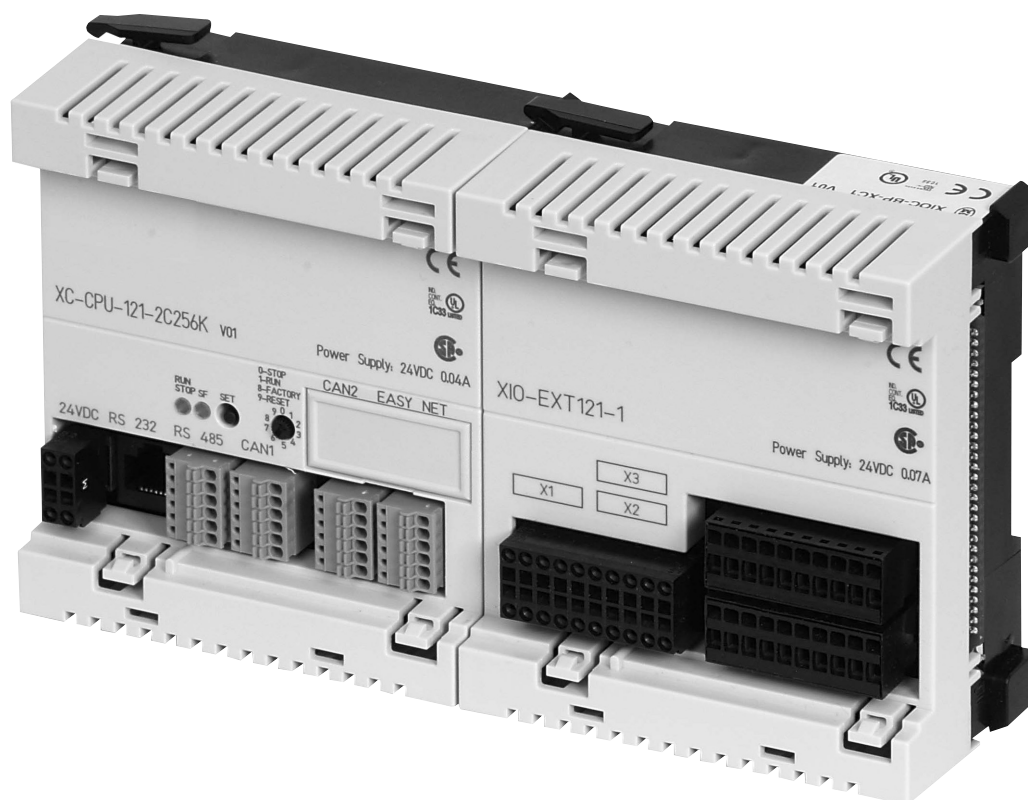


Modular PLC XC-CPU121-2C256K



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

Störfallservice

Bitte rufen Sie Ihre lokale Vertretung an.

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

oder

Hotline After Sales Service

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

AfterSalesEGBonn@eaton.com

Originalbetriebsanleitung

Die deutsche Ausführung dieses Dokuments ist die Originalbetriebsanleitung.

Übersetzung der Originalbetriebsanleitung

Alle nicht deutschen Sprachausgaben dieses Dokuments sind Übersetzungen der Originalbetriebsanleitung.

1. Auflage 2005, Redaktionsdatum 08/05
2. Auflage 2007, Redaktionsdatum 04/07
3. Auflage 2007, Redaktionsdatum 05/07
4. Auflage 2008, Redaktionsdatum 04/08
5. Auflage 2010, Redaktionsdatum 08/10

siehe Änderungsprotokoll im Kapitel „Zu diesem Handbuch“

© 2005 by Eaton Industries GmbH, 53105 Bonn

Autoren: Norbert Mausolf

Redaktion: Thomas Kracht

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 Schutz Erde (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		5
	Änderungsprotokoll	5
	Weiterführende Dokumentationen	5
	Lesekonventionen	5
1 Aufbau der XC121		7
2 XC-CPU121-Funktionen		9
	Betriebsartenschalter (S1)	9
	SET-Taster (S5)	9
	APPLICATION-Schalter (S2)	10
	DIP-Switch-Betriebsart	10
	Speicherkarte – MMC	11
	– Datenzugriff auf die MMC	11
	– Löschfunktionen	11
	LED-Zustandsanzeige RUN/STOP und SF	11
	Echtzeituhr	11
	Grenzwerte für die Speichernutzung	12
	Serielle Schnittstellen COM1/COM2	12
	CANopen-Schnittstellen CAN1/CAN2	13
	– XC121 als CAN-Device (CAN1 und/oder CAN2)	13
	– Einstellung der XC121 als CAN-Master/CAN-Device	14
	– Nutzung der CAN-Bibliotheken	14
	– CAN-Direct	14
	– Busabschlusswiderstände	14
	– Eigenschaften des CANopen-Kabels	14
3 Montage		15
	XC121 auf Hutschiene montieren	15
	XIO-EXT121-1 montieren	15
	APPLICATION-Schalter einstellen	16
	Ein-/Ausgänge verdrahten	16
4 Projektierung		17
	Schaltschrankaufbau	17
	– Belüftung	17
	– Geräteanordnung	17
	Störungen vermeiden	17
	– Schutzbeschaltung von Störquellen	17
	– Schirmung	17
	Blitzschutzmaßnahmen	18
	Anschlüsse	18
	– Spannungsversorgung anschließen	18
	– Sensoren und Aktoren anschließen	19
5 Konfiguration der XIO-EXT121-1		21

6 Betrieb		23
	Einschaltverhalten	23
	– Einschaltverhalten mit Bootprojekt	23
	Startverhalten in der CoDeSys einstellen	24
	Programm START/STOP	24
	– Programmstart (STOP → RUN)	24
	– Verhalten nach dem Ausschalten/Unterbrechen der Spannungsversorgung	24
	– Programmstopp (RUN → STOP)	25
	Programmbearbeitung und Systemzeit	25
	Zykluszeit überwachen	25
	Reset	25
	– Reset (warm)	25
	– Reset kalt	25
	– Reset Ursprung	25
	– Reset zum Wiederherstellen der Werkseinstellungen	25
	– Verhalten der Variablen nach Reset	25
	Test und Inbetriebnahme	26
	– Breakpoint/Einzelschritt-Betrieb	26
	– Einzelzyklus-Betrieb	26
	– Zwangssetzen von Variablen und I/Os (Forcen)	26
	– Zustandsanzeige in der Software CoDeSys	26
	Systemereignisse	27
	Interruptverarbeitung	27
	– Parametrierung der Eingänge	27
	– Beispiel zur Interruptverarbeitung	28
	– Timer-Interrupt	29
	Direkter I/O-Zugriff	30
	– ReadBitDirect	30
	– Fehlercode bei „Direkter Peripheriezugriff“	31
	Bootprojekt erzeugen und transferieren	31
	– Bootprojekt auf MMC abspeichern	31
	– Bootprojekt löschen	31
	Betriebssystem herunterladen/aktualisieren	32
	– Betriebssystem vom PC in die Steuerung übertragen	32
	– BTS vom PC auf die MMC übertragen	33
	– BTS von der MMC in die Steuerung übertragen	33
	Quellcode des Anwenderprogramms	33
7 Browser-Befehle		35
	– reflect	36
	– canload	36
	– setrtc	36
8 Bibliotheken, Funktionsbausteine und Funktionen		37
	Bibliotheken handhaben	37
	Weitere Systembibliotheken installieren	37
	XC121-spezifische Funktionen	38
	– Bibliothek „XC121_Util.lib“	38
	– Funktion CAN_BUSLOAD	38
	– Funktion GETAPPLICATIONSWITCH	38
9 Verbindungsaufbau PC – XC121		39
	Kommunikationsparameter des PC	39
	Kommunikationsparameter (Baudrate) der CPU	39

10 Systemparameter über STARTUP.INI-Datei einstellen		41
	Parameterübersicht	41
	Aufbau der INI-Datei	41
	INI-Datei erstellen	41
	Einträge der INI-Datei	42
	Startverhalten der XC121 bei gesteckter MMC mit INI-Datei	42
	Parameter ändern	42
	INI-Datei löschen	42
11 Programmieren über CANopen-Netzwerk (Routing)		43
	Voraussetzungen	43
	Routingeeigenschaften der XC121	43
	– Einstellung über den Steuerungskonfigurator	43
	– Einstellung über den APPLICATION-Schalter	44
	Routing über XC200	44
	Hinweise	44
	Adressierung	45
	Vorgehensweise	45
	Steuerungskombinationen zum Routing	46
12 RS232-Schnittstelle im Transparent-Modus		47
Anhang		49
	Abmessungen	49
	Technische Daten	50
	– XC-CPU121/XIO-EXT121-1	50
	– XC-CPU121	51
	– XIO-EXT121-1	53
	– 24-V-DC-Netzfilter XT-FIL-1	55
Stichwortverzeichnis		57

Zu diesem Handbuch

Änderungsprotokoll

Redaktionsdatum	Seite	Stichwort	neu	Änderung	entfällt
04/07	12	COM2: Transparent-Modus, Signal		✓	
	14	Nutzung der CAN-Bibliotheken	✓		
	18	Spannungsversorgung anschließen		✓	
	19	Sensoren und Aktoren anschließen		✓	
	53	Verzögerungszeit		✓	
	54	Pt100	✓		
05/07	18	Spannungsversorgung anschließen		✓	
04/08	51- 54	Technische Daten XC-CPU121, XIO-EXT121-1	✓	✓	✓
	13, 16	Grafik		✓	
08/10	20	Pt100-Eichkurve für XC121 (Simulator: burster Precision RTD Simulator 4530)	✓		
	54	Technische Daten XIO-EXT121-1, Genauigkeit	✓		
08/10	12	Schnittstellenbelegung		✓	
	allgemein	Umstellung auf Eaton-Bezeichnungen		✓	

Weiterführende Dokumentationen

In diesem Handbuch wird an verschiedenen Stellen auf ergänzende oder vertiefende Beschreibungen in anderen Dokumentationen hingewiesen. Diese Dokumentationen werden als PDF-Datei bei der Installation der Produkt-CD auf Ihrem PC gespeichert.

Für ein schnelleres Auffinden der gesuchten Dokumentation wählen Sie im Windows-Startmenü:

Programme → Moeller Software → easy Soft CoDeSys → Dokumentation...

Darüber hinaus besteht die Möglichkeit, die PDF-Dateien über den FTP-Server herunterzuladen. Hier stehen immer die aktuellen Daten zur Verfügung:

https://es-assets.eaton.com/DOCUMENTATION/AWB_MANUALS/

Konkrete Informationen über die Kommunikation mit CAN-Teilnehmern und deren Konfiguration finden Sie in den folgend aufgeführten Dokumentationen:

- Kommunikation zwischen zwei Steuerungen mit Netzwerkvariablen über CAN (AN2700K19D.pdf)
 - Kopplung mehrerer autarker Steuerungen (CAN-Device) über CANopen (AN2700K20D.pdf)
 - Projektierung von CAN-Teilnehmern (AN2700K27D.pdf)
- (Zu finden im Windows-Startmenü unter Programme → Moeller Software → easy Soft CoDeSys → Anwendungsbeispiele...)

- MN05010001Z (vormals AWB2786-1554D): Bibliotheksbeschreibung CANUser.lib, CANUser_Master.lib.
Mit den Funktionen der Bibliothek CANUser.lib und CANUser_Master.lib können Sie direkt auf die CAN-Objekte zugreifen.

(Zu finden im Windows-Startmenü unter Programme → Moeller Software → easy Soft CoDeSys → Dokumentation...)

Lesekonventionen

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

**Achtung!**

warnt vor leichten Sachschäden.

**Warnung!**

warnt vor schweren Sachschäden und leichten Verletzungen.

**Gefahr!**

warnt 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.

1 Aufbau der XC121

Die Steuerung XC121 ist für den Einsatz in Maschinen- und Anlagensteuerungen konzipiert. Für die Programmierung ist die Software CoDeSys erforderlich.

Die CPU XC-CPU121-2C256K können Sie autark einsetzen und über die CANopen-Schnittstellen mit Ein-/Ausgangsgeräten verbinden. Zur lokalen Erweiterung der CPU mit Ein-/Ausgängen (E/A) dient das baugleiche E/A-Modul XIO-EXT121-1, das über analoge und digitale E/As verfügt. An das Modul können Sie alle weiteren XI/OC-Signalmodule stecken; ausgenommen die PROFIBUS-DP-Module.

Die Steuerung XC121 besteht aus:

- XC-CPU121 mit Netzteil
- E/A-Modul XIO-EXT121-1 mit Netzteil und digitalen und analogen E/As, die im Folgenden als lokale E/As bezeichnet werden.
- zentralen XI/OC-Signalmodulen.

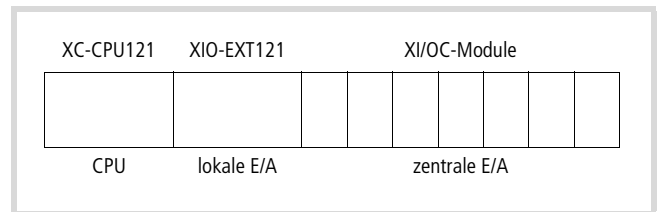


Abbildung 1: Aufbau

Die CPU sowie das CPU-E/A-Modul haben als Plattform den Baugruppenträger mit drei Steckplätzen. Um die CPU mit E/As zu erweitern, verbinden Sie zunächst das XIO-EXT121-Modul mit der CPU. Dann können Sie die Baugruppenträger XIOC-BP-3 und XIOC-BP-2 anknüpfen, auf die Sie die XI/OC-Signalmodule stecken. Eine Erweiterung innerhalb der XI/OC-Baugruppenträger erfolgt durch Einsatz des Baugruppenträgers XIOC-BP-EXT. Im Grundausbau mit den XI/OC-Baugruppenträgern stehen maximal sieben Steckplätze, im Gesamtausbau maximal 15 Steckplätze für die XI/OC-Signalmodule zur Verfügung.

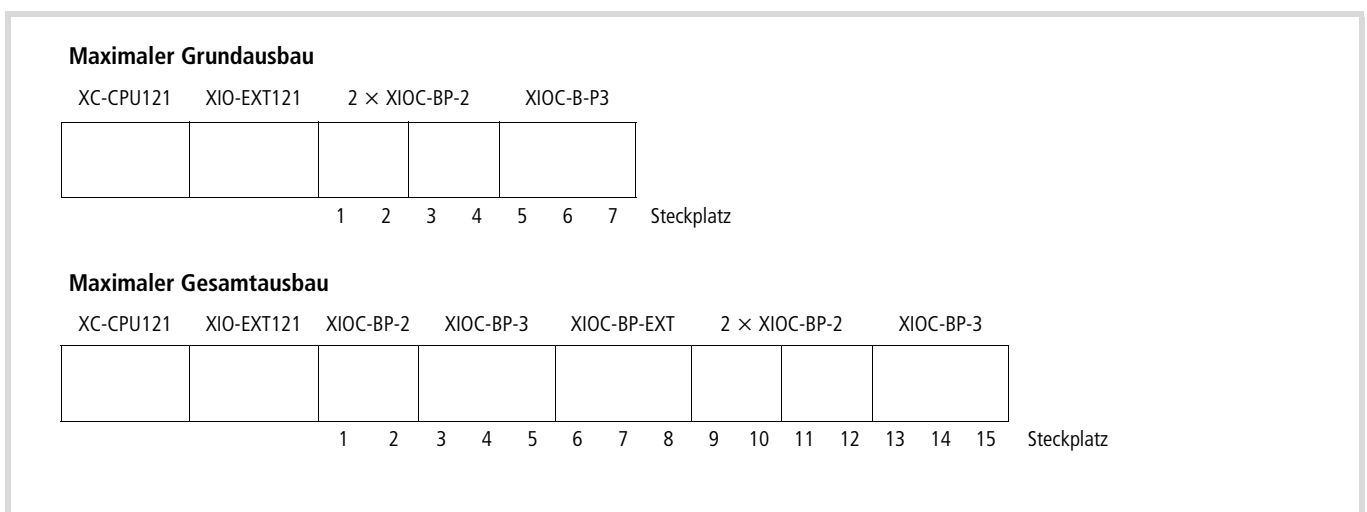


Abbildung 2: Ausbau mit XIOC-Signal-Modulen

Ausführlichere Informationen zu den Baugruppenträgern und XI/OC-Signalmodulen finden Sie in dem Handbuch „XI/OC-Signalmodule“ (MN05002002Z-DE, frühere Bezeichnung AWB2725-1452D).

2 XC-CPU121-Funktionen

Im Folgenden werden die Funktionen näher erläutert.

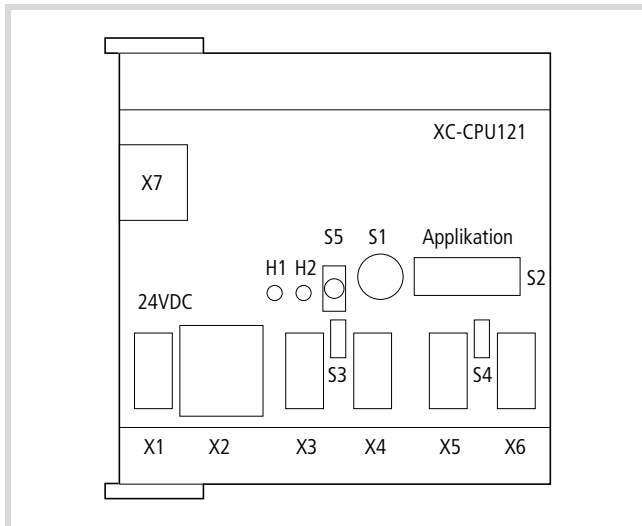


Abbildung 3: Übersicht XC-CPU121

Legende zu Abbildung 3:

H1	LED-Anzeige RUN/STOP
H2	LED-Anzeige SF
S1	Betriebsartenschalter
S2	APPLICATION-Schalter
S3	Schalter für Busabschlusswiderstände für CAN1-Schnittstelle
S4	Schalter für Busabschlusswiderstände für CAN2-Schnittstelle
S5	SET-Taster
X1	Anschluss zur Spannungsversorgung
X2	COM1-Schnittstelle (RS232) zum Anschluss eines Programmiergerätes
X3	COM2-Schnittstelle (RS232/RS485)
X4	CANopen-Schnittstelle CAN1
X5	CANopen-Schnittstelle CAN2
X6	CANopen-Schnittstelle CAN2
X7	Steckplatz für MMC (Multimedia Card)

Betriebsartenschalter (S1)

Mit dem Betriebsartenschalter können Sie die in Tabelle 1 dargestellten Funktionen einstellen.

Tabelle 1: Funktionen des Betriebsartenschalters

Schalterstellung	Funktion
0	STOP
1	RUN Um die CPU zu starten, stellen Sie den Betriebsartenschalter in Stellung „1“ und betätigen anschließend den SET-Taster.
2, 3, ..., 7	STOP
8	Auslieferungszustand wiederherstellen: Wenn Sie den SET-Taster mindestens 3 Sekunden drücken, werden die Werte übernommen. CPU → STOP Achtung! Entnehmen Sie zuvor die Multi Media Card, da sonst das Programm gelöscht wird.
9	„Reset kalt“ durchführen: Wenn Sie den SET-Taster mindestens 3 Sekunden drücken, wird ein Reset durchgeführt. CPU → STOP

Weitere Informationen → Kapitel „Programm START/STOP“ ab Seite 24.

SET-Taster (S5)

Der SET-Taster ist nur in Verbindung mit den Einstellungen 1, 8 und 9 des Betriebsartenschalters aktiv. Wenn Sie den SET-Taster bei Schalterstellung 8 oder 9 drücken, wird die Funktion „Reset kalt“ bzw. „Auslieferungszustand wiederherstellen“ ausgeführt (RUN/STOP-LED blinkt schnell).

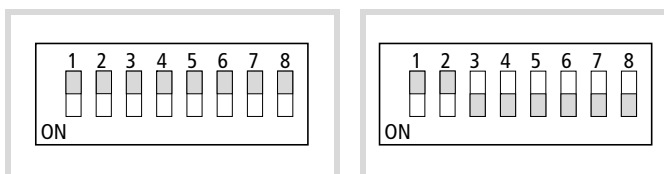
APPLICATION-Schalter (S2)

Über den 8-poligen DIP-Schalter können Sie Werte einstellen, die je nach eingestellter DIP-Switch-Betriebsart unterschiedlich zu bewerten sind. So könnte es sich z. B. bei der Betriebsart „Node ID CAN1“ um die eingestellte Knotenadresse (Node-Id) für den Kanal 1 handeln.

