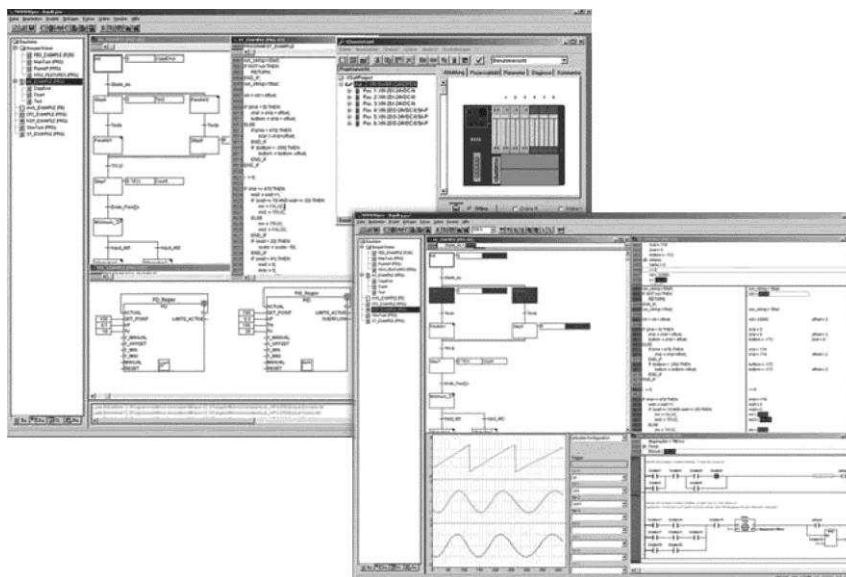


# SPS-Programmierung XV100



## Hersteller

Eaton Automation GmbH  
Spinnereistrasse 8-14  
CH-9008 St. Gallen  
Schweiz  
[www.eaton-automation.com](http://www.eaton-automation.com)  
[www.eaton.com](http://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](http://www.eaton.com)

## Andere Regionen

Bitte kontaktieren Sie Ihren lokalen Lieferanten  
oder senden Sie eine E-Mail an:  
[automation@eaton.com](mailto:automation@eaton.com)

## Originalsprache

Deutsch

## Redaktion

Daniel Lenherr

## Marken- und Produktnamen

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

## Copyright

© Eaton Automation GmbH, 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 GmbH, St. Gallen reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Änderungen vorbehalten

## Inhaltsverzeichnis

<b>1</b>	<b>Allgemein .....</b>	<b>5</b>
1.1	Zweck dieses Dokuments .....	5
1.2	Rückmeldungen zu diesem Dokuments.....	5
1.3	Ergänzende Dokumentation.....	5
<b>2</b>	<b>Installation .....</b>	<b>6</b>
2.1	Lieferumfang .....	6
2.2	Systemvoraussetzung .....	6
2.3	Installieren der SPS-Programmierung .....	7
2.4	Installieren der SPS-Zielsysteme .....	8
<b>3</b>	<b>Deinstallation.....</b>	<b>9</b>
3.1	Deinstallieren der SPS-Programmierung .....	9
3.2	Deinstallieren der SPS-Zielsysteme.....	9
<b>4</b>	<b>Zielsystemeinstellungen .....</b>	<b>10</b>
4.1	Zielplattform .....	10
4.2	Speicheraufteilung .....	11
4.2.1	Übersicht Speicherlayout .....	12
4.3	Allgemein .....	13
4.4	Netzfunktionen .....	14
4.5	Visualisierung.....	15
<b>5</b>	<b>Steuerungskonfiguration .....</b>	<b>16</b>
5.1	Arbeiten in der Steuerungskonfiguration .....	16
5.1.1	Projektspezifische Konfigurationsdatei(en) und Gerätedatei(en) .....	18
5.2	Allgemeine Einstellungen .....	19
5.3	Konfiguration als CAN-Master .....	20
5.3.1	Konfiguration des CAN-Masters.....	21
5.3.2	Konfiguration der CAN-Knoten .....	24
5.3.3	Busdiagnose .....	30
5.4	Konfiguration als CAN-Device.....	31
5.4.1	Konfiguration des CAN-Devices.....	32
5.4.2	Konfiguration des CAN-Devices im CAN-Master .....	36
5.5	Konfiguration als Profibus-Master .....	37
5.5.1	Konfiguration des Profibus-Masters .....	38
5.5.2	Konfiguration der Profibus-Slaves.....	43
5.5.3	Busdiagnose .....	49
5.6	SmartWire-DT Konfiguration .....	50
5.6.1	Konfiguration des SWD-Masters und SWD-Modulauswahl .....	51
5.6.2	Busdiagnose .....	55
5.6.3	Azyklische Kommunikation .....	55
5.6.4	SmartWire-DT LEDs .....	55
5.6.5	Config-Taster .....	56
5.6.6	Browserbefehle .....	57
<b>6</b>	<b>Betrieb.....</b>	<b>58</b>

# Inhaltsverzeichnis

6.1	Einschaltverhalten .....	58
6.2	Ausschaltverhalten .....	58
6.3	Betriebszustände der Steuerung .....	59
6.4	Betriebsartenumschaltung .....	60
6.5	Start-Verhalten .....	61
6.6	Stop-Verhalten .....	62
6.7	Reset-Verhalten .....	62
6.8	Test und Inbetriebnahme .....	63
6.9	Programmtransfer .....	64
6.9.1	Bootprojekt erzeugen .....	65
<b>7</b>	<b>Programmabarbeitung und Systemzeiten .....</b>	<b>66</b>
7.1	Programmabarbeitung .....	66
7.2	Taskkonfiguration .....	67
7.3	Multitasking .....	69
7.4	Tasküberwachung / Zykluszeitüberwachung .....	69
7.5	Datenremanenz .....	72
7.6	Direkter Peripheriezugriff .....	72
7.7	Interruptverarbeitung .....	72
7.8	Systembibliotheken, Funktionsbausteine und Funktionen .....	73
7.9	Prozessabbildverarbeitung / IO-Update .....	73
7.9.1	CAN-Bus .....	74
7.9.2	Profibus .....	74
7.9.3	SmartWire-DT .....	74
<b>8</b>	<b>Verbindungsaufbau Programmier-PC – Steuerung .....</b>	<b>75</b>
8.1	Verbindungsaufbau über Ethernet .....	75
<b>9</b>	<b>Parameter-Manager / Objektverzeichnis .....</b>	<b>78</b>
<b>10</b>	<b>SPS-Browser .....</b>	<b>79</b>
<b>11</b>	<b>Alarmkonfiguration .....</b>	<b>80</b>
<b>12</b>	<b>Anbindung an die Visualisierung / Erstellen der Symboldatei .....</b>	<b>81</b>
12.1	Symboldatei konfigurieren .....	81
12.2	Symboldatei senden .....	84
<b>13</b>	<b>Zielsysteminstallation und Firmware Update .....</b>	<b>85</b>
<b>14</b>	<b>Lizenzierung .....</b>	<b>87</b>
14.1	SPS-Programmierungsumgebung .....	87
14.2	SPS-Laufzeitsystem .....	87
14.3	Target-Visualisierung .....	87
14.4	Web-Visualisierung .....	87
<b>15</b>	<b>Änderungsindex .....</b>	<b>88</b>

# 1 Allgemein

## 1.1 Zweck dieses Dokuments

Dieses Dokument beschreibt den Umgang mit der SPS-Programmierungsumgebung "XSOFTE-CODESYS-2" und dem SPS-Laufzeitsystem für den XV100-Gerätetyp mit Windows CE. Dieses Dokument dient als Ergänzung zum Handbuch der SPS-Programmierung CoDeSys V2.3 der Fa. 3S-Smart Software Solutions GmbH.

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

## 1.2 Rückmeldungen zu diesem Dokument

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

## 1.3 Ergänzende Dokumentation

Ergänzend zu diesem Dokument können weitere Dokumente hilfreich sein.

Folgende Dokumentation finden Sie als pdf-Dateien auf unserer Homepage ([www.eaton-automation.com](http://www.eaton-automation.com)):

- [1] MN05010007Z-DE  
Systembeschreibung Windows CE
- [2] Diverse  
Dokumentationen bezüglich SPS-Funktionsbibliotheken  
MN04802103Z-DE  
SWD\_Util-Bibliothek
- [3] -  
Systembeschreibung CiA Draft Standard DSP301
- [4] -  
Handbuch der SPS-Programmierung CoDeSys V2.3
- [5] MN05006002Z-DE  
Handbuch SmartWire-DT System
- [6] MN05013002Z-DE  
Handbuch SmartWire-DT Gateways
- [7] MN05006001Z-DE  
Handbuch SmartWire-DT Teilnehmer

### 2 Installation

Das Produkt "XSOFTE-CODESYS-2" ist ein eigenständiges Software Paket. Es besteht aus einer PC-seitigen Komponente (SPS-Programmierungsumgebung mit entsprechenden SPS-Zielsystemen), welche auf einem beliebigen PC installiert werden kann und einer zielsystem-seitigen Komponente (SPS-Laufzeitsystem), welche unter der Berücksichtigung von Lizenzpunkten auf SPS-Zielsystemen mit Windows CE installiert werden kann (→ Kap. 14).

➔ Bei Fragen zu Lizenzprodukten wenden Sie sich bitte an Ihren lokalen Eaton Vertriebspartner

#### 2.1 Lieferumfang

Bezeichnung	Version
Software "XSOFTE-CODESYS-2" inkl. Benutzerhandbuch	2.3.9 SP5

Bestehend aus:

Bezeichnung	Version
CoDeSys	2.3.9.47
CoDeSys Gateway	2.3.9.33
CoDeSys OPC-Server <sup>1</sup>	2.3.13.11

SPS-Zielsystem	SPS-Laufzeitsystem	Betriebssystem (OS)
XV-1xx-V2.3.9 SP5	PLCWinCE V 2.4.14 (xxx)	Windows CE 5.0 Image Release 2.28.0 (xxx)

#### 2.2 Systemvoraussetzung

Betriebssystem
Windows XP, Windows Vista, Windows 7/8

<sup>1</sup> Optionale Installation, Komponente ist kein Bestandteil dieser Dokumentation

## 2.3

**Installieren der SPS-Programmierungsumgebung**

Legen Sie die Installations-CD in Ihren PC ein und starten Sie die Installation über 'Setup.exe'.

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

Standardmäßig werden mit der Installation folgende Verzeichnisse angelegt:

Verzeichnis	Beschreibung
<b>Programmiersystem-Verzeichnis</b> C:\Programme\Eaton\XSOFT-CODESYS V2.3.9 SP5	SPS-Programmierungsumgebung
<b>Zielsystem-Verzeichnis</b> C:\Programme\Gemeinsame Dateien\CAA- Targets\Eaton Automation\V2.3.9 SP5	Zielsystemabhängige Komponenten wie Bibliotheken und Konfigurationsdaten inkl. Firmware für die SPS-Zielsysteme

**Verhalten gegenüber bisher installierter SPS-Programmierungsumgebungen**

Bereits installierte Versionen der SPS-Programmierungsumgebung "XSOFT-CODESYS-2" und entsprechende SPS-Zielsysteme werden nicht tangiert.

- ➔ Sämtliche zur Installation dazugehörige SPS-Zielsysteme erhalten eine neue Bezeichnung bzw. Identifikation.
- ➔ Möchte ein mit älterer Version der SPS-Programmierungsumgebung erstelltes SPS-Projekt aktualisiert werden, muss das SPS-Zielsystem umgestellt werden. Nach Schließen und dem erneuten Öffnen des SPS-Projektes wird das SPS-Projekt aktualisiert und es werden die aktuellen Bibliotheken eingebunden.

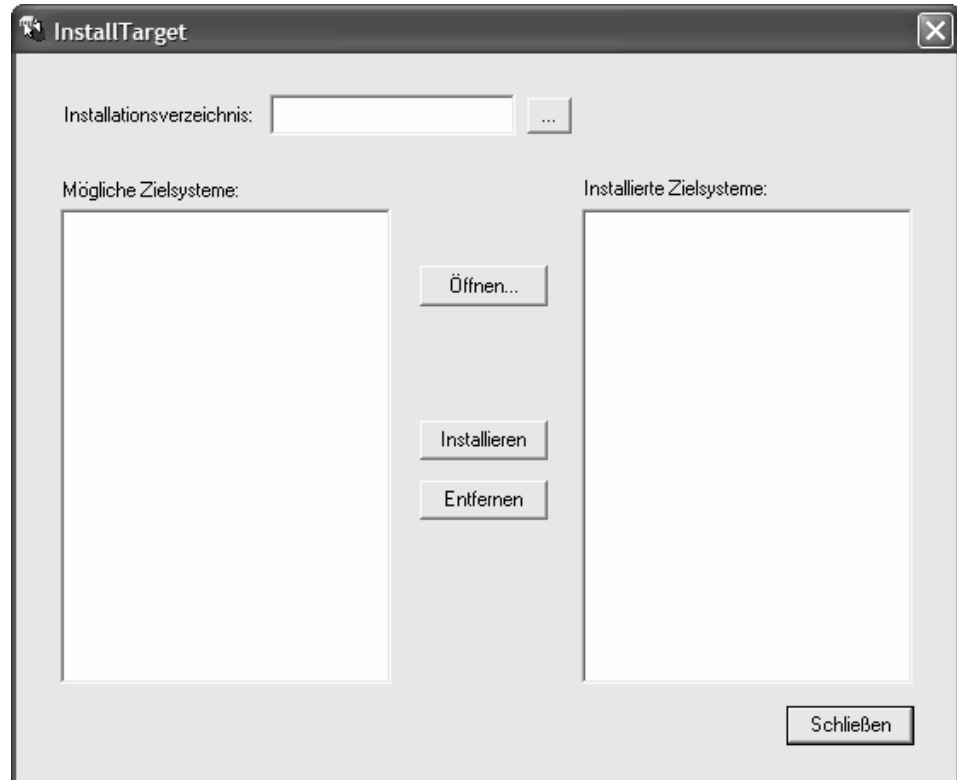
## 2 Installation

### 2.4

#### Installieren der SPS-Zielsysteme

Standardmäßig werden die zur SPS-Programmierungsumgebung entsprechenden SPS-Zielsysteme über den Setup installiert.

Über die Anwendung 'InstallTarget' können allerdings auch nachträglich SPS-Zielsysteme installiert bzw. deinstalliert werden.





