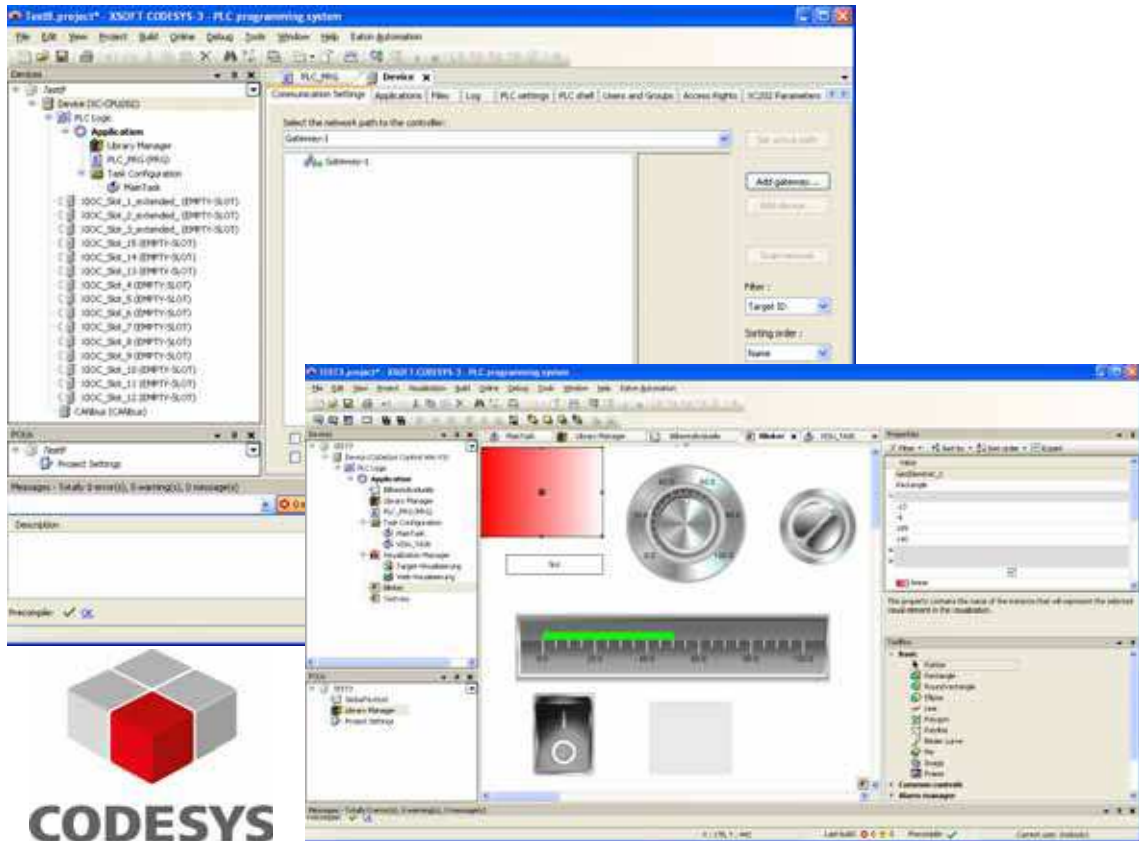


# XSOFT-CODESYS-3 SPS-Programmierung



**Hersteller**

Eaton Automation AG  
Spinnereistrasse 8-14  
CH-9008 St. Gallen  
Schweiz  
www.eaton.eu  
www.eaton.com

**Support**

Region North America  
Eaton Corporation  
Electrical Sector  
1111 Superior Ave.  
Cleveland, OH 44114  
United States  
877-ETN-CARE (877-386-2273)  
www.eaton.com

Andere Regionen  
Bitte kontaktieren Sie Ihren lokalen Lieferanten  
oder senden Sie eine E-Mail an:  
automation@eaton.com

**Originalbetriebsanleitung**

Die deutsche Ausführung dieses Dokuments ist die Originalbetriebsanleitung.

**Übersetzungen der Originalbetriebsanleitung**

Alle nicht deutschen Sprachausgaben dieses Dokuments sind Übersetzungen der Originalbetriebsanleitung.

**Redaktion**

Monika Jahn

**Marken- und Produktnamen**

Alle in diesem Dokument erwähnten Marken- und Produktnamen sind Warenzeichen oder eingetragene Warenzeichen der jeweiligen Titelinhaber.

**Copyright**

© Eaton Automation AG, CH-9008 St. Gallen

Alle Rechte, auch die der Übersetzung, vorbehalten.

Kein Teil dieses Dokuments darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder einem anderen Verfahren) ohne schriftliche Genehmigung der Firma Eaton Automation AG, St. Gallen reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Änderungen vorbehalten.

## Inhaltsverzeichnis

<b>1</b>	<b>Allgemein .....</b>	<b>3</b>
1.1	Zweck dieses Dokuments .....	3
1.2	Rückmeldungen zu diesem Dokuments.....	3
1.3	Auslieferungszustand der Geräte.....	3
<b>2</b>	<b>Zielsystemeinstellungen .....</b>	<b>5</b>
2.1	SPS-Zielsystem definieren.....	5
2.2	Geräte-Editor.....	7
2.2.1	Registerkarte [Kommunikationseinstellungen]: Kommunikation zwischen Steuerung und Programmiersystem konfigurieren .....	8
2.2.2	IP-Adresse einstellen .....	11
2.2.3	Registerkarte [Log]: «Logbuch» der Steuerung anzeigen .....	12
2.2.4	Registerkarte [SPS Shell]: Textbasierter Steuerungs-Monitor (Terminal) .....	13
2.2.5	Registerkarte [Firmware]: Zielsysteminstallation und Firmware-Update .....	14
2.2.6	XC-CPU202: Registerkarte [LOCAL_IO I/O Abbild].....	18
<b>3</b>	<b>Steuerungskonfiguration .....</b>	<b>19</b>
3.1	Allgemeines.....	19
3.2	Arbeiten in der Steuerungskonfiguration.....	20
3.2.1	Steuerungskonfigurationsbaum .....	20
3.2.2	Steuerungskonfigurationsbaum XC-CPU202.....	21
3.3	SmartWire-DT™ Konfiguration .....	26
3.3.1	SmartWire-DT™ Feldbus einrichten .....	27
3.3.2	Busdiagnose .....	31
3.3.3	Azyklische Kommunikation .....	31
3.3.4	Bedienungs- und Anzeigeelemente der SmartWire-DT™ Master Schnittstelle (Hardware).....	32
3.3.5	SmartWire-DT™ spezifische SPS-Shell Befehle .....	33
3.4	easyNET-Konfiguration .....	35
3.4.1	easyNET-Feldbus einrichten.....	36
3.4.2	Funktionen zur Datenübertragung .....	38
<b>4</b>	<b>Betrieb.....</b>	<b>39</b>
4.1	Einschaltverhalten.....	39
4.2	Ausschaltverhalten.....	39
4.3	Betriebszustände der Steuerung.....	39
4.3.1	XC-152: Übersicht Betriebs- und Fehlerzustände.....	40
4.3.2	XC-CPU202: Übersicht Betriebs- und Fehlerzustände .....	41
4.4	Betriebsartenumschaltung .....	42
4.5	Start-, Stop- und Reset-Verhalten.....	43
4.6	Test und Inbetriebnahme .....	45
4.7	Programmtransfer .....	46
4.8	Bootprojekt erzeugen .....	47

<b>5</b>	<b>Programmabarbeitung und Systemzeiten .....</b>	<b>49</b>
5.1	Programmabarbeitung .....	49
5.2	Taskkonfiguration .....	50
5.3	Task-Editor .....	51
5.4	Datenremanenz .....	55
5.5	Systembibliotheken, Funktionsbausteine und Funktionen .....	55
5.6	Prozessabbildverarbeitung / IO-Update .....	56
5.6.1	CAN-Bus .....	56
5.6.2	Profibus .....	57
5.6.3	SmartWire-DT™ .....	57
<b>6</b>	<b>Verbindungsaufbau Programmier-PC – Steuerung .....</b>	<b>59</b>
6.1	Verbindungsaufbau über Ethernet .....	59
6.2	Applikation auf die Steuerung laden .....	59
<b>7</b>	<b>Visualisierung .....</b>	<b>61</b>
7.1	Target- und Web-Visualisierung .....	61
7.2	Symbolkonfigurations-Editor .....	62
7.2.1	Symbolkonfiguration anlegen (z. B. für die Anbindung an eine Visualisierung) .....	62
<b>8</b>	<b>Lizenzierung .....</b>	<b>65</b>
8.1	SPS-Programmierungsumgebung .....	65
8.2	SPS-Laufzeitsystem .....	65
8.3	Target-Visualisierung .....	65
8.4	Web-Visualisierung .....	65

## 1 Allgemein

### 1.1 Zweck dieses Dokuments

Dieses Dokument beschreibt den Umgang mit der SPS-Programmierungsumgebung XSOFT-CODESYS-3 und dem SPS-Laufzeitsystem für folgende Geräte:

- Touch Panel XV100
- Touch Panel XV400
- Touch Panel XVS400
- Kompaktsteuerung XC-152
- Modularsteuerung XC-CPU202

Dieses Dokument dient als Ergänzung zu folgender Dokumentation der Firma 3S-Smart Software Solutions GmbH:

- Anwenderdokumentation CODESYS V3, Installation und Erste Schritte
- Online-Hilfe zum CODESYS V3 Programmiersystem



**Dialoge und Beispiele in diesem Dokument sind standardisiert. Je nach Auswahl des Steuerungstyps können sich deshalb vereinzelt Dialoge unterscheiden.**

### 1.2 Rückmeldungen zu diesem Dokuments

Bitte senden Sie Ihre Kommentare, Empfehlungen oder Anregungen zu diesem Dokument an [info-automation@eaton.com](mailto:info-automation@eaton.com).

### 1.3 Auslieferungszustand der Geräte

Folgende XV-Geräte werden ohne Betriebssystem ausgeliefert:

- XV100
- XV400
- XVS400

Folgende XC-Geräte werden mit der Betriebssystemversion V02 ausgeliefert:

- XC-152
- XC-CPU202



**Die auf dem Gerät installierte Betriebssystemversion muss zur Version der Programmieroberfläche XSOFT-CODESYS passen.**

Informationen zur Installation des Betriebssystems finden siehe Kapitel 2.2.5 Registerkarte [Firmware]: Zielsysteminstallation und Firmware-Update, 14.

## 1 Allgemein

### 1.3 Auslieferungszustand der Geräte

## 2 Zielsystemeinstellungen

### 2.1 SPS-Zielsystem definieren

Das SPS-Zielsystem (Zielgerät, Steuerung) wird definiert, wenn:

- ein neues Standardprojekt angelegt wird

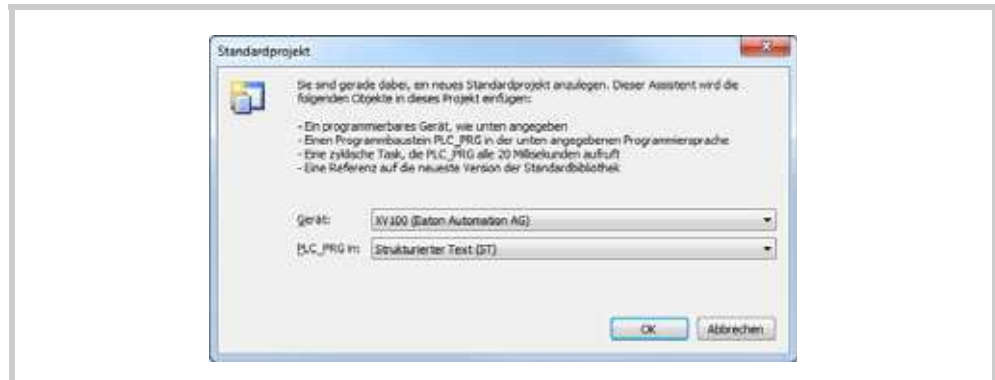


Abb. 1 Fenster «Standardprojekt»

## 2 Zielsystemeinstellungen

### 2.1 SPS-Zielsystem definieren

- oder mittels des Kontextmenüs zum symbolischen Wurzelknoten (Projektname) im Fenster «Geräte» oder «POUs», Befehl [Gerät anhängen].

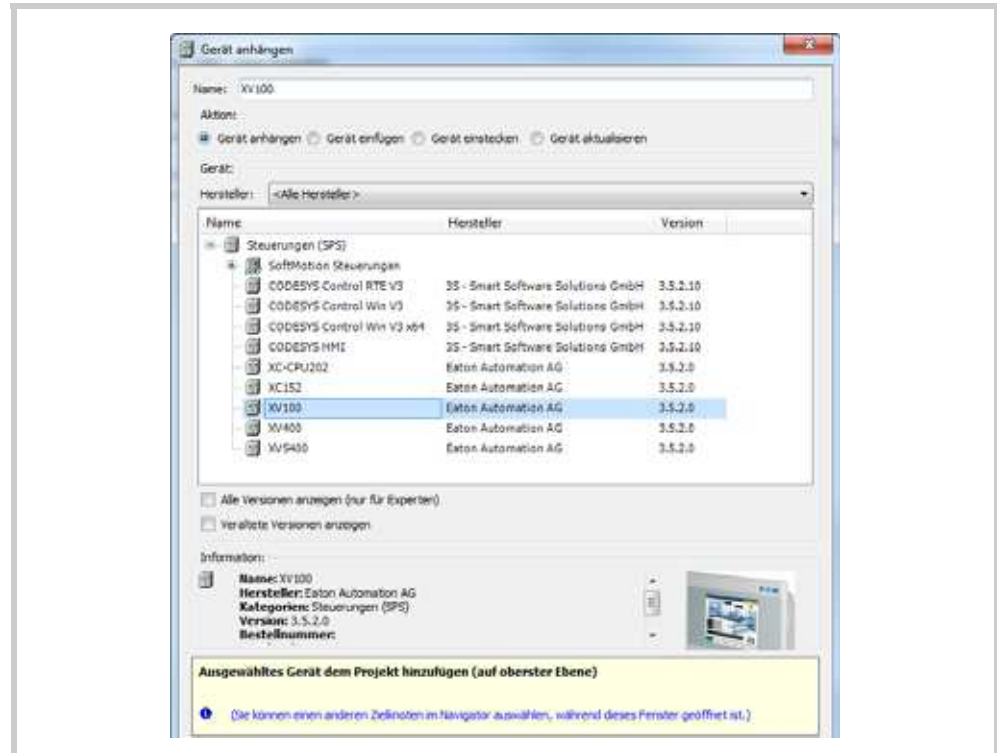


Abb. 2 Fenster «Gerät anhängen»

- ☞ **Um ein übersetzbares Projekt zu erhalten, müssen der Applikation der Steuerung noch folgende Objekte hinzugefügt werden:**
  - «POU» und
  - «Taskkonfiguration».**Innerhalb der Taskkonfiguration muss über die Schaltfläche [Aufruf hinzufügen] die POU als auszuführender Programmteil eingetragen werden.**



## 2.2

### Geräte-Editor

Der Geräte-Editor öffnet sich, wenn der Gerätenamen im Gerätebaum (Fenster «Geräte») doppelgeklickt wird.

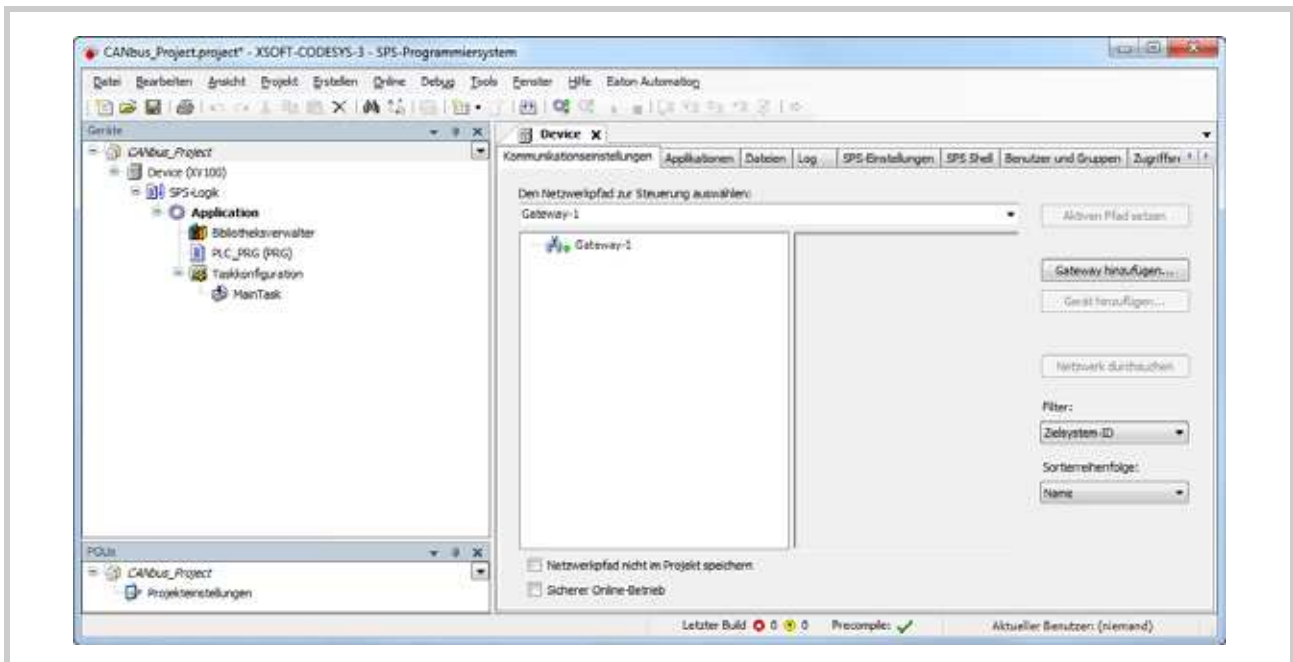


Abb. 3 Geräte-Editor

## 2 Zielseinstellungen

### 2.2 Geräte-Editor

#### 2.2.1

#### Registerkarte [Kommunikationseinstellungen]: Kommunikation zwischen Steuerung und Programmiersystem konfigurieren

Dieser Unterdialog des Geräte-Editors dient der Konfiguration der Parameter für die Kommunikation zwischen Steuerung und Programmiersystem.

#### Gateway-Parameter

Ist im linken Teil des Fensters (im Gateway- und Gerätebaum) ein Gateway markiert, werden im Teil rechts daneben die dazugehörigen Gateway-Parameter angezeigt.

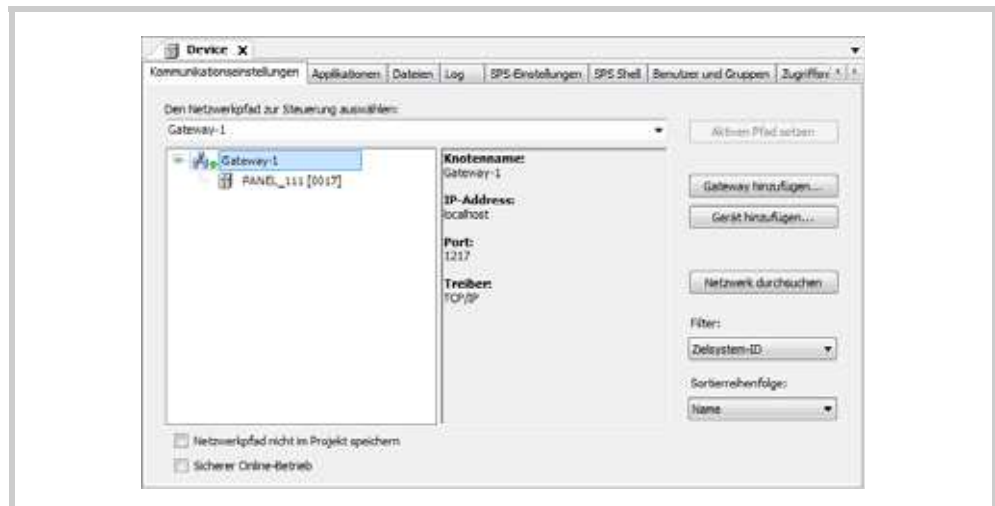


Abb. 4 Gateway-Parameter

#### Geräte-Parameter

Ist im linken Teil des Fensters (im Gateway- und Gerätebaum) ein Geräte-Eintrag markiert, werden im Teil rechts daneben die dazugehörigen Geräte-Parameter angezeigt.