Die DIP-Switch-Betriebsart stellen Sie im Fenster „Steuerungskonfiguration“ ein (siehe folgenden Abschnitt). Den eingestellten Wert am Schalter können Sie im Anwenderprogramm mit Hilfe der Funktion „GetApplicationSwitch“ abfragen. Diese Funktion gehört zur Bibliothek „XC121_Util.lib“. Sie interpretiert die Schalterstellung als binären Wert:

- Schalter 1 = niederwertigstes Bit;
- Schalter 7 = höchstwertigstes Bit;
- Schalter 8 = Auswahl der CAN-Schnittstelle bei Node-Id-Routing: OFF = CAN1; ON = CAN2.

Z. B. liefert die Funktion mit der rechts dargestellten Schaltereinstellung den Wert 3.



Einstellung im
Auslieferungszustand

Einstellung für Adresse 3

Abbildung 4: DIP-Switch-Applikationschalter

DIP-Switch-Betriebsart

Die Betriebsart stellen Sie in der Steuerungskonfiguration ein. Aktivieren Sie das Register „Weitere Parameter“ und wählen Sie im Feld „DIP-Switch-Betriebsart“ eine der folgenden Betriebsarten aus:

- Application
- Node Id CAN1
- Node Id CAN2
- Node Id Routing

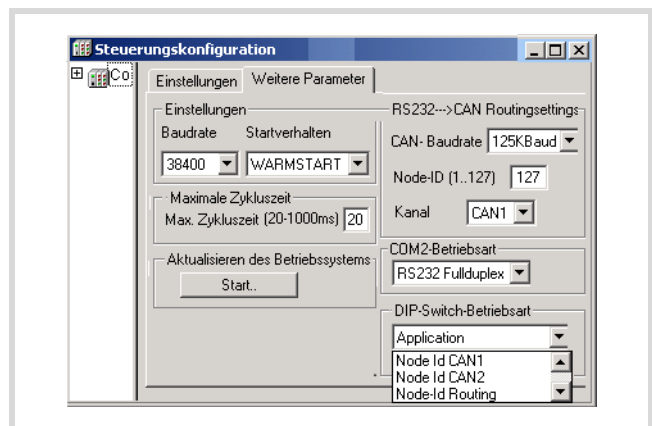


Abbildung 5: Betriebsart des APPLICATION-Schalters

Application: Diese Einstellung bewirkt, dass die DIP-Switch-Betriebsarten „Node Id CAN1/CAN2“ und Node-Id Routing nicht aktiv sind.

Node Id CAN1/CAN2: Die XC121 kann für den CAN1- und/oder CAN2-Kanal (Schnittstellen) als Device-Gerät arbeiten. Für einen der beiden Kanäle „Node Id CAN1“ oder „Node Id CAN2“ können Sie am APPLICATION-Schalter eine Node-Id einstellen. In der Abbildung 4, rechte Darstellung, wurde beispielsweise die Node-Id 3 eingestellt.

Mehr Informationen hierzu finden Sie im Abschnitt „CANopen-Schnittstellen CAN1/CAN2“ auf Seite 13.

Node-Id Routing: Die XC121 können Sie beim Routing als Zielsteuerung einsetzen. Der Zugriff erfolgt über den CAN1- oder CAN2-Kanal (Schnittstellen). Stellen Sie für den Kanal eine Routing-Id am APPLICATION-Schalter ein.

In der Abbildung 4 wurde in der rechten Darstellung die Routing-Id „3“ eingestellt.

Mehr Informationen hierzu finden Sie im Abschnitt „CANopen-Schnittstellen CAN1/CAN2“ auf Seite 13.

Speicherkarte – MMC

Die MMC dient als Massenspeicher. Das Betriebssystem (BTS) unterstützt Speichertypen mit dem FAT16-Dateisystem. Sie können das BTS auch auf die MMC übertragen, um es von dort in andere XC121 zu laden.

→ Abschnitt „Betriebssystem herunterladen/aktualisieren“ auf Seite 32.



Achtung!

Das Dateisystem der Speicherkarten ist nicht transaktionsicher. Stellen Sie sicher, dass Sie alle Dateien des Programms geschlossen haben, bevor Sie die MMC ziehen bzw. stecken oder die Spannung ausschalten.

Datenzugriff auf die MMC

Mit Hilfe von Browser-Befehlen und Funktionen können Sie allgemeine Dateien sowie das Bootprojekt oder den Quellcode des Projekts auf die MMC übertragen bzw. herunterladen. Um z. B. das Bootprojekt auf die MMC zu kopieren, verwenden Sie den Browser-Befehl „copyprojtoMMC“.

Eine Kurzbeschreibung der Browser-Befehle finden Sie ab Seite 35.

Mit Funktionen wie „FileOpen“ oder FileRead“ können Sie aus dem Anwenderprogramm heraus auf die Dateien der MMC zugreifen. Diese Funktionen sind in der Bibliothek „XC121_File.lib“ enthalten und in dem Handbuch „CoDeSys-Funktionsbausteine“ (MN0501002Z-DE, frühere Bezeichnung AWB2786-1456D) beschrieben.

Löschfunktionen

Wenn Sie im Online-Betrieb den Befehl RESET-Ursprung im Steuerungskonfigurator ausführen, wird das Betriebssystem und das Projekt auf der MMC gelöscht. Die Parameter der Datei STARTUP.INI bleiben erhalten.

Sie können auch folgende Browser-Befehle anwenden:

- „format“: löscht den gesamten Inhalt der MMC
 - „removeprojfrommmc“: löscht das Projekt und die INI-Datei auf der MMC. Die Daten auf der CPU bleiben erhalten
- Abschnitt „INI-Datei löschen“ auf Seite 42.

LED-Zustandsanzeige RUN/STOP und SF

Tabelle 2: LED-Zustandsanzeige

LED	Bedeutung
RUN/STP + SF	
aus + rot	Systemtest wird durchgeführt (bis 6 Sekunden nach Start; nach 6 Sekunden, wenn kein Anwenderprogramm vorhanden ist) CPU im NOT READY!
grün + rot	System-Update wird durchgeführt
beide blinkend	Systemtest ergab einen Fehler
grün blinkend + aus	Anwenderprogramm geladen CPU im STOP!
grün + aus	Anwenderprogramm geladen CPU im RUN!
grün blinkend + rot	Sammelfehler/Diagnosemeldung steht an.
grün (während Reset schnell blinkend) + rot (abhängig vom Sammelfehler aus/ein)	Reset über Betriebsartenschalter (Schalterstellung 9)

Echtzeituhr

Die XC121 besitzt eine Echtzeituhr, die Sie im Anwenderprogramm über Funktionen aus der Bibliothek „SysLibRTC“ ansprechen können. Die Funktionen sind in der PDF-Datei „SysLibRTC“ beschrieben. Diese Datei finden Sie im Windows Startmenü unter „Programme → Moeller Software → easy Soft CoDeSys → Dokumentation...“.

Sie können die Uhr auch mit den Browser-Befehlen „getrtc“ und „setrtc“ lesen bzw. setzen. Mehr Informationen finden Sie im Abschnitt „setrtc“ auf Seite 36.

Bei Spannungsausfall wird die Uhr mindestens 72 Stunden gepuffert.

Grenzwerte für die Speichernutzung

Der Datenspeicher der XC121 ist in Speichersegmente gegliedert. Die Speichergröße der einzelnen Segmente geht aus der Abbildung 6 hervor. Die globalen Daten verfügen über mehrere Segmente. Die gewünschte Anzahl können Sie in Abhängigkeit von der Programmgröße festlegen.

Die eingestellte Segmentgröße für die verschiedenen Steuerungstypen erkennen Sie unter «Zielsystem Einstellungen → Speicheraufteilung»:

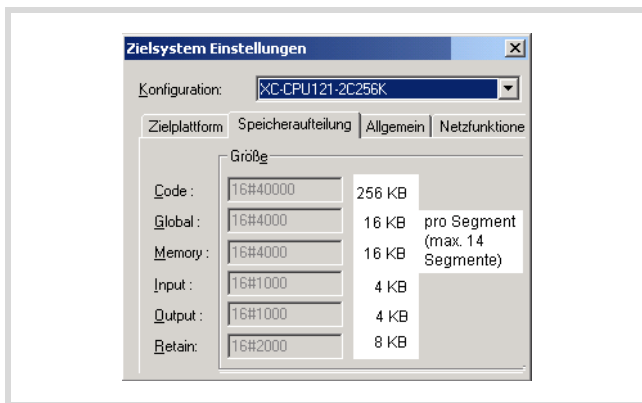


Abbildung 6: Speicheraufteilung

Um den zur Verfügung stehenden maximalen Speicherbereich der globalen Daten maximal ausnutzen zu können, sollten Sie die Anzahl der globalen Datensegmente schon bei der Neuanlage eines Projektes auf 14 setzen. Defaultmäßig steht die Anzahl der Segmente auf 1.

Anzahl der Segmente verändern

Wählen Sie im Verzeichnis «Projekt Optionen → Übersetzungsoptionen» das Feld „Datensegmente“ an und tragen Sie bei ausgewähltem Steuerungstyp XC121 für die Anzahl der Segmente eine „14“ ein.

Serielle Schnittstellen COM1/COM2

Die CPU verfügt über zwei serielle Schnittstellen COM1 und COM2.

COM1: Programmierschnittstelle/Transparent-Modus

Über die Programmierschnittstelle **COM1** (RS232) erfolgt die Kommunikation zwischen der CPU und dem Programmiergerät. Beim Start der Steuerung wird die Schnittstelle mit folgenden Defaultparametern initialisiert.

Datenlänge:	8 Bit
Parität:	keine
Stopbits:	1
Baudrate:	38400 Baud

Tabelle 3: Belegung der Programmierschnittstelle

		Signal
8	8	RxD
7	7	GND
6	6	–
5	5	TxD
4	4	GND
3	3	–
2	2	–
1	1	–

COM2: Transparent-Modus

Nähere Informationen zum Transparent-Modus finden Sie ab Seite 47.

Die COM2-Schnittstelle ist zwischen RS232 (voll duplex) und RS485 (halbduplex) umschaltbar. Die Einstellung erfolgt in der Steuerungskonfiguration.

Sie ist nicht galvanisch getrennt und kann nur über Funktionsbausteine aus dem Anwenderprogramm heraus angesprochen werden. Als Programmierschnittstelle können Sie diese Schnittstelle nicht nutzen. Sie wird mit folgenden Default-Parametern initialisiert:

Datenlänge:	8 Bit
Parität:	keine
Stopbits:	1
Baudrate:	38400 Baud

Weitere Kommunikationsparameter finden Sie im Abschnitt „Technische Daten“ auf Seite 50.

Steuerleitungen der RS232 werden nicht unterstützt.

	RS232	RS485
	Signal	Signal
6	RxD	–
5	TxD	–
4	Vcc	–
3	GND	GND
2	–	Tx–/Rx–
1	–	Tx+/Rx+

① externer Busabschlusswiderstand (120 Ω), 1. und letztes Gerät am Bus

Als Steckertyp wird ein 6-poliger steckbarer Federzugklemmenblock eingesetzt.

→ Nähere Informationen zum Transparent-Modus finden Sie ab Seite 47.

CANopen-Schnittstellen CAN1/CAN2

Die Steuerung verfügt über zwei CANopen-Schnittstellen. Diese werden mit CAN1 und CAN2 bezeichnet. Die CAN2-Schnittstelle ist mehrfach ausgelegt. Sie liegt sowohl am Steckverbinder X5 als auch X6 an.

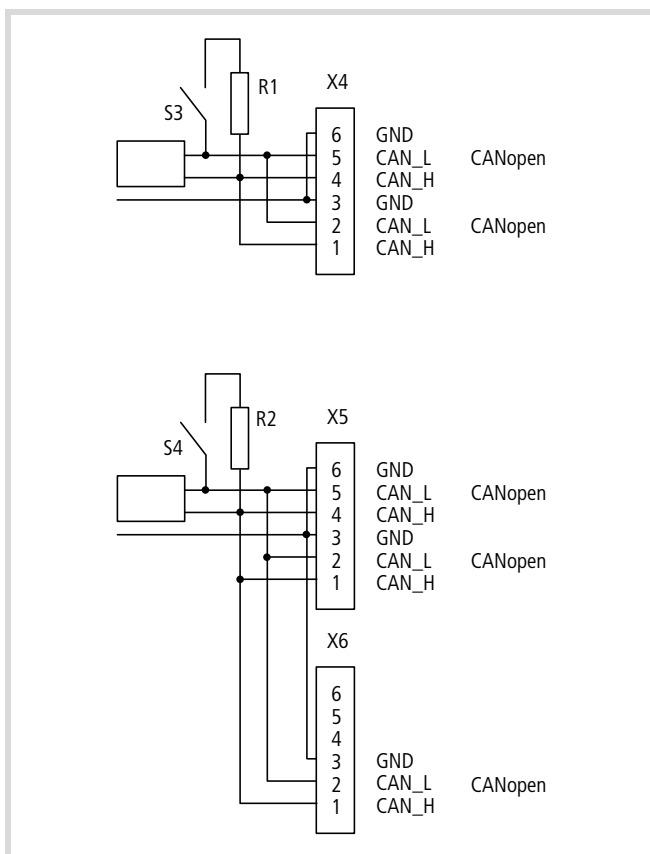


Abbildung 7: Übersicht CANopen-Schnittstellen

- X 4, 5, 6 6 poliger, steckbarer Federzugklemmenblock
- S 3,4 Schalter für Busabschlusswiderstand
- R 1,2 Busabschlusswiderstand 120 Ω

Beide CANopen-Schnittstellen sind nach der CIA-Spezifikation DS 301 V4.0 ausgelegt. Sie können unabhängig voneinander sowohl als CAN-Master als auch als CAN-Device betrieben werden.

XC121 als CAN-Device (CAN1 und/oder CAN2)

In der Abbildung 8 wurde eine XC121 mit zwei CAN-Device-Kanälen konfiguriert. Jedem benutztem Kanal ist eine Node-Id zuzuordnen, die als Adresse dient. Zur Einstellung der Node-Id können Sie zwischen zwei verschiedenen Verfahren wählen.

- Einstellung über den Steuerungskonfigurator
- Einstellung über den APPLICATION-Schalter.

→ Die Einstellungen am APPLICATION-Schalter (für Node-Id und Routing-Id) haben Vorrang gegenüber der Konfiguratoreinstellung.

Einstellung über den Steuerungskonfigurator

- Öffnen Sie im Steuerungskonfigurator das Register „Weitere Parameter“. Im Feld „DIP-Switch-Betriebsart“ (→ Abbildung 5) darf die Betriebsart Node ID CAN1 oder Node ID CAN2 nicht angewählt sein!
- Klicken Sie auf den ersten Ordner CANDevice[VAR]. Das Register „Grundeinstellungen“ wird angezeigt.

→ Dem ersten Ordner CANDevice[VAR] wird grundsätzlich der Kanal CAN1 zugewiesen. Der folgende Ordner erhält den Kanal CAN2.

- Tragen Sie anstelle „CAN1“ einen beliebigen „Namen des Busses“ ein (CAN1 hat keine Bedeutung).
- Wechseln Sie in das Register „CAN-Einstellungen“ und stellen Sie die Node-Id und die Baudrate ein.
- Parametrieren Sie den CAN2-Kanal nach dem gleichen Verfahren.

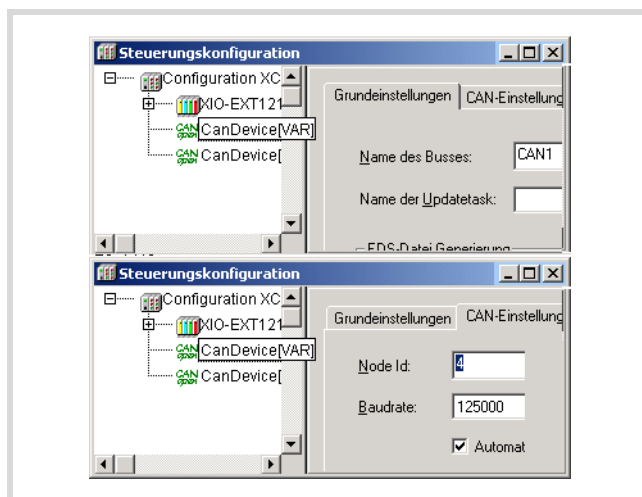


Abbildung 8: Node-Id-Einstellung über Konfigurator (CAN1 = Device)

Einstellung über den APPLICATION-Schalter

Die Node-Id einer Schnittstelle (CAN1 oder CAN2) können Sie über den APPLICATION-Schalter einstellen, die Node-Id der weiteren Schnittstellen müssen Sie im Steuerungskonfigurator einstellen. Dazu muss im Feld „DIP-Switch-Betriebsart“ die Betriebsart „Node Id CAN...“ eingestellt sein (→ Abbildung 5 auf Seite 10). Am APPLICATION-Schalter stellen Sie nun an den Schaltern 1 bis 7 die Node-Id ein; der Schalter 8 ist ohne Funktion.

Einstellungen	
Schalter 1 – 7	Node Id 1 – 127 (Bei ungültiger Adresse 0 wird Default-Node Id 127 verwendet!)
Schalter 8	ohne Funktion

So stellen Sie die Baudrate ein:

- Klicken Sie auf den Ordner „CAN Device [Var]“, öffnen Sie das Register „CAN-Einstellungen“ und geben Sie die Baudrate ein.

Einstellung der XC121 als CAN-Master/CAN-Device

Die Abbildung zeigt ein Beispiel für die XC121 als CAN-Master und als CAN-Device.

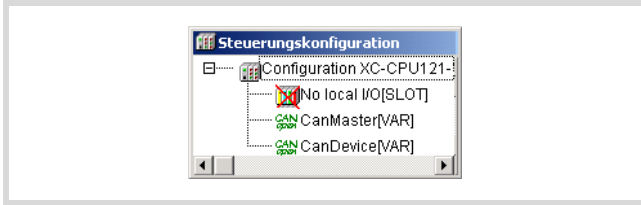


Abbildung 9: XC121 als CAN-Master und als CAN-Device

Dem ersten Ordner „CanMaster[VAR]“ ist grundsätzlich der Kanal CAN 1 zugeordnet. Der zweite Ordner „CanDevice[VAR]“ erhält den Kanal 2. Die Parameter für den Kanal 2 stellen Sie in der Steuerungskonfiguration oder über den Applikationsschalter ein.

➔ Höhere Übertragungsraten als 500 kBit/s werden nicht unterstützt, obwohl sie in der Steuerungskonfiguration auswählbar sind.

Nutzung der CAN-Bibliotheken

Die Bibliotheken CanUser.lib und CanUser_Master.lib stellen Ihnen einen steuerungsübergreifenden Zugriff auf CAN-Objekte zur Verfügung. Hierzu zählen insbesondere CAN-Direkt-Funktionen/Funktionsbausteine, wie das direkte Lesen und Schreiben von CAN-Telegrammen und weitere CANopen-Funktionen wie das Senden und Empfangen von Daten über SDO-Funktionen oder das Abrufen von Diagnoseinformationen aus dem Anwenderprogramm.

CAN-Direct

Über die Funktionsbausteine der Library „SysLibCan“ greifen Sie direkt auf CAN-Objekte zu (➔ Anwenderhinweis AN2700K27). Für jede der beiden CAN-Schnittstellen werden vier unabhängige Ports eingerichtet:

CAN1 – CAN4	Zugriff auf den an CAN1 angeschlossenen CAN-Strang
SECOND_CAN1 – SECOND_CAN4	Zugriff auf den an CAN2 angeschlossenen CAN-Strang

Busabschlusswiderstände

Für jede der beiden CAN-Schnittstellen können Sie den Busabschlusswiderstand zu- oder abschalten. Die Umschalter S3 und S4 befinden sich neben den Steckverbindern (➔ Abbildung 3).



ausgeschaltete
Busabschlusswiderstände

eingeschaltete
Busabschlusswiderstände

Abbildung 10: Busabschlusswiderstände schalten

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 Tabelle 4 sind Standard-Parameter für das CANopen-Netzwerk mit weniger als 64 CANopen-Teilnehmern aufgelistet.

Tabelle 4: Standardparameter für CANopen-Netzwerkabel entsprechend der ISO 11898

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

3 Montage

XC121 auf Hutschiene montieren

- ▶ Haken Sie die XC-CPU121/XIO-EXT121-1 von oben auf die Hutschiene.
- ▶ Ziehen Sie den Verriegelungsschieber nach unten **1**.
- ▶ Drücken Sie die Unterseite des Moduls zur Hutschiene **2**.
- ▶ Drücken Sie den Verriegelungsschieber wieder nach oben **3**.
- ▶ Achten Sie auf den richtigen und festen Sitz auf der Hutschiene.