### 3 Deinstallation

#### 3.1 Deinstallieren der SPS-Programmierungsumgebung

Bei der Deinstallation werden lediglich installierte Dateien und Komponenten entfernt. Wenn nötig müssen nach der Deinstallation Dateien und Verzeichnisse aus dem Programmiersystem-Verzeichnis manuell entfernt werden.

#### 3.2 Deinstallieren der SPS-Zielsysteme

Bei der Deinstallation der SPS-Programmierungsumgebung werden die zielsystemabhängigen Komponenten wie Bibliotheken und Konfigurationsdaten im Zielsystem-Verzeichnis nicht entfernt.

Die installierten SPS-Zielsysteme müssen über die entsprechende Deinstallationsroutine entfernt werden.

➔ Alternativ können installierte SPS-Zielsysteme auch über die Anwendung 'InstallTarget' aus der SPS-Programmierungsumgebung entfernt werden.

Anschließend müssen die zielsystemabhängigen Komponenten aus dem Zielsystem-Verzeichnis manuell entfernt werden.

## 4 Zielsystemeinstellungen

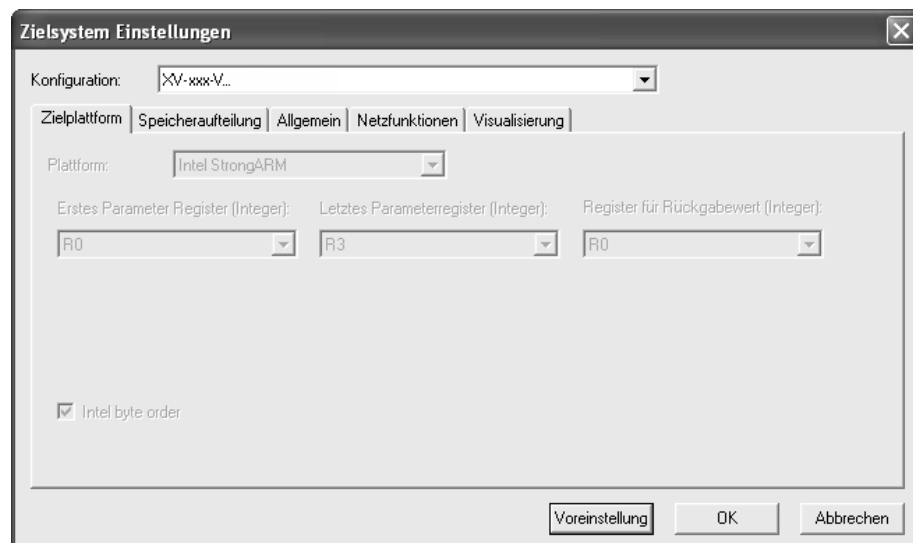
Dieser Dialog öffnet automatisch, wenn ein neues Projekt angelegt wird. Ansonsten wird er über die Funktionalität <Zielsystemeinstellungen> im Registerblatt 'Ressourcen' erreicht.

In den Zielsystemeinstellungen wählen Sie den entsprechenden Steuerungstyp. Anhand dieser Konfiguration werden Prozessortyp und Speichergrößen optimal eingestellt. Der Eintrag '**None**' führt automatisch in den Simulationsmodus.

➔ Veränderungen der voreingestellten Zielsystemeinstellungen können Auswirkungen auf das Verhalten des SPS-Zielsystems haben!

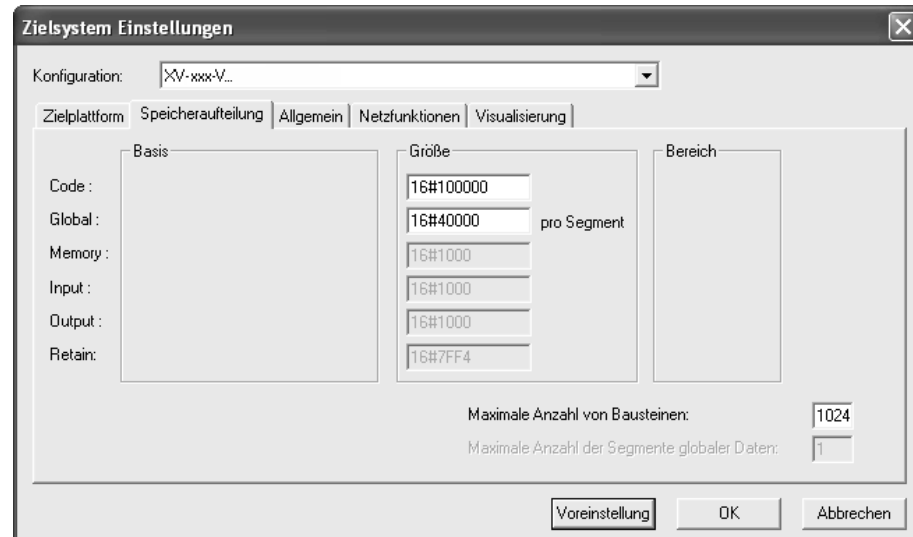
Über die Schaltfläche **Voreinstellung** kann die allenfalls veränderte Konfiguration wieder auf die Standardkonfiguration zurückgesetzt werden.

### 4.1 Zielplattform



## 4.2

## Speicheraufteilung

**Code Grösse**

Standardmässig 2048kB (16#200000) Speicher für den Programmcode. Diese entsprechende Speichermenge wird automatisch alloziert.

**Global Grösse**

Standardmässig 512kB (16#80000) Speicher für die globalen Daten. Diese entsprechende Speichermenge wird automatisch alloziert.

**Max. Anzahl von Programmbausteinen**

Pro Programmbaustein werden automatisch 12Byte Speicher benötigt. Insgesamt werden somit 12kB Speicher für die Funktionspointertabelle benötigt (1024 Bausteine -> 12kB). Dieser Speicher wird an den SPS-Datenspeicher angefügt.



In die Berechnung der im Programm verwendeten Anzahl der Bausteine werden sämtliche Funktionen und Funktionsbausteine eingefügter Bibliotheken miteinbezogen.

## 4 Zielsystemeinstellungen

### 4.2.1

#### Übersicht Speicherlayout

Default Einstellung mit Online-Change

Anzahl Datensegmente = 1

Größe des Datensegments = 512kB

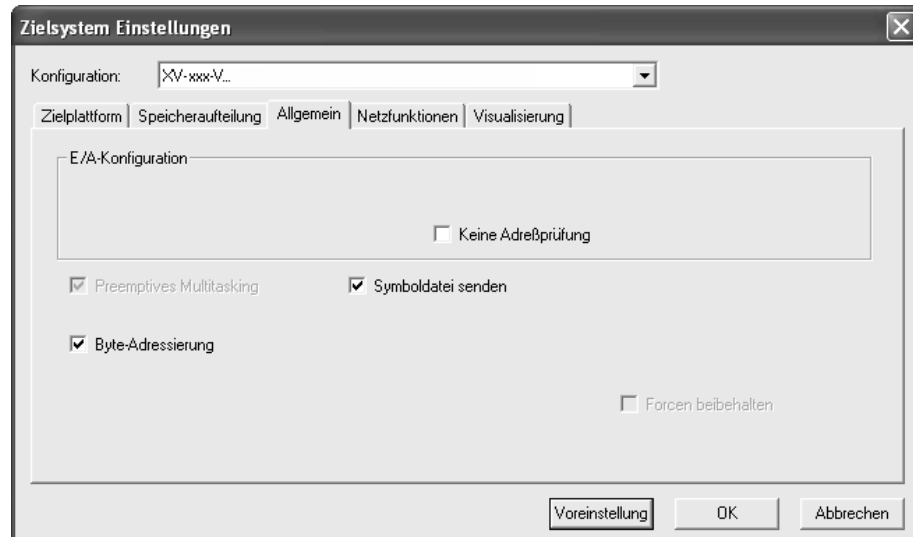
32kB Remanenter Datenbereich
12kB Funktionspointertabelle
12kB Ein-/Ausgangs-/Merkerdaten
1 * 512kB Globaler Datenbereich
2048kB Programmcode (Online Change)
2048kB Programmcode



Beim standardmäßigen Übersetzen des SPS-Programms teilt die SPS-Programmierungsumgebung dem Anwender die Datenbereichs-Größen mit.

Mit dem SPS-Browser kann Online mit dem Kommando 'sysinfo' die SPS-Programmgröße ermittelt werden (→ Kap. 10)

## 4.3

**Allgemein****Keine Adressprüfung**

Bei aktivierter Option wird beim Kompilieren keine Adressprüfung vorgenommen.

**Symboldatei senden**

Bei aktivierter Option wird beim Kompilieren eine Symboldatei erzeugt und beim Programmdownload auf die Steuerung geladen.

**Preemptives Multitasking**

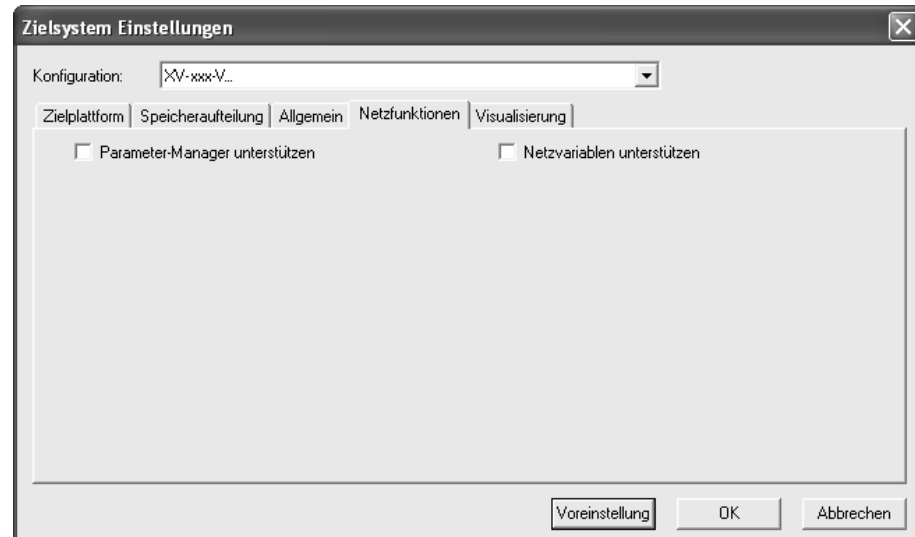
Multitasking wird unterstützt (kann vom Anwender nicht verändert werden).

**Byte-Adressierung**

Bei aktivierter Option erfolgt die Adressierung byteweise (z.B.: %QD4 entspricht %QB4).

### 4.4

### Netzfunktionen



#### Parameter-Manager unterstützen

Bei aktivierter Option erscheint im Registerblatt 'Ressourcen' der Eintrag **Parameter-Manager**. Dieser ermöglicht das Erstellen eines Objektverzeichnisses für Variablen und Parameter, die dem gezielten, aktiven Datenaustausch mit anderen Steuerungen dienen.

- ➔ Die Objektverzeichnis-Funktionalität wird in Verbindung mit einem in der Steuerungskonfiguration eingefügten CAN-Device unterstützt.
- ➔ Detaillierte Informationen sind dem Handbuch CoDeSys V2.3 oder der Online-Hilfe der SPS-Programmierungsumgebung zu entnehmen.

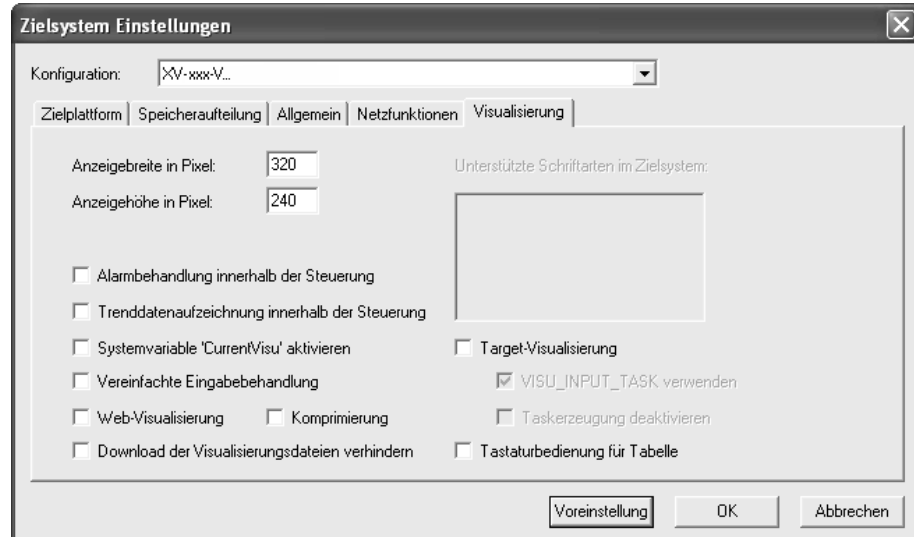
#### Netzvariablen unterstützen

Bei aktivierter Option können Netzwerkvariablen verwendet werden. Sie dienen dem automatischen Datenaustausch mit anderen Steuerungen.

- ➔ Detaillierte Informationen sind dem Handbuch CoDeSys V2.3 oder der Online-Hilfe der SPS-Programmierungsumgebung zu entnehmen.

4.5

Visualisierung



In Abhängigkeit vom verwendeten SPS-Zielsystem werden Target-Visualisierung bzw. Web-Visualisierung unterstützt. Entsprechende Information ist nachfolgender Tabelle zu entnehmen.