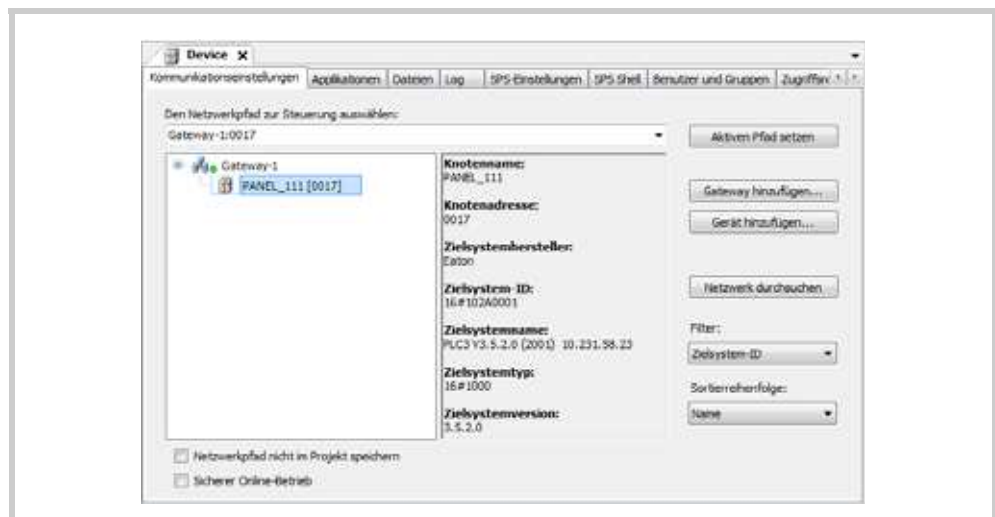




Abb. 5 Geräte-Parameter

**Bedienungs- und Anzeigeelemente**

Die Parameter für die Kommunikation zwischen Steuerung und Programmiersystem werden mit folgenden Bedienelementen konfiguriert:

- Schaltflächen
- Befehle aus dem Kontextmenü
- Kontrollkästchen

Element	Funktion
<b>Anzeigeelemente</b>	
	<p>Zeigt den Eintrag eines Geräts im Gateway- und Gerätebaum. Die Farbe mit der der Kreis im Symbol gefüllt ist zeigt den Status des Gateways an:</p> <ul style="list-style-type: none"> <li>■ Grüner Kreis: Das Gateway läuft korrekt.</li> <li>■ Roter Kreis: Das Gateway läuft <b>nicht</b> korrekt.</li> <li>■ Grauer Kreis: Das Gateway wurde noch nicht verbunden.</li> </ul> <p>Manche Kommunikationsprotokolle erlauben kein regelmässiges Prüfen des Gateways, so dass der Status nicht angezeigt werden</p>
	<p>Zeigt den Eintrag eines Geräts, das über das Gateway erreichbar ist. Diese Einträge werden lokal auf dem System und <b>nicht</b> im Projekt gespeichert. Einträge mit einer Target-ID, die sich von der gerade im Projekt konfigurierten unterscheiden, sind in grauer Schrift dargestellt. Eine aktualisierte Liste erhält man über die Schaltfläche [Netzwerk durchsuchen].</p>
<b>Schaltflächen und Befehle aus dem Kontextmenü</b>	
Aktiven Pfad setzen	Setzt den gerade ausgewählten Kommunikationskanal als den aktiven. Ein Doppelklick auf den Eintrag im Gateway- und Gerätebaum bewirkt dasselbe. Sämtliche Kommunikationszugriffe ausgehend von der aktiven Applikation im Programmiersystem adressieren ausschliesslich die unter dem aktiven Pfad ausgewählte Steuerung.
Gateway hinzufügen	Öffnet den Dialog «Gateway», wo Sie ein Gateway definieren können, der zur aktuellen Konfiguration hinzugefügt werden soll.
Gerät hinzufügen	Öffnet den Dialog «Gerät hinzufügen», wo manuell ein Gerät definiert werden kann, das unter dem gerade ausgewählten Gateway-Eintrag eingefügt werden soll. Diese Funktion ist insbesondere dann hilfreich, wenn ein Gerät mit bekannter/fester IP-Adresse adressiert werden soll.
Suche nach Gerät über Adresse	Durchsucht das Netzwerk nach Geräten mit einer eindeutigen Knotenadresse wie hier im Gateway- und Gerätebaum. Die gefundenen Geräte werden daraufhin unterhalb des Gateways mit der angegebenen Knotenadresse, ergänzt um ihren Namen, dargestellt. Die Suche bezieht sich immer auf die Geräte unterhalb desjenigen Gateways, der gerade ausgewählt ist bzw. unterhalb dem gerade ein Eintrag ausgewählt ist.
Suche nach Gerät über Name	Durchsucht das Netzwerk nach Geräten mit gleichen Namen wie hier im Gateway- und Gerätebaum (Gross-/Kleinschreibung bitte beachten!). Die gefundenen Geräte werden daraufhin unterhalb des Gateways mit dem angegebenen Namen, ergänzt um ihre eindeutige Knotenadresse, dargestellt. Die Suche bezieht sich immer auf die Geräte unterhalb desjenigen Gateways, der gerade ausgewählt ist bzw. unterhalb dem gerade ein Eintrag ausgewählt ist.

## 2 Zielseinstellungen

### 2.2 Geräte-Editor

Element	Funktion
Suche nach Gerät über IP-Adresse	Durchsucht das Netzwerk nach Geräten mit einer eindeutigen IP-Adresse wie hier im Gateway- und Gerätebaum. Die gefundenen Geräte werden daraufhin unterhalb des Gateways mit der angegebenen Knotenadresse, ergänzt um ihren Namen, dargestellt. Die Suche bezieht sich immer auf die Geräte unterhalb desjenigen Gateways, das gerade ausgewählt ist bzw. unterhalb dem gerade ein Eintrag ausgewählt ist.
Netzwerk durchsuchen	Startet eine Suche für verfügbare Geräte im lokalen Netzwerk. Der Konfigurationsbaum des betreffenden Gateways wird entsprechend aktualisiert.
«Filter»	Reduziert die Anzeige der Geräte im Gateway- und Gerätebaum: <ul style="list-style-type: none"><li>■ [Keine]: Alle Geräte werden angezeigt. Wobei Geräte mit nicht passender Zielsystem-ID über ein gegrautes Piktogramm kenntlich gemacht werden.</li><li>■ [Zielsystem-ID]: Es werden nur Geräte angezeigt, die die gleiche Zielsystem-ID haben, wie das aktuell im Projekt konfigurierte Gerät.</li></ul>
«Sortierreihenfolge»	Sortiert die Einträge im Gateway- und Gerätebaum nach: <ul style="list-style-type: none"><li>■ [Name] oder</li><li>■ [Knotenadresse]</li></ul>
Ausgewähltes Gerät löschen	Löscht das ausgewählte Gerät im Gateway- und Gerätebaum.
Gateway bearbeiten	Öffnet den Dialog «Gateway» zum Bearbeiten der Einstellungen für das gerade ausgewählte Gateway.
Zum lokalen Gateway verbinden	Öffnet den Dialog «Gateway-Konfiguration» zur Konfiguration eines lokalen Gateways und bietet daher eine Alternative zur manuellen Bearbeitung der Datei «Gateway.cfg».
<b>Kontrollkästchen</b>	
Netzwerkpfad nicht im Projekt speichern	Wenn diese Option aktiviert ist, wird die aktuelle Netzwerkpfad-Definition nicht im Projekt gespeichert, sondern in den lokalen Optionen-Einstellungen auf Ihrem Programmier-PC. Das bedeutet: <ul style="list-style-type: none"><li>■ Wenn das Projekt auf dem gleichen Programmier-PC wieder geöffnet wird, wird die Einstellung wieder hergestellt</li><li>■ Wird das Projekt auf einem anderen Programmier-PC verwendet, muss der aktive Pfad neu gesetzt werden.</li></ul>
Sicherer Online-Betrieb	Wenn diese Option aktiviert ist, wird der Benutzer aus Sicherheitsgründen beim Aufrufen der folgenden Online-Befehle jeweils zu einer Bestätigung aufgefordert: Werte forcen, Werte schreiben, mehrfaches Laden, Forceliste für <Applikation> aufheben, Einzelzyklus, Start <Applikation> und Stop <Applikation>.

Tab. 1 Bedienungs- und Anzeigeelemente



**Achten Sie darauf, dass die IP-Adressen des Programmier-PCs und der Steuerung der gleichen Adressenfamilie angehören.**

→ Dokument «MN0501009Z-DE Systembeschreibung Netzwerk in Kürze».

## 2.2.2

### IP-Adresse einstellen

Die Einstellung der IP-Adresse für XV-Geräte erfolgt am Display.

Für XC-Geräte erfolgt die Einstellung der IP-Adresse folgendermaßen:

- XC-152:  
Für das Abfragen und Ändern der IP-Adresse steht der «Remote Client» zur Verfügung.  
→ Dokument «MN05010008Z-DE Systembeschreibung Windows CE», Kapitel «Verbindungsaufbau mit XC-150».
- XC-CPU202:  
Die Modularsteuerung XC-CPU202 hat im Auslieferungszustand die IP-Adresse: 192.168.119.202.  
Zum Ändern der IP-Adresse steht der Shell-Befehl „setipconfig“ zur Verfügung.  
Zum Zurücksetzen der IP-Adresse und aller Parameter der XC-CPU202 in den Auslieferungszustand, muss die Betriebsspannung der Steuerung bei gedrücktem Resettaster zugeschaltet werden, siehe «MN05003001Z Modular PLC XC-CPU201, XC-CPU202», Kapitel „Reset Gerät“.
- Über SD-Karte und USB-Stick bei einem Firmware-Update, siehe Kapitel 2.2.5.2 Firmware-Update für XC-CPU202, 16.
- Die IP-Adresse kann ebenfalls über einen Bootprozess eingestellt werden.

## 2 Zielseiteinstellungen

### 2.2 Geräte-Editor

#### 2.2.3

#### Registerkarte [Log]: «Logbuch» der Steuerung anzeigen

Dieser Unterdialog des Geräte-Editors dient dem Anzeigen des «Logbuchs» der Steuerung, d.h. dem Anzeigen von Ereignissen, die auf der Steuerung aufgezeichnet wurden. Dies betrifft:

- Ereignisse beim Systemstart und -beenden (geladene Komponenten mit Version)
- Applikations-Download und Laden des Bootprojekts
- Kundenspezifische Einträge
- Log-Einträge von I/O-Treibern
- Log-Einträge vom DataServer

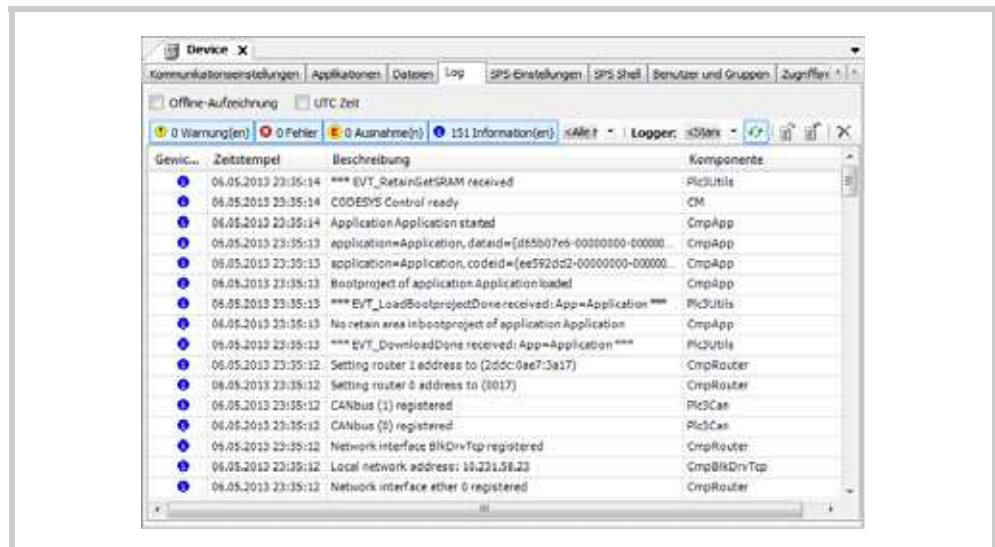


Abb. 6 Geräte-Editor, Registerkarte [Log]



Standardmäßig werden Log-Meldungen in einem flüchtigen Speicherbereich der Steuerung abgelegt. Das heißt, die Log-Meldungen gehen bei einem Spannungsverlust der Steuerung verloren.

2.2.4

**Registerkarte [SPS Shell]:  
Textbasierter Steuerungs-Monitor (Terminal)**

Bei der SPS-Shell handelt es sich um einen textbasierten Steuerungs-Monitor (Terminal). Kommandos zur Abfrage bestimmter Informationen aus der Steuerung werden in einer Eingabezeile eingegeben und als String an die Steuerung geschickt. Der zurückgelieferte Antwortstring wird in einem Ergebnisfenster des Browsers dargestellt. Diese Funktionalität dient Diagnose- und Debugging-Zwecken.

Die zur Verfügung stehenden Kommandos gliedern sich in zwei Gruppen:

- Standard Browser-Befehle
- Zielsystemspezifische Browser-Befehle

Diese Befehle werden in einer Datei verwaltet und sind im SPS-Laufzeitsystem implementiert.

Befehl	Beschreibung
?	Zeigt eine Liste der implementierten und zur Verfügung stehenden Kommandos

Tab. 2 Befehl für Kommandoübersicht

 **Um die SPS Shell Funktionalität zu nutzen, muss die gewünschte Steuerung als aktive Applikation markiert sein. Es wird eine Onlineverbindung zur Steuerung hergestellt.**



Abb. 7 Geräte-Editor, Registerkarte [SPS Shell]

## 2 Zielsystemeinstellungen

### 2.2 Geräte-Editor

#### 2.2.5

#### Registerkarte [Firmware]: Zielsysteminstallation und Firmware-Update

Mittels XSOFT-CODESYS-3 kann folgende Software auf die Steuerung oder eine Speicherkarte installiert werden:

SPS-Zielsystem (Steuerung)	SPS-Laufzeitsystem
XV100	muss installiert werden
XV400	muss installiert werden
XVS400	muss installiert werden
XC-152	Firmware-Update muss durchgeführt werden
XC-CPU202	auf der Steuerung ist ein CODESYS-V02 Runtime-System (RTS) vorinstalliert. Das V03 RTS muss nachinstalliert werden.

Tab. 3 Übersicht Firmware

#### 2.2.5.1

#### Firmware-Update für XV-Geräte und XC-152

Die Aktualisierung des Betriebssystems der XV-Geräte und des XC152 erfolgt innerhalb des XSOFT-CODESYS-3 Programmiersystems. Sobald ein Standardprojekt für das Zielsystem angelegt ist, erscheint das Gerät als „Device“ im Steuerungskonfigurationsbaum. Durch Doppelklick auf „Device“ erscheint ein Konfigurationsfenster. Über die Registerkarte „Firmware“, lässt sich das Update starten.

 **Dateien, die nicht zur aktuellen Installation gehören, können entfernt werden.**

Die Betriebssysteminstallation bzw. ein Firmware-Update wird in der SPS-Programmierungsumgebung im Geräte-Editor, Registerkarte [Firmware], durch Klicken von [Start] aufgerufen.

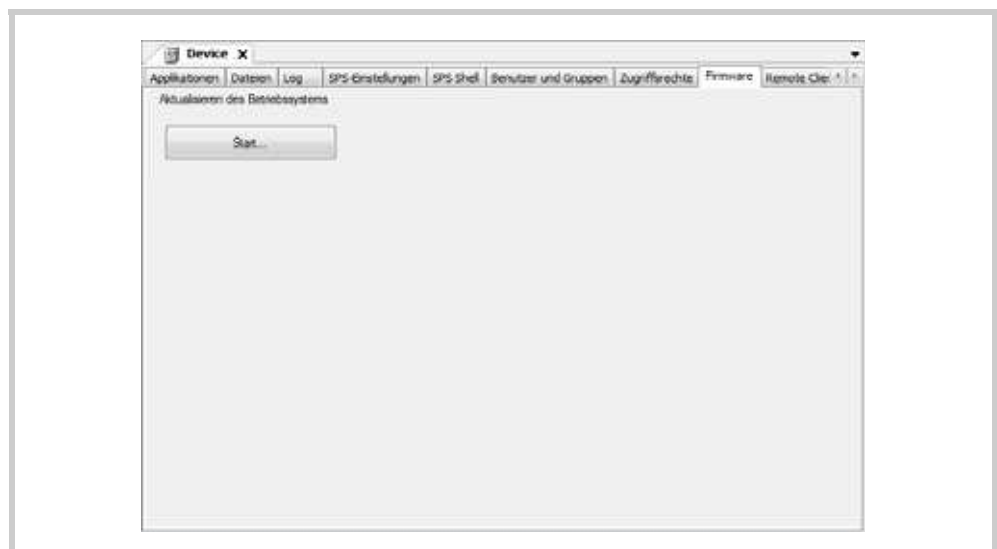


Abb. 8 Geräte-Editor, Registerkarte [Firmware]



- ☞ Ist der Firmwaredownload-Expertenmodus aktiviert ([Tools] > [Optionen], Kategorie «Firmwaredownload»), kann die Firmware-Datei nach dem Klicken von [Start] in einem Dialogfenster ausgewählt werden.

Im Setup-Dialog, wählen Sie den gewünschten Installationstyp.



Abb. 9 Fenster «Setup-Firmware, Installationstyp»

- FTP Installation:  
Die Installation der Verzeichnisse **PicRts** und **PicPrg** erfolgt über FTP.
  - Installation auf einen Wechseldatenträger:  
Der Wechseldatenträger (z.B. CompactFlash™) muss am Programmier-PC über einen entsprechenden Adapter (z.B. PC-Card-Adapter) verfügbar sein. Die Installation der Verzeichnisse **PicRts** und **PicPrg** erfolgt nun direkt auf den Wechseldatenträger
  - Installation in ein lokales Verzeichnis:  
Die Installation der Verzeichnisse **PicRts** und **PicPrg** erfolgt auf ein lokales Verzeichnis des Programmier-PCs. Anschliessend müssen die gesamten Verzeichnisse **PicRts** und **PicPrg** manuell ins Hauptverzeichnis des Wechseldatenträgers (z.B. CompactFlashes™) kopiert werden.
- ☞ Das Firmware-Update ist erst nach einem Bootprozess aktiviert.
- ☞ Das gespeicherte SPS-Programm bleibt bei einem Firmware-Update erhalten.
- ☞ Die Verzeichnisse **PicRts** und **PicPrg** werden bei einer Zielsysteminstallation bzw. einem Firmware-Update nicht gelöscht. Vorhandene Dateien werden lediglich überschrieben.