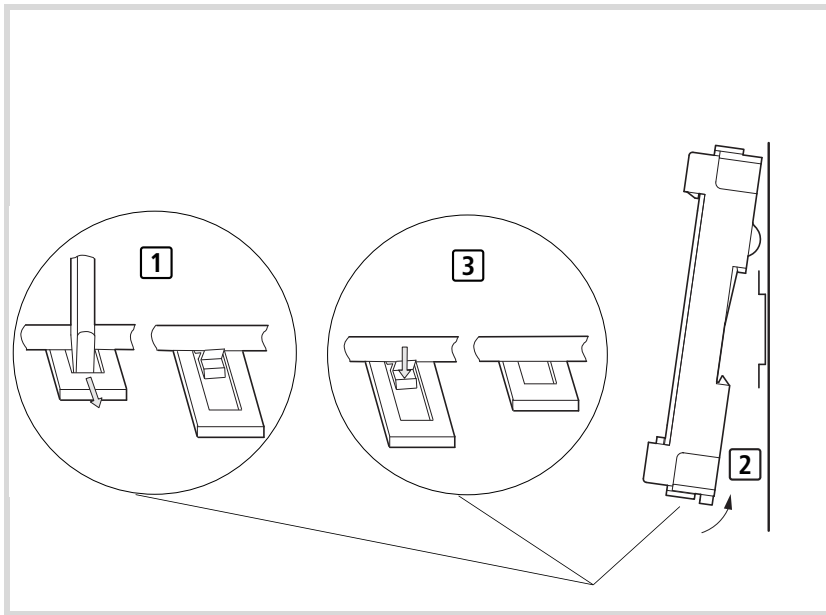


Abbildung 11: Montage der XC121

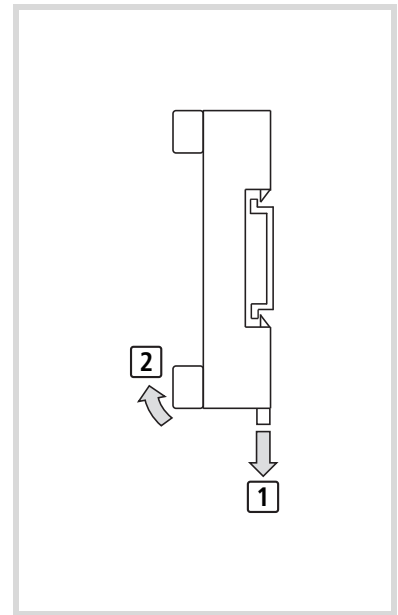


Abbildung 12: XC121 demontieren

XIO-EXT121-1 montieren

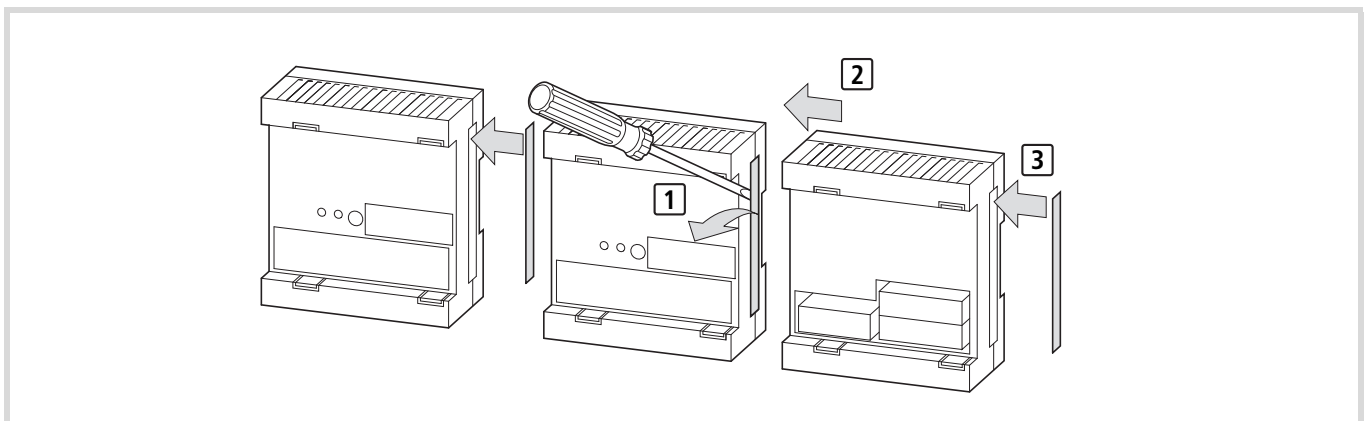


Abbildung 13: XIO-EXT121-1 an XC121 montieren

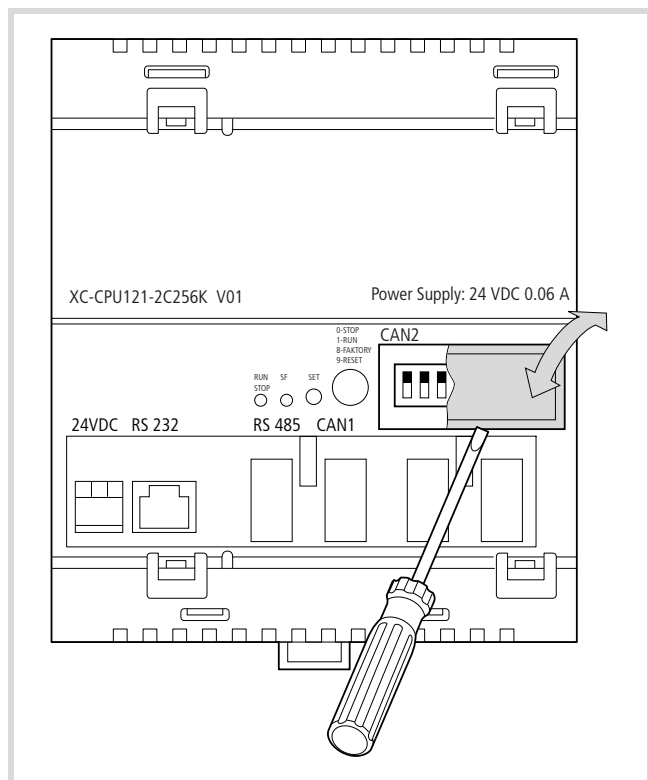
APPLICATION-Schalter einstellen

Abbildung 14: APPLICATION-Schalter einstellen

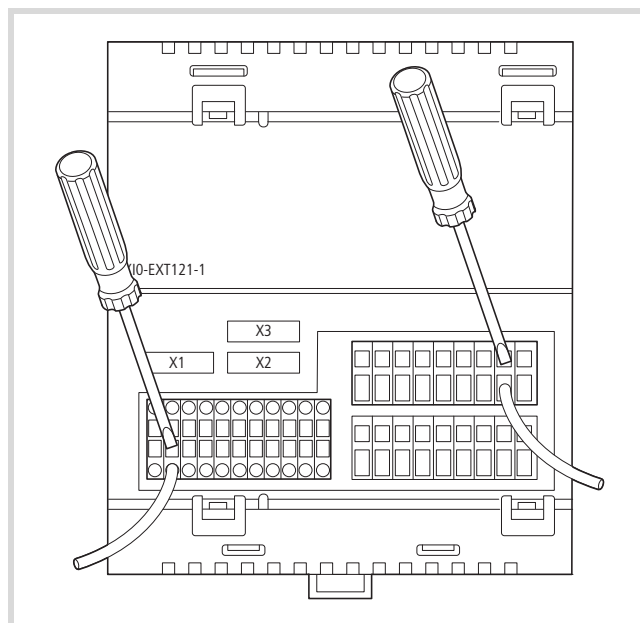
Ein-/Ausgänge verdrahten

Abbildung 15: Ein-/Ausgänge verdrahten

4 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 Steuerung waagrecht in den Schaltschrank ein:

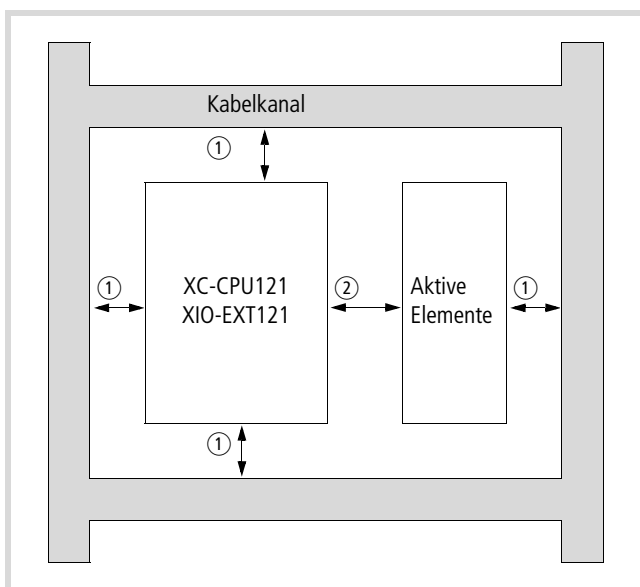


Abbildung 16: Schaltschrankaufbau

- ① Abstand > 50 mm
- ② Abstand > 75 mm

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 schutzbeschaltet 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 z. B. 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 „EMV-Projektierungsrichtlinie für Automatisierungssysteme“.
- TB27-001-D „Elektromagnetische Verträglichkeit (EMV) von Automatisierungsanlagen“.
- TB02-022-D „Elektromagnetische Verträglichkeit (EMV) von Maschinen und Anlagen“.

Anschlüsse

Spannungsversorgung anschließen

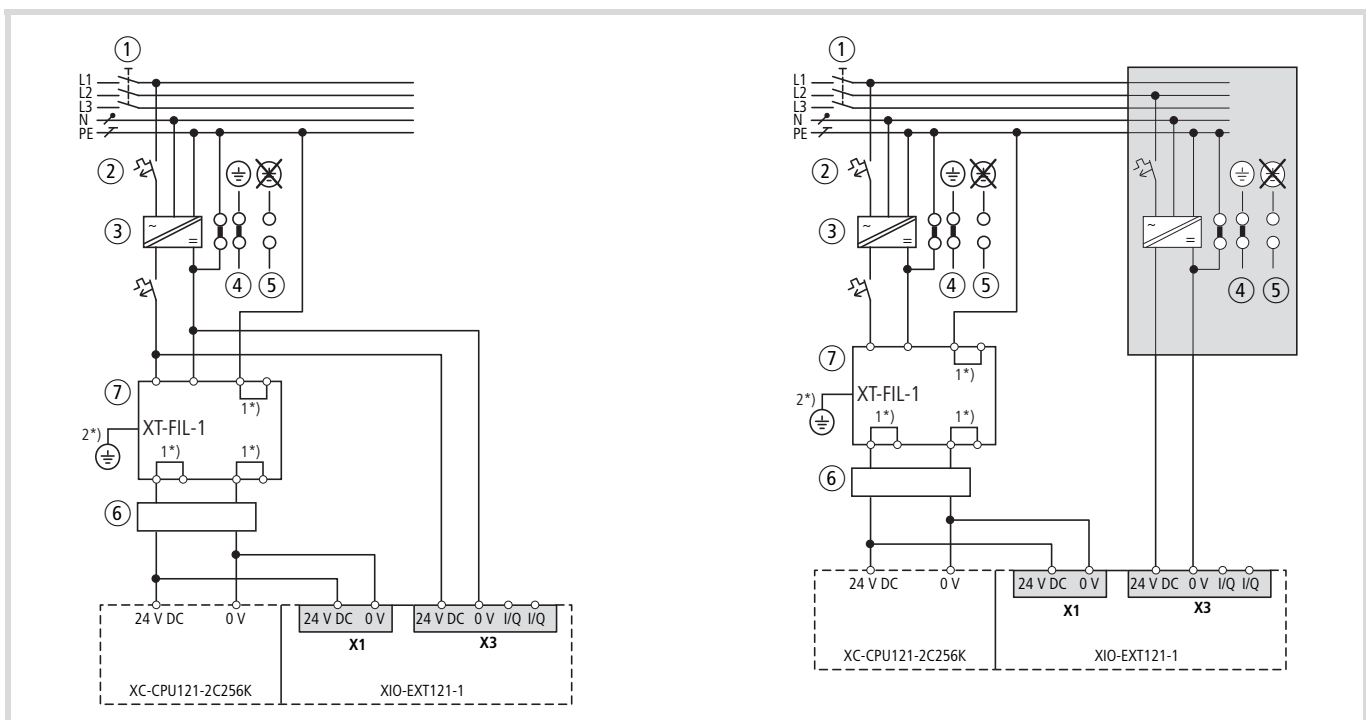


Abbildung 17: Verdrahtungsbeispiel: XC-CPU121 und XIO-EXT121-1 (links: gemeinsame Spannungsversorgung, rechts getrennt)

- 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 Ferrithülse
- ⑦ 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 50
 ➔ „Technische Daten“ auf Seite 55

1*) intern gebrückt

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

→ Beim Einschalten der XC-CPU121, gekoppelt mit der XIO-EXT121:

Jedes Gerät verfügt über einen separaten Anschluss der Versorgungsspannung. Zum Start der CPU müssen Sie an beiden Geräten die Spannung einschalten. Schalten Sie nur ein einzelnes Gerät ein, führt die CPU kein Programm aus und die LEDs bleiben dunkel.

Spannungsversorgung der XC-CPU121 anschließen

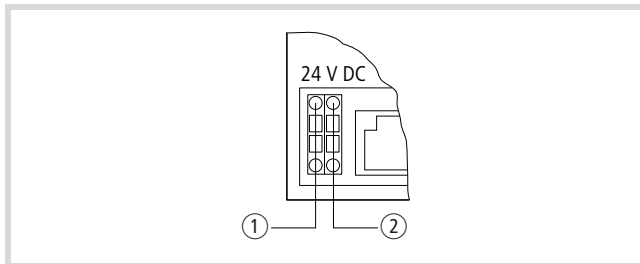


Abbildung 18: Spannungsanschluss der XC-CPU121

- ① 24 V DC
- ② 0 V

Zur Spannungsversorgung der XIO-ETX121-1 siehe Abbildung 21, Stecker X1

Sensoren und Aktoren anschließen

Die digitalen Sensoren/Aktoren schließen Sie an die Steckverbinder X2 und X3 an:

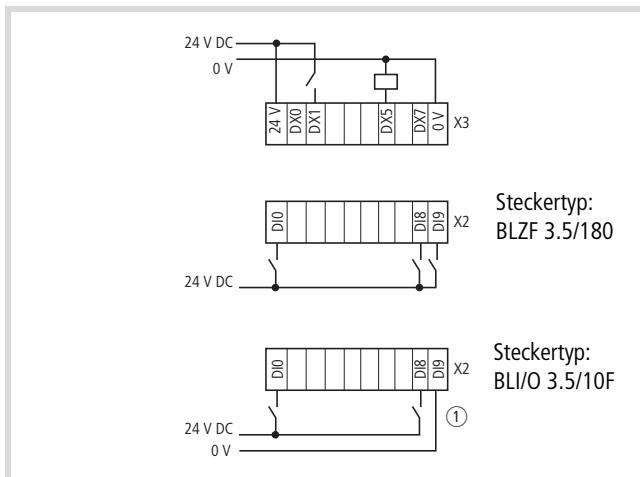


Abbildung 19: Ein-/Ausgänge an X2/X3 anschließen

- ① Wenn Sie den Steckertyp mit LED-Anzeige BLI/O3.5/10F am Stecker X2 einsetzen, müssen Sie die Klemme „DI9“ mit 0 V verbinden.

An den Klemmen DX0 bis DX7 des Steckers X3 können Sie Sensoren und/oder Aktoren anschließen. Jede Klemme (Anschluss) können Sie vom Anwenderprogramm aus setzen (QX0.0 bis QX0.7) oder abfragen (IX0.0 bis IX0.7).

Die Eingänge des Steckers X2 fragen Sie mit IX1.0 (DI0) bis IX1.7 (DI7) und IX2.0 (DI8), IX2.1 (DI9) ab).

Es stehen zwei Steckertypen zur Verfügung:

- ohne LED
- mit LED

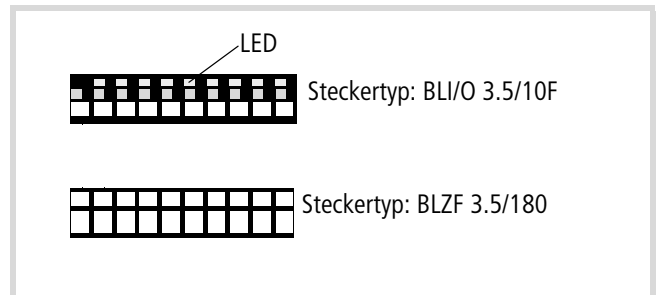


Abbildung 20: Steckertypen

Sie können beide Steckertypen für die Steckerleiste X2 und/oder X3 verwenden.

Die analogen Sensoren/Aktoren schließen Sie an den Steckverbinder X1 an:

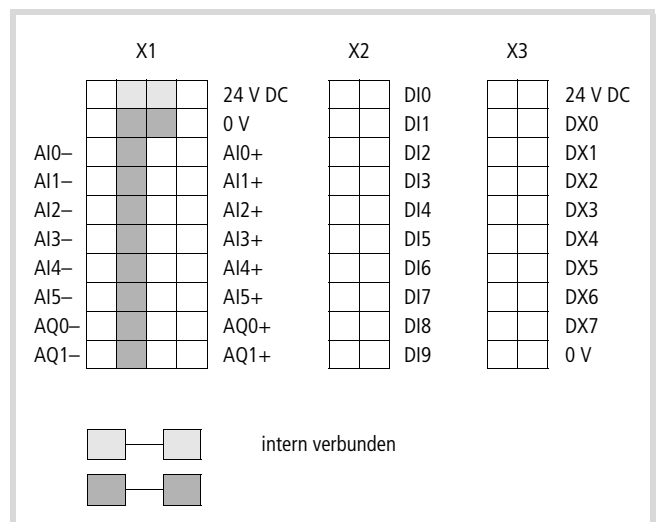


Abbildung 21: Bezeichnung der Steckverbinder und der Ein-/Ausgänge

Tabelle 5: Merkmale Analog-Ein-/Ausgänge

Analogtyp	Auflösung	Operand	Klemme (X1)
Analog-Eingänge			
0 – 10 V	10 Bit	IW4	AI0 -/+
0 – 10 V	10 Bit	IW6	AI1 -/+
0 – 20 mA	10 Bit	IW8	AI2 -/+
0 – 20 mA	10 Bit	IW10	AI3 -/+
Pt100	10 Bit ¹⁾	IW12	AI4 -/+
Pt100	10 Bit ¹⁾	IW14	AI5 -/+
Analog-Ausgänge			
0 – 10 V	12 Bit	QW2	AQ0 -/+
0 – 10 V	12 Bit	QW4	AQ1 -/+

1) Zugeordnete Temperaturwerte → Abbildung 22

Tabelle 6: Analogwerte

0 – 10 V	0 – 20 mA	dez	hex
Analog-Eingänge			
0	0	0	000
5	10	511	1FF
10	20	1023	3FF
Analog-Ausgänge			
0	–	0	000
5	–	2047	7FF
10	–	4095	FFF

①

②

① Simulierte Temperatur [°C]

② Gemessener Dezimalwert XC 121 [dez]

-200	107
-180	155
-160	203
-140	251
-120	297
-100	344
-80	390
-60	435
-40	480
-20	525
0	569
20	614
40	657
60	701
80	745
100	788
120	831
140	873
160	916
180	958
200	1000

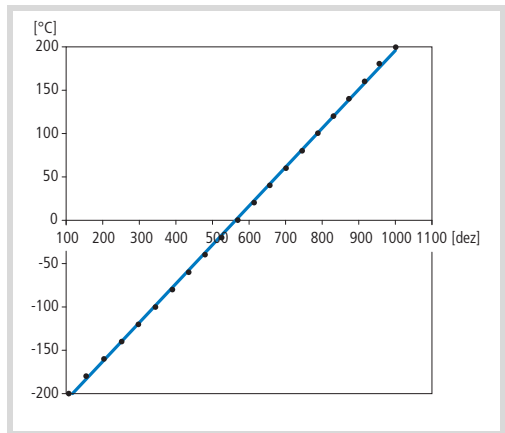


Abbildung 22: Pt100-Eichkurve für XC121
(Simulator: burster Precision
RTD Simulator 4530)

5 Konfiguration der XIO-EXT121-1

Um die XC121 mit dem I/O-Modul XIO-EXT-121-1 zu erweitern, müssen Sie im Steuerungskonfigurator den Ordner „No local I/O“ durch das Element XIO-EXT121-1 ersetzen.

