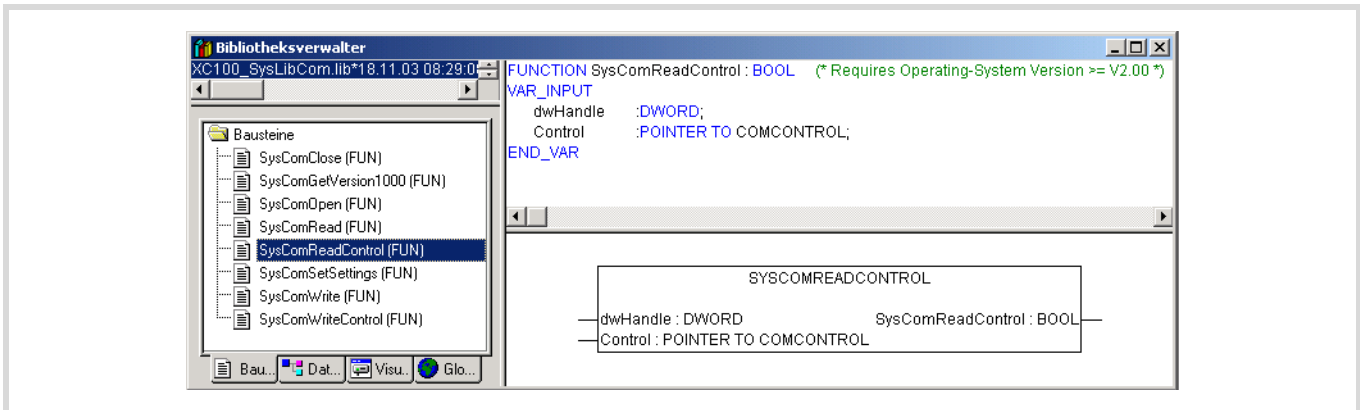


Abbildung 96: Lesezugriff auf die Kontroll-/Schnittstellenleitungen der COM 2- oder COM 3-Schnittstelle

Tabelle 20: Parameter der Funktion „SysComReadControl“

dwHandle	Rückgabewert aus der Funktion „SysComOpen“	
Control	COM 1:	Funktion entfällt
	COM 2, COM 3:	TRUE = Lesebefehl auf die Kontroll-/Steuerleitungen der Hardware-Schnittstelle
SysComReadControl	COM 1:	Funktion entfällt
	COM 2, COM 3:	TRUE = Lesebefehl war erfolgreich; FALSE = Lesebefehl war nicht erfolgreich

Funktion „SysComWriteControl“

Bei der integrierten RS232-Schnittstelle der CPU101 stehen die Kontroll-/Steuerleitungen nicht zur Verfügung. Daher ist der Baustein „SysComWriteControl“ für diese Schnittstelle nicht anzuwenden.

Die Hardware-Schnittstellenbaugruppe XIOC-SER verfügt über Kontroll-/Schnittstellenleitungen. Somit ermöglicht der Baustein „SysComWriteControl“ den Schreibzugriff auf die Kontroll-/Schnittstellenleitungen der COM2- und COM3-Schnittstelle.