## 2 Zielseinstellungen

### 2.2 Geräte-Editor

#### 2.2.5.2

#### Firmware-Update für XC-CPU202

Um mit XSOFT-CODESYS-3 eine Verbindung zu einem Gerät aufbauen zu können und das erstellte Anwenderprogramm starten zu können, muss auf dem Gerät die zugehörige Betriebssystemversion V03 installiert sein.

Dateien zur Installation und zur Aktualisierung des Betriebssystems sind Bestandteil des Zielsystems des zu aktualisierenden Gerätes.

Ein Wechsel der Betriebssystemversion erfolgt zur Zeit über ein Speichermedium, wie z.B. USB-Stick, MMC-Karte oder SD-Karte.



**Die auf dem Gerät installierte Betriebssystemversion muss zur Version der Programmoberfläche XSOFT-CODESYS passen.**

Netzwerkeinstellungen im Auslieferungszustand sind:

IP-Adresse: 192.168.119.202

Netzwerkmaske: 255.255.255.0

#### Von Version V02 nach V03



**Bei der Durchführung eines Updates mittels Speichermedium wird das Betriebssystem nur dann kopiert, wenn es sich von dem bereits installierten unterscheidet. Hierbei werden dann auch die Netzwerkeinstellungen, z.B. die IP-Adresse und Netzwerkmaske, zurückgesetzt.**

- ▶ Speichermedium formatieren; Formatiermodus FAT-Filesystem.
- ▶ Verzeichnisstruktur "CONTROL/XC-CPU202" auf dem Speichermedium anlegen.
- ▶ Betriebssystemdatei „btsxc202\_v03...bin“ in den Ordner XC-CPU202 kopieren.

Das aktuelle Betriebssystem ist in folgendem Verzeichnis abgelegt:

„C:\ProgramData\XSOFT-CODESYS-3\Firmware\4096\102A 0202\3.5.2.0“.

- ▶ Speichermedium auf das SPS-Zielsystem aufstecken.
- ▶ SPS-Zielsystem mit gestecktem Speichermedium neu starten.
- ▶ SPS-Zielsystem NICHT abschalten!  
Die Aktualisierung des Betriebssystems kann bis zu 15 Minuten benötigen. Während des Updatevorgangs blinkt die RUN/STOP-LED grün und die SF-LED rot.

Das SPS-Zielsystem wurde erfolgreich aktualisiert und ist über XSOFT-CODESYS3 ansprechbar. Die RUN/STOP-LED blinkt grün, die SF-LED leuchtet rot<sup>1)</sup>(4.3.2, 41).



**Die Verwendung des V2-Update-Tools ist zur Zeit im aktuellen Codesys-V3 Installer nicht möglich, Bei Aufruf von „XC202Loader.exe“ wird dementsprechend eine Fehlermeldung angezeigt.**

<sup>1</sup> Ist im Gerät eine Batterie gesteckt, leuchtet die SF-LED nicht rot; dann ist sie aus.

### 2.2.5.3

#### Betriebssystemversion V03 nach V02

- ▶ Speichermedium formatieren; Formatiermodus FAT-Filesystem..
- ▶ Verzeichnisstruktur "CONTROL/XC-CPU202" auf dem Speichermedium anlegen.
- ▶ Betriebssystemdatei „btsxc202\_v01...bin“ in den Ordner „XC-CPU202“ kopieren.

Das aktuelle Betriebssystem ist in folgendem Verzeichnis abgelegt:

„<Programm-Verzeichnis>\Common Files\CAA-Targets\Eaton Automation\V2.3.9 SP3\Firmware\XC-202“.

- ▶ Speichermedium auf das SPS-Zielsystem aufstecken.
- ▶ SPS mit gestecktem Speichermedium neu starten.
- ▶ SPS-Zielsystem NICHT abschalten!  
Das Firmware-Update kann bis zu 15 Minuten benötigen. Während des Updatevorgangs blinkt die RUN/STOP-LED grün und die SF-LED rot.
- ▶ Sind beide LEDs aus, kann das Speichermedium gezogen und das SPS-Zielsystem neu gestartet werden.

Das SPS-Zielsystem wurde erfolgreich aktualisiert und ist über XSOFT-CODESYS2 ansprechbar. Die RUN/STOP-LED blinkt grün, die SF-LED leuchtet rot<sup>1)</sup> (4.3.2, 41).

1 Ist im Gerät eine Batterie gesteckt, leuchtet die SF-LED nicht rot; dann ist sie aus.

## 2 Zielsystemeinstellungen

### 2.2 Geräte-Editor

#### 2.2.6

#### XC-CPU202: Registerkarte [LOCAL\_IO I/O Abbild]

Diese Registerkarte dient der Darstellung der lokalen Ein-/Ausgänge der CPU-Baugruppe. Zusätzlich können vorhandene Variablen auf die Ein-/Ausgänge gemappt bzw. neue Variablen erzeugt werden.

Im ONLINE-Zustand wird in einer zusätzlichen Spalte der aktuelle Wert der Ein-/Ausgänge angezeigt.

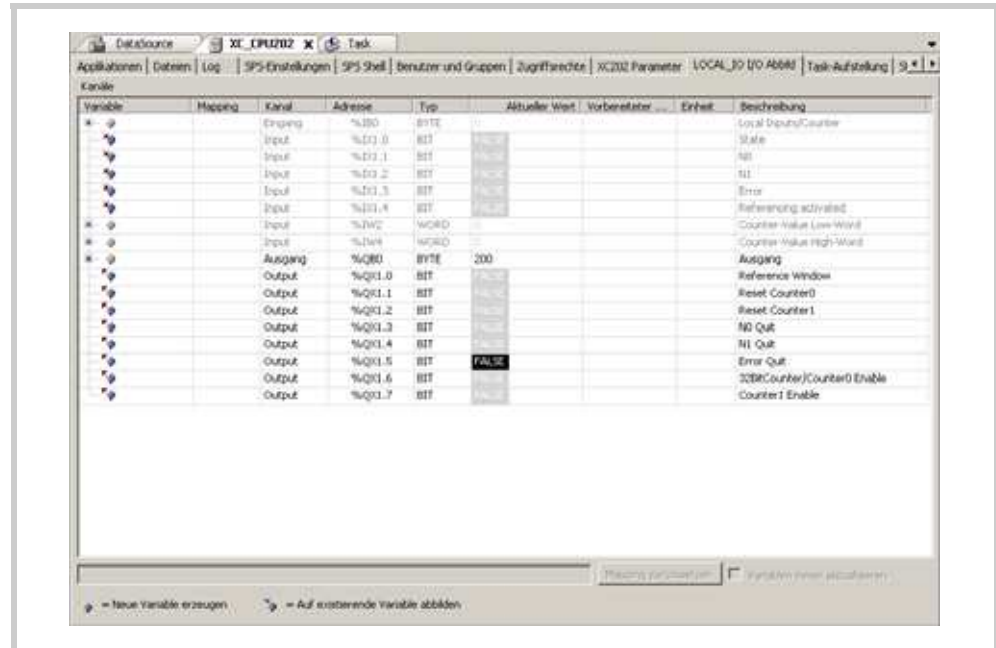


Abb. 10 Fenster «Setup-Firmware, Installationstyp»



Die detaillierte Funktionsweise und Beschreibung der lokalen Ein-/Ausgänge entnehmen Sie bitte dem Dokument "MN05003001Z\_DE.pdf", Kapitel 10.

## 3 Steuerungskonfiguration

### 3.1 Allgemeines

Die «Steuerungskonfiguration» bildet die Ziel-Hardware im Programmiersystem ab, um die Ein- und Ausgänge und die Parameter der Steuerung und der Feldbusgeräte der Applikation zugänglich zu machen. Ausserdem ermöglicht sie das Anzeigen der verfügbaren Geräte-Parameter.

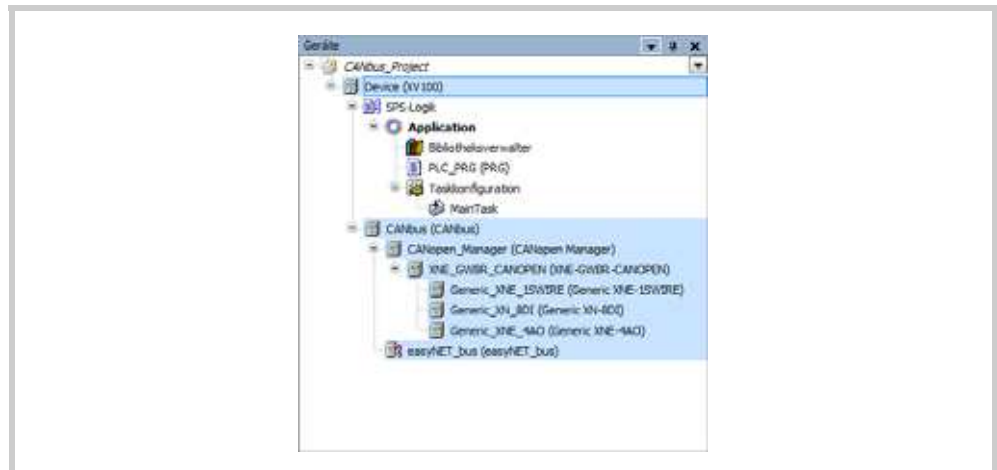


Abb. 11 Beispiel eines Steuerungskonfigurationsbaums

Der Baum der «Steuerungskonfiguration» ist im Gerätebaum integriert, in dem auch die anderen Objekte arrangiert sind, die nötig sind, um eine Applikation auf einer Steuerung laufen zu lassen. Das Abbilden der aktuellen Hardware-Konfiguration im Gerätebaum wird in den Standard-Geräte-Editoren durch eine Scan-Funktionalität erleichtert.

Die Zuweisung der Ein- und Ausgänge der Steuerung zu Projektvariablen geschieht entweder mit Hilfe der «AT Deklaration» im Deklarationseditor oder im «I/O-Abbild»-Dialog des Geräte-Editors, der die Dialoge zur Konfiguration eines Geräts bereitstellt. Wenn unterhalb eines auf oberer Ebene des Gerätebaums positionierten Geräts mehrere Applikationen vorliegen, kann im I/O-Abbild (Mapping) Dialog dieses Geräts festgelegt werden, welche Applikation für das Mapping herangezogen werden soll. Wenn eine Steuerung neu in den Gerätebaum eingefügt wird, wird standardmässig gleich die automatisch mit eingefügte Applikation als «Mapping-Applikation» eingetragen.

## 3 Steuerungskonfiguration

### 3.2 Arbeiten in der Steuerungskonfiguration

## 3.2 Arbeiten in der Steuerungskonfiguration

### 3.2.1 Steuerungskonfigurationsbaum

Im Steuerungskonfigurationsbaum wird unter dem SPS-Zielsystem (Steuerung) die weitere Ziel-Hardware hierarchisch mittels des Befehls [Gerät anhängen] eingefügt. Das Fenster «Gerät anhängen» öffnet sich, wenn im Gerätebaum (Fenster «Geräte») mittels rechter Maustaste das Kontextmenü zu einem Gerät aufrufen und im Kontextmenü [Gerät anhängen] geklickt wird. Welche Objekttypen eingefügt werden können, hängt von der gerade ausgewählten Position im Gerätebaum ab. Beispiel: Module für einen DP Profibus Slave können nicht eingefügt werden, ohne das entsprechende Slave-Objekt vorher eingefügt zu haben. Weiterhin stehen immer nur solche Geräte zum Einfügen zur Auswahl, die korrekt auf dem lokalen System installiert wurden und zur gerade ausgewählten Position im Gerätebaum passen.

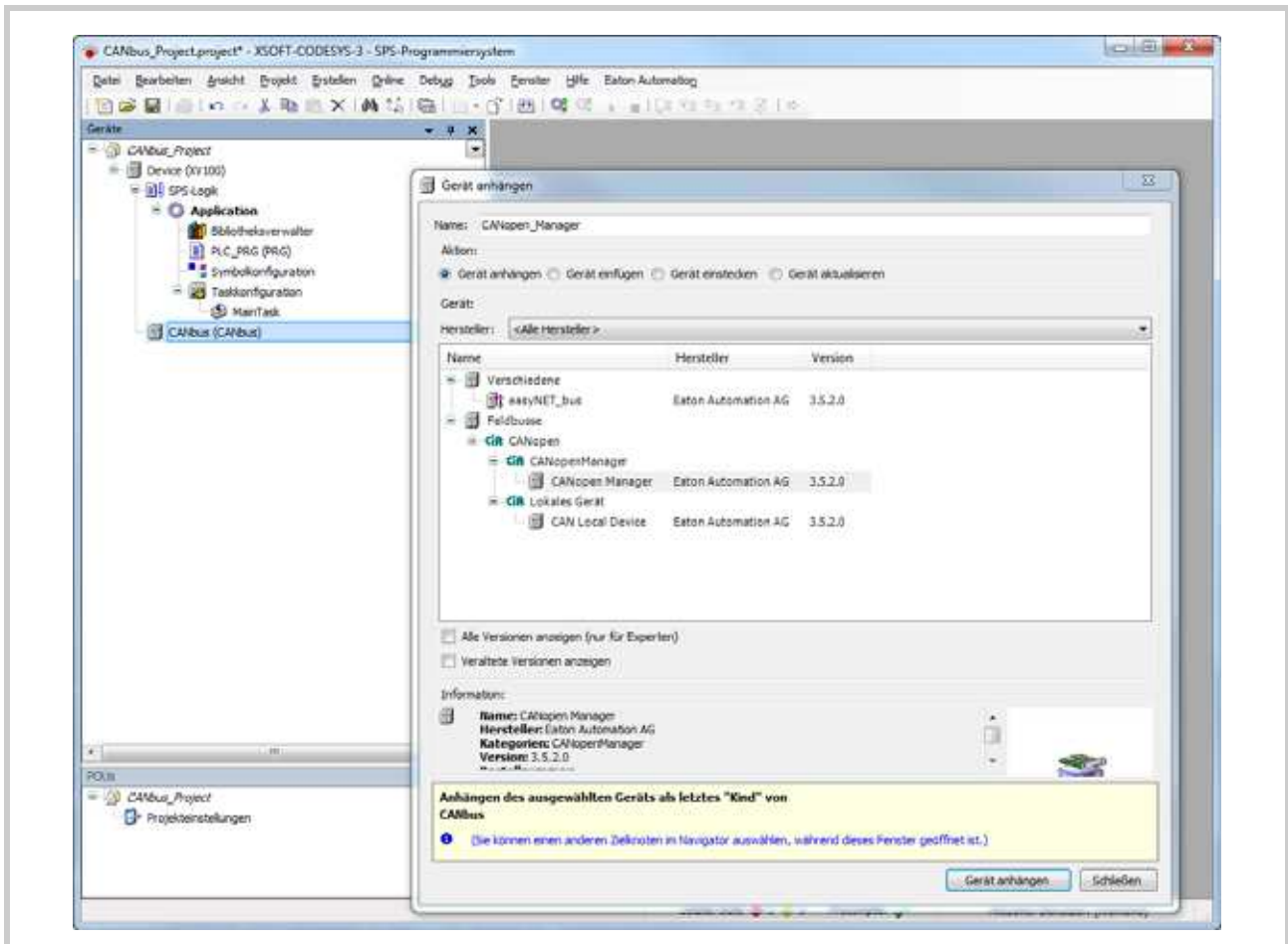


Abb. 12 Gerät anhängen

3.2.2

Steuerungskonfigurationsbaum XC-CPU202

Der Konfigurationsbaum der XC-CPU202 ist standardmäßig um 15 leere XIOC-Module erweitert. Ein XIOC-Modul wird gesetzt, indem das jeweilige Modul markiert wird und mittels rechter Maustaste das Kontextmenü "Gerät setzen..." aufgerufen wird.

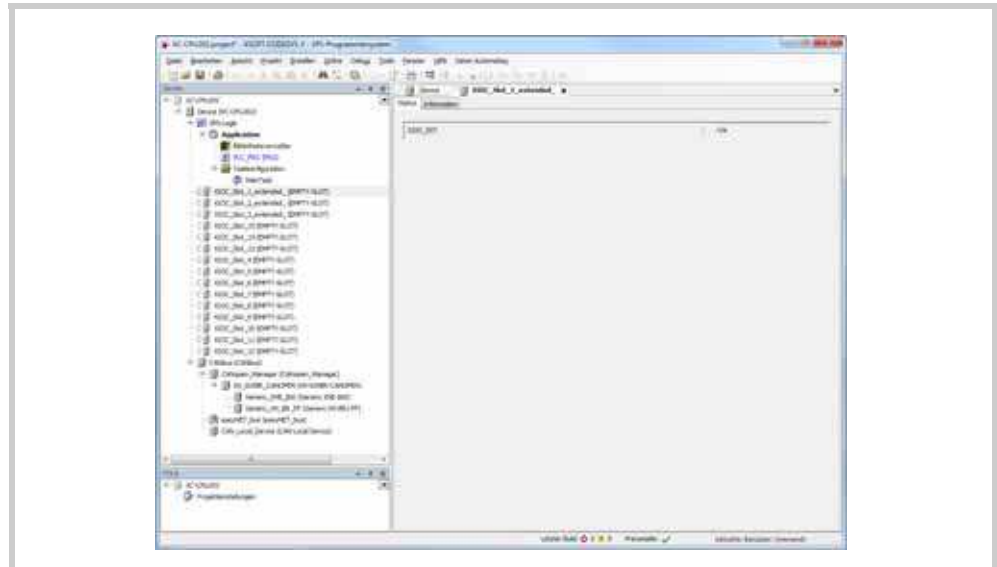


Abb. 13 Steuerungskonfigurationsbaum XC-CPU202

Das entsprechende Modul kann dann ausgewählt werden. Welche Module eingefügt werden können, hängt von der gerade ausgewählten SLOT-Nummer ab. Der Name des XIOC-Moduls kann zusätzlich editiert werden.

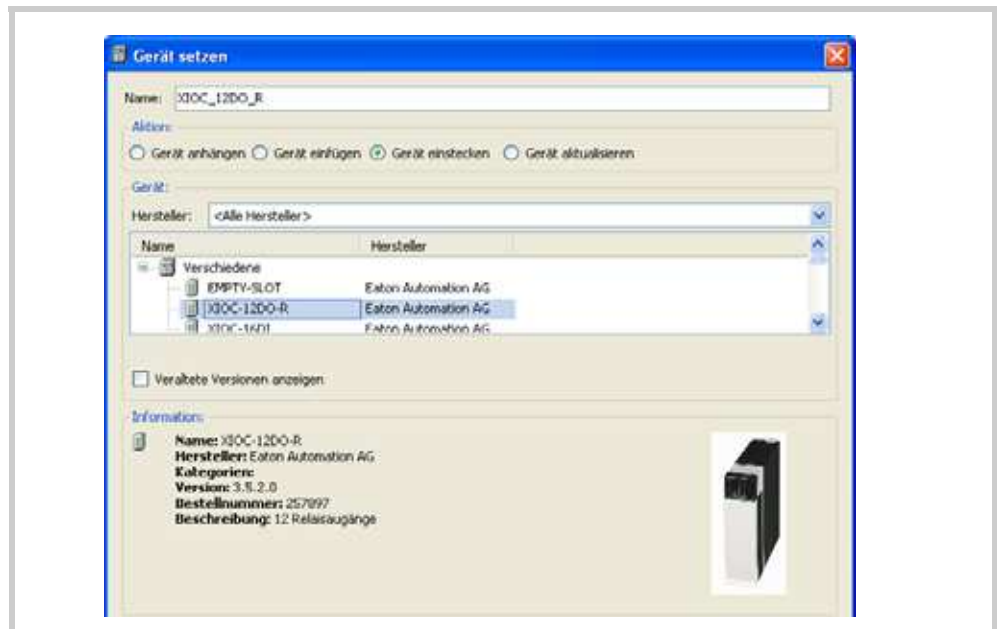


Abb. 14 Fenster Gerät setzen

## 3 Steuerungskonfiguration

### 3.2 Arbeiten in der Steuerungskonfiguration

#### 3.2.3 Feldbus-Konfigurationseditor

Die einzelnen Dialoge des Feldbus-Konfigurationseditors öffnen sich, wenn die entsprechende Knoten im Steuerungskonfigurationsbaum (Fenster «Geräte») doppelgeklickt werden. Jeder dieser Knoten hat einen eigenen Dialog. Die Dialoge enthalten wiederum mehrere Registerkarten, die Informationen und Konfigurationsparameter enthalten, um den Feldbus zu konfigurieren.