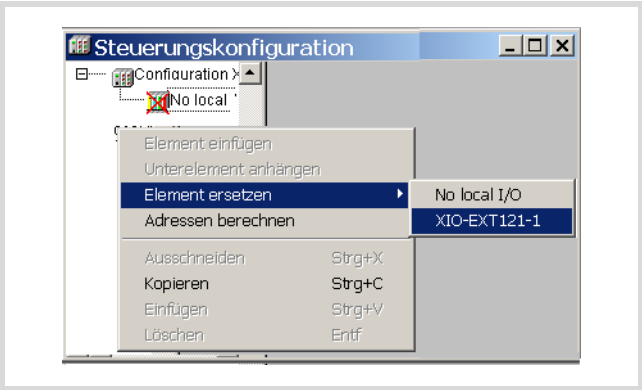


Abbildung 23: XIO-EXT121-1 konfigurieren

Das neue Fenster zeigt Ihnen alle Ein- und Ausgänge des I/O-Moduls.

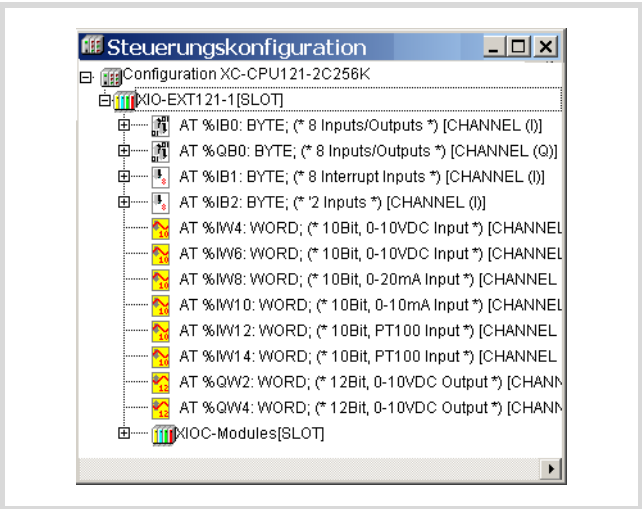


Abbildung 24: Ein-Ausgänge konfigurieren

Aus der folgenden Übersicht geht die Zuordnung der im Konfigurator dargestellten Ein-/Ausgangssyntax zu den Anschlüssen am Stecker hervor.

Tabelle 7: Übersicht der I/O-Signale (analog)

Stecker- typ	Stecker I/Q	Operand I Word	Operand Q Word	Typ I/Q analog
X1	AI0	IW4		I 0 – 10 V DC
	AI1	IW6		I 0 – 10 V DC
	AI2	IW8		I 0 – 20 mA
	AI3	IW10		I 0 – 20 mA
	AI4	IW12		I Pt100
	AI5	IW14		I Pt100
	AQ0		QW2	Q 0 – 10 V DC
	AQ1		QW4	Q 0 – 10 V DC

Tabelle 8: Übersicht der I/O-Signale (digital)

Stecker- typ	Stecker I/Q	Operand I		Operand Q		Typ I/Q digital
		Bit	Byte	Bit	Byte	
X2	DI0	IX1.0	IB1			I
	...	...				
	DI7	IX1.7				
	DI8	IX2.0	IB2			
X3	DI9	IX2.1		QX0.0	QB0	I/Q
	DX0	IX0.0	IB0			
	...	...				
	DX7	IX0.7				

Siehe auch „Bezeichnung der Steckverbinder und der Ein-/Ausgänge“, Abbildung 21 auf Seite 19.

6 Betrieb

Einschaltverhalten

Nach dem Einschalten der Versorgungsspannung führt die CPU einen System-Selbsttest durch. Ist dieser erfolgreich abgeschlossen, startet das Laufzeitsystem. Im Fehlerfall blinkt die rote und grüne LED.

Nach dem Start des Laufzeitsystems prüft die CPU, ob ein Betriebssystem-Update auf einer gesteckten MMC vorhanden ist und ob dieses gegebenenfalls geladen werden muss. Da die Steuerung keine Batterie zum Backup des Arbeits-(SRAM)-Speichers besitzt, wird nach dem Start des Laufzeitsystems auch geprüft, ob ein Bootprojekt verfügbar ist. Ist das der Fall, wird es in den Arbeits-(SRAM)-Speicher der Steuerung geladen und in Abhängigkeit von

der Stellung des Betriebsartenschalters und unter Berücksichtigung des parametrisierten Startverhaltens gestartet. Ist kein Bootprojekt vorhanden, bleibt die Steuerung im Zustand NOT READY.

Einschaltverhalten mit Bootprojekt

Beim Einschalten der Steuerung hat ein auf der MMC vorhandenes Bootprojekt Vorrang vor einem im Systemspeicher (Flash) gespeicherten. Sind die Bootprojekte unterschiedlich, wird das Bootprojekt von der MMC in den Systemspeicher (Flash) kopiert und anschließend zur Ausführung gebracht. Durch den Kopiervorgang verlängert sich die Hochlaufphase der Steuerung einmalig um mehrere Sekunden.

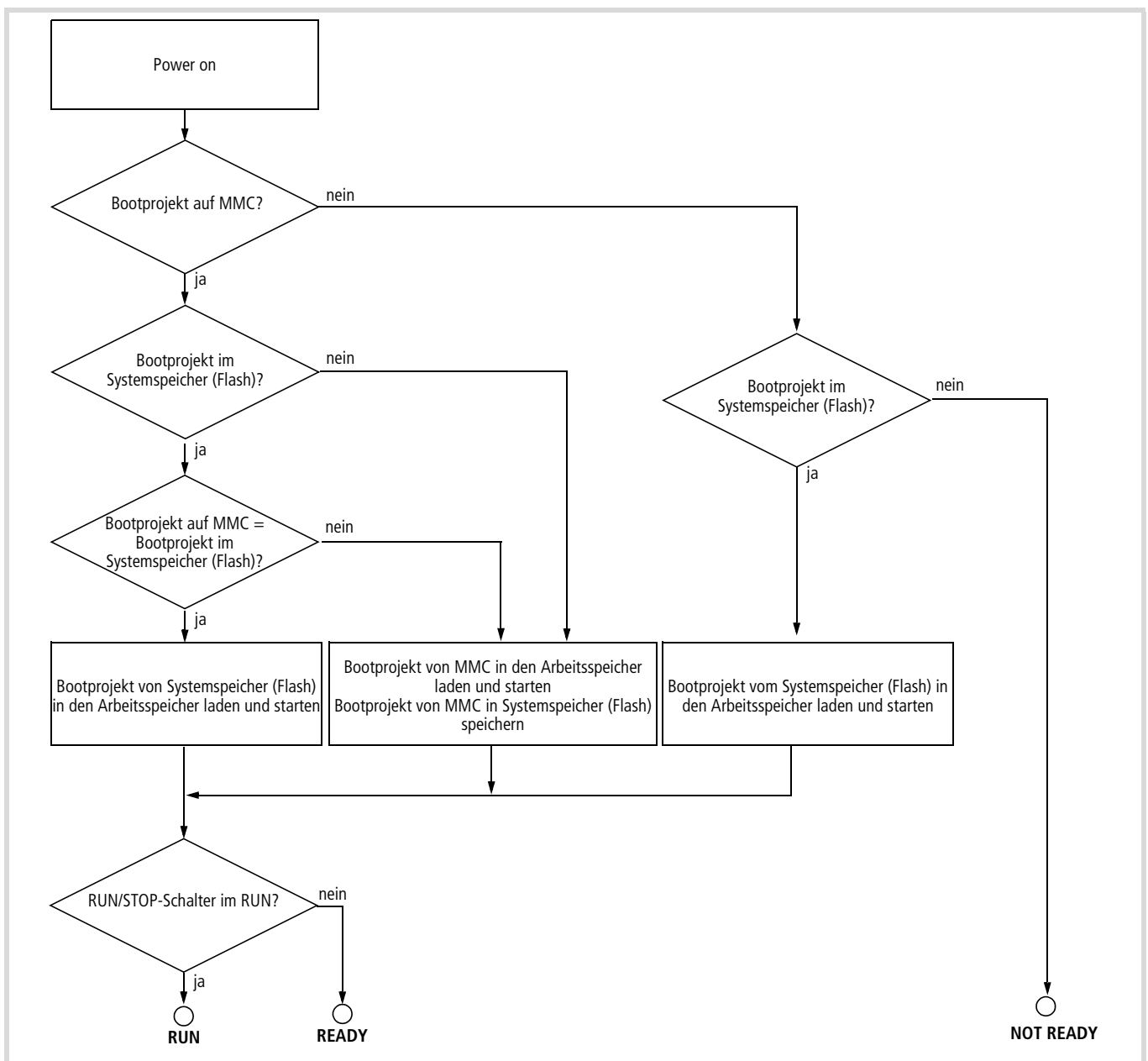


Abbildung 25: Einschaltverhalten mit Bootprojekt

Startverhalten in der CoDeSys einstellen

Mit der Einstellung des Startverhaltens bestimmen Sie das Anlaufverhalten der Steuerung beim Einschalten der Versorgungsspannung.

Die Einstellung nehmen Sie im Steuerungskonfigurator vor. Aktivieren Sie dort das Register „Weitere Parameter“ und wählen Sie aus dem Listenfeld die gewünschte Startbedingung aus.

- HALT
- WARMSTART
- COLDSTART

Zum Verhalten der Variablen je nach eingestellter Startbedingung siehe Tabelle 9.

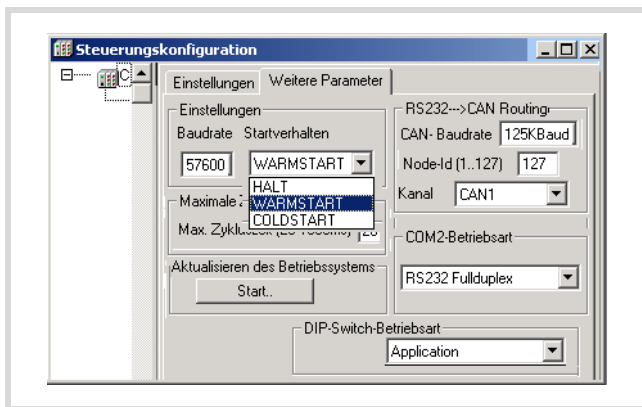


Abbildung 26: Startverhalten festlegen

Programm START/STOP

Programmstart (STOP → RUN)

Sie haben zwei Möglichkeiten, das Programm zu starten:

- Im Online-Betrieb den Befehl START anwählen; z. B. nach dem Laden eines Programms. Dazu muss sich die CPU im Zustand STOP befinden und der Betriebsartenschalter in der Stellung RUN stehen.
- Den Betriebsartenschalter in die Stellung RUN setzen.

→ Die Stellung des Betriebsartenschalters hat Vorrang gegenüber dem Online-Befehl.

Tabelle 9: Verhalten der Variablen beim Start

Startbedingungen	Art der Variablen	
	Nicht remanent	Remanent (Retain)
COLDSTART	Aktivieren der Initialwerte	
WARMSTART	Aktivieren der Initialwerte	Werte bleiben erhalten
Programm im Online-Betrieb laden und starten	Aktivieren der Initialwerte	
Start/Stop/Start...	Werte bleiben erhalten	

Verhalten nach dem Ausschalten/Unterbrechen der Spannungsversorgung

Wenn Sie die (CPU-)Spannung ausschalten oder unterbrechen, wird der Programmzyklus bzw. die Task sofort unterbrochen. Die remanenten Daten sind nicht mehr konsistent! Alle Ausgänge werden auf 0 gesetzt bzw. abgeschaltet.

Das Verhalten remanenter Variablen zeigt Tabelle 9.

Wenn Sie die Spannung wieder einschalten, wird das restliche Programm des Zyklus nicht beendet!

Sind inkonsistente Daten für eine Anwendung nicht praktikabel, können Sie z. B. eine unterbrechungsfreie Stromversorgung mit Akku-Pufferung einsetzen.

Der Start der Steuerung erfolgt gemäß den Einstellungen im Steuerungskonfigurator, → Abbildung 26.

Programmstopp (RUN → STOP)

Wenn Sie den Betriebsartenschalter auf STOP stellen, wechselt die CPU in den Zustand STOP, sobald der Programmzyklus beendet ist. Die Ausgänge werden auf 0 gesetzt.

Sie haben zwei Möglichkeiten, das Programm zu stoppen:

- Im Online-Betrieb den Befehl STOP wählen.
- Den Betriebsartenschalter in die Stellung STOP setzen.

→ Die Stellung des Betriebsartenschalters hat Vorrang gegenüber der Online-Einstellung.

Programmbearbeitung und Systemzeit

Das Anwenderprogramm 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.

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

Zusätzlich gibt es die Möglichkeit, Applikations-Routinen zu programmieren, die beim Auftreten von Systemereignissen gestartet werden → Abschnitt „Systemereignisse“ auf Seite 27.

Zykluszeit überwachen

Die Zyklen des Anwenderprogramms und der einzelnen Ereignis-Tasks überwacht ein Hardware-Timer. Überschreitet die Zykluszeit die von Ihnen parametrisierte Zeit, wird die Steuerung in den Zustand STOP gesetzt und die Ausgänge abgeschaltet.

Die Zeit stellen Sie im Steuerungskonfigurator unter „Weitere Parameter“ ein. Der kleinste Zeitwert beträgt 20 ms (Defaultwert), der größte 1000 ms.

Reset

Es gibt vier verschiedene Reset-Befehle:

- Reset (warm)
- Reset kalt
- Reset Ursprung
- Reset zum Wiederherstellen der Werkseinstellungen

Reset (warm)

- Das Programm wird angehalten
- Die nicht remanenten Variablen werden initialisiert, die „Retain“-Variablen bleiben erhalten.
- Das Programm kann wieder gestartet werden.

Reset kalt

- Das Programm wird angehalten
- Alle Variablen werden initialisiert
- Das Programm kann wieder gestartet werden.

Reset Ursprung

- Das Programm in der Steuerung sowie das Bootprojekt werden gelöscht.
- Bei gesteckter MMC:
 - Alle projektspezifischen Dateien und das Bootprojekt werden gelöscht
 - Alle benutzerspezifischen Dateien sowie die Startup.ini-Datei bleiben unverändert
- Die Steuerung wird in den Zustand NOT READY gesetzt.

Reset zum Wiederherstellen der Werkseinstellungen

Voraussetzung für diesen Reset ist, dass der Betriebsartenschalter in Stellung 8 ist. Wenn Sie dann auf den SET-Taster drücken, werden alle Schnittstellen mit ihren Defaultparametern initialisiert. Ein geladenes Anwenderprogramm, alle Variablen und das Bootprojekt werden im Systemspeicher (Flash) und auf der MMC gelöscht.

Verhalten der Variablen nach Reset

Reset	Art der Variablen	
	Nicht remanent	Retain
Reset warm	Aktivieren der Initialwerte	Werte bleiben erhalten
Reset kalt	Aktivieren der Initialwerte	
Reset Ursprung ¹⁾	Aktivieren der Initialwerte	

1) Nach Reset-Ursprung muss das Programm neu geladen werden. Im Online-Betrieb können Sie danach die Steuerung starten.

Test und Inbetriebnahme

Die Steuerung unterstützt folgende Test- und Inbetriebnahme-Möglichkeiten:

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

Breakpoint/Einzelschritt-Betrieb

Innerhalb des Anwenderprogramms können Breakpoints gesetzt werden. Bei der Ausführung einer mit einem Breakpoint versehenen Anweisung wird das Programm vor Ausführung der Programmzeile angehalten. Die folgenden Programmanweisungen können im Einzelschritt-Betrieb schrittweise ausgeführt werden. Die Zykluszeitüberwachung ist deaktiviert.



Warnung!

Die zu diesem Zeitpunkt gesetzten Ausgänge bleiben gesetzt!



Verwenden Sie „Breakpoint-/Einzelschritt-Betrieb“ und „Einzelzyklusbetrieb“ nur im eigentlichen Hauptprogramm der Applikation. In den Ereignis-Routinen, z. B. für Start-, Stop- und Interrupt-Events, dürfen sie **nicht** verwendet werden; anderenfalls kann dies zu Störungen im Steuerungsablauf führen.

Die Software CoDeSys verhindert nicht, dass Breakpoints in den Ereignis-Routinen gesetzt werden können.

Einzelzyklus-Betrieb

Im Einzelzyklus-Betrieb wird ein einzelner Programmzyklus in Echtzeit ausgeführt. Die Ausgänge sind während des Zyklus freigegeben. Die Zykluszeit-Überwachung ist aktiv.



Warnung!

Die zu diesem Zeitpunkt gesetzten Ausgänge bleiben gesetzt!

Zwangssetzen von Variablen und I/Os (Forcen)

Alle Variablen eines Anwenderprogramms können mittels „Forcen“ zwangsweise auf feste Werte gesetzt werden. Zwangsgesetzte lokale Ausgänge der XI/ON-Module werden nur im Zustand RUN zur Peripherie durchgeschaltet.



Ein Zwangssetzen des über den CANopen-Feldbus angeschlossenen I/Os ist nicht möglich.

Zustandsanzeige in der Software CoDeSys

- Der Signalzustand der physikalischen, booleschen Eingänge wird sowohl im Zustand RUN der CPU als auch im STOP angezeigt.
- Der Signalzustand der physikalischen, booleschen Ausgänge wird nur im RUN-Zustand angezeigt; im STOP-Zustand sind sie mit FALSE gekennzeichnet.
- Alle anderen Variablen werden mit dem aktuellen Variablenwert angezeigt.

Systemereignisse

Auf Systemereignisse der Steuerung können Sie reagieren, indem Sie eine Applikations-Routine (POU) erstellen, die bei Auftreten des Ereignisses einmal ausgeführt wird. Die Ausführung wird zeitlich überwacht. Als Zeitbasis dient der für die maximal zulässige Zykluszeit parametrisierte Wert.

Ereignisse sind:

STOP	Stopp des Anwenderprogramms (gilt nicht bei Zykluszeitüberschreitung oder Hardware-Watchdog)
START	Start des Anwenderprogramms (Kalt- und Warmstart)
COLDSTART	Kaltstart des Anwenderprogramms
WARMSTART	Warmstart des Anwenderprogramms

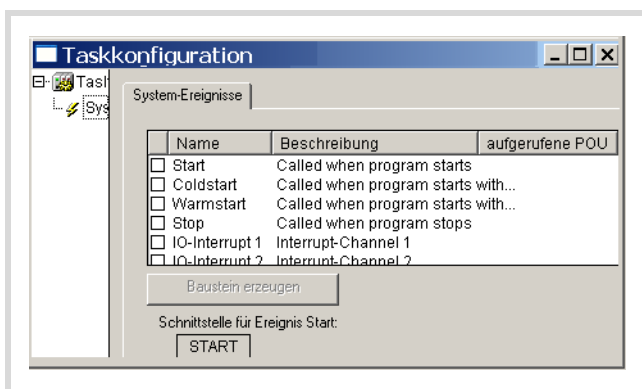


Abbildung 27: Systemereignisse

→ Der Einzelschrittbetrieb ist bei den Programmbausteinen der Systemereignisse nicht möglich.

Interruptverarbeitung

Bei Auftreten eines Interrupts führt das Betriebssystem die mit der Interruptquelle verknüpfte POU aus.



Warnung!

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

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



Achtung!

Alle bis zu diesem Zeitpunkt angesteuerten Ausgänge bleiben aktiv und werden nicht abgeschaltet.

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

Treten Interrupts während eines Programmdurchlaufs zu häufig auf, kann die programmierte Taskzeit überschritten werden; in diesem Fall löst der Watchdog einen RESET aus.

Sie können Interrupts aus dem Programm heraus sperren und wieder freigegeben. Hierzu gibt es die Funktionen „DisableInterrupt“ und „EnableInterrupt“. Ein Aufrufparameter in der Software CoDeSys bestimmt, ob ein einzelner Interrupt oder alle Interrupts gesperrt/freigegeben werden. Die Freigabe eines gesperrten Interrupts muss mit dem gleichen Parameter erfolgen, wie die Sperrung.

Die beiden Funktionen „DisableInterrupt“ und „EnableInterrupt“ sind Bestandteil der Library „XC121_Util.lib“. Diese Library müssen Sie – falls noch nicht vorhanden – in den Bibliotheksverwalter der Software CoDeSys einbinden.

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

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

Parametrierung der Eingänge

Die Eingänge I 1.0 (DI0) bis I 1.7 (DI7) am Steckverbinder X2 können Sie als Standard- oder als Interrupt-Eingänge benutzen. Die Eingänge I 2.0 (DI8) und I 2.1 (DI9) sind Standard-Eingänge. Die einzelnen Interrupt-Eingänge erhalten ihre Interrupt-Funktion erst, nachdem Sie den Eingängen den Typ der Flanke zugewiesen haben, die den Interrupt auslöst.