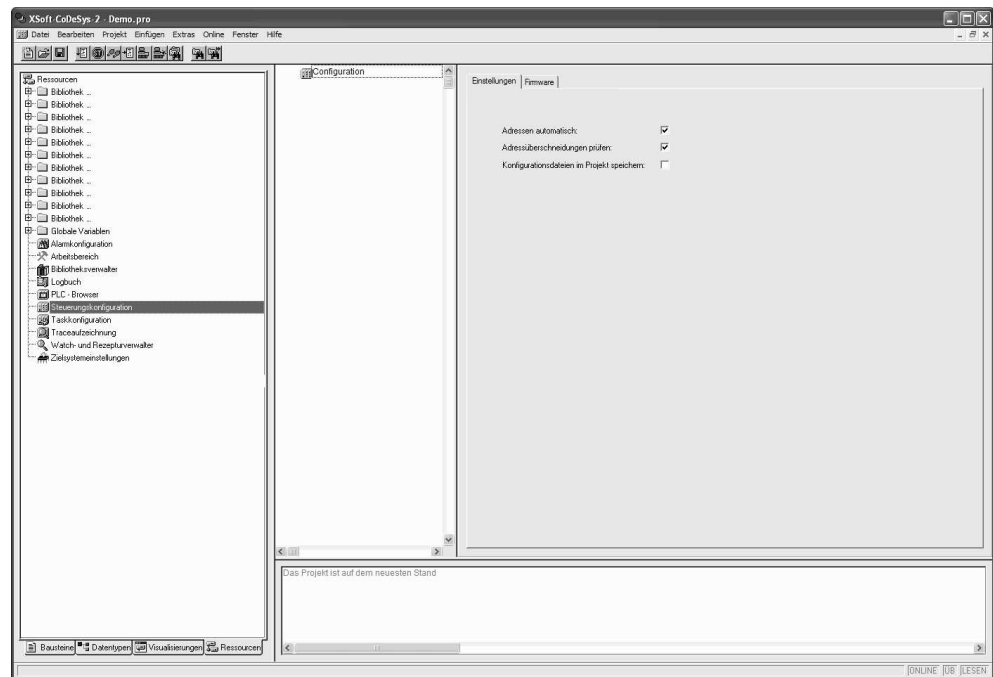
SPS-Zielsystem	SPS-Laufzeitsystem	Target-Visu	Web-Visu
≥ XV-1xx-V2.3.9 SP1	≥ PLCWinCE V 2.4.7 (xxx)	✓	✓

➔ Detaillierte Informationen sind dem Handbuch CoDeSys V2.3, dem Handbuch zur CoDeSys Visualisierung oder der Online-Hilfe der SPS-Programmierungsumgebung zu entnehmen.

### 5 Steuerungskonfiguration

Die Steuerungskonfiguration wird über die Funktionalität <Steuerungskonfiguration> im Registerblatt 'Ressourcen' erreicht.

Die Steuerungskonfiguration erlaubt das Konfigurieren von Ein- und Ausgängen bzw. das Anbinden von busfähigen I/O-Modulen. Grundlage für das Arbeiten in der Steuerungskonfiguration sind die Konfigurationsdateien (\*.cfg) und Geräterdateien (z.B. \*.gsd, \*.eds).



Die Konfiguration wird im Editor in Baumstruktur dargestellt und kann über Menüpunkte und Dialoge bearbeitet werden. Es gibt Elemente, die entweder als Ein- oder Ausgänge dienen oder Verwaltungselemente, die ihrerseits wieder Unterelemente haben (z.B. CAN, PROFIBUS).

Ein- und Ausgänge erscheinen im Editor mit der IEC-Adresse, über die auf sie zugegriffen werden kann. Zur Kennzeichnung kann jedem Ein- und Ausgang ein symbolischer Name vergeben werden, der dann vor der IEC-Adresse steht.

#### 5.1

#### Arbeiten in der Steuerungskonfiguration

Die Steuerungskonfiguration besteht aus zwei Fensterhälften: Links wird der 'Konfigurationsbaum' dargestellt, rechts erscheinen die jeweils zum selektierten Element passenden Konfigurationsdialoge auf einem oder mehreren Registerblättern.

Am Kopf des Konfigurationsbaumes steht das so genannte Root-Modul. Darunter folgen hierarchisch eingerückt die weiteren Elemente der Konfiguration Module verschiedener Typen (CAN, PROFIBUS, I/O), Kanäle oder Bitkanäle.



### **Selektieren von Elementen**

Zum Selektieren von Elementen führen Sie einen Mausklick auf das entsprechende Element aus bzw. bewegen Sie das gepunktete Rechteck mit den Pfeiltasten auf das gewünschte Element.

Elemente, die von einem Pluszeichen angeführt werden, sind Organisationselemente und enthalten Unterelemente. Zum Aufklappen selektieren Sie das Element und führen Sie einen Doppelklick auf dieses Pluszeichen aus oder drücken Sie die <Eingabetaste>. Auf die gleiche Weise werden aufgeklappte Elemente (Minuszeichen vor dem Element) zugeklappt.

### **Einfügen von Elementen, <Einfügen> <Element einfügen>, <Einfügen> <Unterelement anhängen>**

Abhängig von den vor dem Öffnen der SPS-Programmierungsumgebung zur Verfügung stehenden Konfigurationsdatei(en) und Gerätedatei(en) können Sie, wenn Sie ein Element im Konfigurationsbaum selektiert haben, weitere Elemente einfügen. Dazu stehen verschiedene Befehle zur Verfügung:

Menüpunkt <Einfügen> <Element einfügen>: Ein Element kann ausgewählt und vor dem selektierten Element eingefügt werden.

Menüpunkt <Einfügen> <Unterelement anhängen>: Ein Element kann ausgewählt und als letztes Unterelement an das selektierte Element angefügt werden.

Die jeweils wichtigsten Befehle finden Sie im Kontextmenü (rechte Maustaste).

### **Ersetzen/Umschalten von Elementen, <Extras> <Element ersetzen>**

Erlaubt es die Definition in der Konfigurationsdatei, können Sie ein im Konfigurationsbaum selektiertes Element durch ein anderes ersetzen. Dies umfasst auch das Umschalten von Kanälen, die so konfiguriert sind, dass sie als Ein- oder als Ausgang eingesetzt werden können. Verwenden Sie den Menüpunkt <Extras> <Element ersetzen>.

### **Neuberechnen der Modul-Adressen, <Extras> <Adressen berechnen>**

Ist in den allgemeinen Einstellungen der Steuerungskonfiguration die Option **Adressen automatisch** aktiviert, kann über den Menüpunkt <Extras> <Adressen berechnen> eine neue Adressberechnung durchgeführt werden, die ab dem selektierten Modul alle folgenden Elemente erfasst.

### **Zurück zur Standardkonfiguration, <Extras> <Standardkonfiguration>**

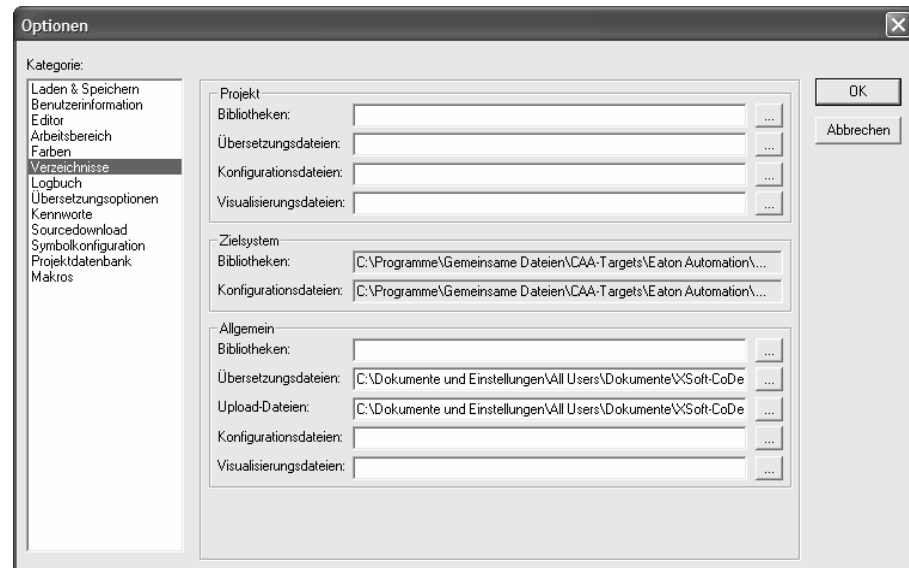
Über den Menüpunkt <Extras> <Standardkonfiguration> kann nach Änderungen im Konfigurationseditor die ursprüngliche Steuerungskonfiguration wiederhergestellt werden. Diese basiert auf der Konfigurationsdatei des entsprechenden SPS-Zielsystems.

## 5 Steuerungskonfiguration

### 5.1.1

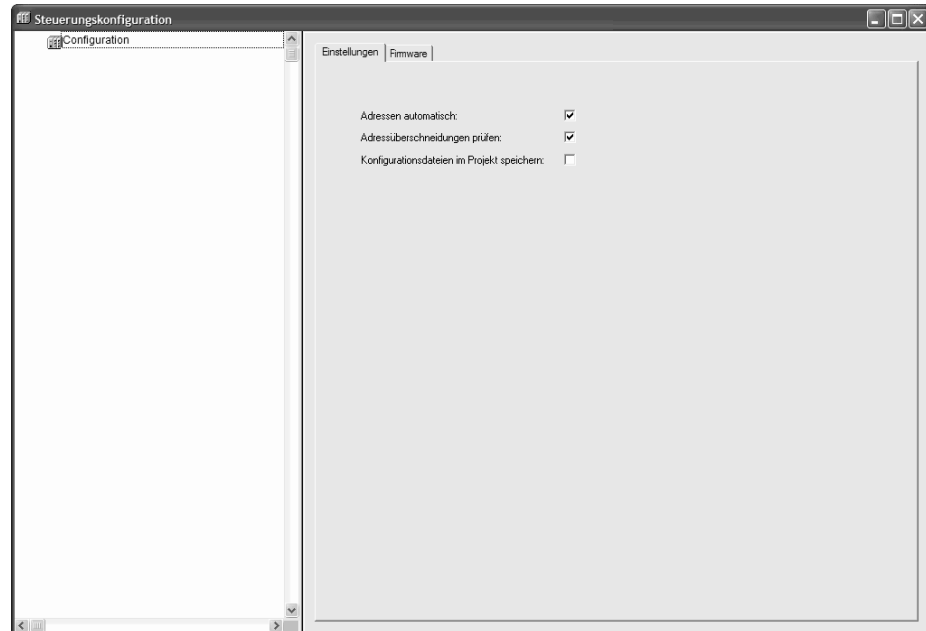
#### Projektspezifische Konfigurationsdatei(en) und Gerätedatei(en)

Kundenspezifische Verzeichnisse für Konfigurationsdatei(en) und Gerätedatei(en) können über den Menüpunkt <Projekt> <Optionen> projektspezifisch parametrierbar werden. Im projektspezifischen Bibliotheks- und Konfigurationspfad können zum Projekt relative Verzeichnispfade parametrierbar werden (z.B. `.\Libraries`, `.\PLCconf`).



- ➔ Nach dem Eintragen von projektspezifischen Verzeichnissen für die Konfigurations- und Gerätedatei(en) muss das Projekt geschlossen werden.
- Nach dem anschließenden Öffnen des Projekts sind die zusätzlichen Konfigurations- und Gerätedatei(en) in der Steuerungskonfiguration sichtbar und können eingefügt werden.

## 5.2

**Allgemeine Einstellungen****Adressen automatisch**

Bei aktivierter Option erhält jedes neu hinzugefügte Modul automatisch eine Adresse, die sich aus der des zuvor eingefügten Moduls plus dessen Größe ergibt. Wird ein Modul aus der Konfiguration entfernt, werden die Adressen der nachfolgenden Module automatisch angepasst. Über den Menüpunkt <Extras> <Adressen berechnen> werden die Adressen ab dem aktuell ausgewählten Knoten (Modul) neu ermittelt.

**Adressüberschneidungen prüfen**

Bei aktivierter Option werden beim Übersetzen des Projekts Adressüberschneidungen überprüft und gemeldet.

**Konfigurationsdateien im Projekt speichern**

Bei aktivierter Option werden über die Steuerungskonfiguration in das Projekt eingefügte Konfigurationsdatei(en) und Gerätedatei(en) im Projekt abgespeichert.

### 5.3

#### Konfiguration als CAN-Master

Die SPS-Programmierungsumgebung unterstützt eine Hardwarekonfiguration gemäß CANopen Draft Standard 301. Voraussetzung ist eine Konfigurationsdatei, die das Einfügen von CAN-Modulen zulässt.

Die im definierten Verzeichnis für die Konfigurationsdateien abgelegten EDS-Dateien (**E**lectronic **D**ata **S**heet) bzw. DCF-Dateien (Device Configuration File) stehen in der Konfiguration zur Verfügung. In einer EDS-Datei sind die Einstellungsmöglichkeiten eines Moduls beschrieben. Wird ein Modul über eine DCF-Datei eingefügt, können nur die IEC-Adressen verändert werden.

Die Module erhalten eine Konfiguration, die Ihr zeitliches und fehlerbezogenes Verhalten im Nachrichtenaustausch beschreibt. Zudem wird für jedes Modul das Mapping der PDOs (Process Data Objects) festgelegt, die zum Senden und Empfangen dienen (Dialoge PDO-Mapping Empfangen bzw. Senden). Die Werte der zur Verfügung stehenden SDOs (Service Data Objects) können angepasst werden (Dialog Service Data Objects).

Zusätzliche in der Gerätedatei angegebene Parameter eines CAN-Moduls bzw. CAN-Masters können im Dialog Modulparameter konfiguriert werden.

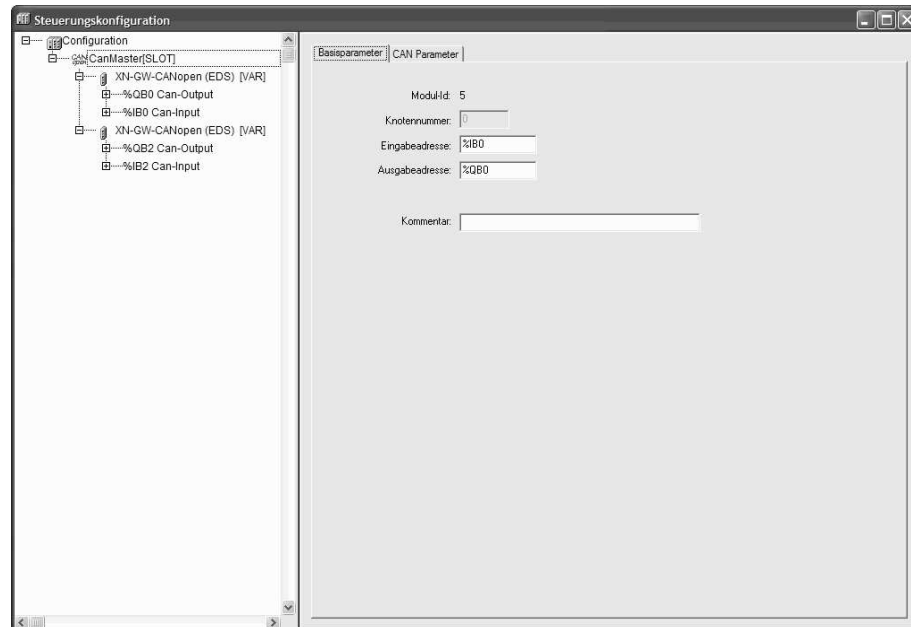
➔ In der Steuerungskonfiguration definierte PDOs, dessen Ein- bzw. Ausgänge im SPS-Programm nicht verwendet werden, werden standardmäßig im Prozessabbild nicht aktualisiert.