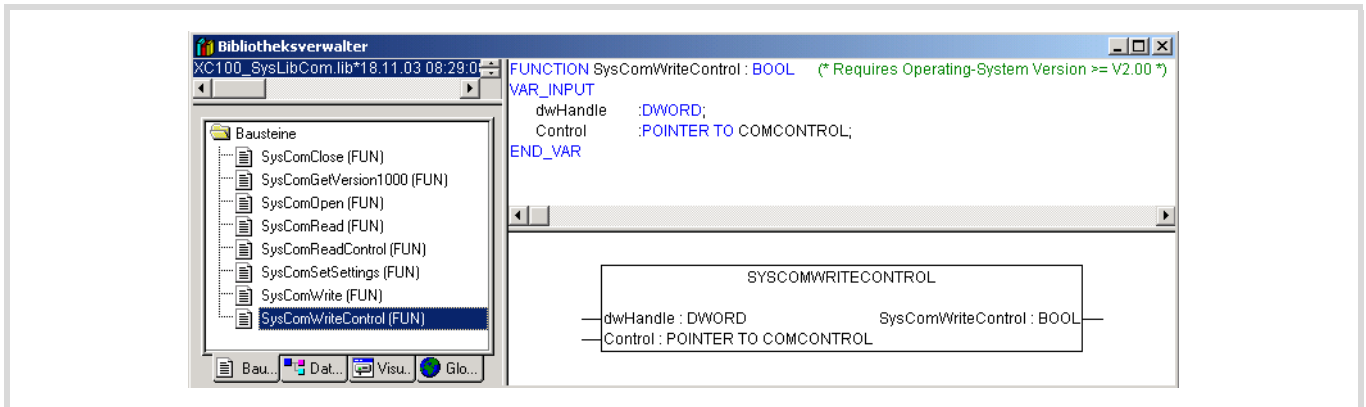


Abbildung 97: Schreibzugriff auf die Kontroll-/Schnittstellenleitungen der COM 2- oder COM 3-Schnittstelle

Tabelle 21: Parameter der Funktion „SysComWriteControl“

dwHandle	Rückgabewert aus der Funktion „SysComOpen“	
Control	COM 1:	Funktion entfällt
	COM 2, COM 3:	TRUE = Schreibbefehl auf die Kontroll-/Steuerleitungen der Hardware-Schnittstelle
SysComReadControl	COM 1:	Funktion entfällt
	COM 2, COM 3:	Rückgabe: TRUE = Lesebefehl war erfolgreich; FALSE = Lesebefehl war nicht erfolgreich

Automatisches Schließen der Schnittstelle

Bei einem SPS-Zustandswechsel der XC100 in STOP wird der Transparent-Modus durch das Betriebssystem automatisch beendet. Die Schnittstelle wird wieder mit den zuletzt eingestellten Schnittstellenparametern initialisiert.

Anhang

Kompatibilität

Der Funktionsumfang der XC-CPU101 ist abhängig von der Hardware-Ausstattung (HW), dem installierten Betriebssystem (BTS) und der Version der Programmiersoftware.

Die nachfolgende Übersicht informiert über die Funktionalitäten in Abhängigkeit zu Hard-, BTS- und easySoft-CoDeSys-Versionen:

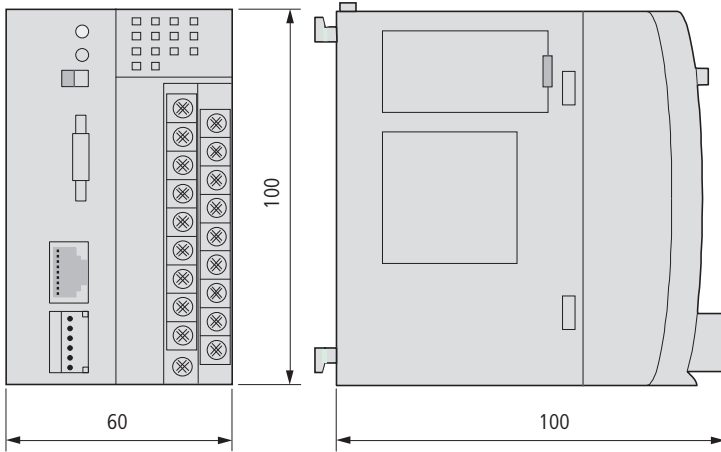


- Die Kompatibilität zu älteren Betriebssystem-Versionen der XC101 wird sichergestellt.
- Durch die geänderte Steuerungskonfiguration kann es bei bestehenden Projekten erforderlich werden, dass die Steuerungskonfiguration neu eingegeben werden muss.
- Wird ein Projekt mit der neuen Steuerungskonfiguration auf eine XC101 mit einer älteren Betriebssystem-Version geladen, stehen nur die Funktionen des älteren Betriebssystems zur Verfügung.