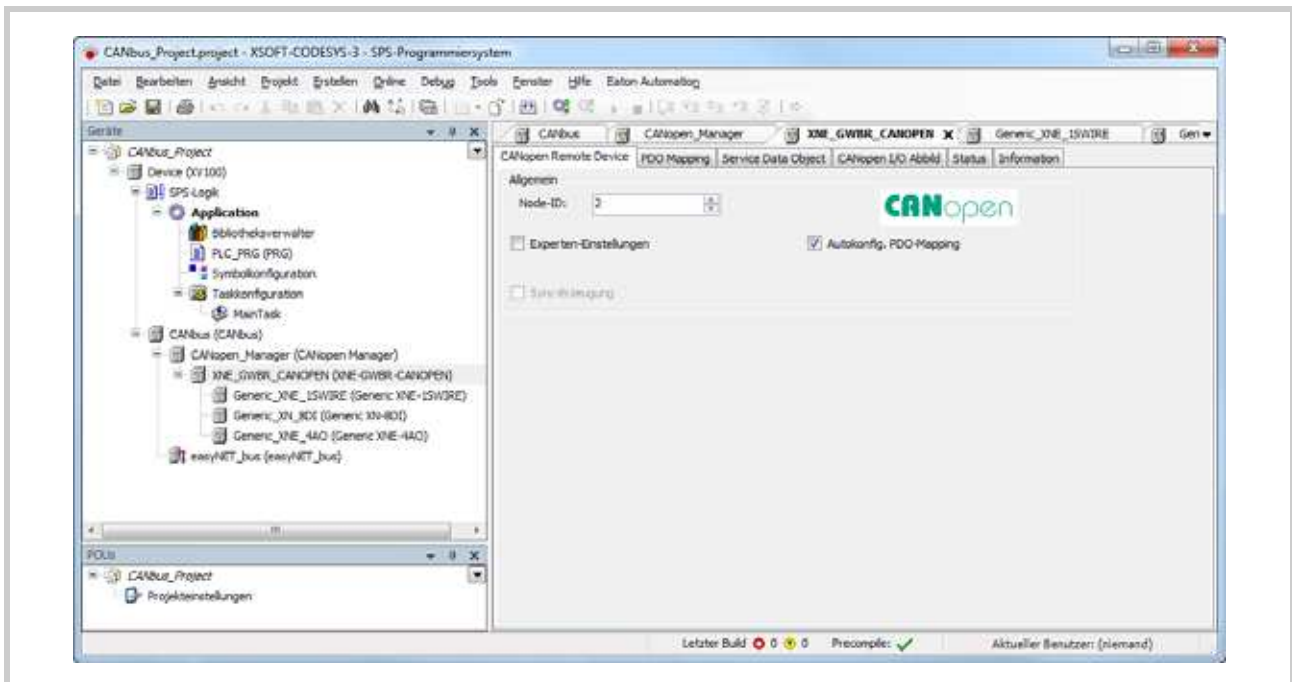


Abb. 15 Dialoge des Feldbus-Konfigurationseditors



3.2.3.1

**Registerkarte [... I/O Abbild]:  
I/O-Abbild konfigurieren**

Dieser Unterdialog des Geräte-Editors ist betitelt mit «<Gerätetyp> I/O Abbild» (z.B. «Profibus DP I/O Abbild»). Er dient dem Konfigurieren eines I/O Abbilds der Steuerung («I/O mapping»). Das heisst, den Eingangs-, Ausgangs- und Speicheradressen des Steuerungsgeräts werden Projektvariablen zugewiesen, die von der Applikation verwendet werden.

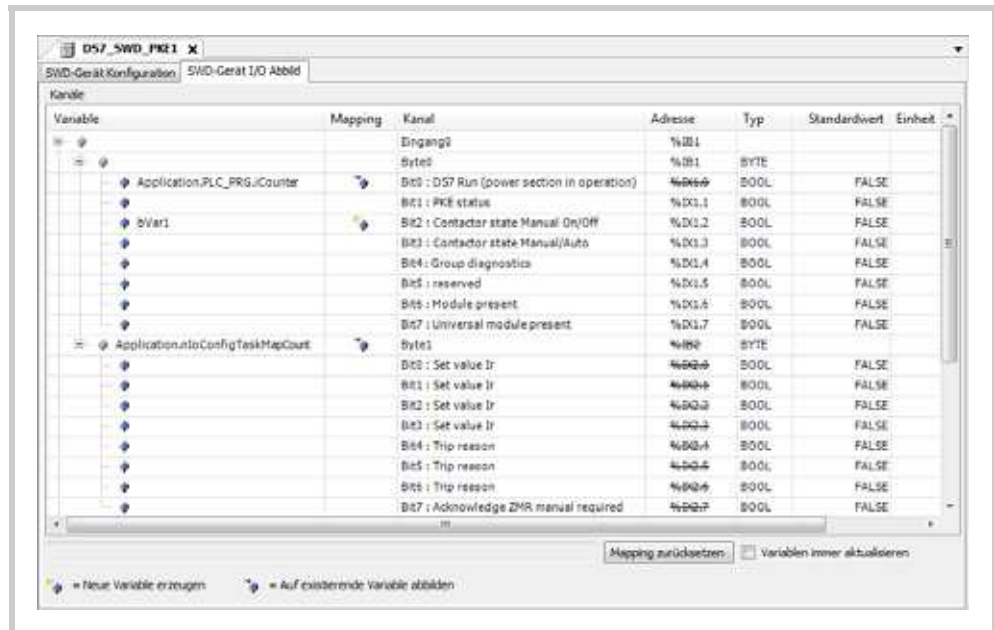






Abb. 16 Konfigurationseditor-Dialog, Registerkarte [... I/O Abbild]

**Bereich Kanäle**

Spalte	Beschreibung
Variable	<p>Projektvariable einem Ein- oder Ausgang zuweisen:</p> <ul style="list-style-type: none"> <li>■ Mapping einer strukturierten Variable: Variablen können Bit-Kanal-Einträgen oder übergeordneten Einträgen in der Variablenbaumstruktur zugeordnet werden.</li> <li>■ Abbilden auf eine bestehende Variable: Beim Abbilden auf eine bestehende Variable muss der komplette Pfad eingegeben werden, d.h. «&lt;Applikationsname&gt;.&lt;Bausteinname&gt;.&lt;Variablenname&gt;»; Beispiel: app1.plc_prg.ivar. Dazu wird am besten die Eingabehilfe über Schaltfläche  geöffnet. In der «Mapping» Spalte erscheint  und der Adresswert wird durchgestrichen dargestellt.</li> <li>■ Definieren einer neuen Variable: Wenn eine neue Variable definiert wird, muss nur ein Variablenname eingegeben werden; Beispiel: «bVar1». In diesem Fall wird in der «Mapping»-Spalte  eingefügt und die Variable automatisch intern als globale Variable deklariert.</li> </ul>

### 3 Steuerungskonfiguration

#### 3.2 Arbeiten in der Steuerungskonfiguration

Spalte	Beschreibung
Mapping	<p>Zeigt, ob auf eine bestehende Variable zugegriffen wird oder ob eine neue Variable definiert wird.</p> <ul style="list-style-type: none"> <li>■  Neue Variable erzeugen (die Variable wird automatisch intern als globale Variable deklariert)</li> <li>■  Auf existierende Variable abbilden</li> </ul>
Kanal	Symbolischer Name des Kanals
Adresse	<p>Adresse des Kanals</p> <ul style="list-style-type: none"> <li>■ Fixieren eines Adresswertes: Der aktuell angezeigte Adresswert der Ein- und Ausgänge (Kanal: Eingang# und Ausgang#) kann hier geändert und fixiert werden. Diese Funktion kann nützlich sein, um die Adressierung an eine vorgegebene Maschinenkonfiguration anzupassen bzw. um den Adresswert auch dann beizubehalten, wenn sich die Anordnung der Module ändert. Dies würde standardmässig zu einer automatischen Anpassung der Adresswerte führen. <ul style="list-style-type: none"> <li>- Fixierte Adresswerte werden mit  gekennzeichnet</li> <li>- Durch Löschen der Adressangabe, wird der fixierte Adresswert zurückgesetzt,</li> </ul> </li> <li>■ Durchgestrichener Adresswert: Wird in der Spalte «Mapping»  angezeigt, ist der Adresswert durchgestrichen. Dies heisst nicht, dass es diese Speicheradresse nicht mehr gibt. Sie wird nur nicht direkt verwendet, weil der Wert einer bestehenden Variablen auf einem anderen Speicherplatz verwaltet wird und es sollte, speziell bei Ausgängen, auf keinen Fall eine andere bereits existierende Variable auf diese Adresse gespeichert werden («%Qxx» im I/O-Abbild), um Uneindeutigkeiten beim Schreiben der Werte zu vermeiden.</li> </ul>
Typ	Datentyp des Kanals
Standardwert	Standardwert des auf dem Kanal anliegenden Parameters
Aktueller Wert	Wird nur im Online-Modus angezeigt: Aktueller Wert des auf dem Kanal anliegenden Parameters
Vorbereiteter Wert	Wird nur im Online-Modus angezeigt: Vorbereiten eines Wertes für eine zu forcende Variable
Einheit	Einheit für den Parameterwert
Beschreibung	Kurze Beschreibung des Parameters

Tab. 4 Konfigurationseditor-Dialog, Registerkarte [... I/O Abbild], Kanäle

#### IEC Objekte

Dieser Teil des Dialogs erscheint nur, wenn implizit eine Instanz des Geräteobjekts angelegt wird, auf die in der Applikation zugegriffen werden kann (beispielsweise, um einen Bus neu zu starten oder Informationen abzufragen). Ob eine solche Instanz zur Verfügung steht und wie sie verwendet werden kann, hängt vom Gerätetyp ab und wird auf den entsprechenden Hilfeseiten für die spezielle Gerätekonfiguration beschrieben.

#### **Buszyklus-Optionen**

Diese Einstellungen sind für Geräte verfügbar, bei denen vor und nach dem Lesen von Ein- oder Ausgängen zyklische Aufrufe erfolgen. Sie erlauben eine busspezifische Task zu definieren.

Standardmäßig gilt die Buszyklus-Einstellung des übergeordneten Busgeräts (Zykluseinstellungen des übergeordneten Busses verwenden), das heisst der Gerätebaum wird nach oben auf die nächste gültige Buszyklus-Taskdefinition hin durchsucht.

Um die spezifische Buszyklus-Task zuzuweisen, wählen Sie die gewünschte aus der Auswahlliste. Die Liste enthält alle Tasks, die aktuell in der Taskkonfiguration der Applikation definiert sind.

### 3.3 SmartWire-DT™ Konfiguration

Die SPS-Programmierungsumgebung unterstützt die Konfiguration und den Betrieb eines SmartWire-DT Strangs.



**Relevante Dokumentation zum Thema SmartWire-DT:**

- Eine allgemeine Beschreibung des SmartWire-DT Systems finden Sie im Dokument «MN05006002Z-DE Handbuch SmartWire-DT Das System»
- Funktionsbeschreibungen sowie Angaben zum Umfang der Ein-/Ausgangsdaten der einzelnen SmartWire-DT Teilnehmer finden Sie im Dokument «MN05006001Z-DE Handbuch SmartWire-DT Teilnehmer».

#### Systemvoraussetzung

SPS-Zielsystem (Steuerung)	Version (SPS-Zielsystem)	Betriebssystem (OS)
XV100	ab V 3.5.2.0	Windows CE 5.0 ab Image Release 2.26.0 (xxx)
XC-152	ab V 3.5.2.0	Windows CE 5.0 ab Image Release 2.26.0 (xxx)

Tab. 5 Systemvoraussetzungen



- In der Steuerungskonfiguration definierte Nachrichten, deren Ein- bzw. Ausgänge im SPS-Programm nicht verwendet werden, werden standardmässig im Prozessabbild nicht aktualisiert.
- Die Updatezeit von SmartWire-DT IOs ist abhängig von der Gesamtanzahl an SmartWire-DT IOs und der verwendeten Baudrate.
- Mindestens eine Variable der SmartWire-DT Teilnehmer muss im SPS-Programm Verwendung finden, damit der SmartWire-DT Strang in Betrieb genommen wird.

3.3.1

SmartWire-DT™ Feldbus einrichten

- 1 Bauen Sie den Steuerungskonfigurationsbaum für den SmartWire-DT Feldbus auf:
  - 1.1 Hängen Sie unter dem SPS-Zielsystem (Steuerung) mittels des Befehls [Gerät anhängen] den SmartWire-DT Master «SWD-Master» ein.
  - 1.2 Hängen Sie unter «SWD-Master» die SmartWire-DT Teilnehmer an.
    - Die SmartWire-DT Teilnehmer müssen in der selben Reihenfolge im Steuerungskonfigurationsbaum positioniert sein, wie sie physisch im SmartWire-DT Strang angeordnet sind.
    - Insgesamt können 99 SmartWire-DT Teilnehmer an einen SmartWire-DT Strang angeschlossen werden.
    - Die Ausdehnung des SmartWire-DT Strangs kann bis zu 600 m betragen.

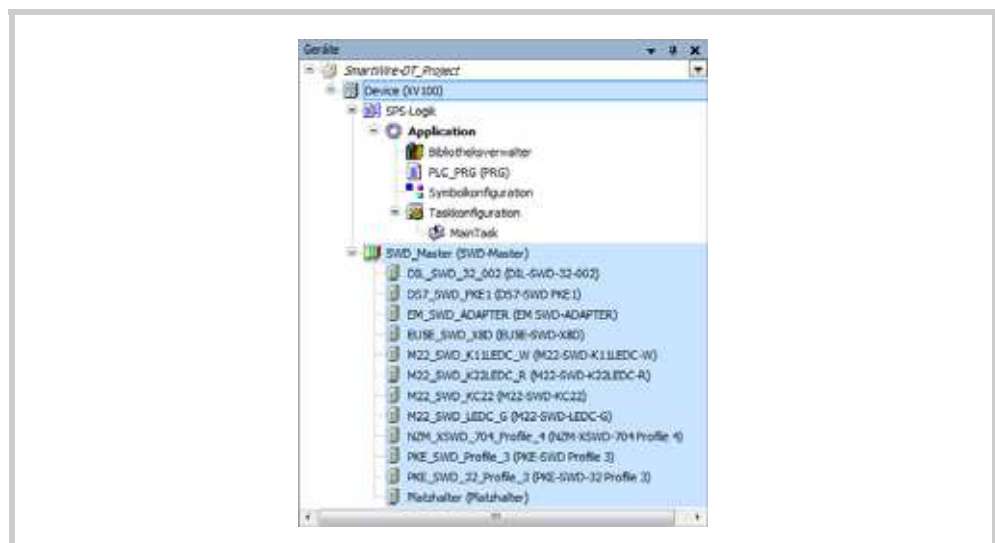


Abb. 17 Steuerungskonfigurationsbaum

### 3 Steuerungskonfiguration

#### 3.3 SmartWire-DT™ Konfiguration

- 2 Konfigurieren Sie den Feldbus mittels der Dialoge des Feldbus-Konfigurationseditors.
  - Konfigurationseditor-Dialog SWD-Master

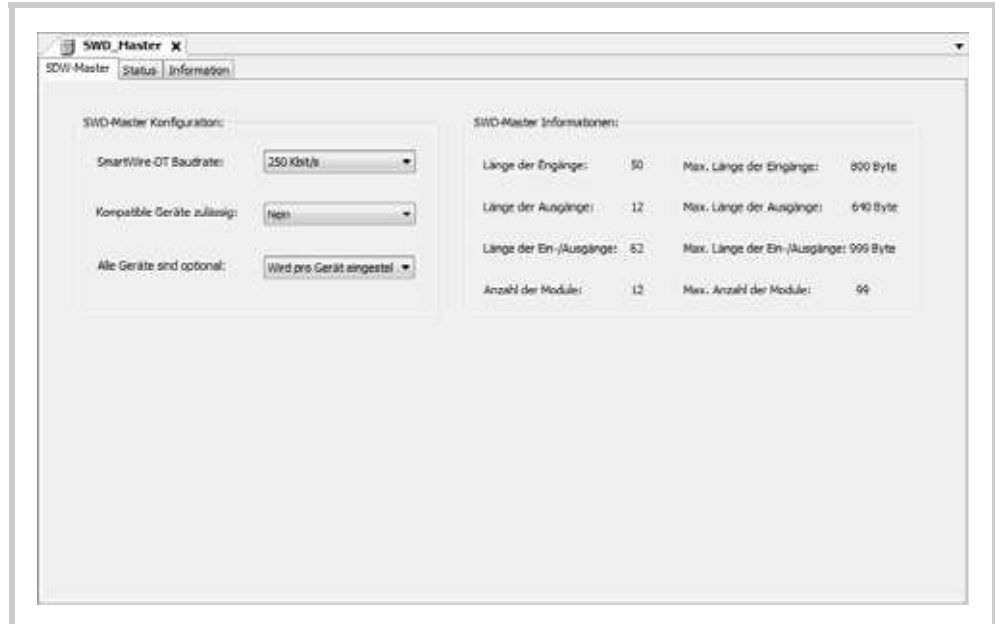


Abb. 18 Konfigurationseditor-Dialog SWD-Master

Konfigurationsparameter	Einstellung
SmartWire-DT Baudrate	<ul style="list-style-type: none"> <li>■ 125 Kbit/s</li> <li>■ 250 Kbit/s (Standardeinstellung)</li> </ul>
Kompatible Geräte zulässig	<ul style="list-style-type: none"> <li>■ Nein (Standardeinstellung)</li> <li>■ Ja</li> </ul> <p>Im SmartWire-DT Strang wird anstelle des projektierten Teilnehmers auch ein zum projektierten Teilnehmer kompatibler Teilnehmer akzeptiert. Als kompatible Teilnehmer werden nur Teilnehmer mit identischem «Family-Code» und grösserem «Device Code» angenommen.</p> <p>Beispiel:</p> <ul style="list-style-type: none"> <li>- M22-SWD-K11LED-W (projektiertes Teilnehmer)</li> <li>- M22-SWD-K22LED-W (kompatibler Teilnehmer)</li> </ul> <p>M22-SWD-K22LED-W wird anstelle von M22-SWD-K11LED-W akzeptiert.</p>

Konfigurationsparameter	Einstellung
Alle Geräte sind optional	<ul style="list-style-type: none"><li>■ Wird pro Gerät eingestellt (Standardeinstellung) Das Verhalten des SmartWire-DT Strangs bei fehlendem Teilnehmer wird für jeden Teilnehmer einzeln konfiguriert (über den Konfigurationseditor-Dialog des entsprechenden SmartWire-DT Teilnehmers, Modulparameter «Teilnehmer am SWD erforderlich»).</li><li>■ Ja Alle Teilnehmer am SmartWire-DT Strang werden als optional angesehen. Die Einstellung in den Konfigurationseditor-Dialogen der SmartWire-DT Teilnehmer wird ignoriert.</li></ul>