## 5.3.1

## Konfiguration des CAN-Masters

## 5.3.1.1

## Basisparameter

**Knotennummer**

Die Knotennummer ergibt sich aus einem Eintrag in der Konfigurationsdatei bzw. aus der Position im Konfigurationsbaum und kann vom Anwender nicht geändert werden (Nicht zu verwechseln mit der Node-Id, die vom Anwender parametrisiert werden kann).

**Eingabeadresse, Ausgabeadresse**

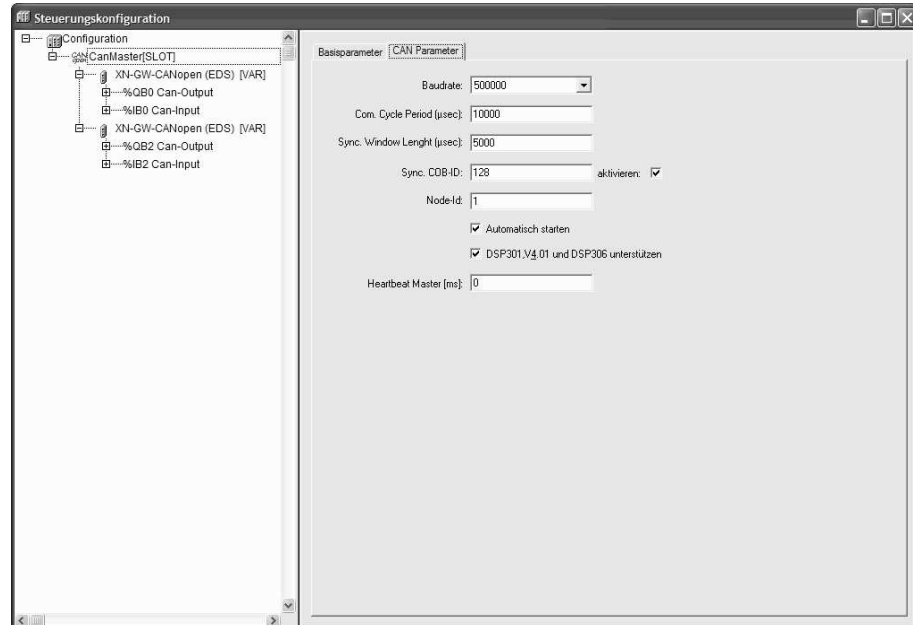
Hier werden die IEC-Adressen eingegeben, ab welchen die PDOs (Process Data Object) im Projekt angesprochen werden können. Welche Adressen bereits vorgegeben, welcher Adressmodus in der Konfiguration verwendet wird und ob an dieser Stelle noch editiert werden kann, hängt von den allgemeinen Einstellungen und den Definitionen in der Konfigurationsdatei ab.

## 5 Steuerungskonfiguration

### 5.3.1.2

#### CAN Parameter

In diesem Dialog werden die globalen Einstellungen und Überwachungsparameter für den CAN-Bus definiert.



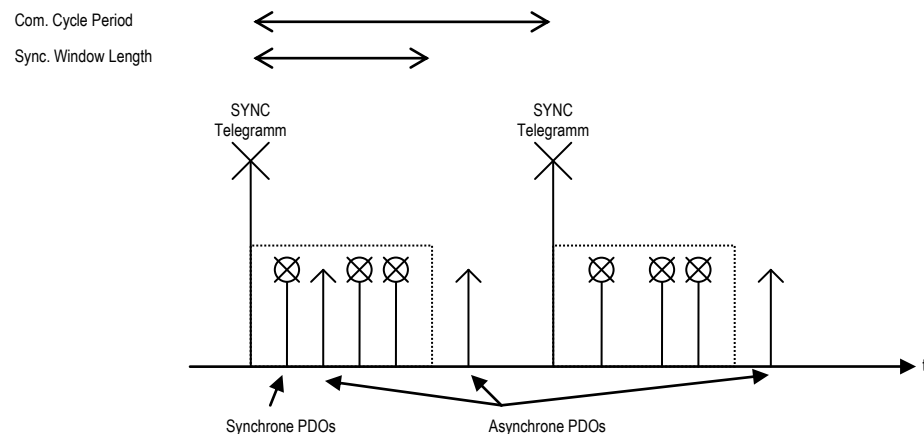
#### Baudrate

Baudrate mit der zwischen dem CAN-Master und den CAN-Modulen kommuniziert wird.

#### Com. Cycle Periode, Sync. Window Length, Sync. COB-ID

Es wird bei den PDOs (Process Data Object) zwischen synchronen und asynchronen Übertragungsarten unterschieden. Die **Communication Cycle Period** [µsec] ist das Zeitintervall in Mikrosekunden, in dem die Synchronisationsnachricht mit der eindeutigen Nummer **Sync. COB-ID** (Communication Object Identifier) versendet wird.

Nur wenn diese Option aktiviert wurde, erfolgt ein Versenden von Synchronisationsnachrichten zwischen CAN-Master und den CAN-Slaves.



Die synchronen PDOs werden direkt nach der Synchronisationsnachricht in dem in **Sync. Window Length** [ $\mu\text{sec}$ ] angegebenen Zeitfenster übertragen. Werden Com. Cycle Period und Sync. Window Length mit 0 belegt, so werden keine Synchronisationsnachrichten versendet.

➔ Wird die Synchronisationsnachricht über diesen Dialog parametrierbar, kann es aufgrund der Softwarearchitektur des SPS-Laufzeitsystems zu zeitlichen Jittern zwischen den einzelnen Synchronisationsnachrichten kommen.

Alternativ kann ein tasksynchrones Versenden der Synchronisations-Nachrichten im SPS-Programm projektiert werden.

### Node-Id

Die **Node-ID** ist die eindeutige Kennung des CAN-Moduls. Sie entspricht der am Modul selbst eingestellten Nummer zwischen 1 und 127 und muss dezimal eingegeben werden. (Nicht zu verwechseln mit der Knotennummer, die auch in der Steuerungskonfiguration verwendet wird)

### Automatisch starten

Ist die Option **Automatisch starten** aktiviert, wird beim Start des SPS-Programms sämtliche CAN-Knoten automatisch initialisiert und gestartet. Auch ein ausgefallener CAN-Knoten wird, nachdem er sich wieder am Bus gemeldet hat, automatisch gestartet. Ohne diese Option müssen die CAN-Knoten im SPS-Programm manuell gestartet werden.

### DSP301, V4.01 und DSP306 unterstützen

Muss aktiviert werden, wenn CAN-Knoten im SPS-Programm implementiert werden, welchen diesen CiA Standard unterstützen.

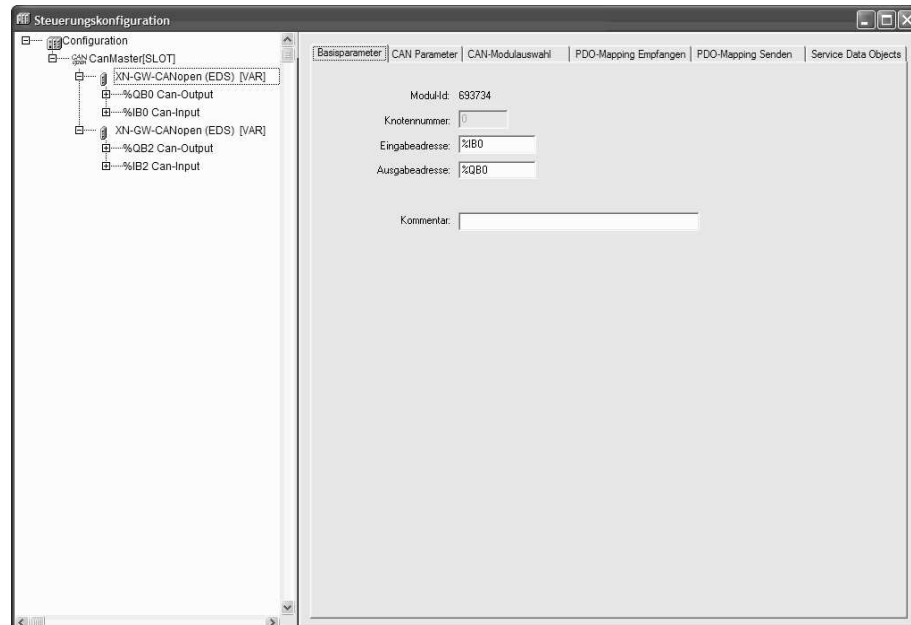
### Heartbeat Master

Überwachungsfunktionalität: Der CAN-Master versendet im entsprechend parametrieren Intervall Heartbeat-Nachrichten mit der **Guard COB-ID** (Communication Object Identifier) auf den Bus. Das Versenden der Heartbeat-Nachrichten ist standardmäßig mit dem Intervall von 0 ms parametrieren und somit deaktiviert.

## 5 Steuerungskonfiguration

### 5.3.2 Konfiguration der CAN-Knoten

#### 5.3.2.1 Basisparameter



#### **Knotennummer**

Die Knotennummer ergibt sich aus einem Eintrag in der Konfigurationsdatei bzw. aus der Position im Konfigurationsbaum und kann vom Anwender nicht geändert werden (Nicht zu verwechseln mit der Node-Id, die vom Anwender parametrisiert werden kann).

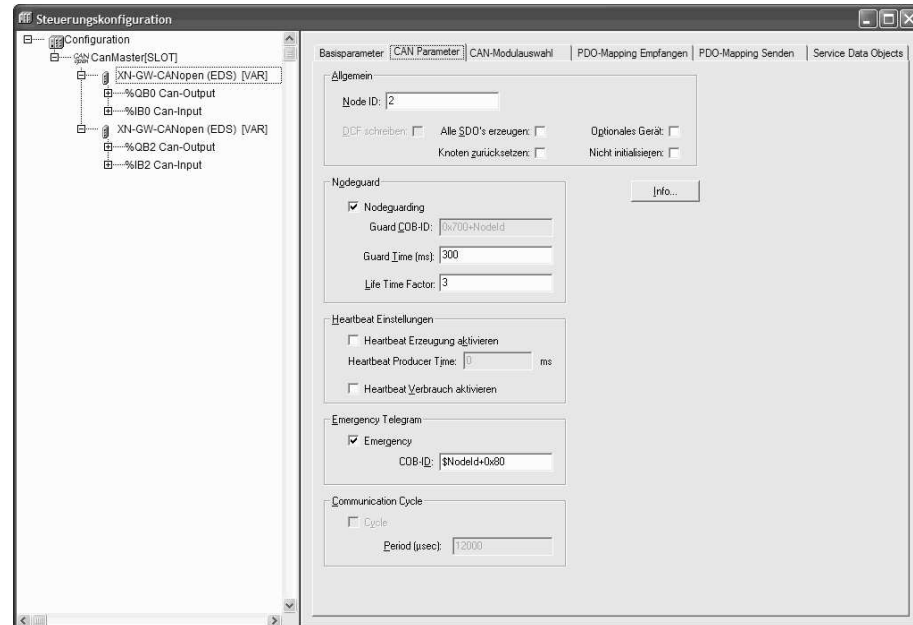
#### **Eingabeadresse, Ausgabeadresse**

Hier werden die IEC-Adressen eingegeben, ab welchen die PDOs (Process Data Object) im Projekt angesprochen werden können. Welche Adressen bereits vorgegeben, welcher Adressmodus in der Konfiguration verwendet wird und ob an dieser Stelle noch editiert werden kann, hängt von den allgemeinen Einstellungen und den Definitionen in der Konfigurationsdatei ab.



## 5.3.2.2

## CAN Parameter

**Node-Id**

Die **Node-ID** ist die eindeutige Kennung des CAN-Moduls Sie entspricht der am Modul selbst eingestellten Nummer zwischen 1 und 127 und muss dezimal eingegeben werden (Nicht zu verwechseln mit der 'Knotennummer', die auch in der Steuerungskonfiguration verwendet wird).

**DCF schreiben**

Bei aktivierter Option **DCF schreiben** wird nach dem Einfügen einer EDS-Datei im eingestellten Verzeichnis für Übersetzungsdateien eine DCF-Datei erstellt. Die Bezeichnung der DCF-Datei setzt sich aus dem Namen der EDS-Datei und der entsprechenden Node-ID zusammen.

**Alle SDOs erzeugen**

Bei aktivierter Option **Alle SDOs erzeugen** werden vor jedem Start des CAN-Knotens alle SDOs übertragen. Ansonsten werden nur die SDOs übertragen, welche sich vom Default-Wert der EDS-Datei unterscheiden.

**Knoten zurücksetzen**

Bei aktivierter Option **Knoten zurücksetzen** wird vor jedem Start des CAN-Knotens die SDO-Nachricht ,restore all default parameters' übertragen. (Index 16#1011 Sub-Index 1 mit Wert "Load", 16#23 11 10 01 6C 6F 61 64).

### Optionales Gerät

Bei aktivierter Option **Optionales Gerät** wird nach dem Start des SPS-Programms nur bedingt geprüft, ob der CAN-Knoten am CAN-Bus aktiv ist.

Ist der CAN-Knoten nicht am CAN-Bus, erfolgt keine Knotenüberwachung. Die Knotenüberwachung im CAN-Diagnosesystem erfolgt ohne den optionalen CAN-Knoten.

Wird der CAN-Knoten vor SPS-Start an den CAN-Bus angeschlossen, wird die Knotenüberwachung aktiviert und im CAN-Diagnosesystem ausgewertet.

Wird der optionale Knoten nachträglich an den CAN-Bus angeschlossen, führt dies im CAN-Diagnosesystem zu einem Knoten-Fehler.

### Nicht initialisieren

Bei aktivierter Option **Nicht initialisieren** erfolgt beim Starten des SPS-Programms weder eine Initialisierung noch ein Start des CAN-Knotens.

### Nodeguarding Einstellungen

Bei aktivierter Option **Nodeguarding** wird im Intervall, das bei **Guard Time** in Millisekunden angegeben ist, eine Nodeguard-Nachricht an das Modul gesendet. Meldet sich das Modul daraufhin nicht mit der angegebenen **Guard COB-ID** (Communication Object Identifier), so erhält es den Status Timeout.