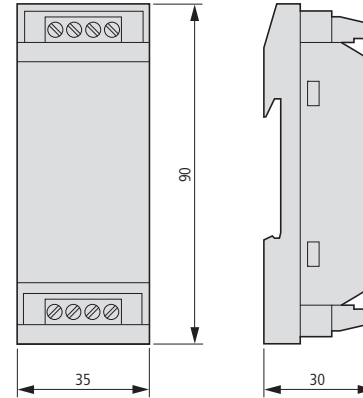
Funktionalität	Hardware-Version:			easySoft-CoDeSys-Version:
	V01	V02	V04	
	ab Betriebssystem-Version:			
Grundfunktionalität	V1.2	–	–	≤ V2.2.5.+Rev. B
PRG-Default Baudrate 57,6 kBit/s	nur V1.2	–	–	V2.3.1
PRG-Default Baudrate 38,4 kBit/s	V1.3	–	–	V2.3.1
Mehrere CAN-Teilnehmer	V1.3	–	–	V2.3.1
CAN Device	V1.3	–	–	V2.3.1
Programmierung über CANopen (Routing)	V1.3	–	–	V2.3.1
MMC-Zugriff	–	V2.0	–	V2.3.1
Interrupt-Funktionalität	–	V2.0	–	V2.3.1
RS232-Schnittstelle im Transparent-Modus	–	V2.0	–	V2.3.1
Peripherie-Direktzugriff	–	V2.0	–	V2.3.1
CAN Direktzugriff	–	V2.0	–	V2.3.1
CAN-Baudrate bis 500 kBit/s	–	V2.0	–	V2.3.1
XIOC-Buserweiterung mit XIOC-BP-EXT	V1.3	–	–	V2.3.2
Einführen der CAN-Buslast „canload“	V1.3	–	–	V2.3.2
Ablage der Textdatenbank auf der MMC	V1.3	–	–	V2.3.2
Ablage der Projektinformationen auf der MMC	V1.3	–	–	V2.3.2
Systemereignisse Warm- und Cold-Start	V1.3	–	–	V2.3.2
Timer-Interrupt	V1.3	–	–	V2.3.2
Serielltes Schnittstellenmodul XIOC-SER	V1.3	–	–	V2.3.2
PROFIBUS-DP-M-Modul	–	–	V3.01	V2.3.2
Suconet-K-Slave-Modul	–	–	V3.02	V2.3.2
Update des Betriebssystems von MMC	–	–	V3.03	V2.3.2
PROFIBUS-DP-S-Modul	–	–	V3.10	V2.3.2
Suconet-K-Master-Modul	–	–	V3.10	V2.3.2
Diagnose	–	–	V3.10	V2.3.2
Freie Adressvergabe DP	–	–	V3.10	V2.3.2

Abmessungen

XC-CPU101...



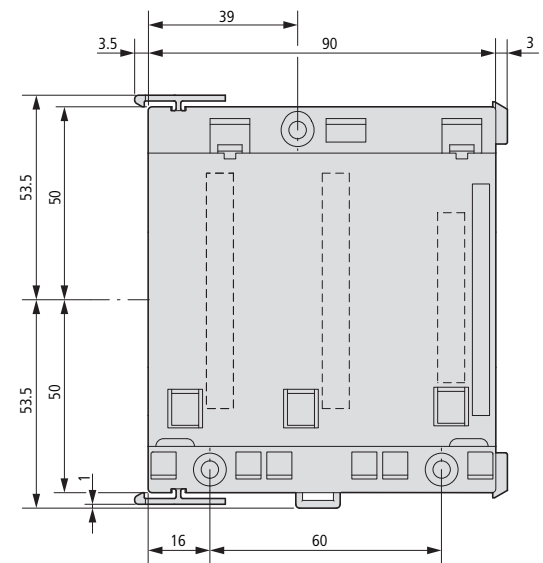
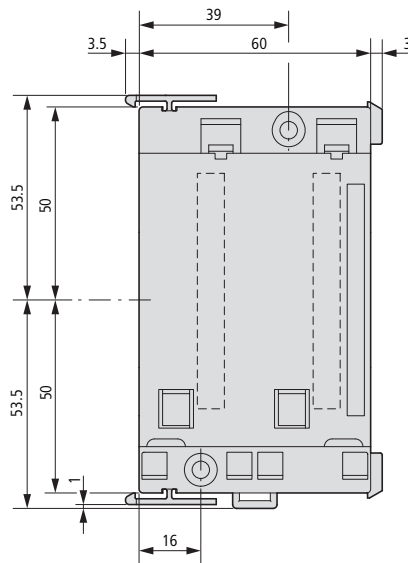
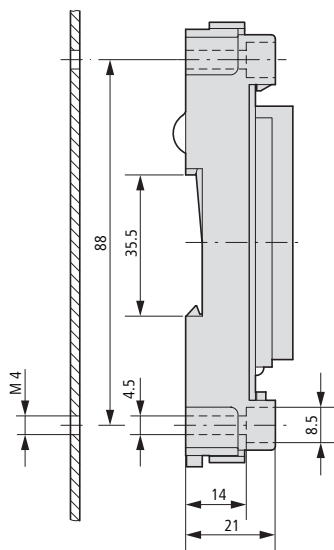
Netzfilter XT-FIL-1