Tab. 6 Konfigurationseditor-Dialog SWD-Master

### 3 Steuerungskonfiguration

#### 3.3 SmartWire-DT™ Konfiguration

- Konfigurationseditor-Dialog SWD-Teilnehmer
  - Registerkarte [SWD-Gerät Konfiguration]

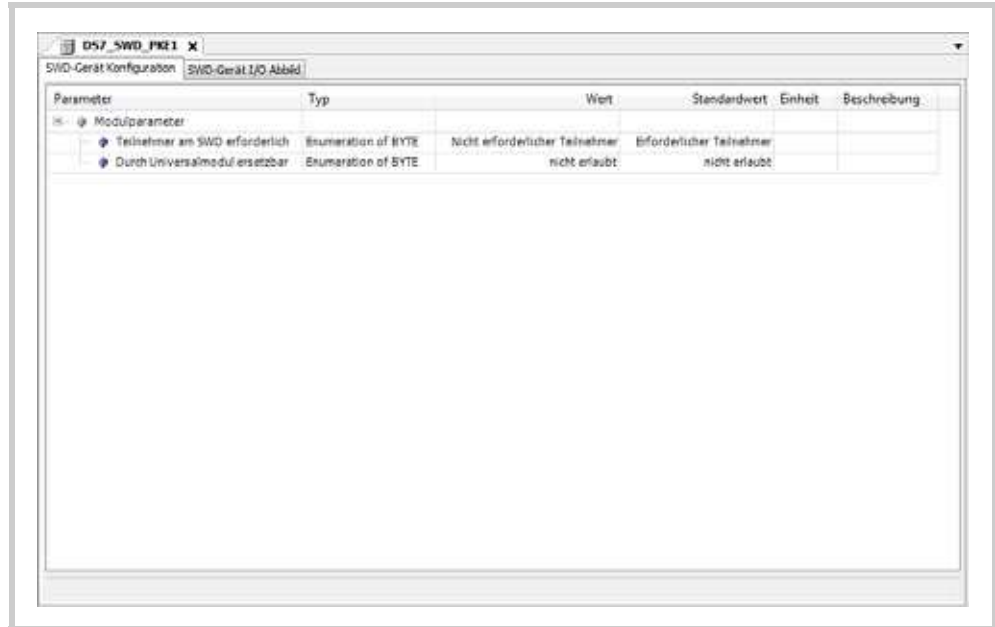


Abb. 19 Konfigurationseditor-Dialog SWD-Teilnehmer, Registerkarte [SWD-Gerät Konfiguration]

Modulparameter	Wert
Teilnehmer am SWD erforderlich	Diese Parametereinstellung hat nur eine Wirkung, wenn im Konfigurations-Dialog SWD-Master der Konfigurationsparameter «Alle Geräte optional» auf «Wird pro Gerät eingestellt» gesetzt ist. <ul style="list-style-type: none"> <li>■ Erforderlicher Teilnehmer (Standardeinstellung) Fehlt dieser Teilnehmer oder hat er eine Störung, wird der SmartWire-DT Strang in den Zustand FAILSAFE zurückgesetzt.</li> <li>■ Nicht erforderlicher Teilnehmer Fehlt dieser Teilnehmer oder hat er eine Störung, hat dies keinen Einfluss auf den Zustand des SmartWire-DT Strangs.</li> </ul>
Durch Universalmodul ersetzbar	<ul style="list-style-type: none"> <li>■ nicht erlaubt (Standardeinstellung)</li> <li>■ erlaubt</li> </ul> Dieser Teilnehmer kann im SmartWire-DT Strang auch durch einen Universalteilnehmer ersetzt werden.

Tab. 7 Konfigurationseditor-Dialog SWD-Teilnehmer, Registerkarte [SWD-Gerät Konfiguration]

- Registerkarte [SWD-Gerät I/O Abbild]  
Siehe Kapitel 3.2.3.1 Registerkarte [... I/O Abbild]: I/O-Abbild konfigurieren, 23



3.3.2

**Busdiagnose**

Jeder SmartWire-DT Teilnehmer liefert in seinem ersten Input-Byte grundsätzliche Statusinformationen:

Byte0	Bedeutung
Bit0 ... 3	Teilnehmerspezifische Nutzdaten
Bit4	TRUE = Für den Teilnehmer liegt eine erweiterte Diagnosemeldung vor
Bit5	Reserviert
Bit6	TRUE = Der Teilnehmer ist vorhanden
Bit7	TRUE = Der Teilnehmer wird durch ein Universalmodul ersetzt

Tab. 8 Grundsätzliche Statusinformationen der SmartWire-DT Teilnehmer

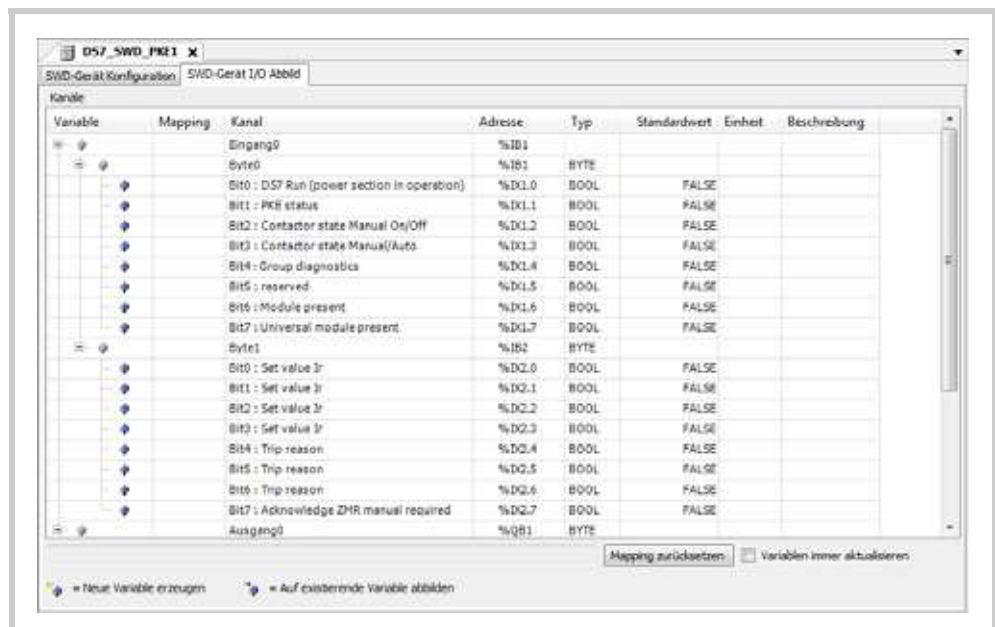


Abb. 20 Konfigurationseditor-Dialog SWD-Teilnehmer, Registerkarte [SWD-Gerät I/O Abbild]

Weitere Statusmeldungen zum SmartWire-DT Koordinator und den angeschlossenen Teilnehmern können über Funktionen und Funktionsblöcke der «EA23\_SwdUtil»-Bibliothek abgerufen werden.

3.3.3

**Azyklische Kommunikation**

Eine azyklische Kommunikation zu SmartWire-DT Teilnehmern kann über den Funktionsblock SWD\_ACYCLICREQUEST der «EA23\_SwdUtil»-Bibliothek gestartet werden.

### 3 Steuerungskonfiguration

#### 3.3 SmartWire-DT™ Konfiguration

##### 3.3.4

#### Bedienungs- und Anzeigeelemente der SmartWire-DT™ Master Schnittstelle (Hardware)



Abb. 21 SmartWire-DT Master Schnittstelle von XV100 und XC-152

Element	Funktion
A POW/AUX-Schnittstelle	Stromversorgungs-Schnittstelle für SmartWire-DT
B POW-LED	Leuchtet, wenn der SmartWire-DT Strang mit Strom gespeist wird (Spannungsversorgung des lokalen SmartWire-DT ASIC OK).
C SWD-LED	Zeigt, ob der physikalische Aufbau des SmartWire-DT Strangs mit der in der Steuerung (SWD-Master) gespeicherten Sollkonfiguration (Targetkonfiguration) übereinstimmt. Die Konfigurationen werden bei jedem Einschalten der Stromversorgung verglichen.
Aus	Keine Targetkonfiguration vorhanden.
Rot Dauerlicht	<ul style="list-style-type: none"> <li>■ Kurzschluss auf der 15 VDC-Spannungsversorgung.</li> <li>■ Kein SmartWire-DT Teilnehmer gefunden.</li> </ul>
Rot blinkend	<ul style="list-style-type: none"> <li>■ Die im SmartWire-DT Strang gefundenen Teilnehmer entsprechen nicht der Sollkonfiguration.</li> <li>■ Ein als notwendig projektierter SmartWire-DT Teilnehmer fehlt.</li> </ul>
Orange blinkend	Der physikalische Aufbau des SmartWire-DT Strangs wird als neue Sollkonfiguration im Gerät eingelesen und abgespeichert.
Grün blinkend	<ul style="list-style-type: none"> <li>■ Der physikalische Aufbau des SmartWire-DT Strangs wird mit der Sollkonfiguration verglichen.</li> <li>■ Die SmartWire-DT Teilnehmer werden adressiert.</li> </ul>
Grün Dauerlicht	Die am SmartWire-DT Strang gefundenen Teilnehmer entsprechen der Sollkonfiguration. Der SmartWire-DT Strang ist bereit für den Datenaustausch.
D Config-LED	Zeigt, ob die in der SPS definierte Projekt-Konfiguration des SWD-Masters mit der in der Steuerung (SWD-Master) gespeicherten Sollkonfiguration des SmartWire-DT Strangs übereinstimmt (Voraussetzung für einen Datenaustausch). Die Konfigurationen werden bei jedem Einschalten der Stromversorgung verglichen.

Element	Funktion
Aus	<ul style="list-style-type: none"> <li>■ Keine Projekt-Konfiguration vorhanden.</li> <li>■ Fehlerhafte Sollkonfiguration (siehe SWD-LED).</li> </ul>
Rot Dauerlicht	Die Projekt-Konfiguration und die gespeicherte Sollkonfiguration sind nicht kompatibel.
Grün blinkend	Die Projekt-Konfiguration ist kompatibel mit der gespeicherten Sollkonfiguration.
Grün Dauerlicht	Die Projekt-Konfiguration ist identisch mit der gespeicherten Sollkonfiguration.
E Konfigurationstaste «Config»	<p>SmartWire-DT Strang konfigurieren. Speichert den physischen Aufbau des SmartWire-DT Strangs in der Steuerung (SWD-Master) remanent als Sollkonfiguration (Targetkonfiguration).</p> <ul style="list-style-type: none"> <li>■ Funktioniert nicht, wenn die Config-LED grün leuchtet und die SPS-Programm im Status «RUN» ist. (SmartWire-DT im Zustand «SWD_RUN_NORMAL»)</li> <li>■ Die Sollkonfiguration muss nach jedem Teilnehmertausch am SmartWire-DT Strang durchgeführt werden.</li> <li>■ Ein Einlesen und Speichern der Sollkonfiguration ist auch über den Browser-Befehl «swd_setnew» möglich.</li> </ul>
F SWD-Schnittstelle	SmartWire-DT Schnittstelle

Tab. 9 Bedienungs- und Anzeigeelemente der SmartWire-DT Master Schnittstelle

### 3.3.5

#### SmartWire-DT™ spezifische SPS-Shell Befehle

Der SmartWire-DT Master unterstützt folgende SPS-Shell Befehle:

Befehl	Beschreibung
?	Zeigt eine Liste der implementierten und zur Verfügung stehenden Kommandos
swd_targetconfig	Zeigt eine Liste der in der aktuell gespeicherten Targetkonfiguration enthaltenen SmartWire-DT Teilnehmer.
swd_projectconfig	Zeigt eine Liste der in der aktuell geladenen Projektkonfiguration enthaltenen SmartWire-DT Teilnehmer.
swd_setnew	Speichert die aktuell am SmartWire-DT Strang gefundenen Teilnehmer als Sollkonfiguration (Targetkonfiguration).
swd_version <arg>	Zeigt Versionsinformationen für einen / alle SmartWire-DT Teilnehmer.
swd_data <arg>	Zeigt den aktuellen Stand der Nutzdaten auf dem SmartWire-DT Strang.
swd_state	Zeigt Statusinformationen des SmartWire-DT Strangs.
swd_diag <arg>	Zeigt erweiterte Diagnoseinformationen von SmartWire-DT Teilnehmern.
swd_param <arg>	Zeigt erweiterte Parametereinstellungen von SmartWire-DT Teilnehmern.

## 3 Steuerungskonfiguration

### 3.3 SmartWire-DT™ Konfiguration

<b>Befehl</b>	<b>Beschreibung</b>
swd_umodules	Zeigt durch Universalmodule ersetzte SmartWire-DT Teilnehmer.

Tab. 10 SmartWire-DT spezifische SPS-Shell Befehle

## 3.4

### easyNET-Konfiguration

Die SPS-Programmierungsumgebung unterstützt die Konfiguration und den Betrieb eines easyNET-Netzwerks.



Relevante Informationen zum Thema easyNET finden Sie im Dokument:

- Eine allgemeine Beschreibung des SmartWire-DT Systems finden Sie im Dokument «MN05006004Z-DE Datenübertragung zwischen easy- und IEC-Steuerungen»

#### Systemvoraussetzung

SPS-Zielsystem (Steuerung)	Version (SPS-Zielsystem)	Betriebssystem (OS)
XV100	ab V 3.5.2.0	Windows CE 5.0 ab Image Release 2.26.0 (xxx)
XV400	ab V 3.5.2.0	Windows CE 5.0 ab Image Release 2.26.0 (xxx)
XC-152	ab V 3.5.2.0	Windows CE 5.0 ab Image Release 2.26.0 (xxx)
XC-CPU202	ab V 3.5.2.0	ab btsxc202_v030504.bin

Tab. 11 Systemvoraussetzungen

## 3 Steuerungskonfiguration

### 3.4 easyNET-Konfiguration

#### 3.4.1

#### easyNET-Feldbus einrichten

- 1 Bauen Sie den Steuerungskonfigurationsbaum für den easyNET-Feldbus auf:
  - 1.1 Hängen Sie unter dem SPS-Zielsystem (Steuerung) mittels des Befehls [Gerät anhängen] den «CANbus» ein.
  - 1.2 Hängen Sie unter «CANbus» den «easyNET bus» an.

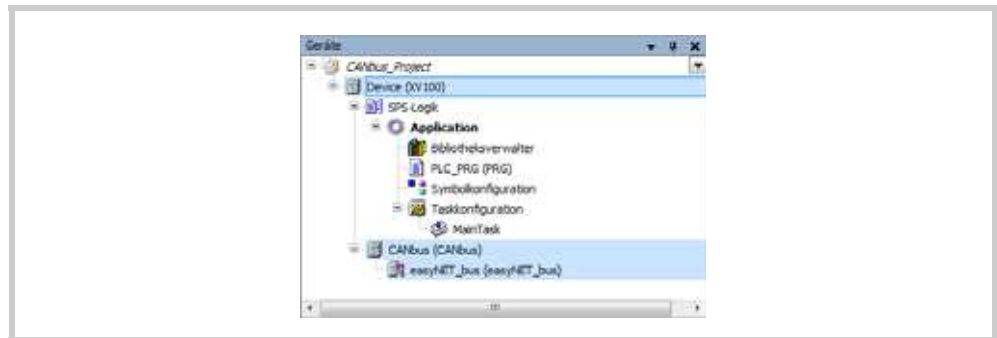


Abb. 22 Steuerungskonfigurationsbaum

- 2 Konfigurieren Sie den Feldbus mittels der Dialoge des Feldbus-Konfigurationseditors.
  - Konfigurationseditor-Dialog CANbus
  - Registerkarte [CANbus]

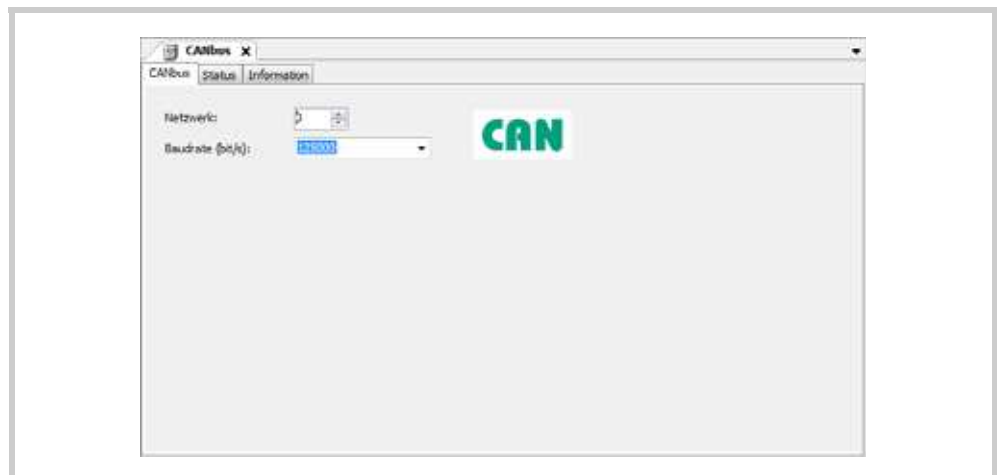


Abb. 23 Konfigurationseditor-Dialog CANbus, Registerkarte [CANbus]

Konfigurationsparameter	Einstellung
Netzwerk	Nummer des CAN-Netzwerks, das über die CANbus-Schnittstelle angebunden werden soll.
Baudrate (bit/s)	Baudrate für die Datenübertragung auf dem Bus.