Wird die unter **Life Time Factor** parametrisierte Anzahl von ungültigen Nodeguarding-Zyklen erreicht, so gilt das Modul als nicht OK. Der Status des Moduls wird im CAN-Diagnosesystem hinterlegt und kann im SPS-Programm ausgewertet werden.

Wird weder Guard Time noch Life Time Factor angegeben (0), erfolgt keine Überwachung des Moduls.

➔ Die Nodeguard-Funktionalität dient als Alternative zur Heartbeat-Funktionalität.

### Heartbeat Einstellungen

Bei aktivierter Option **Heartbeat Erzeugung aktivieren** versendet das Modul im entsprechend parametrisierten Intervall **Heartbeat Producer Time** seine Heartbeat-Nachrichten mit der **Guard COB-ID** (Communication Object Identifier). Der CAN-Master erwartet in diesem Intervall diese Heartbeat-Nachricht. Empfängt der CAN-Master diese Heartbeat-Nachricht nicht, gilt das Modul als nicht OK, Der Status des Moduls wird im CAN-Diagnosesystem hinterlegt und kann im SPS-Programm ausgewertet werden.

Bei aktivierter Option **Heartbeat Verbrauch aktivieren**, erwartet das Modul im entsprechend parametrisierten Intervall **Heartbeat Master** des CAN-Masters Heartbeat-Nachrichten mit der **Guard COB-ID** (Communication Object Identifier).

➔ Die Heartbeat-Funktionalität dient als Alternative zur Nodeguard-Funktionalität.

### Emergency Telegramm

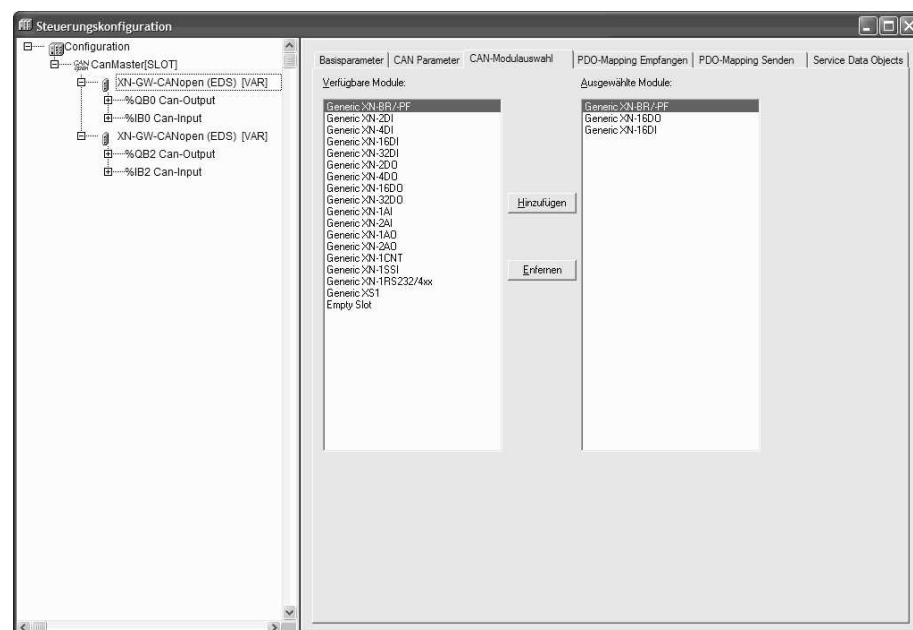
Ein Modul sendet bei internen Fehlern **Emergency**-Nachrichten mit einer eindeutigen **COB-ID**. Diese je nach Modul unterschiedlichen Nachrichten werden im CAN-Diagnosesystem hinterlegt und können im SPS-Programm ausgewertet werden.

### Info-Schaltfläche

Hinter der **Info**-Schaltfläche verbergen sich die Einträge "FileInfo" und "DeviceInfo" der EDS- bzw. DCF-Datei des jeweiligen Modulherstellers.

#### 5.3.2.3

### Modulauswahl

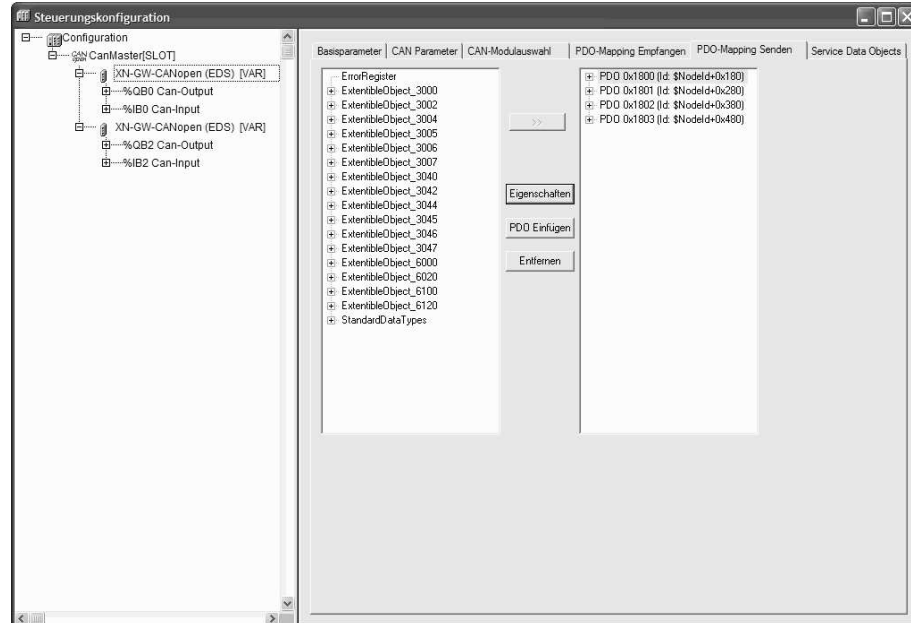


Ist das eingefügte CAN-Modul modular aufgebaut und unterstützt es die entsprechenden Standards (**DSP 301, V4.01 und DSP 306**), so erscheint die Registerkarte **CAN-Modulauswahl**. Die Konfiguration des entsprechenden Aufbaus kann durch **Hinzufügen** oder **Entfernen** der Module vorgenommen werden.

## 5 Steuerungskonfiguration

### 5.3.2.4

#### PDO-Mapping (Empfangen, Senden)



Die Registerkarten **PDO-Mapping Empfangen** und **PDO-Mapping Senden** ermöglichen, das in der EDS-Datei beschriebene "Mapping" des Moduls zu verändern.

Auf der linken Seite stehen alle "map-baren" Objekte der EDS-Datei zur Verfügung und können zu den PDOs (Process Data Object) über die Schaltfläche >> der rechten Seite hinzugefügt bzw. über die Schaltfläche **Entfernen** wieder entfernt werden.

Die StandardDataTypes können eingefügt werden, um im PDO Zwischenräume zu erzeugen.

Mit der Schaltfläche **PDO Einfügen** können weitere PDOs erzeugt und mit entsprechenden Objekten belegt werden. Über die eingefügten PDOs erfolgt die Zuordnung der Ein- und Ausgänge zu den IEC-Adressen. In der Steuerungskonfiguration werden die vorgenommenen Einstellungen nach Verlassen des Dialoges sichtbar. Die einzelnen Objekte können dort mit symbolischen Namen belegt werden.

➔ Die Kommunikationskanäle Senden und Empfangen sind auch Sicht des CAN-Moduls zu betrachten. Das heißt, im SPS-Programm projizierte Eingänge sind aus Sicht des CAN-Moduls im Sendekanal bzw. Ausgänge sind im Empfangskanal zu betrachten.

Über die Schaltfläche **Eigenschaften** lassen sich die in der Norm definierten Eigenschaften der PDOs editieren. Wird eine Option von dem Modul nicht unterstützt bzw. darf der Wert nicht verändert werden, so erscheint das Feld grau und kann nicht editiert werden.

### COB-ID

Jede PDO-Nachricht benötigt eine eindeutige **COB-ID** (Communication Object Identifier).

### Inhibit Time

Die **Inhibit Time** ist die minimale Zeit zwischen zwei Nachrichten dieses PDOs. Damit wird verhindert, dass PDOs, welche bei Änderungen des Wertes übertragen werden sollen, zu häufig versendet werden.

### Transmission Type, Number of Sync's, Event-Time

Bei **Transmission Type** erhalten Sie eine Auswahl von möglichen Übertragungsarten für dieses PDOs:

**azyklisch – synchron** : das PDO wird synchron übertragen, aber nicht periodisch.

**zyklisch – synchron** : das PDO wird synchron übertragen, wobei **Number of Sync's** die Anzahl der Synchronisationsnachrichten angibt, die zwischen zwei Übertragungen dieses PDOs liegen.

**synchron – nur RTR** : das PDO wird nach einer Synchronisationsnachricht aktualisiert, aber nicht versendet. Übertragen wird es nur auf eine explizite Anfrage (**Remote Transmission Request**)

**asynchron – nur RTR** : das PDO wird nur auf eine explizite Anfrage (**Remote Transmission Request**) aktualisiert und übertragen

**asynchron – herstellerepezifisch** und **asynchron – geräteprofilsspezifisch** : das PDO wird nur nach bestimmten Ereignissen übertragen. Dabei kann mit der **Event-Time** ein zusätzliches Ereignis definiert werden, welches ein Übertragen auslöst.

## 5 Steuerungskonfiguration

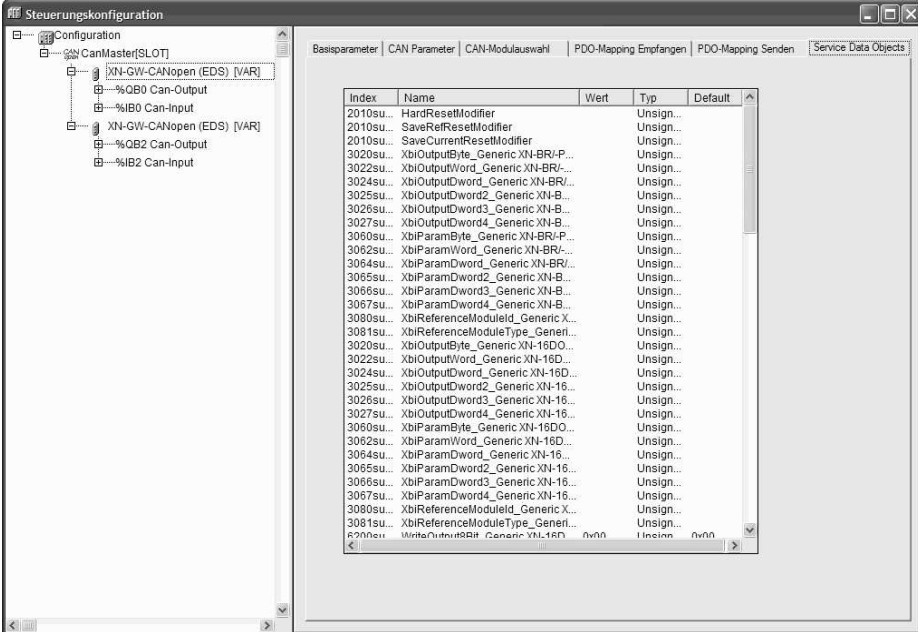
### 5.3.2.5

#### Service Data Objects

Hier werden alle Objekte der EDS- bzw. DCF-Datei aufgelistet, die im Bereich von Index 0x2000 bis 0x9FFF liegen und als beschreibbar definiert sind.

Zu jedem Objekt werden **Index**, **Name**, **Wert**, **Typ** und **Default** angegeben. Der Wert kann verändert werden. Markieren Sie den Wert und drücken Sie die <Leertaste>. Nach Änderung können Sie den Wert über die <Eingabetaste> bestätigen oder mit der <Escape-Taste> verwerfen.

Bei der Initialisierung des CAN-Bus werden die eingestellten Werte in Form von SDOs (Service Data Object) an das CAN-Modul übertragen.



The screenshot shows the 'Steuerungskonfiguration' window with the 'Service Data Objects' tab selected. The left pane shows a configuration tree with 'CanMaster[SLOT]' expanded to show two CANopen nodes (XN-GW-CANopen) and their associated I/O objects. The right pane displays a table of SDOs with columns for Index, Name, Wert, Typ, and Default.

Index	Name	Wert	Typ	Default
2010su...	HardResetModifier		Unsign...	
2010su...	SaveRefResetModifier		Unsign...	
2010su...	SaveCurrentResetModifier		Unsign...	
3020su...	XbiOutputByte_Generic XN-BR/P...		Unsign...	
3022su...	XbiOutputWord_Generic XN-BR/P...		Unsign...	
3024su...	XbiOutputDword_Generic XN-BR/P...		Unsign...	
3025su...	XbiOutputDword2_Generic XN-B...		Unsign...	
3026su...	XbiOutputDword3_Generic XN-B...		Unsign...	
3027su...	XbiOutputDword4_Generic XN-B...		Unsign...	
3060su...	XbiParamByte_Generic XN-BR/P...		Unsign...	
3062su...	XbiParamWord_Generic XN-BR/P...		Unsign...	
3064su...	XbiParamDword_Generic XN-BR/P...		Unsign...	
3065su...	XbiParamDword2_Generic XN-B...		Unsign...	
3066su...	XbiParamDword3_Generic XN-B...		Unsign...	
3067su...	XbiParamDword4_Generic XN-B...		Unsign...	
3080su...	XbiReferenceModuleId_Generic X...		Unsign...	
3081su...	XbiReferenceModuleType_Generi...		Unsign...	
3020su...	XbiOutputByte_Generic XN-16D...		Unsign...	
3022su...	XbiOutputWord_Generic XN-16D...		Unsign...	
3024su...	XbiOutputDword_Generic XN-16D...		Unsign...	
3025su...	XbiOutputDword2_Generic XN-16...		Unsign...	
3026su...	XbiOutputDword3_Generic XN-16...		Unsign...	
3027su...	XbiOutputDword4_Generic XN-16...		Unsign...	
3060su...	XbiParamByte_Generic XN-16D...		Unsign...	
3062su...	XbiParamWord_Generic XN-16D...		Unsign...	
3064su...	XbiParamDword_Generic XN-16...		Unsign...	
3065su...	XbiParamDword2_Generic XN-16...		Unsign...	
3066su...	XbiParamDword3_Generic XN-16...		Unsign...	
3067su...	XbiParamDword4_Generic XN-16...		Unsign...	
3080su...	XbiReferenceModuleId_Generic X...		Unsign...	
3081su...	XbiReferenceModuleType_Generi...		Unsign...	
6900su...	WriteOutputSR#_Generic XN-16D...	0x00	Unsign...	0x00

### 5.3.3

#### Busdiagnose

Für die Projektierung der Busdiagnose im SPS-Programm stehen Ihnen diverse Funktionsbibliotheken zur Verfügung.