Die Zuweisung erfolgt im Steuerungskonfigurator im Register „Weitere Parameter“ unter „Lokale Interrupts“.

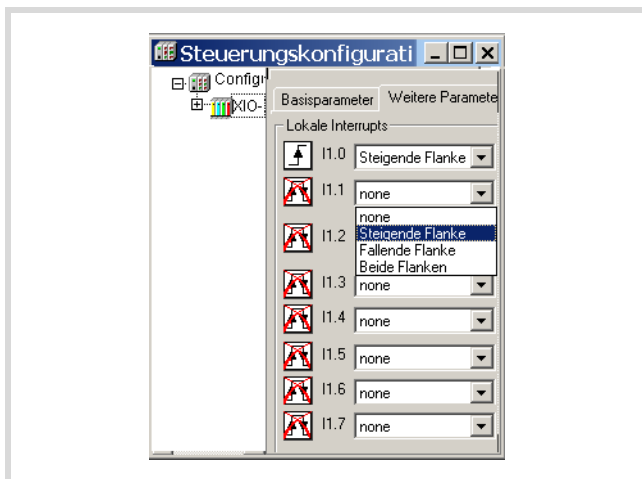


Abbildung 28: Flankentyp der lokalen Interrupts festlegen

Die Verbindung zwischen IO-Interrupt und POU treffen Sie im Taskkonfigurator.

Die Interrupt-Kanäle 1 bis 8 (Channels) sind den Eingängen I1.0 bis I1.7 direkt zugeordnet. Die Priorität der Eingänge ist fest vorgegeben: Channel 1 (Eingang I1.0) hat die höchste Priorität, Channel 8 (Eingang I1.7) hat die niedrigste Priorität (Siehe Beispiel).

Beispiel zur Interruptverarbeitung

Eine Task „Basic“ enthält eine POU „PLC_PRG“. Eine weitere POU „Fastprog“ soll abgearbeitet werden, wenn eine L → H-steigende Flanke am Eingang I1.2 einen Interrupt erzeugt.

- Erstellen Sie die POUs „PLC_PRG“ und „Fastprog“ gemäß Abbildung 29.

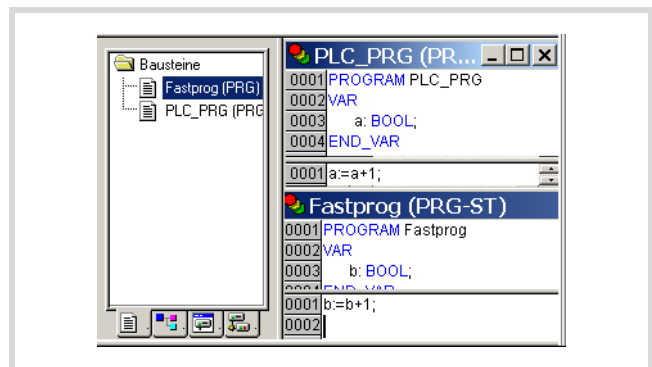


Abbildung 29: Programm erstellen

- Wechseln Sie in die Steuerungskonfiguration und weisen Sie dem Eingang I1.2 den Typ „steigende Flanke“ zu.



Abbildung 30: Interrupt-Flanke auswählen

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

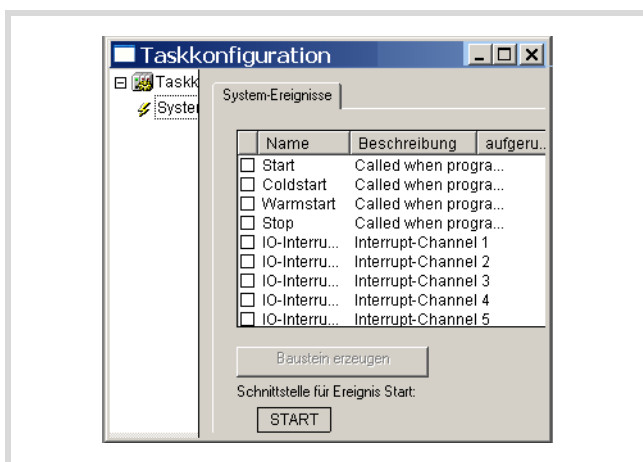


Abbildung 31: Interrupt freigeben

- Geben Sie den IO-Interrupt 3 frei, indem Sie auf das Kästchen links neben dem Namen „IO-Interrupt3“ klicken. Die Bestätigung wird durch einen Haken dargestellt.
- Markieren Sie den Bereich von Spalte „aufgerufene POU“ und Zeile „IO-Interrupt3“.
- Stellen Sie den Cursor auf den markierten Bereich und betätigen Sie die Funktionstaste F2.

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

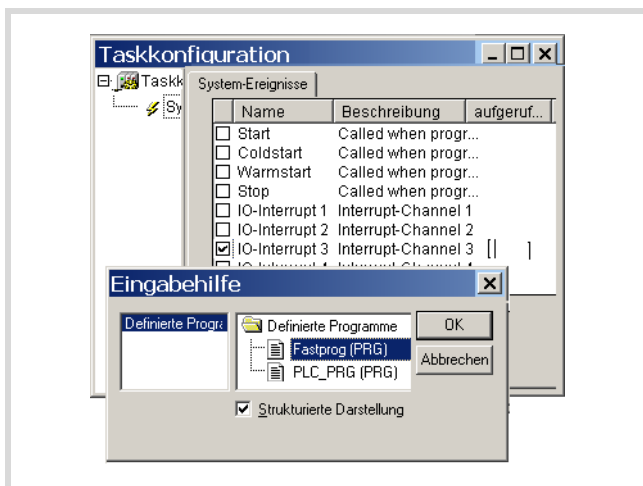


Abbildung 32: Zuweisung Interruptquelle → POU

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

Mit jeder steigenden Flanke am Eingang I1.2 wird die Variable „b“ um eins erhöht.

Timer-Interrupt

Mit dem Timer-Interrupt lösen Sie einen Interrupt aus, der periodisch aktiv wird. Die Periodendauer kann von 500 – 2500000 Mikrosekunden eingestellt werden. In Abhängigkeit einer booleschen Variablen startet der Timer. Nach Ablauf der eingestellten Zeit wird das Anwenderprogramm unterbrochen und arbeitet die von Ihnen festgelegte Applikations-Routine ab.

Zur Programmierung der Funktion „TimerInterruptEnable“ müssen Sie die Bibliothek XC121_Util.lib in Ihr Anwenderprogramm mit einbinden.

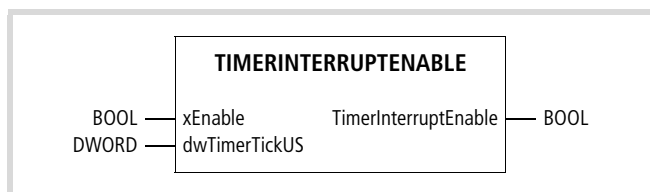


Abbildung 33: Funktion TimerInterruptEnable

Am Eingang „dwTimerTickUS“ geben Sie die Verzögerungszeit an.

Der Wert wird beim Start des Timers übernommen und kann während des Ablaufs nicht geändert werden. Wird der Wert von 500 unter- bzw. der Wert von 2500000 überschritten, liefert die Funktion als Rückgabewert FALSE und der Timer wird nicht gestartet.

Applikations-Routine „Zeit_Int“ erstellen:

- Öffnen Sie im Verzeichnis „Ressourcen“ das Unterverzeichnis „Task-Konfiguration“ mit einem Doppelklick.
- Klicken Sie hier den Ordner „Systemereignisse“ an. Das Register „System-Ereignisse“ ist aktiv:

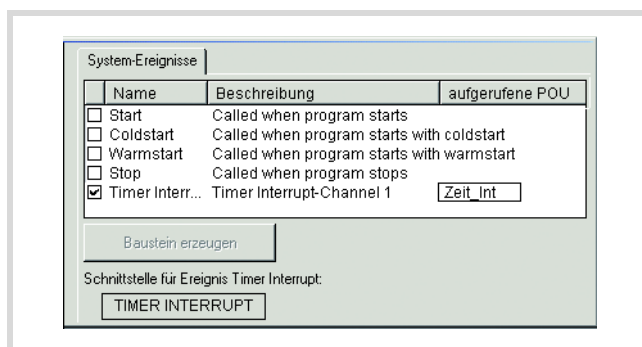


Abbildung 34: Systemereignis „Timer-Interrupt“ wählen

- Aktivieren Sie den Timer-Interrupt, indem Sie auf das Kästchen links neben dem Namen „Timer-Interrupt“ klicken.
- Tragen Sie im Feld „aufgerufene POU“ den Namen „Zeit_Int“ der Applikations-Routine ein.
- Klicken Sie nochmals auf den Namen „Timer-Interrupt“. Jetzt wird die Schaltfläche „Baustein erzeugen“ aktiv und zeigt den Namen der POU an.
- Klicken Sie auf diese Schaltfläche. Im Fenster „Bausteine“ wird ein Ordner (POU) mit dem Namen eingefügt.
- Öffnen Sie die POU und schreiben Sie Ihr Applikationsprogramm:

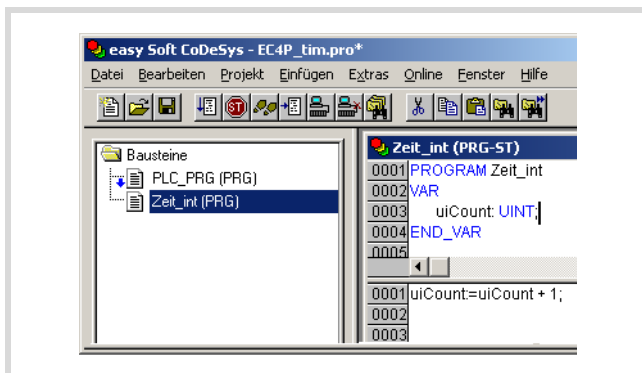


Abbildung 35: Applikations-Routine schreiben

Der Interrupt kann durch höher priorisierte Systeminterrupts unterbrochen werden. Während der Ausführung des Timer-Interrupts ist die Zykluszeitüberwachung aktiv.

Der Timer-Interrupt kann aus dem Anwenderprogramm heraus gesperrt und freigegeben werden. Hierzu stehen die Funktionen „DisableInterrupt“ und „EnableInterrupt“ aus der Bibliothek XC121_UTIL.lib zur Verfügung.

Direkter I/O-Zugriff

Über die Funktion „Direkter IO-Zugriff“ greift die CPU direkt auf die lokalen Ein- und Ausgänge des XIO-EXT121-1-Moduls zu. Der Zugriff erfolgt nicht über das Ein-/Ausgangsabbild.

→ Der direkte Zugriff auf die Daten der XI/OC-Module wird nicht unterstützt.

Für den „direkten Zugriff“ auf die aktuellen IO-Zustände und Daten verwenden Sie die Funktionen wie z. B. „ReadBitDirect“ der Library „XC121_Util.lib“.

Die Funktion „ReadBitDirect“ wird beispielhaft für alle weiteren Funktionen beschrieben:

ReadBitDirect

Mit dieser Funktion können Sie das Bit eines Eingangs direkt lesen. Der Zustand des Eingangsbits wird in der Variablen abgelegt, auf die der parametrisierte Pointer „ptr_xValue“ zeigt. Die Pointervariable wird nicht verändert, wenn bei der Bearbeitung ein Fehler auftritt.

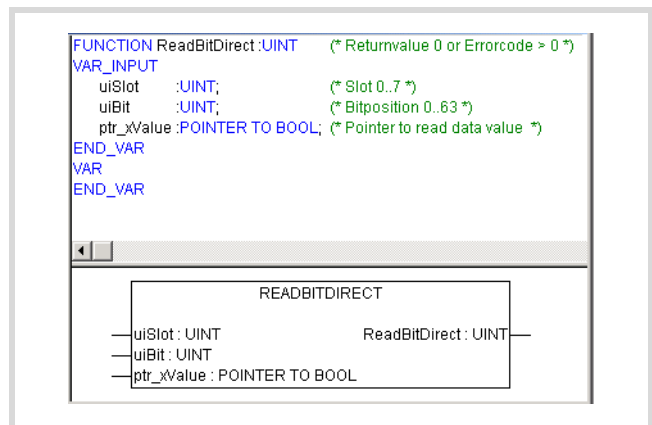


Abbildung 36: Funktion READBITDIRECT

Bit-Zugriff

Funktion: ReadBitDirect(uiSlot, uiBit, ptr_xValue)

IX0.0 – IX0.7: uiSlot = 0, uiBit 0 – 7

IX1.0 – IX1.7: uiSlot = 0, uiBit 8 – 15

IX2.0 – IX2.7: uiSlot = 0, uiBit 16 – 23

Funktion: WriteBitDirect(uiSlot, uiBit, xValue)

QX0.0 – QX0.7: uiSlot = 0, uiBit 0 – 7

Weitere Funktionen sind:

Byte-Zugriff

Funktion: ReadByteDirect(uiSlot, uiByte, ptr_byValue)

IB0: uiSlot = 0, uiByte 0

IB1: uiSlot = 0, uiByte 1

IB2: uiSlot = 0, uiByte 2

Funktion: WriteByteDirect(uiSlot, uiByte, byValue)

QB0: uiSlot = 0, uiByte 0

WORD-Zugriff

Funktion: ReadWordDirect(uiSlot, uiOffset, ptr_wValue)

IW4: uiSlot = 0, uiOffset 2

IW6: uiSlot = 0, uiOffset 3

IW8: uiSlot = 0, uiOffset 4

IW10: uiSlot = 0, uiOffset 5

IW12: uiSlot = 0, uiOffset 6

IW14: uiSlot = 0, uiOffset 7

Funktion: WriteWordDirect(uiSlot, uiOffset, wValue)

QW2: uiSlot = 0, uiOffset 1

QW4: uiSlot = 0, uiOffset 2

Fehlercode bei „Direkter Peripheriezugriff“

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

Folgende Rückgabewerte sind möglich:

Tabelle 10: Fehlercodes beim direkten Peripherie-Zugriff

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

Bootprojekt erzeugen und transferieren

Die CPU arbeitet das im Arbeitsspeicher gespeicherte Anwenderprogramm ab. Da der Arbeitsspeicher nicht gepuffert ist, wird das Programm bei Spannungsausfall gelöscht. Um das Programm remanent zu sichern, müssen Sie ein Bootprojekt erstellen. Dazu sind folgende Schritte nötig:

- Wechseln Sie in das Menü „Online“ und wählen Sie den Befehl „Einloggen“.
- Wählen Sie den Befehl „Bootprojekt erzeugen“.

Es erscheint folgende Abfrage:

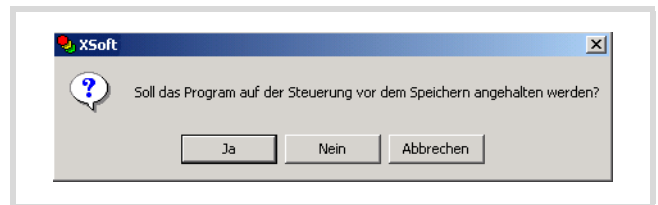


Abbildung 37: Bootprojekt erzeugen

- Beantworten Sie diese Frage mit „Ja“.

Es erscheint für eine kurze Zeit das folgende Fenster:

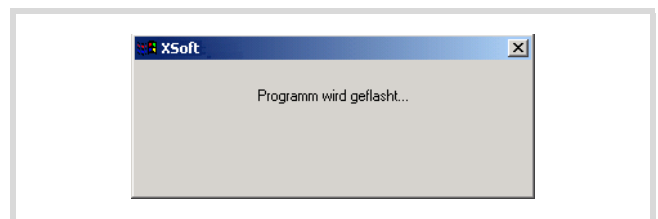


Abbildung 38: Bootprojekt erstellen

Mit dem automatischen Ausblenden des Fensters ist das Bootprojekt erstellt. Sie können nun die Steuerung wieder starten.

Siehe Abschnitt „Einschaltverhalten mit Bootprojekt“ auf Seite 23.

Bootprojekt auf MMC abspeichern

Das im Systemspeicher (Flash) abgelegte Bootprojekt kann zusätzlich auf der MMC abgespeichert werden. Das geschieht durch den Aufruf des Browserkommandos „copyprojtoMMC“.

Bootprojekt löschen

Der Browser-Befehl „Remove“ löscht sowohl das im Systemspeicher (Flash) gespeicherte Bootprojekt als auch ein auf der MMC vorhandenes. Mit dem Browser-Befehl „removeprojfrommmc“ wird nur das Bootprojekt auf der MMC gelöscht.

Betriebssystem herunterladen/aktualisieren

Bei der XC121 haben Sie die Möglichkeit, das gespeicherte Betriebssystem (BTS) durch ein aktuelles zu ersetzen. Eaton bietet die jeweils aktuelle BTS-Version im Internet als Download an (<http://www.eaton.com/moeller/support>). Zusätzlich ist das aktuelle BTS auch auf der jeweils neuesten „CoDeSys-CD“ enthalten.



Achtung!

Beim Download des Betriebssystems werden alle auf der Steuerung befindlichen Dateien gelöscht. Das gilt sowohl für das bestehende Bootprojekt als auch für das Anwenderprogramm.

Sie haben zwei Möglichkeiten, das Betriebssystem zu übertragen:

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

Betriebssystem vom PC in die Steuerung übertragen

- Öffnen Sie ein CoDeSys-Projekt und aktivieren Sie unter «Ressourcen → Steuerungskonfigurator» das Register „Weitere Parameter“ → Abbildung 5.
- Klicken Sie auf die Schaltfläche „Start“.

Das Fenster „Betriebssystem übertragen“ öffnet sich.

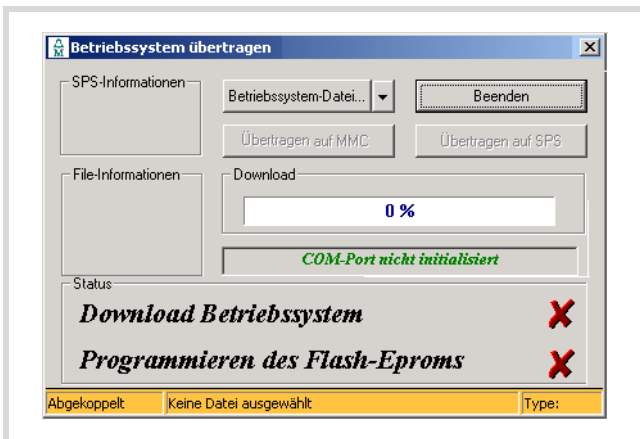


Abbildung 39: Betriebssystem übertragen

Das System meldet, dass der COM-Port nicht initialisiert ist.

- Klicken Sie auf die Schaltfläche „Betriebssystem-Datei“ und wählen Sie die gewünschte Betriebssystem-Datei (*.hex) aus.



Zuletzt geöffnete Dateien können Sie aus dem Listenfeld (Drop-Down-Menü) auswählen.

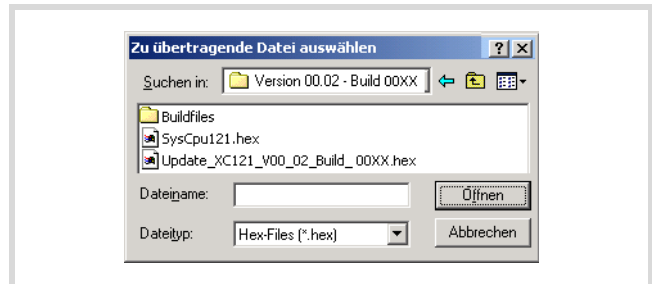


Abbildung 40: Betriebssystem-Datei auswählen

Nach der Auswahl der BTS-Datei erhalten Sie eine Information über den Zieltyp und die Fileversion.

- Betätigen Sie die Schaltfläche „Übertragen auf SPS“.

Die Übertragung beginnt. Das Programmieren des Flash-EPROMs dauert ca. 20 bis 30 Sekunden.



Während der Übertragung oder wenn ein Warnzeichen im Feld „Programmieren des Flash-Eproms“ erscheint, darf die Spannung nicht abgeschaltet werden!