Tab. 12 Konfigurationseditor-Dialog CANbus, Registerkarte [CANbus]

■ Konfigurationseditor-Dialog easyNET-Bus

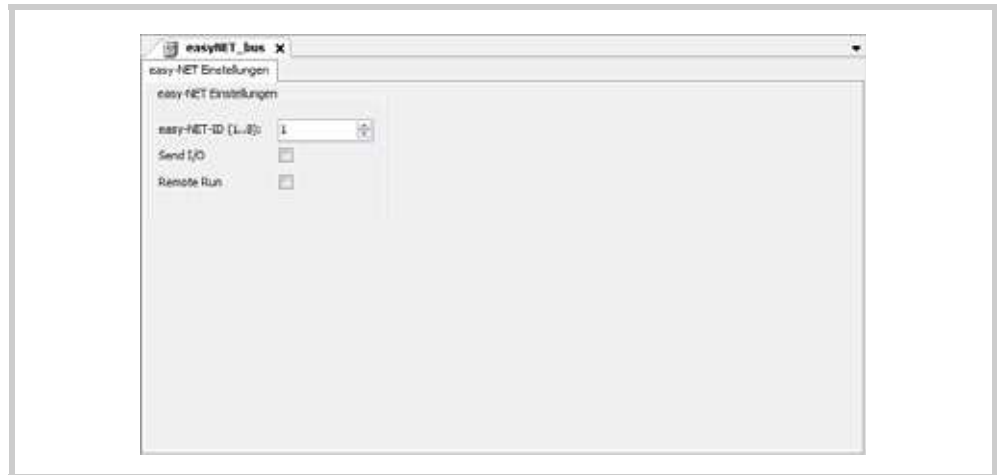


Abb. 24 Konfigurationseditor-Dialog easyNET-Bus

Konfigurationsparameter	Einstellung
easy-NET-ID (1..8)	Adresse der Steuerung (easyNET-Teilnehmer) im easyNET-Netzwerk.
Send I/O	<p>Jede Änderung der Ein-/Ausgänge senden:</p> <ul style="list-style-type: none"> <li>■ Diese Option sollte aktiviert sein: <ul style="list-style-type: none"> <li>- Wenn eine Änderung eines Ein- oder Ausganges unverzüglich allen anderen Netzwerkteilnehmern mitgeteilt werden soll.</li> <li>- Wenn von intelligenten Teilnehmern lesend auf Ein- und Ausgänge von anderen Teilnehmern direkt zugegriffen wird (2I 02, 8Q 01, etc.).</li> </ul> </li> </ul> <p>Dies bedeutet, dass das Nachrichtenaufkommen auf dem Netzwerk stark wachsen kann.</p> <ul style="list-style-type: none"> <li>■ Diese Option sollte deaktiviert sein, wenn schnelle Zähler benutzt werden. Grund: Ist diese Option aktiviert, werden die Eingangsdaten sehr schnell auf das Netzwerk geschrieben, da diese sich ständig ändern und das Netzwerk wird unnötigerweise belasten.</li> </ul>
Remote Run	<p>Automatischer Wechsel der Betriebsart «RUN» und «STOP»:</p> <p>Diese Option sollte aktiviert sein, wenn während des Betriebs die Teilnehmer 2 bis 8 automatisch den Betriebsartenwechsel des Teilnehmers 1 folgen sollen.</p>

Tab. 13 Konfigurationseditor-Dialog easyNET-Bus

## 3 Steuerungskonfiguration

### 3.4 easyNET-Konfiguration

#### 3.4.2

#### Funktionen zur Datenübertragung

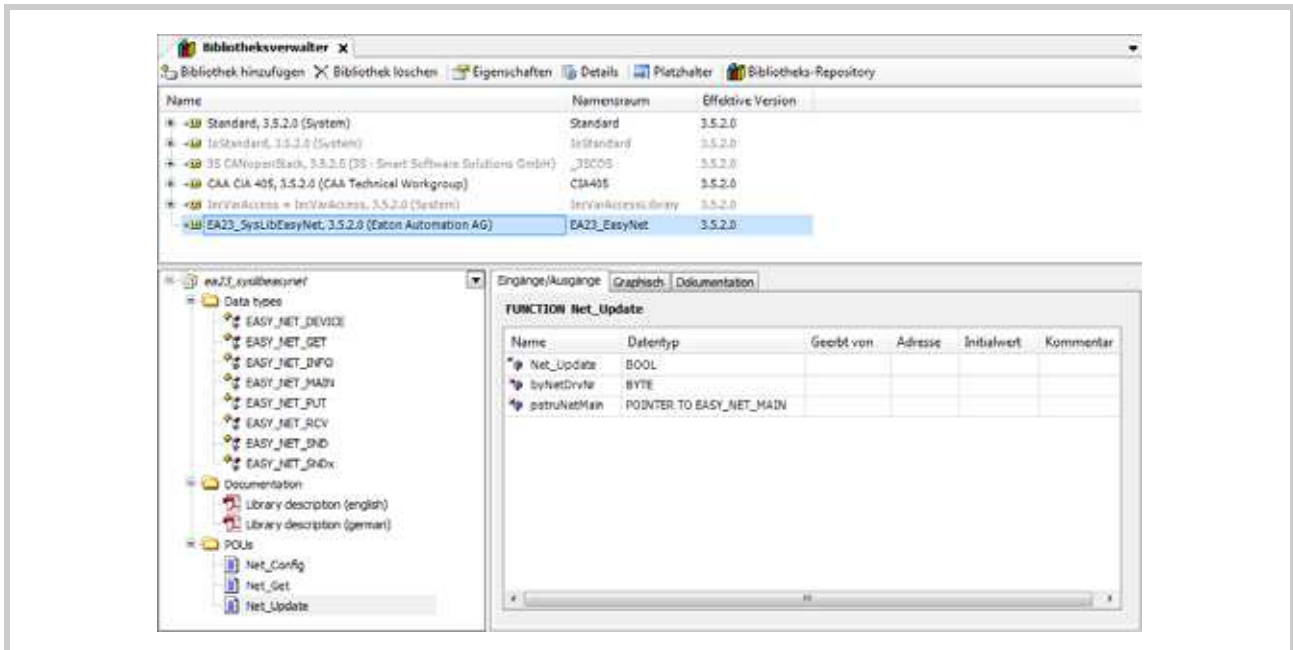


Abb. 25 Bibliothek «EA23\_SysLibEasyNet»

Die Bibliothek «EA23\_SysLibEasyNet» stellt die beiden Funktionen NET\_UPDATE und NET\_GET zum Senden und Empfangen der Daten zur Verfügung. Die Funktionalität von Net\_Config (zur Konfiguration des Netzwerks) wird von den derzeit verfügbaren Steuerungen nicht unterstützt. Die Funktionen NET\_UPDATE und NET\_GET bieten folgende Übertragungsmöglichkeiten:

#### NET\_UPDATE

- Ausgangszustände (Bit) des Teilnehmers mit der Net-ID = 1 an einen Teilnehmer ohne Programm (Remote I/O) senden (Operand Q/S).
- Ein-/Ausgangszustände (Bit) von jedem Teilnehmer (auch Remote I/O) abfragen (Operand Q/S und I/R).
- Bit-Informationen zwischen zwei Teilnehmern mit Programm austauschen. Der Sendeteilnehmer überträgt das Bit mit dem Operand SN. Der Empfangsteilnehmer erfasst das Bit mit dem Operand RN.
- Werte (DWORD-Format) ins Netzwerk stellen, die von anderen Teilnehmern mit Hilfe der NET\_GET-Funktion abgeholt werden.

#### NET\_GET

- Werte (DWORD-Format) aus dem Netzwerk abholen, die ein anderer Teilnehmer mit der Funktion NET\_UPDATE ins Netz gestellt hat.



**Die detaillierten Funktionsbeschreibungen entnehmen Sie bitte den entsprechenden Dokumentationen zu den Funktionsbibliotheken.**



## 4 Betrieb

### 4.1 Einschaltverhalten

Nach dem Einschalten führt die Steuerung einen Systemtest durch. Die Steuerung nimmt den Betriebszustand «STOP» oder «RUN» erst an, wenn sie keinen Hardwarefehler festgestellt hat. Der Systemtest beinhaltet folgende Tests:

- Speichertest
- SPS-Programmtest

Im Weiteren hängt der Zustand der Steuerung vom parametrisierten Startverhalten ab (→ Kapitel 4.4 Betriebsartenumschaltung, 42).

### 4.2 Ausschaltverhalten

Bei einem Spannungseinbruch wird die Programmabarbeitung sofort beendet und alle für den Wiederanlauf erforderlichen Informationen werden gespeichert. Nach einem erneuten Einschalten der Spannung führt die Steuerung einen Neustart aus.

### 4.3 Betriebszustände der Steuerung

#### STOP

Der Betriebszustand «STOP» ist durch folgende Merkmale charakterisiert:

- Es befindet sich ein SPS-Programm in der Steuerung.
- Das SPS-Programm wird nicht durchlaufen.

Der Betriebszustand «STOP» wird angenommen:

- Nach Einschalten der Spannung mit parametrisiertem Startverhalten «STOP» (→ Kapitel 4.4 Betriebsartenumschaltung, 42)
- Über die SPS-Programmierung im PC
- Nach einer Zykluszeitüberschreitung

#### RUN

Im Betriebszustand «RUN» wird das SPS-Programm zyklisch abgearbeitet.

Der Betriebszustand «RUN» wird eingenommen:

- Nach Einschalten der Spannung mit parametrisiertem Startverhalten «Warm»
- Über die SPS-Programmierung im PC

## 4 Betrieb

### 4.3 Betriebszustände der Steuerung

#### 4.3.1

#### XC-152: Übersicht Betriebs- und Fehlerzustände

LED-Stati	Bedeutung
	Kein SPS-Programm vorhanden (RUN/STOP-LED blinkt orange/grün)
	SPS läuft...
	SPS läuft und kommuniziert (CAN, Profibus)
	SPS-Systemfehler
	SPS-Systemfehler (unresolved External)
	SPS-Retain Datenfehler
	Steuerung im Detect-Modus gestartet (gedrückter CTRL-Taster)




Tab. 14 Betriebs- und Fehlerzustände XC-152





Weitere Informationen zu weiteren Betriebs- und Fehlerzuständen sind der Systembeschreibung zum Betriebssystem Windows CE zu entnehmen.

4.3.2

XC-CPU202: Übersicht Betriebs- und Fehlerzustände

LED-Stati	Bedeutung	CPU-Zustand
	RUN/STOP-LED grün; SF-LED aus	RUN
	RUN/STOP-LED blinkt grün; SF-LED aus	STOP
	RUN/STOP-LED blinkt grün; SF-LED rot	NOT READY

Tab. 15 Betriebs- und Fehlerzustände XC-CPU202

- 
**Der Zustand NOT READY wird durch die LEDs RUN/STOP und SF signalisiert. Er tritt ein, wenn während des Starts ein Fehler aufgetreten ist. Die CPU verbleibt dann im Zustand STOP. Nach Beseitigung des Fehlers muss die CPU erneut gestartet werden.**
  
- 
**Die SF-LED leuchtet auch dann rot, wenn keine Batterie im Gerät steckt.**


## 4 Betrieb

### 4.4 Betriebsartenumschaltung

#### 4.4

#### Betriebsartenumschaltung

Die Betriebsartenumschaltung erfolgt über:

- das Fenster des SPS-Laufzeitsystems auf der Steuerung (mittels )

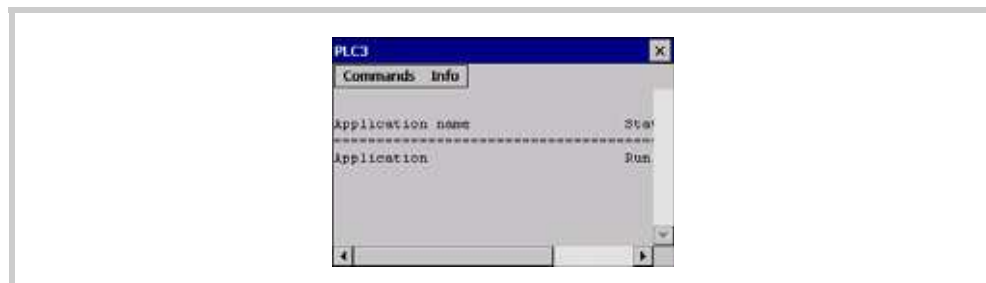


Abb. 26 Fenster des SPS-Laufzeitsystems

- die SPS-Programmierungsumgebung oder
- bei XC-152 und XC-CPU202: auch über den RUN/STOP-Schalter

4.5 Start-, Stop- und Reset-Verhalten

Ablauf

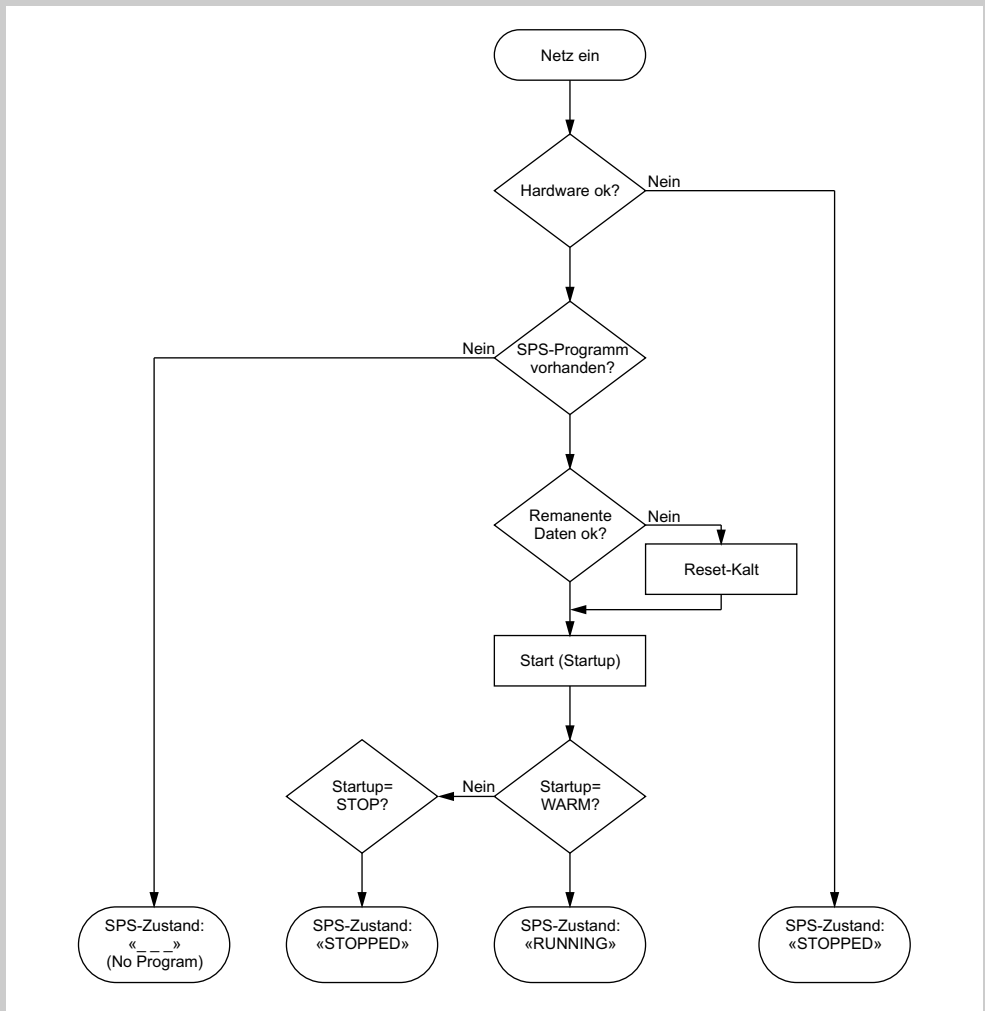


Abb. 27 Ablauf

## 4 Betrieb

### 4.5 Start-, Stop- und Reset-Verhalten

#### Verhalten der Variablenwerte

Aktion	VAR	VAR RETAIN	VAR PERSISTENT (nicht unterstützt)	VAR RETAIN PERSISTENT	Bemerkung
Reboot	initialisiert	bleibt erhalten	–	bleibt erhalten	
Warmstart bzw. Reset / Reset-Warm	initialisiert	bleibt erhalten	–	bleibt erhalten	
Kaltstart bzw. Reset-Kalt	initialisiert	initialisiert	–	bleibt erhalten	
Reset-Ursprung	initialisiert	initialisiert	–	initialisiert	Das SPS-Programm wird auf der Steuerung gelöscht. Die Steuerung wird in den Urzustand zurückversetzt.
Start → Stop	bleibt erhalten	bleibt erhalten	–	bleibt erhalten	
Online Change	bleibt erhalten	bleibt erhalten	–	bleibt erhalten	
Download	initialisiert	initialisiert	–	bleibt erhalten	

Tab. 16 Verhalten der Variablenwerte



**Variablen ohne explizite Initialisierungswerte werden mit ihren Standardinitialisierungswerten initialisiert.**

#### **Kaltstart**

Ein Kaltstart erfolgt beim ersten Start nach dem Laden des SPS-Programms auf die Steuerung bzw. nach jedem Reset-Kalt. Alle Variablen des SPS-Programms, ausser RETAIN PERISTENT Variablen, werden mit ihren Initialisierungswerten initialisiert und das SPS-Programm wird gestartet.

#### **Warmstart**

Ein Warmstart erfolgt nach jedem Reset / Reset-Warm. Die remanenten (RETAIN, RETAIN PERSISTENT) Variablen behalten ihre Werte, die übrigen Variablen werden mit ihren Initialisierungswerten initialisiert.

#### **Reset / Reset-Warm**

Entspricht der Initialisierung eines Warmstarts.

#### **Reset-Kalt**

Entspricht der Initialisierung eines Kaltstarts.

#### **Reset-Ursprung**

Dieser Befehl setzt alle Variablen, auch die remanenten (RETAIN, RETAIN PERSISTENT) Variablen auf den Initialisierungswert zurück und löscht das SPS-Programm auf der Steuerung. Die Steuerung wird in den Urzustand zurückversetzt.

#### **Stop-Verhalten**

Die Abarbeitung des SPS-Programms wird am Ende des Programmzyklus gestoppt.

## 4.6

### Test und Inbetriebnahme

Die Steuerung unterstützt folgende Tests- und Inbetriebnahmemöglichkeiten:

- Breakpoint / Einzelschritt-Betrieb
- Einzelzyklus-Betrieb
- Forcen
- Online-Änderung

#### **Breakpoint / Einzelschritt-Betrieb**

Innerhalb eines SPS-Programms können Breakpoints gesetzt werden. Bei der Ausführung einer mit einem Breakpoint versehenen Anweisung wird das SPS-Programm an dieser Stelle taskspezifisch angehalten. Die folgenden Anweisungen können im Einzelschritt-Betrieb ausgeführt werden. Die Zykluszeitüberwachung ist in diesem Falle deaktiviert.

#### **Einzelzyklus-Betrieb**

Im Einzelzyklus-Betrieb wird ein einzelner Programmzyklus taskspezifisch ausgeführt. Die Ausgänge sind während dem Programmzyklus freigegeben. Am Ende des Programmzyklus wird das Ausgangsbild gelöscht und die Ausgänge abgeschaltet. Die Zykluszeitüberwachung ist in diesem Falle aktiviert.

#### **Forcen**

Alle Variablen eines SPS-Programms können zwangsgesetzt werden.

## 4.7 Programmtransfer

Wenn das SPS-Programm in der SPS-Programmierungsumgebung (Programmier-PC) fehlerfrei übersetzt wurde, kann es in den Arbeitsspeicher der Steuerung geladen und anschliessend gestartet werden. (→ Kapitel 6 Verbindungsaufbau Programmier-PC – Steuerung, 59)

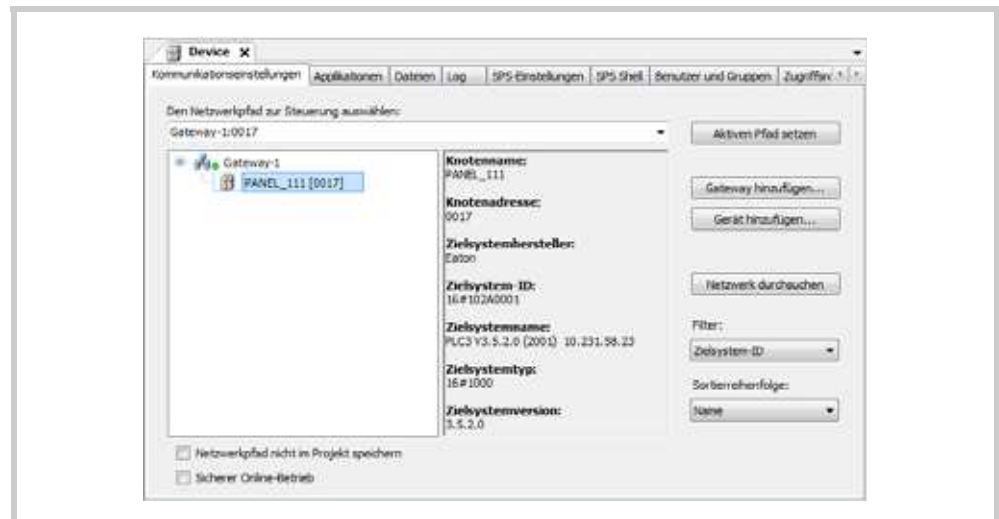


Abb. 28 Geräte-Editor, Registerkarte [Kommunikationseinstellungen]

Bei einem Programmtransfer vom Programmier-PC auf die Steuerung wird das Programm auf der Steuerung mit dem Programm auf dem Programmier-PC verglichen.