Baugruppenträger

XIOC-BP-XC

XIOC-BP-XC1



Technische Daten

		XC-CPU101-C...K-8DI-6DO(-XV)
Allgemeines		
Normen und Bestimmungen		IEC/EN 61131-2 EN 50178
Umgebungstemperatur	°C	0 bis +55
Lagerung	°C	-25 bis +70
Einbaulage		waagrecht
relative Luftfeuchte, keine Betauung (IEC/EN 60068-2-30)	%	10 bis 95
Luftdruck (Betrieb)	hPa	795 bis 1080
Schwingfestigkeit		10 bis 57 Hz ±0,075 mm 57 bis 150 Hz ±1,0 g
Schockfestigkeit		15 g/11 ms
Überspannungskategorie		II
Verschmutzungsgrad		2
Schutzart		IP20
Bemessungsisolationsspannung	V	500
Störaussendung		EN 50081-2, Klasse A
Störfestigkeit		EN 50082-2
Batterie		Lithium SL-350, 3.6 V, 0.6 mA, Tadiran
Batteriegewicht		9.6 g
SVHC Substanz		Lithium Thionylchlorid
Substanzgewicht		0.35 g
Lebensdauer		Worst case 3 Jahre, typ. 5 Jahre
Gewicht	kg	0,23
Maße (B × H × T)	mm	90 × 100 × 100
Anschlussklemmen		steckbarer Klemmenblock
Anschlussquerschnitte		
Schraubklemmen		
feindrätig mit Aderendhülse	mm ²	0,5 bis 1,5
eindrätig	mm ²	0,5 bis 2,5
Federzugklemmen		
feindrätig	mm ²	0,14 bis 1,0
eindrätig	mm ²	0,34 bis 1,0
Elektromagnetische Verträglichkeit (EMV)		
elektrostatische Entladung (IEC/EN 61000-4-2, Level 3, ESD)		
Kontaktentladung	kV	4
elektromagnetische Felder (IEC/EN 61000-4-3, RFI)		
Burst Impulse (IEC/EN 61000-4-4, Level 3)		
Versorgungsleitungen	kV	2
Signalleitungen	kV	1
energiereiche Impulse (Surge) (IEC/EN 61000-4-5)		

		XC-CPU101-C...K-8DI-6DO(-XV)
Einströmung (IEC/EN 61000-4-6)	V	10

			XC-CPU101-C...K-8DI-6DO(-XV)
Spannungsversorgung der CPU (24 V/0 V)			
Netzausfallüberbrückung			
Dauer des Einbruchs		ms	10
Wiederholrate		s	1
Eingangsspannung		V DC	24
zulässiger Bereich		V DC	20,4 bis 28,8
Eingangsleistung		W	max. 26
Restwelligkeit		%	≤ 5
maximale Verlustleistung (ohne lokale E/A)	<i>P_v</i>	W	6
Überspannungsschutz			ja
Verpolungsschutz			ja
Netzfilter, extern			Typ: XT-FIL-1, siehe Technische Daten auf Seite 77
Netzfilter, intern			ja
Einschaltstrom		× <i>I_n</i>	keine Begrenzung (Begrenzung nur durch vorgeschaltetes 24-V-DC-Netzteil)
Ausgangsspannung für die Signalmodule			
Nennwert		V DC	5
Ausgangsstrom		A	3,2
Leerlauffestigkeit			ja
Kurzschlussfestigkeit			ja
galvanische Trennung zur Versorgungsspannung			nein
CPU			
Mikroprozessor			
			Infineon C164
Speicher			
Programmcode (C64K/C128K/C256K)		kByte	64/128/256
Programmdaten (C64K/C128K/C256K)		kByte	64/128/256
Merker und/oder Retaindaten (C64K/C128K/C256K)		kByte	4/8/16
Zykluszeit für 1 k Anweisungen (Bit, Byte)		ms	0,5
Schnittstellen			
Multi Media Card			
			ja, optional, 16 MB oder 32 MB, seperat zu bestellen
Serielle Schnittstelle (RS232) ohne Handshake-Leitungen			
Datenübertragungsraten		kBit/s	38,4
Anschlusstechnik			RJ45
galvanische Trennung			nein
in der Betriebsart „Transparentmodus“:			
Datenübertragungsraten			300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600 Bit/s
Zeichenformate			8E1, 8O1, 8N1, 8N2, 7E2, 7O2, 7N2, 7E1
Anzahl der Sendebyte in einem Block			190 Byte
Anzahl der Empfangsbyte in einem Block			190 Byte