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

### 5.4

#### Konfiguration als CAN-Device

Ein mit der SPS-Programmierungsumgebung "XSOFTE-CODESYS-2" programmiertes Gerät kann in einem CAN-Netzwerk auch als CANopen-Slave (im Folgenden als CAN-Device bezeichnet) erscheinen.

Anhand Parameter-Manager, Steuerungskonfiguration und Bibliotheksfunktionalitäten steht dem Anwender folgendes zur Verfügung:

- Konfigurieren von Variablen- oder Parameterlisten, welche bei der Datenkommunikation zwischen CAN-Device und CAN-Master ausgetauscht werden (Parameter-Manager, erstellen des Objektverzeichnisses).
- Konfigurieren der Nodeguarding/Heartbeat-Funktionalität, der Emergency-Nachricht, der Knotennummer Node-Id und der Baudrate
- Konfigurieren des Default PDO-Mappings anhand der Variablen- oder Parameterlisten.
- Erstellen der EDS-Datei, welche das CAN-Device beschreibt und ins SPS-Programm des CAN-Masters eingefügt werden kann.
- Bibliotheksfunktionalitäten CAN-Device-seitig für Überwachung und Verwaltung des Objektverzeichnisses

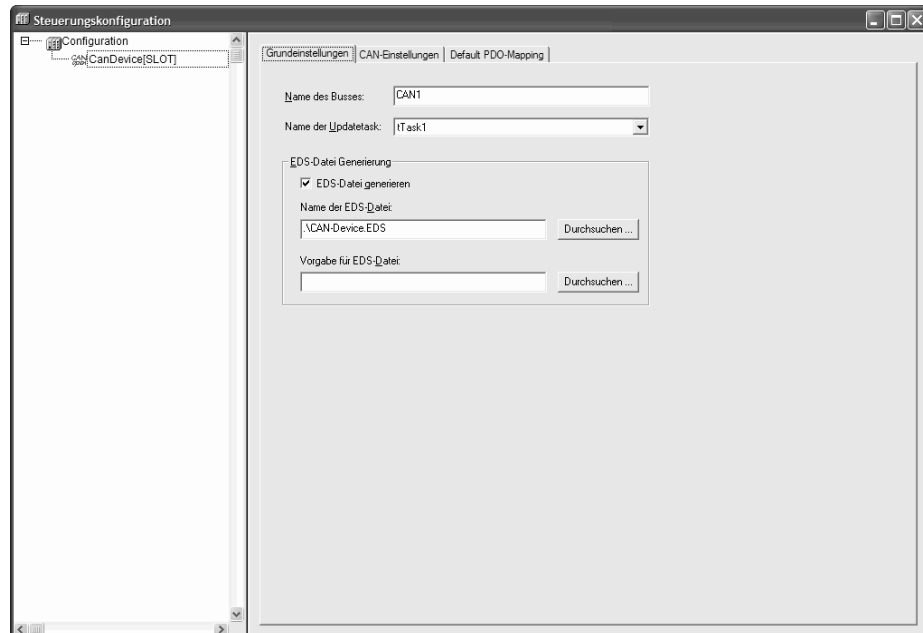
Folgende Funktionalitäten stehen nicht zur Verfügung:

- Dynamische SDO- bzw. SDO-Identifizierung
- SDO-Blocktransfer
- Implizite Erzeugung von Emergency-Nachrichten. Emergency-Nachrichten müssen immer von der Applikation erzeugt werden. Die Bibliothek stellt dazu einen FB zur Verfügung, der von der Applikation benutzt werden kann.
- Dynamische Änderungen der PDO-Eigenschaften/Laufzeiteigenschaften

## 5 Steuerungskonfiguration

### 5.4.1 Konfiguration des CAN-Devices

#### 5.4.1.1 Grundeinstellungen



#### Name des Busses

Wird zur Zeit nicht verwendet!

#### Name des Updatetasks

Name der Task, in der der Aufruf des CAN-Devices erfolgt.

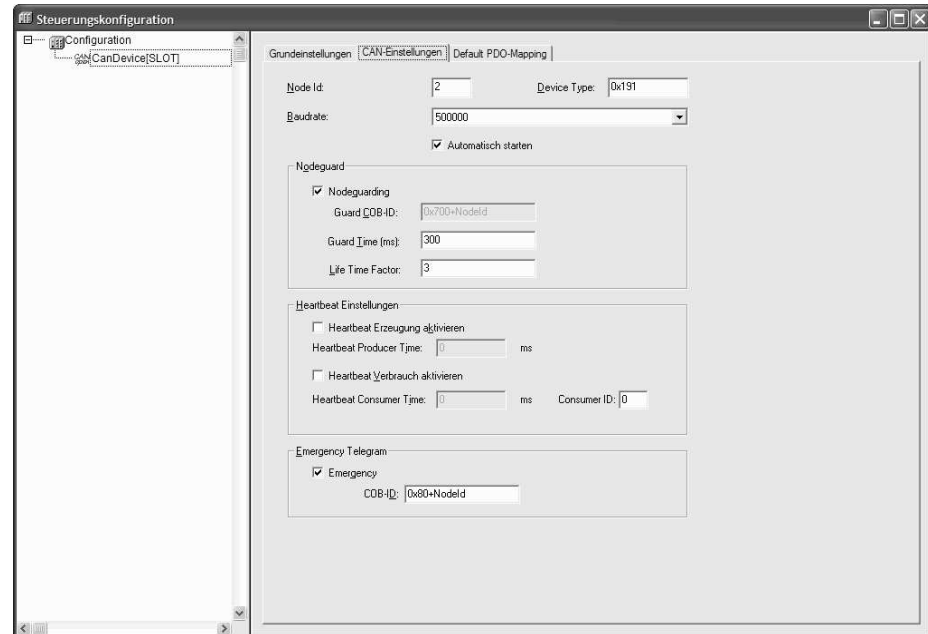
#### EDS-Datei Generierung

Bei aktivierter Option **EDS-Datei generieren** wird eine EDS-Datei mit **Name der EDS-Datei** generiert. Diese EDS-Datei kann anschließend in die Steuerungskonfiguration des CAN-Masters eingefügt werden.



## 5.4.1.2

## CAN-Einstellungen

**Node-Id**

Die **Node-ID** ist die eindeutige Kennung des CAN-Devices. Sie entspricht der am CAN-Device selbst eingestellten Nummer zwischen 1 und 127 und muss dezimal eingegeben werden (Nicht zu verwechseln mit der 'Knotennummer', die auch in der Steuerungskonfiguration verwendet wird).

➔ Die Node-ID kann mittels Bibliotheksfunktionen im SPS-Programm überlagert werden. Somit ist es möglich ein SPS-Programm für mehrere CAN-Devices zu schreiben, ohne dabei die CAN-Device Konfiguration zu ändern.

**Device Type**

Der **Device Type** (Defaultwert des Objekts 0x1000) des Geräts wird mit 0x191 (Standard IO Device) vorbelegt und kann vom Benutzer beliebig geändert werden.

**Baudrate**

Baudrate mit der zwischen dem CAN-Master und dem CAN-Device kommuniziert wird.

**Automatisch starten**

Ist die Option **Automatisch starten** aktiviert, wird beim Start des SPS-Programms der CAN-Bus automatisch initialisiert und gestartet.

## 5 Steuerungskonfiguration

### Nodeguarding Einstellungen

Parametrierung der **Guard COB-ID** (Communication Object Identifier), **Guard Time** in Millisekunden und **Life Time Factor**. Diese Parameter werden als Defaultwerte in die EDS-Datei des CAN-Devices eingefügt und können anschließend in der Steuerungskonfiguration des CAN-Masters geändert werden.

➔ Die Nodeguard-Funktionalität dient als Alternative zur Heartbeat-Funktionalität.

### Heartbeat Einstellungen

Parametrierung der **Heartbeat Producer Time** und **Heartbeat Consumer Time** in Millisekunden. Diese Parameter werden als Defaultwerte in die EDS-Datei des CAN-Devices eingefügt und können anschließend in der Steuerungskonfiguration des CAN-Masters geändert werden.

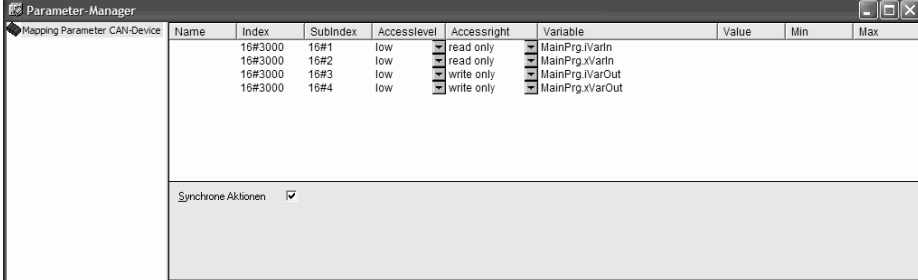
➔ Die Heartbeat-Funktionalität dient als Alternative zur Nodeguard-Funktionalität.

### Emergency Telegramm

Parametrierung der **Emergency**-Nachricht mit eindeutiger **COB-ID**. Diese Parameter werden als Defaultwerte in die EDS-Datei des CAN-Devices eingefügt und können anschließend in der Steuerungskonfiguration des CAN-Masters geändert werden.

### Default PDO-Mapping

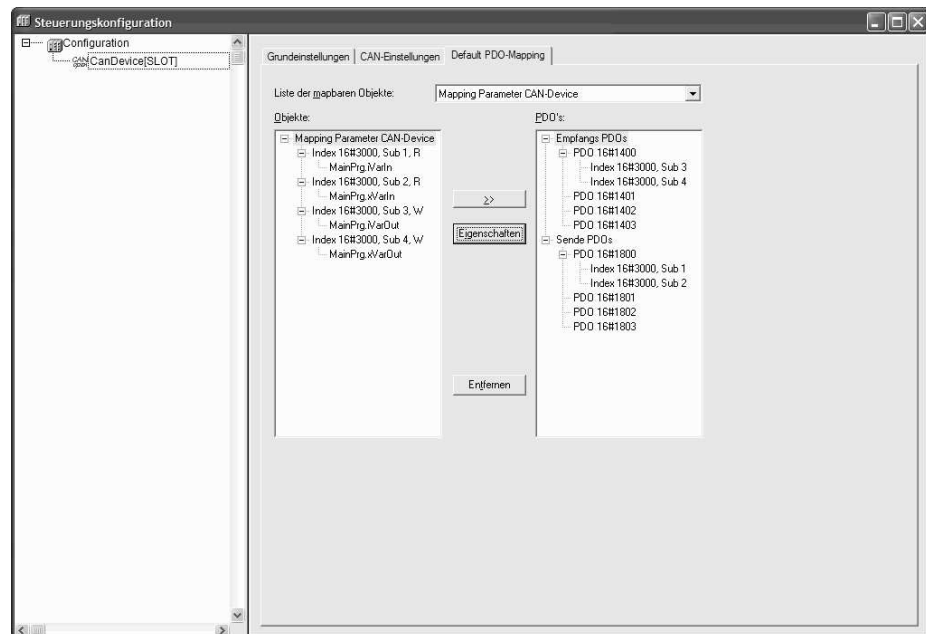
Im Parameter-Manager werden die Variablen- oder Parameterlisten erstellt, die danach im PDO-Mapping dem CAN-Device zur Verfügung stehen



Name	Index	Subindex	Accesslevel	Accessright	Variable	Value	Min	Max
	16#3000	16#1	low	read only	MainPrg.iVarIn			
	16#3000	16#2	low	read only	MainPrg.xVarIn			
	16#3000	16#3	low	write only	MainPrg.iVarOut			
	16#3000	16#4	low	write only	MainPrg.xVarOut			

Synchrone Aktionen

➔ Um den Parameter-Manager öffnen zu können, muss in der Zielsystemeinstellung im Registerblatt 'Netzfunktionen' der Parameter-Manager aktiviert und mit sinnvollen Index/Subindexbereichen parametrisiert werden.

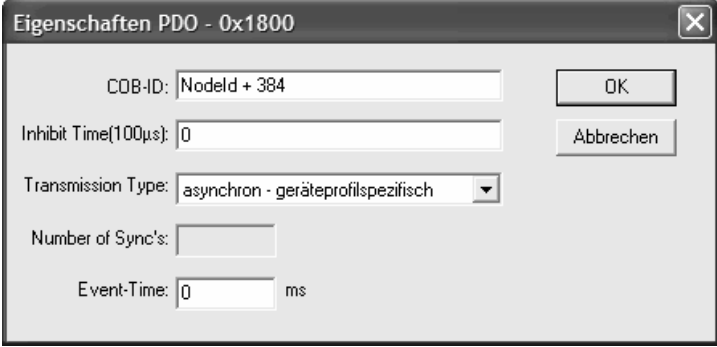


Die Registerkarte **Default-PDO-Mapping** ermöglicht, die im Parameter-Manager definierten Einträge dem "Mapping" des CAN-Devices zuzuordnen bzw. zu verändern.

Auf der linken Seite stehen alle "map-baren" Objekte zur Verfügung und können zu den PDOs (Process Data Object) über die Schaltfläche >> der rechten Seite hinzugefügt bzw. über die Schaltfläche **Entfernen** wieder entfernt werden.

## 5 Steuerungskonfiguration

Über die Schaltfläche **Eigenschaften** lassen sich die Eigenschaften der PDOs editieren. Diese können anschließend in der Steuerungskonfiguration des CAN-Masters geändert werden.



The screenshot shows a dialog box titled "Eigenschaften PDO - 0x1800". It contains the following fields and controls:

- COB-ID:
- Inhibit Time(100µs):
- Transmission Type:
- Number of Sync's:
- Event-Time:  ms
- Buttons: OK, Abbrechen

### 5.4.2

#### Konfiguration des CAN-Devices im CAN-Master

Nach dem Einfügen des CAN-Devices steht dem Anwender in der Steuerungskonfiguration folgendes zur Verfügung:

- Konfigurieren der Nodeguarding/Heartbeat-Funktionalität, der Emergency-Nachricht und der Knotennummer Node-Id und der Baudrate
- Konfigurieren des PDO-Mappings anhand des Default PDO-Mappings der EDS-Datei

➔ Detaillierte Informationen entnehmen Sie bitte dem Kapitel 'Konfiguration der CAN-Knoten' (➔ Kap. 5.3.2).