- Befindet existiert noch keine Applikation auf der Steuerung, wird dies gemeldet.

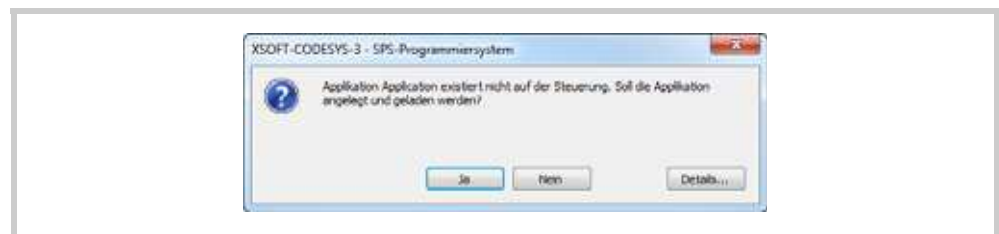


Abb. 29 Dialog «Code geändert»



- Bei Ungleichheit erfolgt eine Abfrage ob das Programm überschrieben werden soll. Wird diese Abfrage bestätigt, so geht die Steuerung in den Zustand «Stop» und das neue Programm wird in den Arbeitsspeicher geladen.

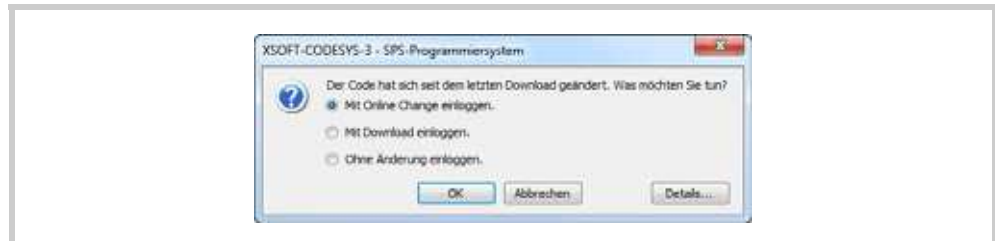


Abb. 30 Dialog «Code geändert»

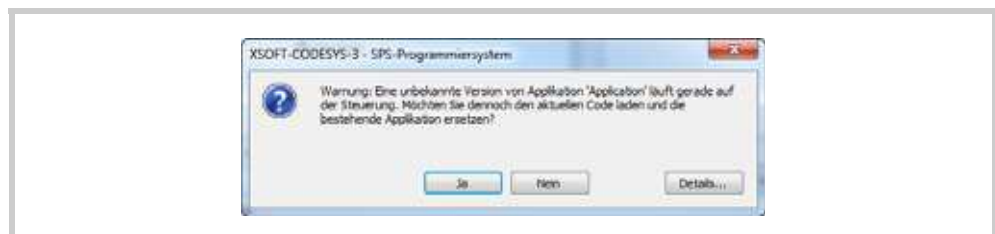


Abb. 31 Dialog «Warnung: unbekannte Version»

### Keine Verbindung

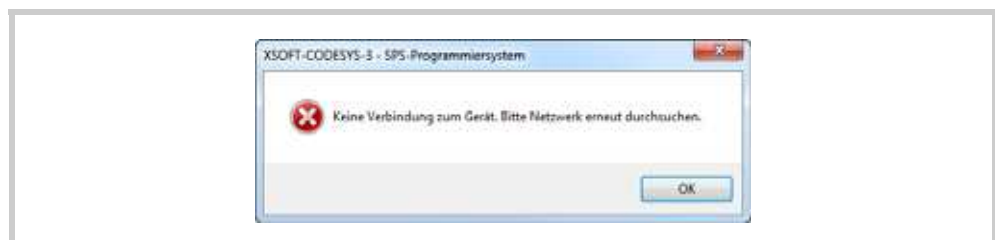


Abb. 32 Dialog «Keine Verbindung»

Kann zwischen der SPS-Programmierungsumgebung und der Steuerung keine Verbindung aufgebaut werden sind folgende Punkte zu überprüfen:

- Physikalische Verbindung
- Kommunikationseinstellungen in der SPS-Programmierungsumgebung (im Geräte-Editor)
- TCP/IP-Einstellungen in den Systemeinstellungen seitens des Programmier-PCs
- TCP/IP-Einstellungen in den Systemeinstellungen seitens der Steuerung

## 4.8

### Bootprojekt erzeugen

Ein SPS-Programm wird nur dann netzausfallsicher gespeichert, wenn vor dem Spannungsausfall ein Bootprojekt erzeugt und anschliessend auf die Steuerung übertragen wurde. Das Bootprojekt wird auf der Steuerung erzeugt.

## 4 Betrieb

### 4.8 Bootprojekt erzeugen

## 5 Programmabarbeitung und Systemzeiten

### 5.1 Programmabarbeitung

Die zeitliche Ablaufeinheit eines SPS-Programms wird mit «Task» bezeichnet. Zusätzlich zum eigentlichen SPS-Programm werden alle anfallenden Systemaktivitäten bearbeitet.

Hierunter fallen z.B. folgende Systemaktivitäten:

- Kommunikation mit der SPS-Programmierungsumgebung
- Online-Änderungen
- Bearbeitung des CANopen-Protokollstacks
- Bearbeitung des Prozessabbilds (IO-Update)

## 5 Programmabarbeitung und Systemzeiten

### 5.2 Taskkonfiguration

#### 5.2

#### Taskkonfiguration

Die Task-Konfiguration definiert eine oder mehrere Tasks, die die Ausführung eines Applikations-Programms kontrollieren. Dies ist eine unabdingbare Ressource für eine Applikation und somit **muss** ein Objekt «Taskkonfiguration» unterhalb einer Applikation im Gerätebaum eingefügt sein.

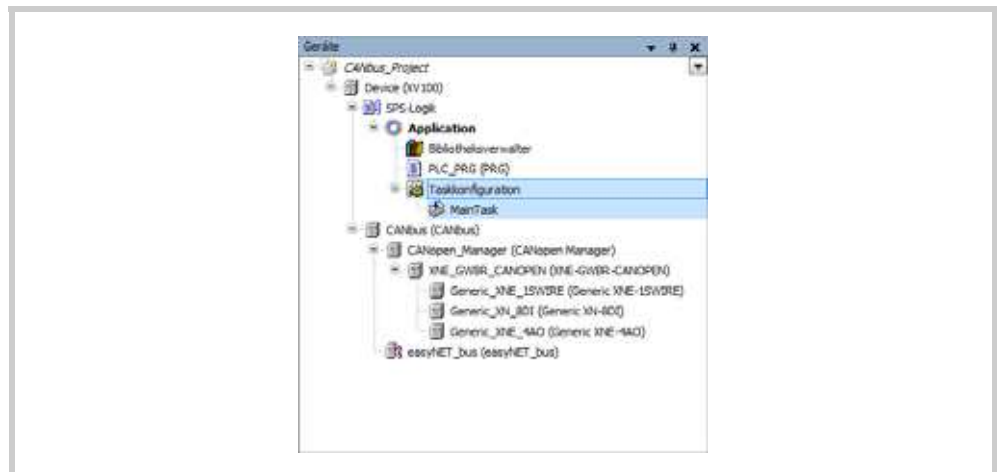


Abb. 33 Objekte für die Taskkonfiguration im Gerätebaum

#### Task

Eine Task ist eine zeitliche Ablaufeinheit eines IEC-Programms. Sie ist definiert durch:

- einen Namen,
- eine Priorität und
- einen Typ, der festlegt, welche Bedingung den Start des Tasks auslöst (Zyklisch, Ereignis, Freilaufend, Status)

Diese Bedingung kann entweder zeitlich definiert sein (Zyklusintervall, freilaufend) oder durch ein internes oder ein externes Ereignis, bei dessen Eintreten die Task ausgeführt werden soll; beispielsweise die steigende Flanke einer globalen Projektvariable oder ein Interrupt-Event der Steuerung.

Jeder Task kann eine Folge von Programmen zugeordnet werden, die beim Ausführen der Task pro Zyklus abgearbeitet werden sollen. Durch die Kombination von Taskpriorität und Taskbedingung wird festgelegt, in welcher zeitlichen Abfolge die Tasks ausgeführt werden. Ein SPS-Programm kann aus mehreren Tasks gleicher oder verschiedener Priorität bestehen, die zyklisch in parametrierbaren Zeitintervallen oder beim Eintreffen von Ereignissen abgearbeitet werden.

Für jede Task kann eine Zeitüberwachung (Watchdog) konfiguriert werden; die dabei möglichen Einstellungen hängen von der Steuerung ab. Zusätzlich gibt es die Möglichkeit, (z.B. Start, Stop, Reset) direkt mit der Ausführung eines Programmierbausteins zu koppeln.

Im Online-Modus kann die Task-Abarbeitung in einer speziellen Ansicht überwacht werden.



## 5 Programmabarbeitung und Systemzeiten

### 5.3 Task-Editor

#### 5.3.0.1

#### Dialog «Taskkonfiguration», Registerkarte [Eigenschaften]

Diese Registerkarte zeigt allgemeine Einstellungen zur Taskkonfiguration, die von der Steuerung vorgegeben werden.



Tab. 17 Dialog «Taskkonfiguration», Registerkarte [Eigenschaften]

## 5.3.0.2

**Task-Dialog, Registerkarte [Konfiguration]:  
Einzelner Task konfigurieren**

Im Task-Dialog, Registerkarte [Konfiguration] werden die Taskparameter eingestellt.

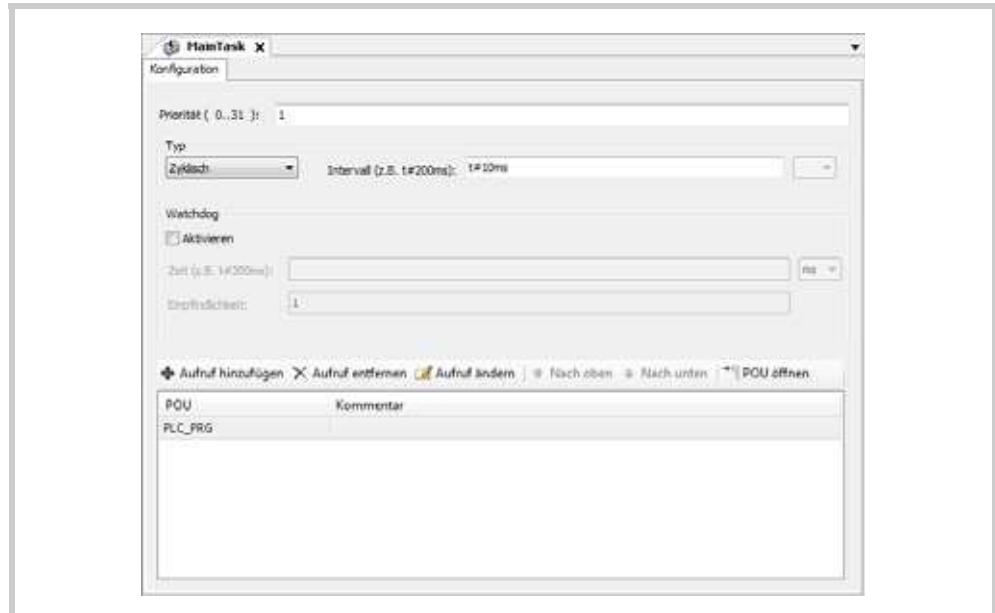


Abb. 35 Task-Dialog, Registerkarte [Konfiguration]

Parameter	Einstellung
Priorität (0...31)	<p>Priorität des Tasks (0 ... 31)</p> <ul style="list-style-type: none"> <li>■ 0: höchste Priorität</li> <li>■ 31: niedrigste Priorität</li> </ul> <p>Wenn mehrere Tasks gleichzeitig die Bedingung zur Abarbeitung erfüllen, wird zuerst diejenige mit der höchsten Priorität abgearbeitet. Wenn mehrere Tasks gleichzeitig die Bedingung zur Abarbeitung erfüllen und gleiche Prioritätsstufe haben, wird zuerst diejenige, die bereits am längsten gewartet hat, abgearbeitet.</p>
Typ	<p>Startbedingung einer Task</p> <ul style="list-style-type: none"> <li>■ Zyklisch: Die Task wird zyklisch abgearbeitet, mit einer Zykluszeit wie im Eingabefeld «Intervall (z.B. t#200ms)» definiert.</li> <li>■ Freilaufend: Es ist keine Zykluszeit definiert. Die Abarbeitung der Task wird bei Programmstart gestartet und am Ende eines Durchlaufs automatisch neu gestartet (in einer fortlaufenden Schleife).</li> <li>■ Status: Die Abarbeitung der Task wird gestartet, wenn die Variable, die im Eingabefeld «Ereignis» definiert ist, den booleschen Wert «TRUE» hat.</li> </ul>

## 5 Programmabarbeitung und Systemzeiten

### 5.3 Task-Editor

Parameter	Einstellung
Typ	<ul style="list-style-type: none"> <li>■ Ereignis: Die Abarbeitung der Task wird gestartet, sobald die Variable, die im Eingabefeld «Ereignis» definiert ist, eine steigende Flanke erhält. Ist die Abtastrate des Task Scheduler zu niedrig, können steigende Flanken des Ereignisses allerdings unbemerkt bleiben.</li> </ul>
Watchdog	<p>Zeitüberwachung (Watchdog) einer Task konfigurieren.</p> <ul style="list-style-type: none"> <li>■ Aktivieren: Ist diese Option gesetzt, ist die Watchdog-Funktion aktiv. Die Task wird im Fehlerstatus «Ausnahmefehler» beendet: <ul style="list-style-type: none"> <li>- Wenn mehrere Zyklen in Folge (Anzahl ist abhängig von «Empfindlichkeit»), die unter «Zeit (z.B. t#200ms)» definierte Zeit überschritten wurde.</li> <li>- Bei einmaliger Zeitüberschreitung, wenn die Zykluszeit des aktuellen Zyklus länger ist als: «Zeit (z.B. t#200ms)» x «Empfindlichkeit»</li> </ul> <p><b>Beispiel:</b> «Zeit (z.B. t#200ms)»: t#10ms «Empfindlichkeit»: 5 Die Task wird im Fehlerstatus «Ausnahmefehler» beendet, sobald die Task einmalig länger als 50 ms läuft.</p> </li> <li>■ Zeit (z.B. t#200ms): Watchdog-Zeit</li> <li>■ Empfindlichkeit: Kriterium für die Auslösung eines Ausnahmefehlers: <ul style="list-style-type: none"> <li>- Kriterium bei mehreren aufeinanderfolgenden Zeitüberschreitungen: 0,1,2: Ausnahmefehler in Zyklus 1 3: Ausnahmefehler in Zyklus 2 ... n: Ausnahmefehler in Zyklus n-1</li> <li>- Faktor bei einmaliger Zeitüberschreitung</li> </ul> <p><b>Hinweis:</b> Ein Watchdog kann mit Hilfe der Bibliotheksfunktionen Cmpl-ecTask.library für bestimmte SPS-Zyklen abgeschaltet werden. Das ist nützlich für Zyklen, die aufgrund von Initialisierungen mehr Zeit beanspruchen können.</p> </li> </ul>
POU	<p>Programmierbausteine zuordnen, die von der Task kontrolliert werden sollen.</p>

Tab. 18 Task-Dialog, Registerkarte [Konfiguration]



#### 5.4

#### Datenremanenz

Die Steuerung besitzt einen Speicherbereich für remanente Daten (RETAIN, RETAIN PERSISTENT). Diese Daten werden bei einem Abschalten der Steuerung gespeichert und bleiben somit erhalten.

Da unter Umständen bei einem Spannungsaus-/abfall die Programmabarbeitung inmitten des Programmzyklus abgebrochen wird, sind die Daten des aktuellen Programmzyklus nicht konsistent.

Beim nächstfolgenden Start wird die Steuerung mit den unmittelbar bei Spannungsaus-/abfall noch korrekt geschriebenen Daten initialisiert.



**Variablenverhalten, siehe Tab. 16, 44.**

#### 5.5

#### Systembibliotheken, Funktionsbausteine und Funktionen

Für die Projektierung des SPS-Programms stehen Ihnen diverse Funktionsbibliotheken zur Verfügung.



**Die detaillierten Funktionsbeschreibungen entnehmen Sie bitte den entsprechenden Dokumentationen zu den Funktionsbibliotheken.**

#### 5.6

#### Prozessabbildverarbeitung / IO-Update

Beim Übersetzen des SPS-Programms wird jeder Task die Konfiguration der in der Task verwendeten Ein- bzw. Ausgänge angehängt.

##### Eingangsabbild

Eine Anweisung «myVar := %IX0.0» führt in der Konfigurationsdatei zu einem Eintrag, welcher beschreibt, dass der Wert des Eingangs %IX0.0 verarbeitet werden muss. Wird gleichzeitig das Byte %IB0 verwendet, wird anstelle des bitweisen Zugriffs das ganze Byte verarbeitet.

Diese Konfiguration wird taskspezifisch erstellt und der Task angehängt. Anhand dieser Konfiguration werden nun am Anfang der Task die Eingänge gelesen.

##### Ausgangsabbild

Eine Anweisung «%QX0.0 := myVar» führt in der Konfigurationsdatei zu einem Eintrag, welcher beschreibt, dass der Wert des Ausgangs %QX0.0 verarbeitet werden muss. Wird gleichzeitig das Byte %QB0 verwendet, wird anstelle des bitweisen Zugriffs das ganze Byte verarbeitet.

Diese Konfiguration wird taskspezifisch erstellt und der Task angehängt. Anhand dieser Konfiguration werden nun am Ende der Task die Ausgänge geschrieben.



**Bei der Projektierung soll konkurrierender Zugriff vermieden werden (d.h. Zugriff auf die gleiche Ein-/Ausgangsadresse aus mehrerer Tasks). Dies kann zu einem inkonsistenten Ein-/Ausgangsabbild führen!**

**Bei konkurrierendem Zugriff werden beim Kompilieren in der SPS-Programmierung Warnungen ausgegeben.**

Infolge unterschiedlichen Bus-Systemen und den entsprechenden Übertragungsmechanismen unterscheidet sich die Verarbeitung des Ein-/Ausgangsabbilds.

#### 5.6.1

#### CAN-Bus

Die Ein- bzw. Ausgänge werden jeweils in PDOs von bis zu 8 Byte Datenlänge zusammengefügt. Die Rx- und Tx-PDOs werden jeweils in der höchstpriorien aller Tasks, die das PDO referenzieren, aktualisiert.

- Die jeweils verwendeten Eingänge bzw. Rx-PDOs werden am Anfang der Task gelesen und ins Eingangsabbild übernommen.
- Die jeweils verwendeten Ausgänge bzw. Tx-PDOs werden am Ende der Task vom Ausgangsabbild übernommen und anschliessend geschrieben.



**Werden dieselben Rx- bzw. Tx-PDOs in verschiedenen Tasks verwendet, kann es zu zyklusinkonsistenten Daten kommen:**

**Bei Eingängen kann der Fall auftreten, dass Werte innerhalb eines niederpriorien Taskzyklus geändert werden, da die Eingänge durch einen hochpriorien Task neu gelesen werden.**

**Ausgänge können durch einen anderen Task frühzeitig gesendet werden, da die Ausgänge durch einen hochpriorien Task neu geschrieben werden.**

#### 5.6.2

#### Profibus

##### 5.6.2.1

#### Profibus-Master

##### **DP-Master CMM-MPI**

Die Eingänge und Ausgänge werden in einem zusätzlichen Systemtask (Treiber-Task) zyklisch aktualisiert. Dies erfolgt asynchron zu den Tasks des SPS-Programms.