			XC-CPU101-C...K-8DI-6DO(-XV)
CANopen			
maximale Datenübertragungsrate	Bits/s		Betriebssystem-Version < 2.0: maximal 250 000 Betriebssystem-Version = 2.0: maximal 1 000 000
Potentialtrennung			ja
Geräteprofil			nach DS301V4
PDO-Art			asyn., cyc., acyc.
Anschluss			steckbarer Federzug-Klemmenblock, 6-polig
Busabschlusswiderstände			extern
Teilnehmer	Anzahl		maximal 126
Watchdog			ja
RTC (Real Time Clock)			ja
Spannungsversorgung der lokalen Ein-/Ausgänge (24 V _Q /0 V _Q)			
Eingangsspannung	V DC		24
Spannungsbereich	V DC		19,2 bis 30, Polarität beachten
Potentialtrennung			
Spannungsversorgung gegen CPU-Spannung			ja
Überspannungsschutz			ja
Verpolungsschutz			ja
Digital-Eingänge			
Eingangsstrom pro Kanal bei Nennspannung	mA		typ. 3,5
Verlustleistung pro Kanal			typ. 85 mW
Schaltpegel nach EN 61131-2			
Grenzwerte-Typ „1“			low < 5 V DC, high > 15 V DC
Eingangsverzögerung			
Aus → Ein	ms		typ. 0,1
Ein → Aus	ms		typ. 0,1
Eingänge	Anzahl		8
Kanäle mit gleichem Bezugspotential	Anzahl		8
Zustandsanzeige			LED
Digital-Ausgänge ¹			
Verlustleistung pro Kanal			
QX0.0 bis QX0.5	W		0,08
Laststromkreis			
QX0.0 bis QX0.5	A		0,5
Ausgangsverzögerung			
Aus → Ein			typ. 0,1 ms
Ein → Aus			typ. 0,1 ms
Kanäle	Anzahl		6
Kanäle mit gleichem Bezugspotential	Anzahl		6
Zustandsanzeige			LED
Einschaltdauer	% ED		100
Gleichzeitigkeitsfaktor	g		1

1) Beachten Sie die Einschränkungen beim Einsatz der XC-CPU101 im ABS-Gehäuse, → Tabelle 1 auf Seite 8.

			24-V-DC-Filter XT-FIL-1
Allgemeines			
Normen und Bestimmungen			IEC/EN 61131-2 EN 50178
Umgebungstemperatur	°C		0 bis +55
Lagerung	°C		-25 bis +70
Einbaulage			waagrecht/senkrecht
relative Luftfeuchte, keine Betauung (IEC/EN 60068-2-30)	%		10 bis 95
Luftdruck (Betrieb)	hPa		795 bis 1080
Schwingfestigkeit			10 bis 57 Hz ±0,075 mm 57 bis 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 50081-2, Klasse A
Störfestigkeit			EN 50082-2
Gewicht	g		95
Maße (B × H × T)	mm		35 × 90 × 30
Anschlussklemmen			Schraubklemme
Anschlussquerschnitte			
Schraubklemmen			
feindrätig mit Aderendhülse	mm ²		0,2 bis 2,5 (AWG22-12)
eindrätig	mm ²		0,2 bis 2,5 (AWG22-12)
Spannungsversorgung			
Eingangsspannung	V DC		24
zulässiger Bereich	V DC		20,4 bis 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