### 5.5

#### Konfiguration als Profibus-Master

Die SPS-Programmierungsumgebung unterstützt eine Hardwarekonfiguration gemäß Profibus-DP Standard. Voraussetzung ist eine Konfigurationsdatei, die das Einfügen von Profibus-Modulen zulässt.

Die im definierten Verzeichnis für die Konfigurationsdateien abgelegten GSD-Dateien (**Geräte Stamm Datei**) stehen in der Konfiguration zur Verfügung. In einer GSD-Datei sind die Einstellungsmöglichkeiten eines Moduls beschrieben.

Die Module erhalten eine Konfiguration, die ihr zeitliches und fehlerbezogenes Verhalten im Nachrichtenaustausch beschreibt.

➔ In der Steuerungskonfiguration definierte Nachrichten, dessen Ein- bzw. Ausgänge im SPS-Programm nicht verwendet werden, werden standardmäßig im Prozessabbild nicht aktualisiert.

## 5 Steuerungskonfiguration

### 5.5.1

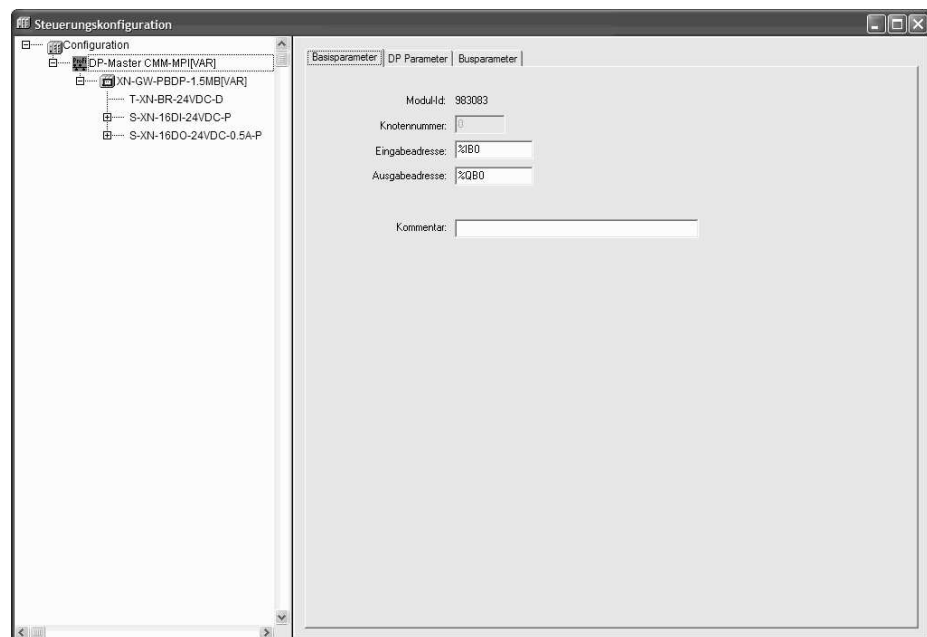
#### Konfiguration des Profibus-Masters

In der Steuerungskonfiguration sind folgende Profibus-Master implementiert:

Bezeichnung	Beschreibung
DP-Master CMM-MPI (MPDP4D09.GSD)	Profibus-Master DP-V1 für Profibus DP / FMS und MPI

#### 5.5.1.1

#### Basisparameter



#### Knotennummer

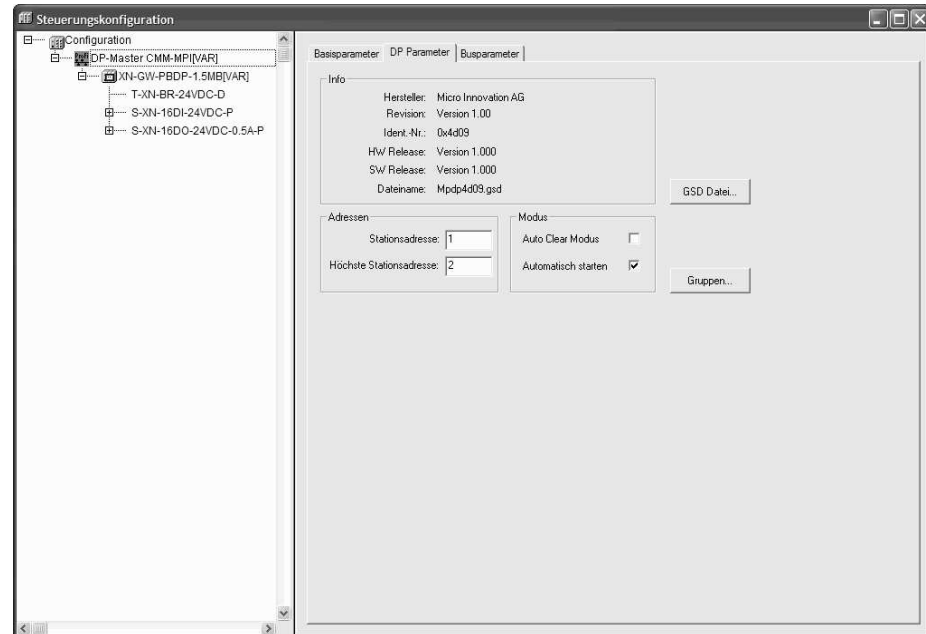
Die Knotennummer ergibt sich aus einem Eintrag in der Konfigurationsdatei bzw. aus der Position im Konfigurationsbaum und kann vom Anwender nicht geändert werden (Nicht zu verwechseln mit der Node-Id, die vom Anwender parametrisiert werden kann).

#### Eingabeadresse, Ausgabeadresse

Hier werden die IEC-Adressen eingegeben, ab welchen die Eingänge bzw. Ausgänge im Projekt angesprochen werden können. Welche Adressen bereits vorgegeben, welcher Adressmodus in der Konfiguration verwendet wird und ob an dieser Stelle noch editiert werden kann, hängt von den allgemeinen Einstellungen und den Definitionen in der Konfigurationsdatei ab.

## 5.5.1.2

## DP Parameter

**Stationsadresse**

Die **Stationsadresse** ist die eindeutige Kennung des Profibus-Masters. Sie ist mit dem Wert 1 vordefiniert und muss vom Anwender nicht editiert werden.

**Höchste Stationsadresse**

Sie entspricht der höchsten auf dem Profibus vorhandenen Stationsadresse (vom Anwender einstellbar, oder berechnet durch die SPS-Programmierungsumgebung).

➔ Vergewissern Sie sich, dass eine korrekte HSA eingetragen ist!  
 Folgende HSAs werden unterstützt: 15, 31, 63 und 126.

**Auto Clear Modus**

Jeder Slave hat einen Watchdog, der überwacht, ob der Slave regelmäßig vom Master angesprochen wird. Ist dies nicht der Fall, geht der Slave eigenständig in den sicheren Zustand. Bedeutet der Ausfall von Slaves ein hohes Sicherheitsrisiko in einer Anlage, wird mit der Option **Auto Clear Modus** bewirkt, dass der Master bei Ausfall eines Slaves automatisch vom Zustand Operate in den Zustand Clear wechselt. Dies führt wiederum dazu, dass alle anderen in den sicheren Zustand gehen, d.h. digitale Ausgänge werden auf 0 gesetzt und analoge geben vorher projizierte Ersatzwerte aus (FailSafe).

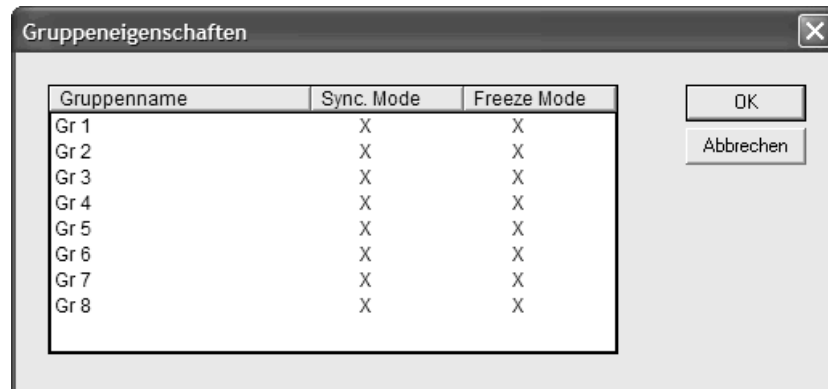
## 5 Steuerungskonfiguration

### Automatisch Starten

Bei aktivierter Option **Automatisch starten** wird beim Start des SPS-Programms der Profibus automatisch initialisiert und gestartet.

➔ Muss immer angewählt sein (ein manueller Start wird nicht unterstützt)

### Gruppen



➔ Freeze- und Sync-Mode werden nicht unterstützt.

Die Schaltfläche **Gruppen** führt zum Dialog 'Gruppeneigenschaften'. Die Gruppeneigenschaften beziehen sich auf die dem Master zugeordneten Slaves. Bis zu acht Gruppen können eingerichtet werden. Stellen Sie für jede Gruppe ein, ob sie im **Freeze-Mode** und/oder **Sync-Mode** betrieben werden soll. Durch die Zuordnung der Slaves zu verschiedenen Gruppen kann der Datenaustausch vom Master über ein Global-Control-Kommando synchronisiert werden. Mit einem Freeze-Kommando veranlasst ein Master einen Slave oder eine Gruppe, die Eingänge im momentanen Zustand 'einzufrieren' und diese Daten beim darauf folgenden Datenaustausch zu übertragen. Mit einem Sync-Kommando werden die Slaves veranlasst, die im folgenden Datenaustausch vom Master empfangenen Daten mit dem nächsten Sync-Kommando zeitlich synchron an die Ausgänge durchzuschalten.

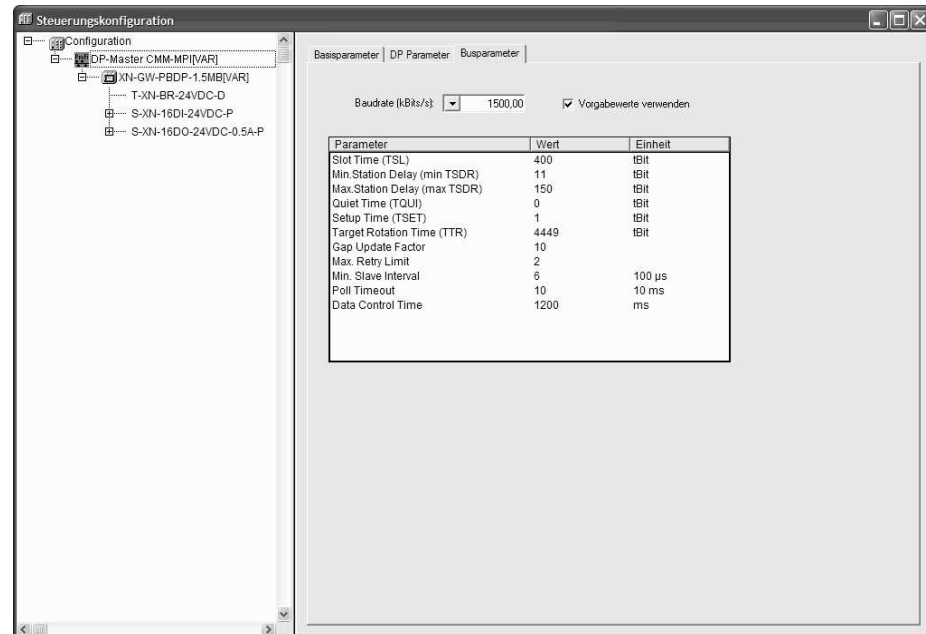
Im Dialog 'Gruppeneigenschaften' lassen sich die Gruppennamen editieren sowie die entsprechenden Übertragungsarten aktivieren bzw. deaktivieren.



## 5.5.1.3

**Busparameter**

Die Busparameter beschreiben das Zeitverhalten der Kommunikation.

**Baudrate**

Baudrate mit der zwischen dem Profibus-Master und den Profibus-Slaves kommuniziert wird.

**Vorgabewerte verwenden**

Ist die Option **Vorgabewerte verwenden** aktiviert, wird ein Default-Parametersatz für diese Baudrate verwendet. Wird diese Option nicht verwendet, können die Parameter manuell verändert werden.

➔ Diese Option nimmt keine Rücksicht auf die konfigurierten I/Os. Wenn optimierte Parameter unter Berücksichtigung der Bustopologie benötigt werden, so müssen diese Parameter mit externen Profibus-Konfiguratoren (z.B. Sycon von Hilscher) ermittelt werden.

## 5 Steuerungskonfiguration

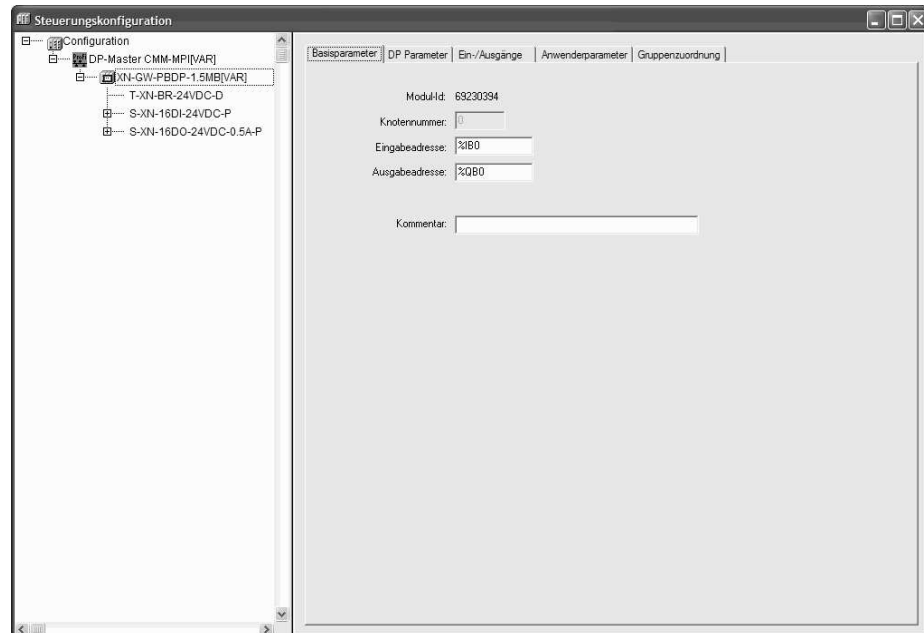
### Parameter

Zu jedem Objekt werden **Name**, **Wert** und **Einheit** angegeben. Der Wert kann verändert werden. Markieren Sie den Wert und drücken Sie die <Leertaste>. Nach Änderung können Sie den Wert über die <Eingabetaste> bestätigen oder mit der <Escape-Taste> verwerfen.