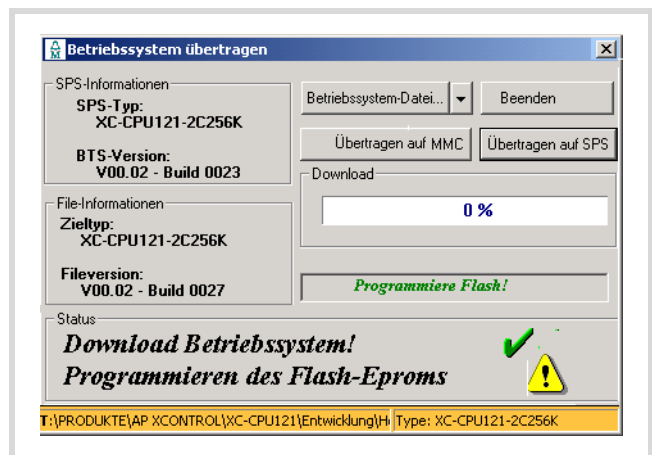


Abbildung 41: Warnung beim Download

Warten Sie die folgende Anzeige ab.

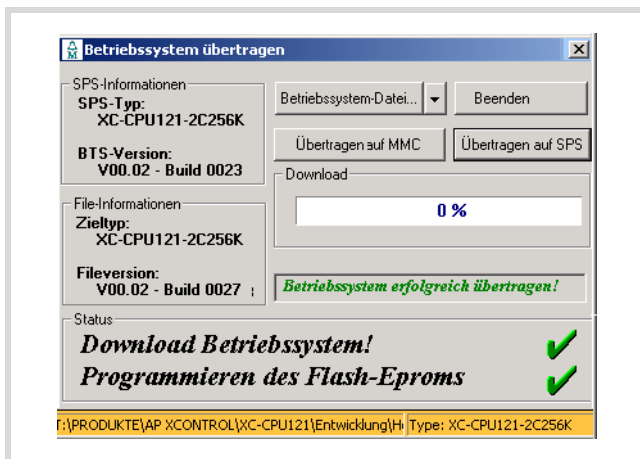


Abbildung 42: Betriebssystem auf SPS übertragen

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

BTS vom PC auf die MMC übertragen

Wenn Sie ein BTS auf die MMC laden, 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“. Klicken Sie in diesem Fall auf die Schaltfläche „Übertragen auf MMC“, → Abbildung 39 auf Seite 32.

BTS von der MMC in die Steuerung übertragen

- Stecken Sie die MMC im ausgeschalteten Zustand in die Steuerung.
- Schalten Sie die Steuerung ein.

Beim Einschalten wird das BTS der Steuerung aktualisiert und ein Bootprojekt in die Steuerung geladen. Die Übertragung kann über 30 Sekunden dauern, da die CPU mehrfach gebootet wird.

- Unterbrechen Sie den Vorgang nicht, indem Sie z. B. die Versorgungsspannung ausschalten.

Quellcode des Anwenderprogramms

Es ist möglich, den Quellcode (Source-Code) des Anwenderprogramms auf der MMC abzulegen.

7 Browser-Befehle

Der PLC-Browser ist ein textbasierter Steuerungsmonitor. Hier geben Sie Kommandos zur Abfrage bestimmter Informationen aus der Steuerung in einer Eingabezeile ein und schicken Sie als String in die Steuerung. Den Antwortstring zeigt ein Ergebnisfenster des Browsers an. Diese Funktionalität können Sie zur Diagnose und zum Debugging benutzen.

→ Die Browser-Befehle sind nur online anwendbar.

Um diese Befehle aufzurufen, sind folgende Schritte notwendig:

- Wählen Sie in der Software CoDeSys im Verzeichnis „Ressourcen“ mit einem Doppelklick den „PLC-Browser“ aus.

Es erscheint im Feld rechts ein neues Fenster mit der Bezeichnung „PLC-Browser“.

- Klicken Sie die Schaltfläche  an.

Im Auswahlfeld werden die zur Verfügung stehenden Browser-Befehle angezeigt.

- Wählen Sie den gewünschten Befehl mit einem Doppelklick aus.

Jetzt erscheint im Fenster „PLC-Browser“ der gewählte Befehl.

- Betätigen Sie die Eingabe-Taste, um die Antwort der Steuerung auf den Browser-Befehl im Ergebnisfenster zu sehen.

→ Weiterführende Informationen zu dem ausgewählten Browser-Befehl erhalten Sie, wenn Sie vor den ausgewählten Browser-Befehl ein „?“ mit einem anschließenden Leerzeichen setzen und die Eingabetaste (RETURN-Taste) drücken.

Die Beschreibung der Befehle finden Sie auch im Handbuch zur Programmiersoftware (MN05010003Z-DE) im Kapitel «Die Ressourcen → PLC-Browser».

Die XC-CPU121 unterstützt die Browser-Befehle aus Tabelle 11.

Tabelle 11: Browser-Befehle

?	Liste der implementierten Kommandos holen
reflect*	Aktuelle Kommandozeile spiegeln, zu Testzwecken
mem	memory-dump, Syntax: mem <start-addr> <end-addr>
memc	wie mem, Adressen + Startadresse des Code-Bereichs
memd	wie mem, Adressen + Startadresse des Datenbereichs
pinf	Projektinformationen ausgeben
ppt	Bausteinzeiger-Tabelle ausgeben
dpt	Datenzeiger-Tabelle ausgeben
pid	Projekt-ID ausgeben
cycle	Zykluszeit ausgeben
canload*	Belastung der lokalen CAN-Busse (CAN1 + CAN2) anzeigen
copyprojtommc	Kopieren des aktuellen Bootprojekts auf die MMC
createstartupini	Erzeugen der Initialisierungsdatei auf der MMC
format	Formatieren der MMC-Speicherkarte
GetNodeId	Anzeigen der CANopen Node-Ids der beiden CAN-Schnittstellen
GetRoutingId	Anzeigen der Routing Node-Id und der Routing-Schnittstelle
metrics	SPS-Informationen ausgeben
reload	Bootprojekt vom FLASH in die SPS laden
remove	Bootprojekt im FLASH löschen
removeprojfrommmc	Löschen des Bootprojekts von der MMC
removestartupini	Löschen der Initialisierungsdatei von der MMC
getswitchpos	Schalterstellung ausgeben
getrtc	Echtzeituhr auslesen
setrtc*	Echtzeituhr setzen

Zu den mit * markierten Befehlen finden Sie gleich anschließend noch weitere Informationen.

reflect

Reflektiert die Kommandozeile, um die Kommunikation vom Browser zur Steuerung und zurück zu testen.

Dieser Befehl wird nicht zur Steuerung übertragen!

Beispiel:

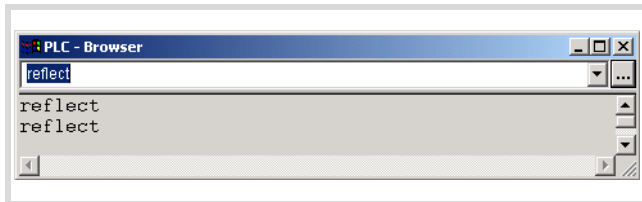


Abbildung 43: Browser-Befehl „reflect“

canload

Zeigt die Auslastung des CANopen-Feldbusses an.

Beispiel:

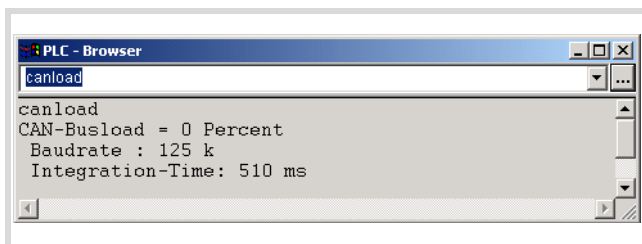


Abbildung 44: Browser-Befehl „canload“

Nach dem Aufruf des Browser-Befehls erhalten Sie z. B. folgende Information:

- CAN-Busload = 0 Percent
- Baudrate 125 kBaud
- Integration-Time: 510 ms



Achtung!

Bei einer Busbelastung von 75 Prozent oder mehr erscheint zusätzlich der Warnhinweis ATTENTION: HIGH BUSLOAD. Die Überlastung des lokalen CAN-Busses im Zusammenhang mit weiteren kurzzeitigen Lastspitzen kann zu CAN-Datenverlust führen.



Zusätzlich zum Browser-Befehl steht die Funktion CAN_BUSLOAD zur Ermittlung der CAN-Busauslastung aus dem Anwenderprogramm heraus zur Verfügung, siehe Abschnitt „Funktion CAN_BUSLOAD“ auf Seite 38.

setrtc

Stellt oder verändert das Datum und/oder die Uhrzeit in der Steuerung.

Syntax:

```
<setrtc_YY:MM:DD:DW_HH:MM:SS>
```

Legende:

- _ Leerzeichen
- YY die beiden letzten Ziffern der Jahreszahl (00 F YY F 99)
- MM Monat (01 F YY F 12)
- DD Tag (01 F DD F 31)
- DW Wochentag (01 F DW F 07; 01 = Montag, 07 = Sonntag)
- HH Stunde (00 F HH F 23)
- MM Minute (00 F MM F 59)
- SS Sekunde (00 F SS F 59)

8 Bibliotheken, Funktionsbausteine und Funktionen

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

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

Die Bibliotheken finden Sie in den Verzeichnissen

- Lib_Common für alle Steuerungen,
- Lib_CPU121 für die Steuerung XC121.

Bibliotheken handhaben

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

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

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

Dokument	Bibliothek
AWB 2700-1437	Standard.lib Util.lib XC121_Util.lib
Online-Hilfe oder PDF-Dateien im Windows-Startmenü unter Moeller Software → easy Soft CoDeSys → Dokumentation...	SysLib... lib
MN05010002Z-DE (vormals AWB2786-1456)	XS40_MoellerFB. Lib/ Visu.lib/...
AN2700K20	3S_CANopenDevice.lib 3S_CANopenManager.lib
AN2700K19	3S_CANopenNetVar.lib
AN2700K27	SysLibCan.lib
MN05010001Z-DE (vormals AWB2786-1554)	CANUserLib.lib CANUser_Master.lib

Weitere Systembibliotheken installieren

Zum Nachinstallieren von Bibliotheken sind folgende Schritte notwendig:

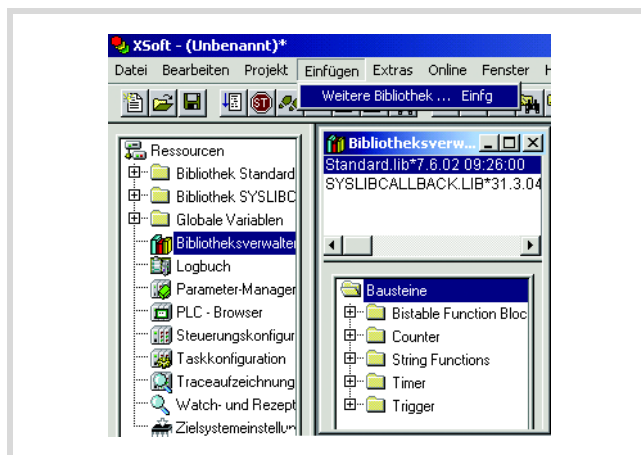


Abbildung 45: Bibliotheken nachinstallieren

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

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

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

Die Bibliothek erscheint danach im Bibliotheksverwalter.

XC121-spezifische Funktionen

Bibliothek „XC121_Util.lib“

Diese Bibliothek stellt die in der folgenden Abbildung dargestellten Funktionen zu Verfügung:

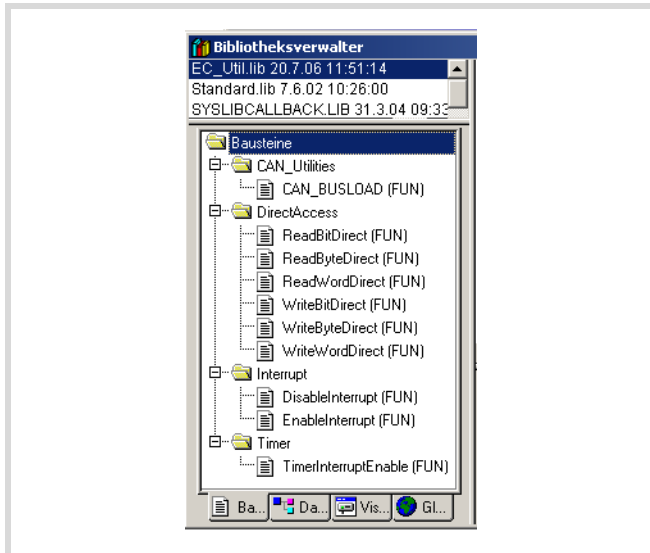


Abbildung 46: Funktionen der Bibliothek XC121_Util.lib

Funktion GETAPPLICATIONSWITCH

Mit dieser Funktion können Sie die Stellung des APPLICATION-Schalters abfragen. Nach einem H-Signal am Eingang „xEnable“ wird am Ausgang der eingestellte Wert des Schalters angezeigt.

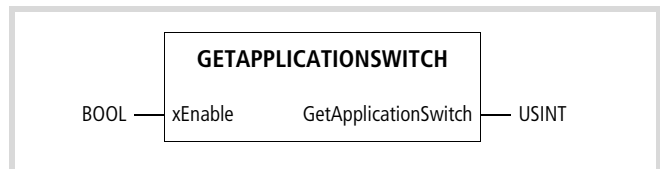


Abbildung 48: Funktion GETAPPLICATIONSWITCH

→ Die „Timer“-Funktionen sind im Abschnitt „Timer-Interrupt“ auf Seite 29 beschrieben.

Funktion CAN_BUSLOAD

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

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

Informationen zur Auswertung des Rückgabewertes finden Sie beim Browser-Befehl „canload“ auf Seite 36.

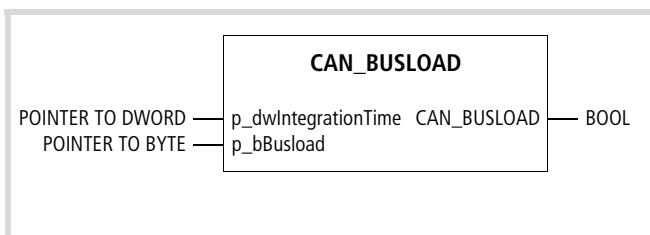


Abbildung 47: Funktion CAN_BUSLOAD

9 Verbindungsaufbau PC – XC121

Zur Herstellung der Verbindung zwischen PC und XC121 müssen die Kommunikationsparameter beider Geräte übereinstimmen.

Hierzu passen Sie zunächst die Kommunikationsparameter des PC an die CPU-Parameter an. Gehen Sie dabei von den Default-Parametern der CPU aus und übertragen Sie diese wie in Abbildung 49 dargestellt.

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

Anschließend können Sie die Parameter der CPU neu festlegen (→ Abbildung 50). Diese Parameter-Änderungen müssen Sie dann für den PC erneut anpassen.

Kommunikationsparameter des PC

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

- ▶ Wählen Sie den Menüpunkt «Online → Kommunikationsparameter» aus.
- ▶ Legen Sie den Port (COM1- oder COM2-Schnittstelle) fest, → Abschnitt „Parameter ändern“
- ▶ Übernehmen Sie die weiteren Parameter aus der Abbildung 49.
- ▶ Bestätigen Sie die Parameter mit OK.
- ▶ Loggen Sie sich in die Steuerung ein.

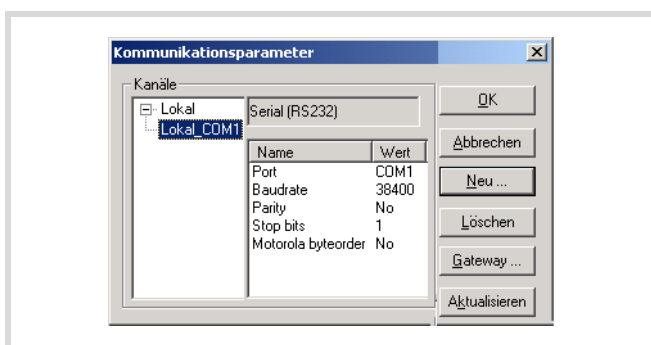


Abbildung 49: Kommunikationsparameter des PC festlegen

Parameter ändern

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

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

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

Kommunikationsparameter (Baudrate) der CPU

- ▶ Öffnen Sie das Dialogfeld «Ressourcen → Steuerungskonfiguration».
- ▶ Wählen Sie die Registerkarte „Weitere Parameter“.
- ▶ Wählen Sie im Listenfeld „Baudrate“ die Baudrate aus (z. B. 57600 Bit/s wie in Abbildung 50).

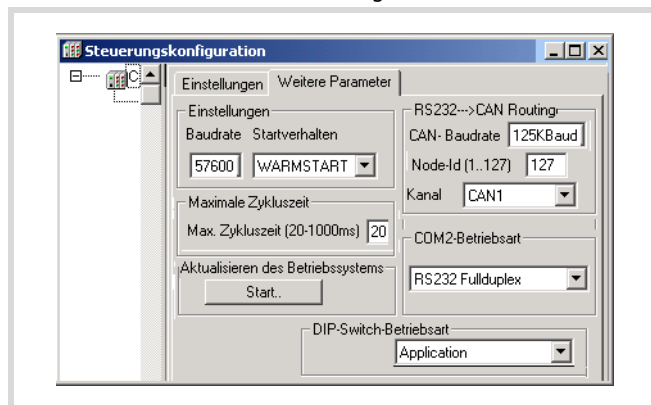


Abbildung 50: Kommunikationsparameter der CPU festlegen

- ▶ Loggen Sie sich in die Steuerung ein.

Sie erhalten folgende Abfrage:



Abbildung 51: Abfrage nach Programmänderung

- ▶ Beantworten Sie diese Frage mit „Ja“.

Das Programm wird geladen. Nach einer Verzögerung erhalten Sie die Fehlermeldung für Kommunikationsfehler, da die Baudrate zwischen CPU und PC nicht mehr übereinstimmt:

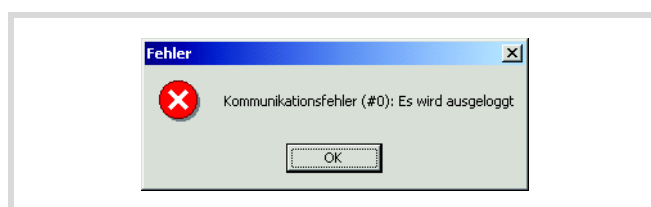


Abbildung 52: Kommunikationsfehler

- ▶ Bestätigen Sie die Fehlermeldung mit OK.

Um den PC wieder ankoppeln zu können, müssen Sie die Baudrate des PC erneut anpassen.

10 Systemparameter über STARTUP.INI-Datei einstellen

Parameterübersicht

Mit Hilfe der INI-Datei können Sie Systemparameter definieren und auf der MMC abspeichern. Beim Start übernimmt die Steuerung die Parameter. Die INI-Datei wird immer mit allen steuerungsspezifischen Einträgen erstellt (→ Tabelle 12). Parameter, die Sie in der Steuerungskonfiguration spezifizieren, erhalten zunächst keinen Eintrag.

Tabelle 12: Parameter in der INI-Datei

```
Einträge
COM_BAUDRATE
CAN1_BAUDRATE
CAN1_NODEID
CAN2_BAUDRATE
CAN2_NODEID
CAN_ROUTINGID
CAN_ROUTING_CHANNEL
```

Aufbau der INI-Datei

Eine INI-Datei ist eine Textdatei mit einem festgelegten Datenformat. Ab einer namentlich benannten Sektion (in eckigen Klammern) wie z. B. [STARTUP] werden die Systemparameter aufgeführt, gefolgt von einem Gleichheitszeichen und deren Wert. Die Zeile schließen Sie mit der Eingabetaste (Return) ab.

```
COM_BAUDRATE=38400 Return
```

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

```
; CAN_ROUTING_CHANNEL=CAN1
```

Beispiel: Default STARTUP.INI für XC121

```
[STARTUP]
TARGET=XC-CPU121-2C256K
COM_Baudrate=38400
CAN1_Baudrate=125
CAN1_NODEID=2
CAN2_Baudrate=125
CAN2_NODEID=2
CAN_ROUTINGID=127
CAN_ROUTING_CHANNEL=CAN1
```

INI-Datei erstellen

Die INI-Datei erstellen Sie mit dem Browser-Befehl „createstartupini“ auf der MMC-Karte. Dabei werden für die Systemparameter die aktuell eingestellten Parameterwerte übernommen. Für die Programmierschnittstelle wird folgender Parameter übernommen:

```
COM_BAUDRATE=38400
```

Die Parameter aus der Programm-Steuerungskonfiguration haben keine Default-Werte, z. B.:

```
CAN_ROUTINGID=
CAN_ROUTING_CHANNEL=
CAN1_BAUDRATE=
CAN1_NODEID=
```

Eine bereits vorhandene Datei können Sie nicht über das Browserkommando „createstartupini“ verändern oder überschreiben. Falls Sie das Kommando trotzdem eingeben, erscheint eine Warnung. Um eine neue Datei zu erstellen, löschen Sie die bestehende Datei zuerst → Abschnitt „INI-Datei löschen“ auf Seite 42.