Stichwortverzeichnis

A	Abkürzungen	6	D	Datenremanenz	41
	Adressbereich	43		Datensicherung	10, 15
	Adressieren, Ein-/Ausgänge und Merker	43		Datenübertragungsrate ändern	12
	Adressierung			Deklaration	54
	Steuerung am CAN-Bus	56		Diagnose	44
	Adressüberschneidungen	43		über CAN	57
	Akku-Pufferung	41		Direkter Peripheriezugriff	
	Anschluss			Fehlercode	33
	Netzteil und lokales I/O	9			
	Ausgänge adressieren	43	E	Echtzeituhr	15
	Ausschaltverhalten	20		Einbaulage	17
	Automatische Adressierung	43		Eingänge adressieren	43
				Eingangsmodul einfügen (in easySoft-CoDeSys)	52
B	Batterie	15		Einschaltverhalten	19
	Batteriewechsel	15		Einzelschritt-Betrieb	20
	Baudrate	56		Einzelzyklus-Betrieb	20
	Baudrate einstellen	12		Elektromagnetische Beeinflussung	17
	Baudraten, für CAN-Verbindung	57		EMPTY-SLOT	52
	Bausteintyp auswählen	50		Ethernet-Patch-Kabel	12
	Beispielprojekt	47			
	Belüftung	17	F	Fehlercode, bei „Direkter Peripheriezugriff“	33
	Betriebsarten-Vorwahlschalter	11		Forcen	20
	Betriebssystem laden	22		Funktionen	25
	Betriebszustände	41		Funktionen, XIOC	
	Blitzschutzmaßnahmen	18		DisableInterrupt	36
	Blockgröße, für Datenübertragung	55		EnableInterrupt	36
	Blockschaltbild			ReadBitDirect	30
	CPU	11		ReadWordDirect	31
	Versorgungsmodul	8		WriteBitDirect	32
	Bootprojekt			WriteWordDirect	33
	erzeugen	41		Funktionsbausteine	25
	Bootprojekt erzeugen				
	nachOnline-Änderung	21	G	Geräteanordnung	17
	transferieren	21			
	Breakpoint	20	H	Handshake-Leitungen	12
	Browser-Befehle	40		Hardware-Timer	25
C	CAN		I	Induktivitäten	17
	-Deviceparameter	57		Interrupts	34
	-Master-Routingeinstellungen	56			
	CAN-Telegramme		J	Jittern	25
	aus Anwenderprogramm senden/Empfangen	14			
	CoDeSys-Gateway-Server	55			
	CPU-Modul	7			
	CRC-Prüfungen	19			

K	Klemmenbelegung	9	S	Schaltschrankaufbau	17
	Knotennummer	56		Schirmung	17
	Kommunikation mit der Zielsteuerung	57		Schnittstelle	
	Kommunikationskanal	58		CANopen	14
	Kommunikationsparameter	45		Kommunikation (Programmiergerät)	12
	Konfiguration XC100	50		Schnittstellenbelegung, Programmiergeräte-Schnittstelle	12
				Schutzbeschaltung von Störquellen	17
L	LED-Anzeigen	9		Segmente	42
	Leitungsführung	17		Spannungsaus/-abfall	41
	Löschen			Speicher, Applikationsprogramm	10
	Betriebssystem	11		Speichernutzung, Grenzwerte	42
	Bootprojekt	11		Startverhalten	20
	Inhalt der Multi Media Card	11		Startverhalten, nach PowerOn	21
				Stopp-Verhalten	20
M	Memory-Card	11		Störeinflüsse	17
	Merker adressieren	43		Symbole	6
	Montage, CPU	16		Systembibliotheken	25
	Multi Media Card	11		Systemzeit	25
			T	TCP/IP-Verbindung (beim Routing)	55
N	Netzteil	7		Test und Inbetriebnahme	20
	Node-ID	56		Textdisplays anschließen	15
			U	Überwachung, Systemspannung	10
P	Parametrierung	21		Ungerade Wort-Adressen	43
	PC anschließen	45		Unterbrechungsfreie Stromversorgung	41
	Programm erstellen (Beispielprojekt)	54	V	Verdrahtung	17
	Programmbearbeitung	25		Verdrahtungsbeispiel	
	Programmbearbeitung unterbrochen	20		Netzteil	18
	Programmiergerät anschließen	45		Spannungsversorgung der Digital-Ein-/Ausgänge .	18
	Programmierkabel			Versionen, CPU	10
	bei RS232-Schnittstelle	45	W	Warmstart	20
	Programmtransfer	41		Wortadressen, ungerade	43
	Projektierung	17	X	XIOC-Module	7
R	Reset	21	Z	Zustandsanzeige, easySoft-CoDeSys	20
	Restzyklus	41		Zykluszeit, max.	21
	RJ45-Stecker	12		Zykluszeitüberwachung	25
	RS232-Schnittstelle	12			