- Die jeweils verwendeten Eingänge werden am Anfang der Task von der Treiber-Task gelesen und in das Eingangsabbild übernommen.
- Die jeweils verwendeten Ausgänge werden am Ende der Task vom Ausgangsabbild übernommen, an die Treiber-Task übergeben und anschliessend geschrieben.

##### **DP-Master DPM-MC2 (optionale Kommunikationsbaugruppe zu XV400)**

Die Eingänge und Ausgänge werden auf der optionalen Kommunikationsbaugruppe zyklisch aktualisiert. Dies erfolgt asynchron zu den Tasks des SPS-Programms.

- Die jeweils verwendeten Eingänge werden am Anfang der Task von der optionalen Kommunikationsbaugruppe gelesen und in das Eingangsabbild übernommen.
- Die jeweils verwendeten Ausgänge werden am Ende der Task vom Ausgangsabbild übernommen, an die optionale Kommunikationsbaugruppe übergeben und anschliessend geschrieben.

##### 5.6.2.2

#### Profibus-Slave

##### **Profibus DP Device PDP-TP (optionale Kommunikationsbaugruppe zu XV400)**

Die Eingänge und Ausgänge werden auf der Feldbuskarte zyklisch aktualisiert. Dies erfolgt asynchron zu den Tasks des SPS-Programms.

- Die jeweils verwendeten Eingänge werden am Anfang der Task von der optionalen Kommunikationsbaugruppe gelesen und in das Eingangsabbild übernommen.
- Die jeweils verwendeten Ausgänge werden am Ende der Task vom Ausgangsabbild übernommen, an die optionale Kommunikationsbaugruppe übergeben und anschliessend geschrieben.

##### 5.6.3

#### SmartWire-DT™

Die Eingänge und Ausgänge werden in einer zusätzlichen Systemtask (Treiber-Task) zyklisch aktualisiert. Dies erfolgt asynchron zu den Tasks des SPS-Programms.

Die Eingänge werden am Anfang derjenigen Task des SPS-Programms in das Eingangsabbild übernommen, in der sie in der Taskkonfiguration zuerst verwendet werden.

Die Ausgänge werden am Ende derjenigen Task des SPS-Programms in das Ausgangsabbild übergeben, in der sie in der Taskkonfiguration zuerst verwendet werden.

## 5 Programmabarbeitung und Systemzeiten

### 5.6 Prozessabbildverarbeitung / IO-Update

## 6 Verbindungsaufbau Programmier-PC – Steuerung



### 6.1 Verbindungsaufbau über Ethernet

Die Kommunikation vom Programmier-PC auf die Steuerung erfolgt mit Ethernet und dem TCP/IP-Protokoll. Für die Vernetzung sind geschirmte TwistedPair-Kabel (STP) zu verwenden:

- Für die Verbindung von Gerät zu Gerät: ausgekreuzte Kabel (Crossover-Kabel)
- Für die Verbindung zu Ethernet-Hub/-Switch: 1:1-Patchkabel

### 6.2 Applikation auf die Steuerung laden

#### Vorbedingungen

- Die Kommunikation zwischen Steuerung und Programmiersystem ist konfiguriert. Siehe Kapitel 2.2.1,  8.
- Das SPS-Laufzeitsystem ist auf der Steuerung installiert. Siehe Kapitel 2.2.5,  14.
- Falls mehrere Applikationen in einem Projekt verwaltet werden, muss die zu übertragende Applikation als aktive Applikation gesetzt sein ([Projekt] > [Aktive Applikation setzen]).
- Das SPS-Programm ist fehlerfrei kompiliert worden ([Erstellen] > [Übersetzen]).
- Eine Ethernet-Verbindung zwischen dem Gerät und dem Projektierungs-PC ist hergestellt.

#### Vorgehen

- 1 Klicken Sie [Erstellen] > [Übersetzen], um das SPS-Programm zu kompilieren.
- 2 Stellen Sie sicher, dass der Simulation-Modus deaktiviert ist (der Menüeintrag [Online] > [Simulation] darf kein Häkchen haben).
- 3 Klicken Sie [Online] > [Einloggen], um die Applikation mit der Steuerung zu verbinden und um das SPS-Programm auf die Steuerung zu übertragen.
- 4 Klicken Sie [Debug] > [Start], um das SPS-Programm auf der Steuerung zu starten.
- 5 Klicken Sie [Online] > [Bootapplikation erzeugen], um das SPS-Programm dauerhaft auf der Steuerung zu speichern.



**Ein SPS-Programm wird nur dann netzausfallsicher gespeichert, wenn vor dem Spannungsausfall ein Bootprojekt «online» erzeugt und anschliessend auf die Steuerung übertragen wurde.**

## 6 Verbindungsaufbau Programmier-PC – Steuerung

### 6.2 Applikation auf die Steuerung laden

## 7 Visualisierung

### 7.1 Target- und Web-Visualisierung

In Abhängigkeit von der verwendeten Steuerung werden Target-Visualisierung bzw. Web-Visualisierung unterstützt. Entsprechende Information ist nachfolgender Tabelle zu entnehmen.

SPS-Zielsystem (Steuerung)	Version (SPS-Zielsystem)	Target-Visu	Web-Visu
XV100	ab V 3.5.2.0	✓	✓
XV400	ab V 3.5.2.0	✓	✓
XVS400	ab V 3.5.2.0	✓	✓
XC-152	ab V 3.5.2.0	✓	✓
XC-CPU202	ab V 3.5.2.0	–	✓

Tab. 19 Target- und Web-Visualisierung



Detaillierte Informationen siehe Online-Hilfe zum Programmiersystem.

## 7 Visualisierung

### 7.2 Symbolkonfigurations-Editor

#### 7.2 Symbolkonfigurations-Editor

##### Für den Zugriff von Visualisierungssystemen und OPC-Servern auf Variablen

Als Basis für die Kommunikation zwischen Steuerung und einer allfällig verwendeten Visualisierung dient die Symboldatei. Der Inhalt dieser Symboldatei wird in der SPS-Programmierungsumgebung konfiguriert. Beim Kompilieren wird diese Symboldatei erstellt.

Die Symbolkonfiguration wird verwendet, um Symbole mit bestimmten Zugriffsrechten zu erzeugen, über die von extern auf Projekt(applikations)variablen zugegriffen werden kann, z.B. von einem OPC-Server oder einer Visualisierung aus. Eine Beschreibung der Symbole wird in einer XML-Datei (Symboldatei) des Projektverzeichnisses zur Verfügung gestellt und kann zu Importzwecken für Visualisierungssysteme verwendet werden.

#### 7.2.1 Symbolkonfiguration anlegen (z. B. für die Anbindung an eine Visualisierung)

Der Symbolkonfigurations-Editor wird wie folgt geöffnet:

- Erstmaliges Öffnen:
  - Im Gerätebaum (Fenster «Geräte») mittels rechter Maustaste das Kontextmenü zur Applikation aufrufen.
  - Im Kontextmenü [Objekt hinzufügen] > [Symbolkonfiguration] klicken.
  - Im Fenster «Symbolkonfiguration hinzufügen» [Hinzufügen] klicken.
- Im Gerätebaum existiert unter Applikation bereits der Eintrag «Symbolkonfiguration»:
  - Im Gerätebaum den Eintrag «Symbolkonfiguration» doppelklicken.

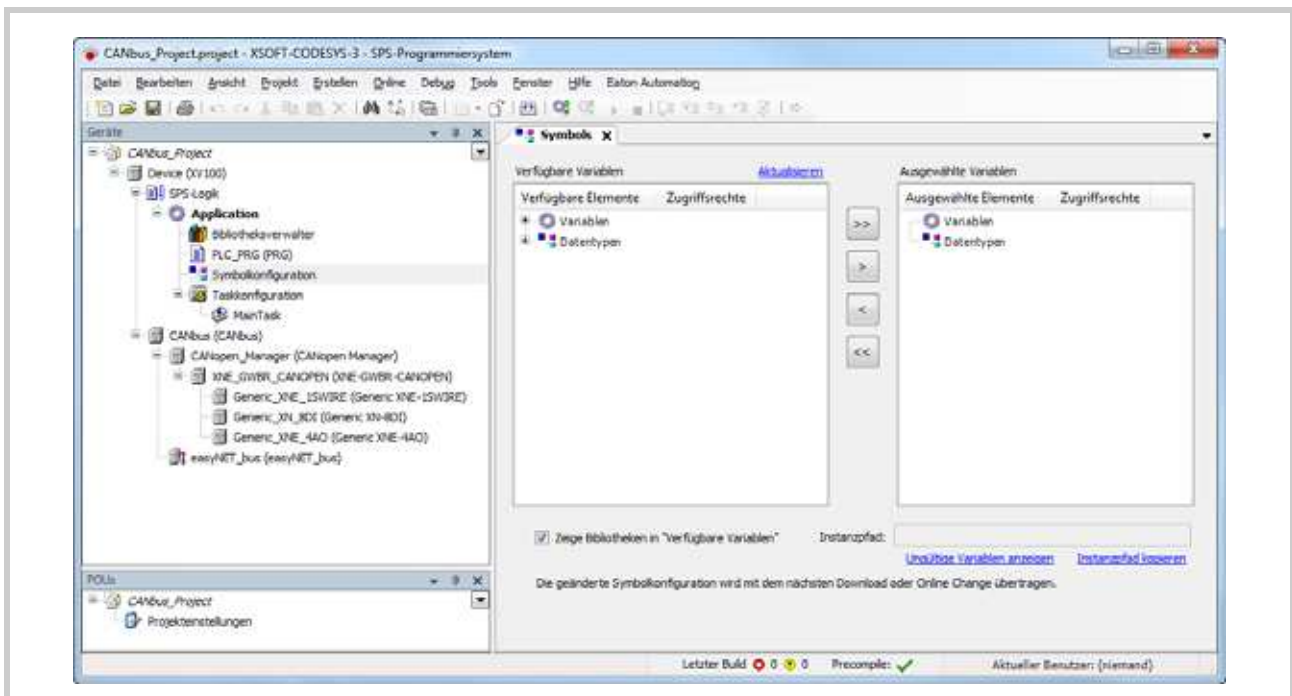


Abb. 36 Symbolkonfiguration-Editor



## 7 Visualisierung

### 7.2 Symbolkonfigurations-Editor

Mittels der Pfeil-Schaltflächen [>] und [>>] können Variablen und Datentypen vom «Verfügbare Variablen»-Baum in den «Ausgewählte Variablen»-Baum kopiert werden.

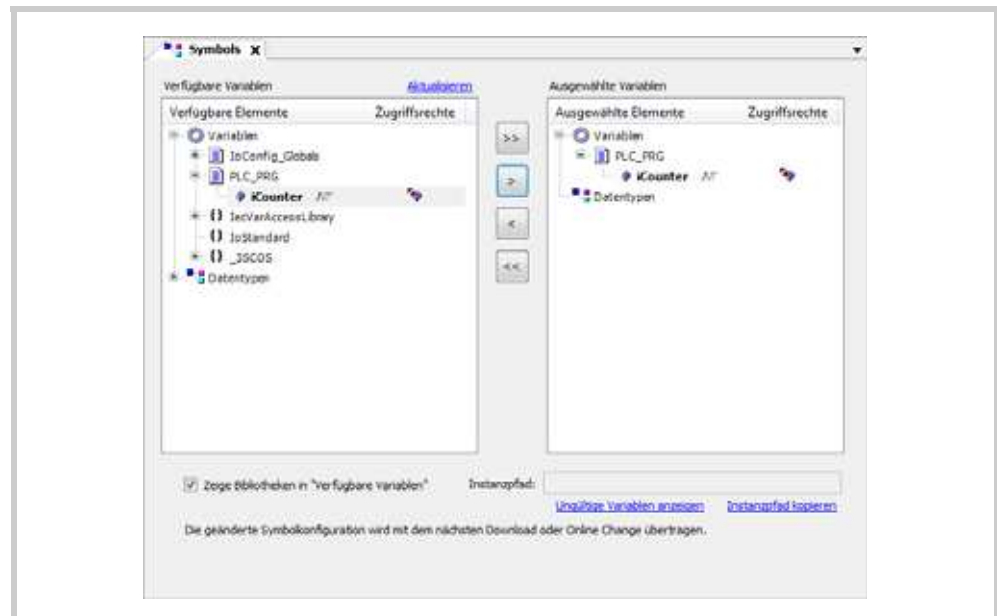
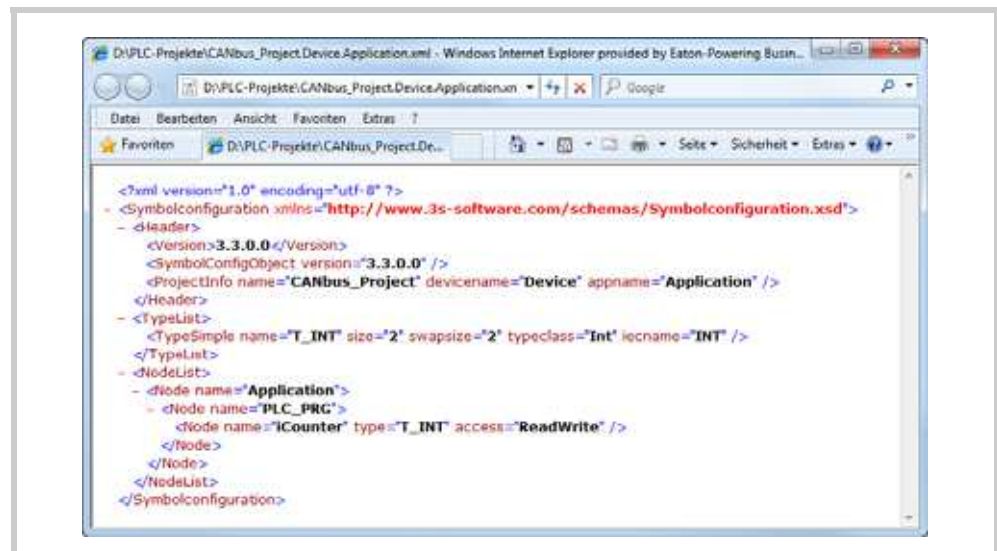


Abb. 37 Variablen und Datentypen auswählen

Die Symbole, welche für eine Applikation definiert wurden, werden in das Projektverzeichnis (Symboldatei) exportiert wenn die Applikation auf die SPS geladen wird. Diese Datei ist laut folgender Syntax benannt: <Projektname>.<Name der Steuerung>.<Applikationsname>.xml




## 7 Visualisierung

### 7.2 Symbolkonfigurations-Editor

## 8 Lizenzierung


### 8.1 SPS-Programmierungsumgebung


Die SPS-Programmierungsumgebung ist lizenzpflichtig.

 **Ist bei der Installation der SPS-Programmierungsumgebung keine Seriennummer bzw. Lizenz vorhanden, werden die SPS-Zielsysteme im Demomode installiert.**

### 8.2 SPS-Laufzeitsystem

Das SPS-Runtime-System ist lizenzpflichtig und benötigt auf der Steuerung **100 Lizenzpunkte**. Für XC-CPU202. ist das SPS-Runtime-System nicht lizenzpflichtig.


 **XC-152:**  
**Auf der Steuerung ist das SPS-Laufzeitsystem bereits vorinstalliert. Eine allenfalls nötige Aktualisierung des Betriebs- bzw. des SPS-Laufzeitsystems kann gemäss Kapitel 2.2.5 Registerkarte [Firmware]: Zielsysteminstallation und Firmware-Update, 14 durchgeführt werden.**

 **Sind beim Start des SPS-Laufzeitsystems nicht genügend Lizenzpunkte vorhanden, beginnt keine Abarbeitung des SPS-Programms. Das SPS-Programm startet im Betriebszustand «Stop».**

Die Abarbeitung des SPS-Programms kann anschliessend in der SPS-Programmierungsumgebung über den Menüpunkt [Debug] > [Start] erfolgen.

### 8.3 Target-Visualisierung

Die Target-Visualisierung ist lizenzpflichtig und benötigt auf der Steuerung jeweils **100 Lizenzpunkte**.

 **Sind beim Start des SPS-Laufzeitsystems nicht genügend Lizenzpunkte vorhanden, beginnt keine Abarbeitung des SPS-Programms. Das SPS-Programm startet im Betriebszustand «Stop».**

Die Abarbeitung des SPS-Programms kann anschliessend in der SPS-Programmierungsumgebung über den Menüpunkt [Debug] > [Start] erfolgen.

### 8.4 Web-Visualisierung

Die Web-Visualisierung ist nicht lizenzpflichtig und benötigt auf der Steuerung **keine Lizenzpunkte**.

8 Lizenzierung  
8.4 Web-Visualisierung

## Stichwortverzeichnis

### A

Applikation auf die Steuerung laden .....	59
Ausgangsabbild .....	56
Ausschaltverhalten .....	39
Azyklische Kommunikation .....	31

### B

Betriebs- und Fehlerzustände	
XC-152 .....	40
XC-CPU202 .....	41
Betriebsartenumschaltung .....	42
Betriebszustände der Steuerung .....	39
Bootprojekt erzeugen .....	47
Busdiagnose .....	31
Buszyklus-Optionen .....	25

### C

CAN-Bus .....	56
---------------	----

### D

Datenremanenz .....	55
---------------------	----

### E

easyNET .....	35
easyNET-Konfiguration .....	35
Eingangsabbild .....	56
Einschaltverhalten .....	39
Einzelschritt-Betrieb .....	45
Einzelzyklus-Betrieb .....	45
Ethernet .....	59
Expertenmodus .....	15

### F

Feldbus-Konfigurationseditor .....	22
Firmware-Update	
XC-152 .....	14
XC-CPU20 .....	16
XV-Geräte .....	14
Forcen .....	45

### G

Geräte-Editor .....	7
---------------------	---

### I

IEC Objekte .....	24
IP-Adresse	
XC-CPU202 im Auslieferungszustand .....	16
XV-Geräte .....	11
IP-Adresse einstellen	
XC-152 .....	11
XC-CPU202 .....	11
XC-Geräte .....	11

### K

Kaltstart .....	44
Konfigurationseditor .....	22

### L

Lizenzierung .....	65
--------------------	----

### N

Netzwerkmaske XC-CPU202 .....	16
-------------------------------	----

### P

Profibus .....	57
Programmtransfer .....	46
Prozessabbildverarbeitung / IO-Update .....	56

### R

Registerkarte	
CANbus .....	36
Eigenschaften .....	52
Firmware .....	14
I/O Abbild .....	23
Kommunikationseinstellungen .....	8, 46
Konfiguration .....	53
LOCAL_IO I/O Abbild .....	18
Log .....	12
SPS Shell .....	13
SWD-Gerät I/O Abbild .....	30
SWD-Gerät Konfiguration .....	30
Remanente Daten .....	55
Reset / Reset-Warm .....	44
Reset-Kalt .....	44
Reset-Ursprung .....	44
RETAIN PERSISTENT .....	44

### S

Setup-Dialog .....	15
SmartWire-DT .....	57

SmartWire-DT Konfiguration .....	26
SmartWire-DT Master Schnittstelle .....	32
SPS-Zielsystem .....	5
Start-, Stop- und Reset-Verhalten .....	43
Steuerungskonfiguration .....	19
Stop-Verhalten .....	44
Symbolkonfigurations-Editor .....	62

## **T**

Target- und Web-Visualisierung .....	61
Task .....	50
Task-Editor .....	51
Taskkonfiguration .....	50

## **V**

VAR .....	44
PERSISTENT .....	44
RETAIN .....	44
RETAIN PERSISTENT .....	44
Verhalten der Variablenwerte .....	44
Visualisierung .....	61

## **W**

Warmstart .....	44
Watchdog .....	50, 54

## **Z**

Zielsystemeinstellungen .....	5
Zielsysteminstallation .....	14