Die Parameter können Sie mit einem Text-Editor ändern, wenn Sie die MMC in den MMC-Slot des PC stecken. Die Datei STARTUP.INI befindet sich im Verzeichnis „MOELLER/XC-CPU121-2C256K/BOOTPRJ/“. Während des Online-Betriebes können Sie im Menü „Online“ die Befehle „Datei aus Steuerung laden“ und „Datei in Steuerung schreiben“ ausführen.

Einträge der INI-Datei

- Eintrag zur Spezifizierung des Zielsystems.

```
TARGET=XC-CPU121-2C256K
```

- Parameter zur Programmierung über die serielle Kommunikation mit CoDeSys:

```
COM_BAUDRATE=38400
```

Die Parameter aus der INI-Datei haben Vorrang vor den Parametern aus der Steuerungskonfiguration. Nach einem Programm-Download oder nach Laden des Bootprojektes werden die Parameter aus der Steuerungskonfiguration nicht übernommen:

- Parameter der Steuerungskonfiguration für CAN Level 2 Route:

```
CAN_ROUTINGID=127
CAN_ROUTING_CHANNEL=CAN1
CAN1_BAUDRATE=125
CAN1_NODEID=
CAN2_BAUDRATE=
CAN2_NODEID=
```

Startverhalten der XC121 bei gesteckter MMC mit INI-Datei

Beim Einschalten der XC121 werden die Daten der INI-Datei in die XC121 übertragen. Sie bleiben auch nach dem Laden eines neuen Programms aktiv, bis Sie den Browser-Befehl „removestartupini“ ausführen.

Parameter ändern

Wenn Sie eine MMC mit einer INI-Datei in die XC121 stecken und die XC121 einschalten, übernimmt die XC121 die Parameter der INI-Datei der MMC. Es werden keine Parameter von geladenen Projekten übernommen.

Die Parameter bleiben erhalten, bis Sie den Browser-Befehl „removestartupini“ eingeben und dann die XC121 aus-/einschalten. Die XC121 arbeitet nun mit den Parametern des Projekts.

INI-Datei löschen

Es stehen zwei Browser-Befehle zur Verfügung, die auf die MMC zugreifen.

- removestartupini:
Löscht die INI-Datei auf der MMC und die Daten auf der CPU. Beim nächsten Einschalten werden die Daten aus dem Projekt übernommen.
- removeprojfrommmc:
Löscht das Projekt und die INI-Datei auf der MMC. Die Daten auf der CPU bleiben erhalten.

➔ Wenn Sie im Online-Betrieb den Befehl RESET-Ursprung im Steuerungskonfigurator ausführen, wird das BTS und das Projekt auf der MMC gelöscht. Die INI-Datei bleibt erhalten ➔ Abschnitt „Speicherkarte – MMC“ auf Seite 11.

11 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 Sie von einer Steuerung, die mit dem Programmiergerät verbunden ist, Zugriff auf alle routingfähigen Steuerungen am CAN-Bus erhalten. 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 über die gleiche aktive CAN-Baudrate verfügen.
- Auf beiden Steuerungen muss eine gültige Routing-Node-Id eingestellt sein.

Routeigenschaften der XC121

Die XC121 unterstützt das Routing über den CAN-Bus.

Das Routing kann ohne vorherigen Download eines Anwenderprogramms ausgeführt werden (Default: CAN1, 125 kBaud, Node-Id 127). Die Zielsteuerung muss hierfür nicht als CAN-Master bzw. CAN-Device konfiguriert werden.

Sie können z. B. ein Programm vom PC über eine Steuerung der XC-Gerätefamilie in die XC121 laden. Ordnen Sie in diesem Fall der XC121 (Zielsteuerung) eine Routing-Node-Id zu.

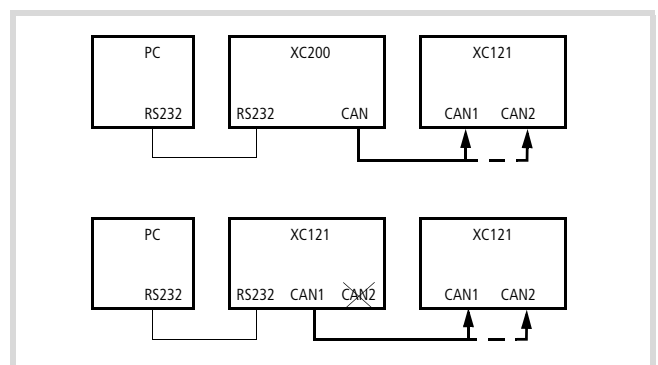


Abbildung 53: Programm-Download per Routing

Sie können die Zielsteuerung wahlweise über die Schnittstellen CAN1 oder CAN2 anschließen.

➔ Wenn Sie die XC121 als routende Steuerung einsetzen, dürfen Sie für die weiterführende Verbindung nur den CAN1-Kanal verwenden.

Zur Einstellung der Routing-Node-Id können Sie zwischen zwei verschiedenen Verfahren wählen:

- Einstellung über den Steuerungskonfigurator
- Einstellung über den APPLICATION-Schalter

Einstellung über den Steuerungskonfigurator

- Klicken Sie im Ordner „Configurator“ das Register „Weitere Parameter“ an (➔ Abbildung 5).

Im Feld DIP-Switch-Betriebsart darf „Node-Id-Routing“ nicht angewählt sein!

- Geben Sie im Feld „RS232->CAN Routing“ die CAN-Baudrate, die Node-Id und den Kanal CAN1 oder CAN2 und ein.

Einstellung über den APPLICATION-Schalter

- Klicken Sie im Steuerungskonfigurator das Register „Weitere Parameter“ an (→ Abbildung 5 auf Seite 10).
- Stellen Sie im Feld „DIP-Switch-Betriebsart“ die Betriebsart „Node-Id Routing“ ein.
- Stellen Sie im Feld „RS232->CAN Routing“ die Baudrate ein. Folgende Baudraten sind möglich: 50, 100, 125, 250 und 500 kBaud (Default: 125 kBaud)
- Am APPLICATION-Schalter stellen Sie nun die Routing-Node-Id und die CAN-Schnittstelle ein:

Schalter 1 – 7: (Routing-)Node-Id 1 – 127; bei ungültiger Adresse 0 wird Default-ID 127 verwendet.

Schalter 8: OFF = CAN1, ON = CAN2

→ Die Einstellungen am APPLICATION-Schalter haben Vorrang gegenüber der Konfiguratoreinstellung.

Weitere Informationen im Kapitel „APPLICATION-Schalter (S2)“ auf Seite 10.

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 13.

Tabelle 13: Blockgröße für die Datenübertragung

	Programm-/Datei-Transfer		Routing	
	BTS < V1.03.03	BTS ≥ V1.03.03	BTS < V1.03.03	BTS ≥ V1.03.03
Blockgröße Default: 128 kByte	128 kByte	4/128 kByte	Routing nicht möglich	4 kByte



Achtung!

Der Programmdownload mit einer Blockgröße von 4 kByte auf eine Steuerung mit dem Betriebssystem < V1.03.03 führt zu einem Fehlverhalten!

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

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

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

Einstellung der Blockgröße:

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

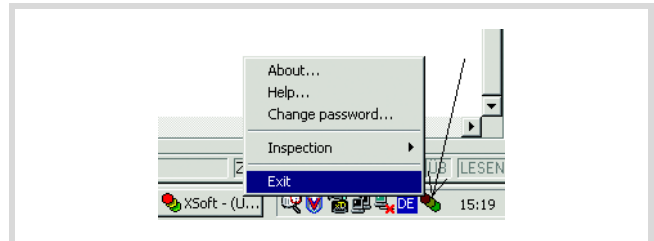


Abbildung 54: CoDeSys-Gateway-Server schließen

- Ändern Sie die Blockgröße auf den gewünschten Wert.

Zum Eintrag der Blockgröße in die Registry stehen im 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 1000_{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 CANopen-Bus können als Master oder als Device konfiguriert werden. Zur eindeutigen Identifikation erhalten die Steuerungen eine Node-Id/Knotennummer (Adresse). Möchten Sie mit Hilfe der Routing-Funktion auf eine (Ziel-)Steuerung zugreifen, müssen Sie der Ziel-Steuerung eine weitere (Routing-) Node-Id zuordnen.

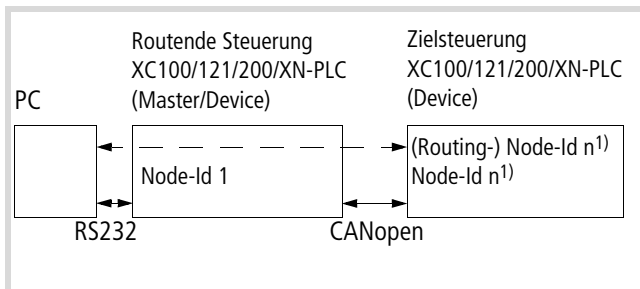


Abbildung 55: XC100/121/200, XN-PLC am CANopen-Bus, Routingprinzip

- 1) Für die Node-Id der Device-Funktion und die Node-Id der Routing-Funktion gilt:
 - Die (Routing-) Node-Id muss **ungleich** der Node-Id (Device) sein!

Vorgehensweise

- Verbinden 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:

Die Node-Id/Knotennummer und die Baudrate der Zielsteuerung zur Routingfunktion legen Sie in der Steuerungskonfiguration im Fenster „Weitere Parameter“ fest, → Abbildung 56:

- Geben Sie dort im Feld „RS232 → CAN Routing“ die Baudrate am CANopen-Bus und die Node-Id/Knotennummer an.

Node-Id und Baudrate werden mit dem Projektdownload in die Steuerung übertragen.

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

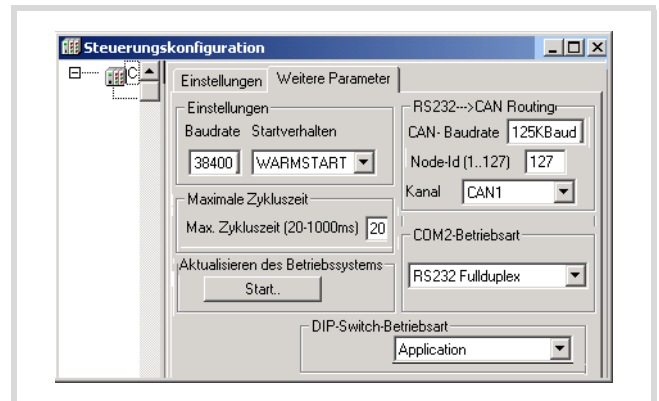


Abbildung 56: CANopen-RoutingEinstellungen

Die folgenden Abbildungen zeigen unabhängig von den RoutingEinstellungen, wo Sie die Baudrate und die Node-Id der Steuerungen eintragen, die als Master oder Device konfiguriert wurden. Die Einstellungen erfolgen bei der Master-Steuerung im Register „CAN-Parameter“ bzw. bei der Device-Steuerung im Register „CAN-Einstellungen“.

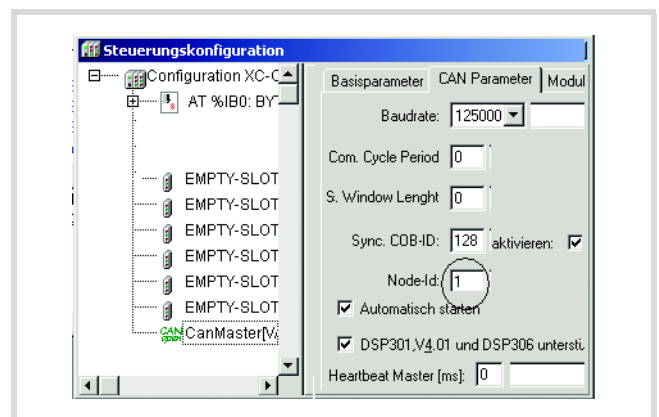


Abbildung 57: CAN-Masterparameter

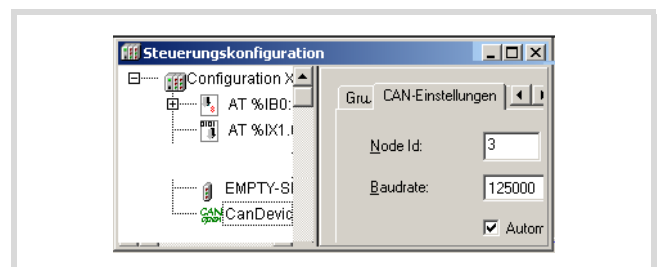


Abbildung 58: CAN-Deviceparameter

Beispiel

Im folgenden Beispiel wird die Vorgehensweise eines Zugriffs auf ein Steuerungsprogramm erklärt.

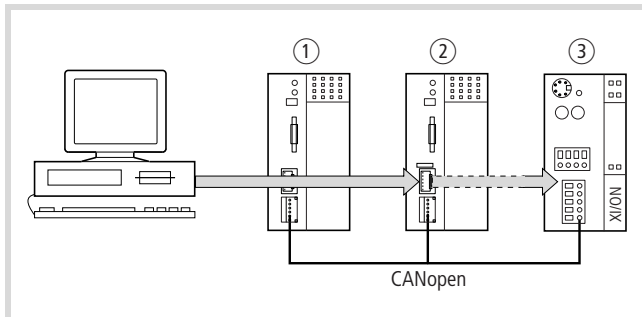


Abbildung 59: Diagnosemöglichkeiten

- ① XC100 mit Node-Id 1
- ② XC200 mit Node-Id 2
- ③ XN-PLC mit Node-Id 3 (Anstelle der XN-PLC können Sie auch eine XC121 einsetzen)

Den PC haben Sie an die Steuerung mit der Node-Id „2“ angeschlossen und Sie wollen auf die Zielsteuerung mit der Node-Id „3“ zugreifen.

- Öffnen Sie das Projekt der Zielsteuerung (Node-Id 3), 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.
XC200: 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. Sie gelangen zurück zum Ausgangsfenster.

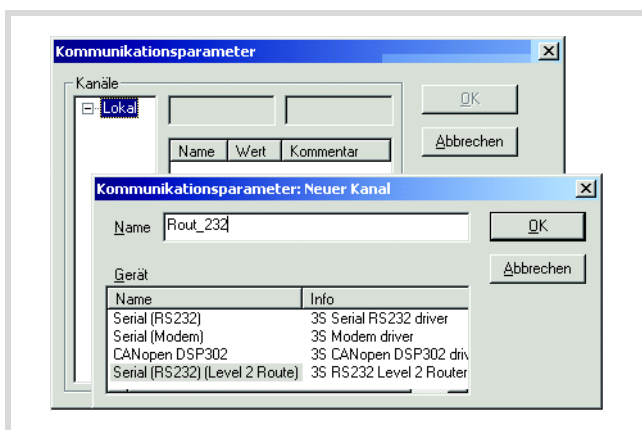


Abbildung 60: Kanal parametrieren

Sie haben nun die Parameter für die Hardwareverbindung 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 Node-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.

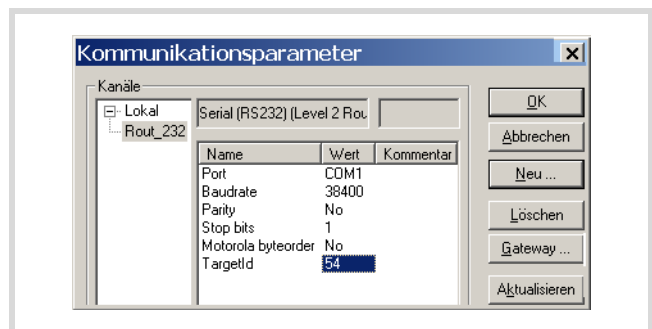


Abbildung 61: Target-Id der Zielsteuerung einstellen

Steuerungskombinationen zum Routing

Folgende Steuerungen unterstützen das Routing:

Von → Nach ↓	XC100	XC121	XC200	XN-PLC-CANopen
XC100	×	×	×	×
XC121	×	×	×	×
XC200	×	×	×	×
XN-PLC-CANopen	×	×	×	×

12 RS232-Schnittstelle im Transparent-Modus

Im Transparent-Modus erfolgt der Datenaustausch zwischen der XC121 und Datenendgeräten (z. B. Terminals, Drucker, PC, Messgeräte) ohne Interpretation der Daten. Dazu ist die serielle Schnittstelle RS232 (COM1) der XC121 per Anwenderprogramm in den Transparent-Modus zu schalten.

Zum Ausführen des Transparent-Modus stehen Funktionen zum Öffnen und Schließen der Schnittstelle, zum Senden und Empfangen von Daten und zum Setzen der Schnittstellenparameter zur Verfügung.

→ Da die Kontrollleitungen der Schnittstelle nicht aktiv sind, können Sie die Funktionen „SysComReadControl“ und „SysComWriteControl“ nicht benutzen.

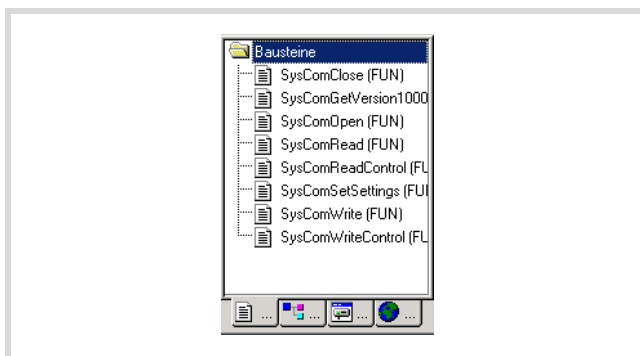


Abbildung 62: Funktionsübersicht

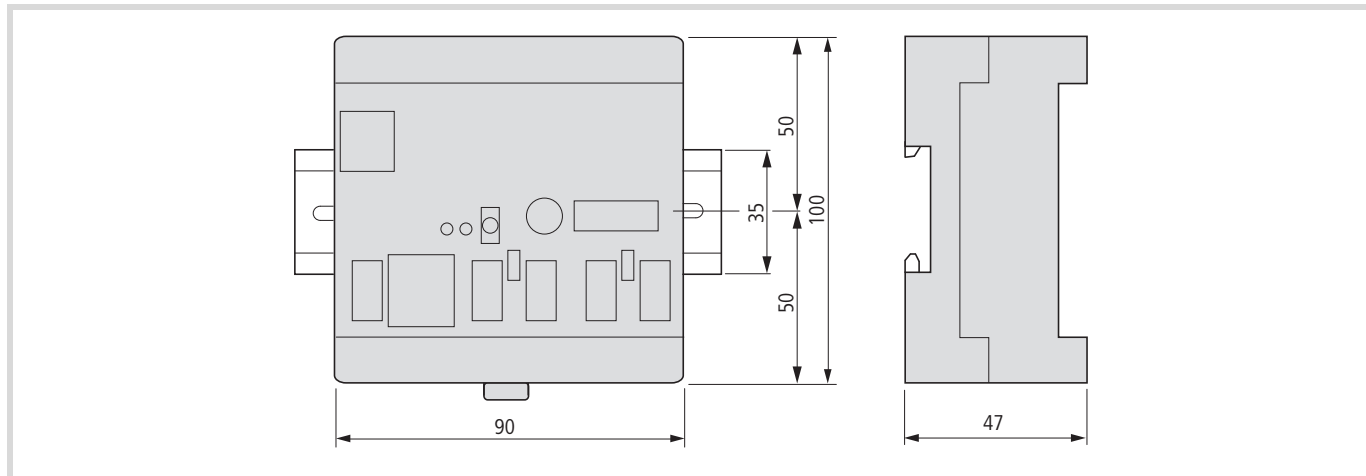
Die Funktionen des Transparent-Modus sind in der Bibliothek „XC121_SysLibCom.lib“ enthalten. Die Bibliothek muss daher in den Bibliotheksverwalter eingebunden sein. Eine Beschreibung der Funktionen finden Sie im Handbuch „CoDeSys-Funktionsbausteine“ (MN05010002Z-DE, früher AWB2786-1456D).

→ Befindet sich die RS232-Schnittstelle (COM1) der XC121 im Transparent-Modus, ist eine Programmierung über diese Schnittstelle nicht möglich. Der Transparent-Modus muss erst geschlossen werden. Mit Schließen des Transparent-Modus werden die ursprünglichen Kommunikationsparameter wieder initialisiert. Das Schließen des Transparent-Modus erfolgt zwangsweise bei einem SPS-Zustandswechsel in den STOP oder mit dem Aufruf der Funktion „SysComClose“.

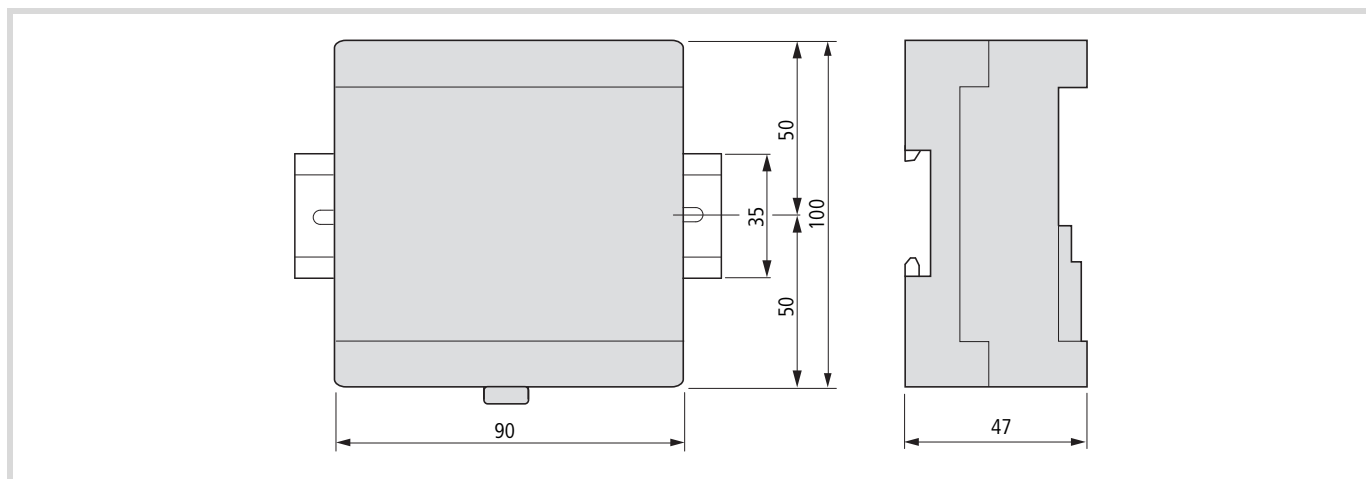
Anhang

Abmessungen

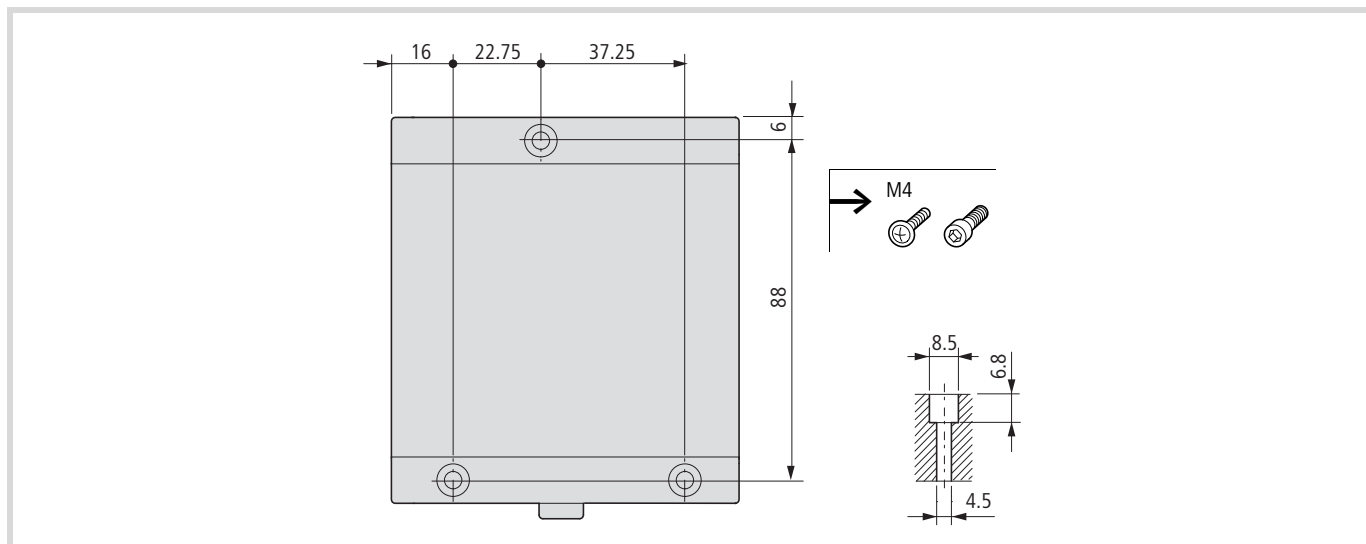
XC-CPU121



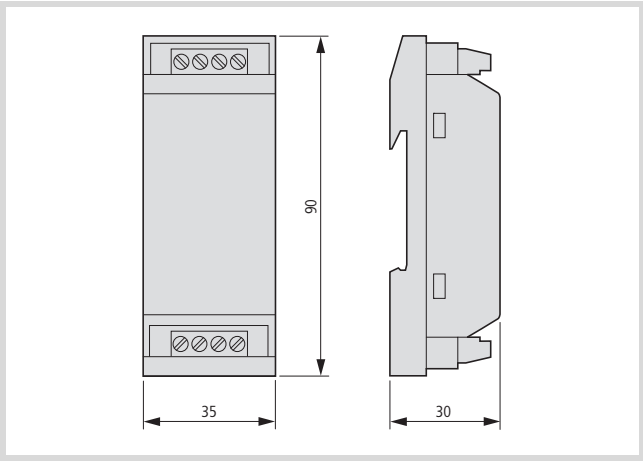
XIO-EXT121-1



XC-CPU121, XIO-EXT121-1



24-V-DC-Netzfilter XT-FIL-1



Technische Daten

XC-CPU121/XIO-EXT121-1

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		Frequenz 5 bis 9 Hz; 3,5 mm Amplitude 9 bis 150 Hz; 1,0 g konstante Beschleunigung
Schockfestigkeit		15 g/11 ms
Überspannungskategorie		II
Verschmutzungsgrad		2
Schutzart		IP20
Bemessungsisolationsspannung	V	500
Störaussendung (Industrie)		EN 61000-6-4
Störfestigkeit (Industrie)		EN 61000-6-2
Pufferung der Uhr		mindestens 72 Stunden
Gewicht	kg	0,15
Maße (B x H x T)	mm	90 × 100 x 47

Elektromagnetische Verträglichkeit (EMV)		
elektrostatische Entladung (IEC/EN 61000-4-2, Level 3, ESD)		
Kontaktentladung	kV	4
Luftentladung	kV	8
elektromagnetische Felder (IEC/EN 61000-4-3, RFI)	V/m	10
Burst Impulse (IEC/EN 61000-4-4, Level 3)		
Versorgungsleitungen	kV	2
Signalleitungen	kV	1
energiereiche Impulse (Surge) (IEC/EN 61000-4-5)	kV	0,5
Einströmung (IEC/EN 61000-4-6)	V	10

XC-CPU121

Anschlussstechnik		
Versorgungsspannung		
Stecker		Federzugklemmenblock, 4 polig
Anschlussquerschnitt	mm ²	0,14 bis 1 (AWG28-18)
COM1-Schnittstelle		
Stecker		RJ45
COM2-, CAN1-, CAN2-Schnittstelle		
Stecker		Federzugklemmenblock, 6 polig
Anschlussquerschnitt	mm ²	0,14 bis 0,5 (AWG28-20)
Spannungsversorgung (24 V/0 V)		
Eingangsspannung	V DC	24
zulässiger Bereich	V DC	20,4 bis 28,8
Eingangsleistung	W	max. 1,44
Eingangsstrom	mA	60
Restwelligkeit	%	≤ 5
Überspannungsschutz		ja
Verpolungsschutz		ja
Einschaltstrom		keine Begrenzung (Begrenzung nur durch vorgeschaltetes 24-V-DC-Netzteil)
Netzausfallüberbrückung		
Dauer des Einbruchs	ms	10
Wiederholrate	s	1
Netzfilter, extern		Typ: XT-FIL-1, siehe Technische Daten auf Seite 55
Netzfilter, intern		ja

CPU		
Mikroprozessor		Infineon XC161
Speicher		
Programmcode	kByte	256
Programmdaten	kByte	14 Segmente a 16KB
Merker /Input/Output/Retaindaten	kByte	16/4/4/8
Zykluszeit für 1 k Anweisungen (Bit, Byte)	ms	< 0,3
Schnittstellen		
COM1 (RS232) ohne Handshake-Leitungen		
Datenübertragungsrate für die Programmierung Zeichenformat: 8 Bit Daten, keine Parität, 1 Stopbit	Bits/s	19 200, 38 400 (Default), 57 600 Bit/s
Anschlusstechnik		RJ45-Buchse
Galvanische Trennung		keine
In der Betriebsart Transparentmode		
Datenübertragungsraten	Bits/s	300, 600, 1 200, 2 400, 4 800, 9 600, 19 200, 38 400, 57 600, 115 200 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
COM2 (RS232/RS485) ohne Handshake-Leitungen		
Datenübertragungsrate (Transparentmode) Einstellung über Funktionsbausteine	Bits/s	300, 600, 1 200, 2 400, 4 800, 9 600, 19 200, 38 400, 57 600
Zeichenformate		8E1, 8O1, 8N1, 8N2, 7E2, 7O2, 7N2, 7E1 Einstellung über FB
Galvanische Trennung		Keine
Busabschlusswiderstände für RS485		Extern
CAN1-/CAN2-Schnittstelle		
Datenübertragungsrate	kBit/s	10 bis 500 ¹⁾
Potentialtrennung		Nein
Teilnehmer		126
Busabschluss		Für jede Schnittstelle (CAN1/CAN2) einschaltbar
PDO-Art		Asynchron, zyklisch, azyklisch

1) Höhere Übertragungsraten, die in der Steuerungskonfiguration wählbar sind, werden nicht unterstützt!

XIO-EXT121-1

Anschlussstechnik		
X1 Stecker		
Steckertyp		Federzugklemmenblock, 20 polig, B2L 3.5(Weidmüller)
Anschlussquerschnitt (eindrätig)	mm ²	0,5 bis 1
X2/X3-Stecker		
Steckertyp		Federzugklemmenblock, 10 polig, BLZF 3.5/180 oder BLI/O 3.5/10F mit LEDs (Weidmüller)
Anschlussquerschnitt (eindrätig)	mm ²	0,5 bis 1
Spannungsversorgung (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. 1,68 + Bedarf der XI/OC-Baugruppen
Eingangsstrom	mA	70 + Bedarf der XI/OC-Baugruppen
Ausgangsspannung für die Signalmodule		
Nennwert	V DC	5
Ausgangsstrom	A	2,5
Kurzschlussfestigkeit		ja
galvanische Trennung zur Versorgungsspannung		nein
Restwelligkeit	%	≤ 5
Überspannungsschutz		ja
Verpolungsschutz		ja
Einschaltstrom	A	max. 1
Digital-Eingänge		
Anzahl bei X2		9 bei Stecker BLI/O 3.5/10F 10 bei Stecker BLZF 3.5/180
Anzahl bei X3		8 (auch als Ausgänge nutzbar)
Bemessungsspannung	V DC	24
für Signal 0	V	< 5
für Signal 1	V	> 15
Bemessungsstrom bei Signal 1	mA	3,3
Verzögerungszeit		
X2: DI0...DI3	μs	20
X2: DI4...DI9	μs	250
X2: DX0...DX7	ms	20
Potentialtrennung		nein
Digital-Ausgänge		
Anzahl bei X3		8 (auch als Eingänge nutzbar)
Bemessungsspannung U_e	VDC	24
zulässiger Bereich	VDC	20,4 bis 28,8
Restwelligkeit	V	≤ 5 %
Bemessungsstrom I_e bei Signal 1	A	0,5 bei 24 V DC

Gleichzeitigkeitsfaktor		1
relative Einschaltdauer	ms	100 %
Lampenlast ohne R_v	W	5
Potentialtrennung		ja
Reststrom bei Signal 0 pro Kanal	mA	< 0,1
Max. Ausgangsspannung		
bei 0 ext. Last < 10 M Ω	V	2,5
bei 1 bei $I_e = 0,5$ A		$U = U_e - 1$ V
Kurzschlusschutz		ja
Kurzschlussauslösestrom		
für $R_a < 10$ M Ω		$0,7 \leq I_e \leq 2$ pro Ausgang
gesamter Kurzschlussstrom	A	16
Spitzenkurzschlussstrom	A	32
max. Schaltfrequenz	S/h	40000
Parallelschaltbarkeit		ja (gruppenweise: Q0 - Q3 oder Q4 - Q7)
Daten zu analogen E/A		
analoge Eingänge 0 ... 10 V		
Anzahl Kanäle		2
Eingangsspannungsbereich	V	0 - 10
Auflösung	Bit	10
Wandlungszeit	ms	≤ 5
Gesamtgenauigkeit		$\leq \pm 1$ % (vom Skalenendwert)
Eingangswiderstand	k Ω	200
analoge Eingänge 0 ... 20 mA		
Anzahl Kanäle		2
Eingangsspannungsbereich	mA	0 ... 20
Auflösung	Bit	10
Wandlungszeit	ms	≤ 5
Gesamtgenauigkeit		$\leq \pm 1$ % (vom Skalenendwert)
Eingangswiderstand	Ω	50
Pt100		
Anzahl Kanäle		2
Temperaturbereich	°C	-200...+200
Widerstandsbereich	Ω	18,5...175,8
Genauigkeit		± 10 Digits ¹⁾
analoge Ausgänge		
Anzahl Kanäle		2
Ausgangsspannungsbereich	V	0 ... 10
Auflösung	Bit	12
Wandlungszeit	ms	≤ 5
Gesamtgenauigkeit		$\leq \pm 1$ % (vom Skalenendwert)
Externer Lastwiderstand	k Ω	10

1) Zugeordnete Temperaturwerte → Abbildung 22

24-V-DC-Netzfilter 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		Frequenz 5 bis 9 Hz; 3,5 mm Amplitude 9 bis 150 Hz; 1,0 g konstante Beschleunigung
Schockfestigkeit		15 g/11 ms
Schlagfestigkeit		500 g/Ø 50 mm ±25 g
Überspannungskategorie		II
Verschmutzungsgrad		2
Schutzart		IP20
Bemessungsstoßspannung	V	850
Störaussendung (Industrie)		EN 61000-6-4
Störfestigkeit (Industrie)		EN 61000-6-2
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	Aderquerschnitt, CANopen-Kabel	14	D	Datenspeicher	12
	Adressierung, Steuerung am CANopen-Feldbus	45		Datenzugriff, auf MMC	11
	Akku-Pufferung	24		Demontage, XC121 von Hutschiene	15
	Aktoren anschließen	19		Diagnosemöglichkeiten	46
	Analog-Ein-/Ausgänge	20		DIP-Schalter	10
	APPLICATION-Schalter	10		DIP-Switch-Betriebsart	10
	einstellen	16		Direkter IO-Zugriff	30
	Applikations-Routine	25, 29, 30		Direkter Peripheriezugriff	31
	Aufbau, XC121	7		Download, Betriebssystem	32
	Auslieferungszustand wiederherstellen	9			
	Ausschalten der Spannungsversorgung	24	E	E/A-Modul	7
B	Baudrate festlegen/ändern	39		Echtzeituhr	11
	Baudrate, CAN-Bus	13		Einbaulage, Steuerung im Schaltschrank	17
	Baugruppenträger	7		Einschaltverhalten	23, 42
	Belüftung	17		Einzelschritt-Betrieb	26
	Betriebsart	10		Einzelzyklus-Betrieb	26
	Betriebsartenschalter	9		Elektromagnetische Beeinflussung	17
	Betriebssystem herunterladen/aktualisieren	32		Ereignis-Task	25
	Betriebssystem-Update	23	F	Forcen	26
	Bibliotheken			format (Browser-Befehl mit Löschfunktion)	11
	CANUser.lib, CANUser_Master.lib	5		Funktion	
	SysLibRTC	11		CAN_BUSLOAD	38
	XC121_File.lib	11		EnableInterrupt	27
	XC121_SysLibCom.lib	47		FileOpen	11
	XC121_Util.lib	10, 27, 29, 38		FileRead	11
	Bibliotheken installieren	37		GETAPPLICATIONSWITCH	38
	Blitzschutzmaßnahmen	18		GetApplicationSwitch	10
	Blockgröße, für Datenübertragung	44		Read...Direct	30
	Bootprojekt	23		removeprojfrommmc	11
	Breakpoint	26		Reset kalt	9
	Browser-Befehle	35		TimerInterruptEnable	29
	Busabschlusswiderstand	13, 14		Funktionen	37
	Busauslastung, CANopen-Feldbus	36, 38		Betriebsartenschalter	9
	Buslänge, CANopen	14		zum Transparent-Modus	47
C	CAN-Device (XC121 konfigurieren)	13		zur Echtzeituhr	11
	CAN-Deviceparameter	45		Funktionsbausteine	37
	CAN-Direct (Direktzugriff auf CAN-Objekte)	14	G	Geräteanordnung	17
	CAN-Einstellungen (Register in der Software CoDeSys)	13		Gesamtausbau, maximal	7
	canload, Browserbefehl	36		Grundausbau, maximal	7
	CAN-Master (XC121 konfigurieren)	14	H	Hardware-Timer	25
	CAN-Masterparameter	45	I	Inbetriebnahme	26
	CANopen			Induktivitäten	17
	-Kabel, Eigenschaften	14		INI-Datei	41
	-Netzwerk, Anforderungen	14		Initialwerte aktivieren	25
	-Routingeinstellungen	45		Interrupt	27, 29
	-Schnittstelle	13		Interruptquelle	29
	CoDeSys-Gateway-Server	44		IO-Zugriff, direkter	30
	CPU starten	9			

K	Kabel, CANopen (Eigenschaften)	14		Schnittstelle	
	Kanal parametrieren	46		CANopen	13
	Knotennummer	45		Kommunikationsparameter festlegen	39
	Kommunikation testen (vom Browser zur PLC und zurück)	36		seriell (COM1/COM2)	12
	Kommunikationsparameter	39		Übersicht	9
	Konfiguration			Verbindung bei Routing	43
	XC121 als CAN-Device	13		Schutzbeschaltung von Störquellen	17
	XC121 als CAN-Master	14		Segmente	12
	XIO-EXT121-1	21		Sensoren anschließen	19
L	Laufzeitsystem	23		Serielle Schnittstellen COM1/COM2	12
	LED-Zustandsanzeige	11		SET-Taster	9
	Leitungsführung	17		Source-Code	33
	lokale Erweiterung	7		Spannungsversorgung anschließen	18
	Löschen, Inhalt der Speicherkarte	11		Spannungsversorgung ausschalten/unterbrechen	24
M	MMC	11		Speicherkarte	11
	Montage			Speichernutzung, Grenzwerte	12
	XC121 auf Hutschiene	15		STARTUP.INI	41
	XIO-EXT121-1 an XC121	15		Startverhalten	23, 42
N	Node-Id	45		Startverhalten, Einstellen in der Software CoDeSys ...	24
	CAN1/CAN2	10		STOP	9, 25
	einstellen	13		Störeinflüsse	17
	Routing	10		Systemereignisse	25, 27
P	PLC-Browser	35		Systemparameter einstellen	41
	Programmbearbeitung	25		System-Selbsttest	23
	Programmierschnittstelle	12		Systemuhr, Pufferung	11
	Programmiersoftware	7		Systemzeit	25
	Programmstopp	25			
	Projektierung	17		T	
	Pufferzeit, Batterie	11		Target-Id	45
Q	Quellcode	33		TCP/IP-Verbindung (beim Routing)	44
R	ReadBitDirect	30		Testfunktionen	26
	reflect	36		Timer-Interrupt	29
	remanente Variablen	24		Transparent-Modus	12, 47
	Reset	25, 27		U	
	Reset kalt	9		Übertragungsrate, CANopen	14
	Routing-Id	10		Unterbrechen der Spannungsversorgung	24
	RUN	9		Unterbrechungsfreie Stromversorgung	24
S	Schaltschrankaufbau	17		V	
	Schirmung	17		Variablen	
	Schleifenwiderstand, CANopen-Kabel	14		Verhalten beim Start	24
				Verhalten nach Reset	25
				Verbindungsaufbau PC – XN-PLC	39
				Verdrahten, Ein-/Ausgänge (Montagebeispiel)	16
				Verdrahtung	17
				Verdrahtungsbeispiel, Spannungsversorgung anschließen	18
				Verzögerungszeit eintragen	29
				Z	
				Zustandsanzeige	
				an der LED	11
				in der Software CoDeSys	26
				Zwangssetzen, Variablen und I/Os	26
				Zykluszeit überwachen	25