Parameter	Beschreibung
Slot Time (TSL)	Zeit, die der Master nach Aussendung eines Aufruf-Telegramms maximal auf den Empfang des ersten Zeichens des Antwort-Telegramms eines Slaves wartet.
Min. Station Delay (Min. TSDR)	Minimale Reaktionszeit, nach der ein Slave antworten darf.
Max. Station Delay (Max. TSDR)	Maximale Reaktionszeit, innerhalb der ein Slave antworten muss.
Quiet Time (TQUI)	Ruhezeit, die beim Umsetzen von NRZ-Signalen (Non Return to Zero) auf andere Kodierungen zu berücksichtigen ist (Umschaltzeit für Repeater).
Setup Time (TSET)	Zeit, die zwischen dem Empfang eines Telegramms und der darauf folgenden Reaktion im Slave vergehen darf.
Target Rotation Time (TTR)	Token-Soll-Umlaufzeit: Projektiertes Zeitintervall, in dem ein Master den Token erhalten soll. Ergibt sich aus der Summe der Token-Halte-Zeiten aller Master am Bus.
Gap Update Factor (GAP)	GAP-Aktualisierungsfaktor: Anzahl der Busumläufe, nach denen im GAP des Masters (Adressbereich von der eigenen Busadresse bis zur Adresse des nächsten aktiven Teilnehmers) nach einer weiteren, neu hinzugekommenen, aktiven Station gesucht wird.
Max. Retry Limit	Maximale Anzahl der erneuten Aufrufversuche des Masters, wenn er vom Slave keine gültige Antwort empfangen hat.
Min. Slave Interval	Zeit zwischen zwei Buszyklen, in denen ein Slave eine Anforderung des Masters bearbeiten kann (Zeitbasis 100µs). Der hier eingetragene Wert muss mit den jeweiligen Vorgaben in den GSD-Dateien der Slaves abgestimmt sein.
Poll Timeout	Maximale Zeit, nach der die Antwort des Masters bei einer Master-Master-Kommunikation vom Requester (DP-Master Klasse 2) abgeholt sein muss (Zeitbasis 1ms).
Data Control Time	Zeit, in der der Master den zugeordneten Slaves seinen Betriebszustand mitteilt. Gleichzeitig überwacht der Master, ob mit den Slaves innerhalb dieser Zeit jeweils mindestens ein Nutzdatenaustausch stattgefunden hat und aktualisiert die Data_Transfer_List.

## 5.5.2 Konfiguration der Profibus-Slaves

### 5.5.2.1 Basisparameter



#### Knotennummer

Die Knotennummer ergibt sich aus einem Eintrag in der Konfigurationsdatei bzw. aus der Position im Konfigurationsbaum und kann vom Anwender nicht geändert werden (Nicht zu verwechseln mit der Stationsadresse, die vom Anwender parametrisiert werden kann).

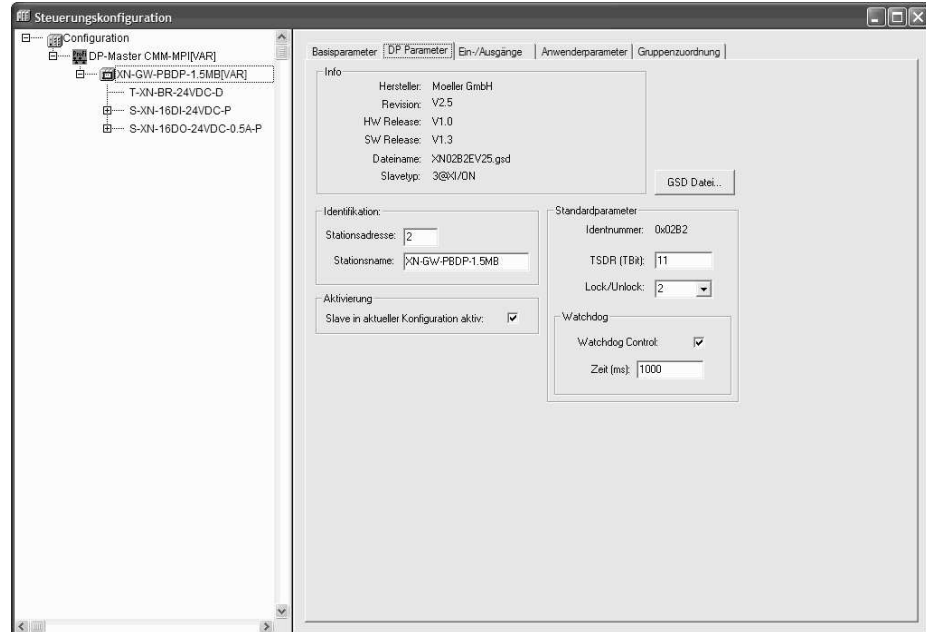
#### Eingabeadresse, Ausgabeadresse

Hier werden die IEC-Adressen eingegeben, ab welchen die Eingänge bzw. Ausgänge im Projekt angesprochen werden können. Welche Adressen bereits vorgegeben, welcher Adressmodus in der Konfiguration verwendet wird und ob an dieser Stelle noch editiert werden kann, hängt von den allgemeinen Einstellungen und den Definitionen in der Konfigurationsdatei ab.

## 5 Steuerungskonfiguration

### 5.5.2.2

#### DP Parameter



#### Stationsadresse

Die **Stationsadresse** ist die eindeutige Kennung des Profibus-Moduls. Sie entspricht der am Modul selbst eingestellten Nummer zwischen 0 und 126 und muss dezimal eingegeben werden.

#### Aktivierung

Bei deaktivierter Option **Aktivierung**, werden die Konfigurationsdaten des Slaves beim Download an den Koppler übertragen, ein Datenaustausch über den Bus findet jedoch nicht statt.

#### Standardparameter

Die **Identnummer** wird von der PNO vergeben und ist die eindeutige Identifikationsnummer des entsprechenden Gerätetyps. Sie stellt die eindeutige Referenz zwischen Slave und der zugehörigen GSD-Datei her. Der Parameter **TDSR** (Time Station Delay Responder) entspricht der Reaktionszeit nach der der Slave frühestens dem Master antworten darf. Zeiteinheit für die Übertragung eines Bits über PROFIBUS: Kehrwert der Übertragungsrage; z.B. 1 TBit bei 12MBaud=1/12.000.000 Bit/sek=83ns. Über den Parameter **Lock/Unlock** wird der Slave für andere Master gesperrt oder freigegeben:

- 0: min. TSDR und slave-spezifische Parameter dürfen überschrieben werden
- 1: Slave für andere Master freigegeben
- 2: Slave für andere Master gesperrt, alle Parameter werden übernommen
- 3: Slave für andere Master erneut freigegeben

### Watchdog

Bei aktivierter Option **Watchdog-Contol** gilt die parametrisierte Watchdogzeit. Wird ein Slave nicht innerhalb dieser Zeit vom Master angesprochen, wechselt der Slave in den Initialisierungszustand.

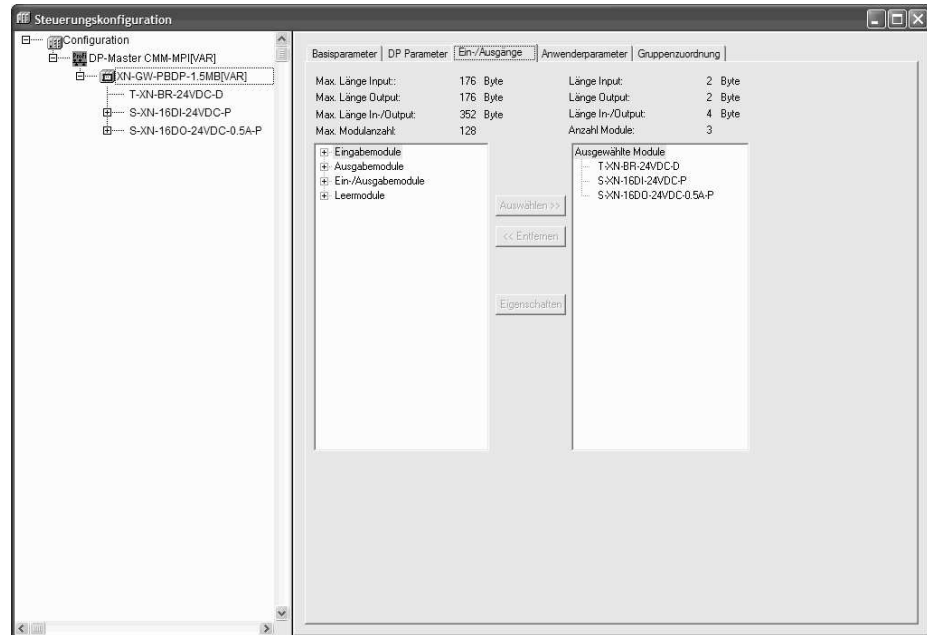
➔ Die Watchdog Control Zeit muss bei allen Slaves gleich eingestellt sein.

## 5 Steuerungskonfiguration

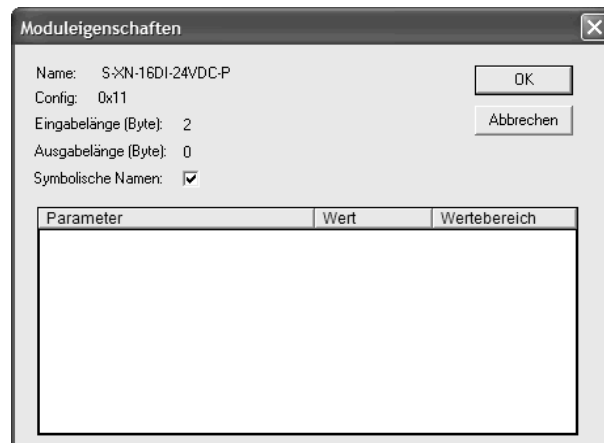
### 5.5.2.3

#### Modulauswahl

Ist der eingefügte Profibus-Knoten modular aufgebaut und unterstützt er die entsprechenden Standards, so erscheint die entsprechende Registerkarte. Die Konfiguration des entsprechenden Aufbaus kann durch **Auswählen** oder **Entfernen** der Module vorgenommen werden.



Die Schaltfläche **Eigenschaften** führt zum Dialog 'Moduleigenschaften' des aktuell selektierten Ein- oder Ausgangsmoduls. Er zeigt den Namen, die Config (Kodierung der Modulbeschreibung nach PROFIBUS-Norm) und die Ein- und Ausgabelänge des Moduls in Byte. Enthält die Modulbeschreibung in der GSD-Datei neben dem Standardsatz zusätzliche spezifische Parameter, werden diese hier mit Wert und Wertebereich aufgelistet. Ist die Option **Symbolische Namen** aktiviert, werden dabei die symbolischen Namen verwendet.



## 5.5.2.4

**Anwenderparameter**

Hier sind verschiedene in der GSD-Datei definierte erweiterte Parameter eines Slaves aufgelistet. Die Spalte **Parameter** zeigt den Namen des Parameters. Die in der Spalte **Wert** eingetragenen Parameterwerte können verändert werden. Außerdem ist der **Wertebereich** angegeben.

Sind in der GSD-Datei für die Parameter auch symbolische Namen vergeben, kann die Option **Symbolische Namen** aktiviert werden, sodass die Werte mit diesen dargestellt werden. Zur Information ist außerdem über der Tabelle die Länge der Anwenderparameter angegeben.

Steuerungskonfiguration

Configuration

- DP-Master CMM-MPI[VAR]
  - KN-GW-PBDP-1.5MB[VAR]
    - T-XN-BR-24VDC-D
    - S-XN-18DI-24VDC-P
    - S-XN-18DO-24VDC-0.5A-P

Basisparameter | DP Parameter | Ein-/Ausgänge | **Anwenderparameter** | Gruppenzuordnung

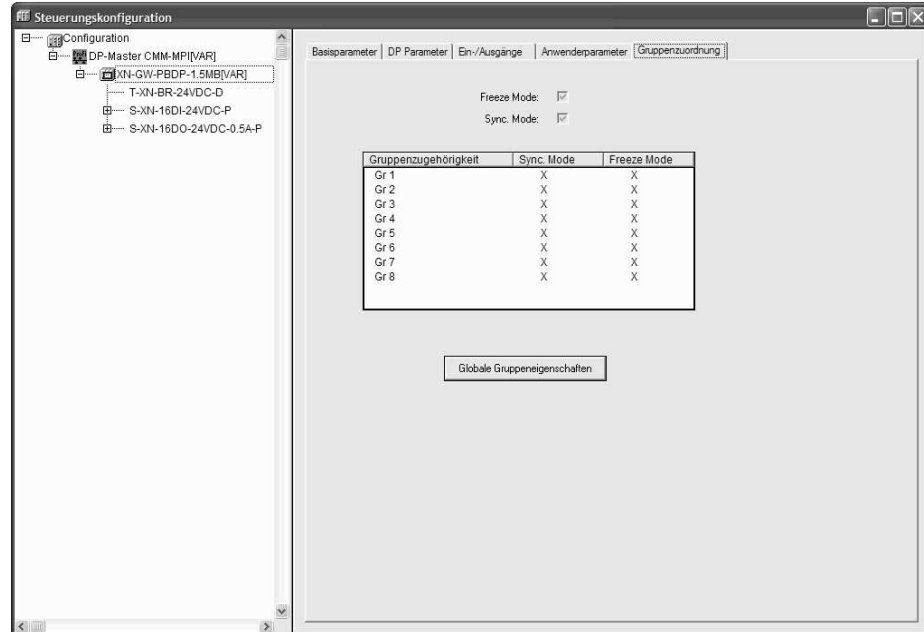
Länge der Anwenderparameter in Byte: 5      Symbolische Namen:

Parameter	Wert	Wertebereich
"outputs fieldbus error:"	output 0	BitArea
"outputs module exchange error:"	output 0	BitArea
"outputs module exchange:"	output 0	BitArea
"integer data format:"	LSB first	Bit (0) 0
"diagnostics from modules:"	activate	Bit (1) 0
"gateway diagnostics:"	device related diagnostics	Bit (5) 0
"station configuration:"	do not allow changes	Bit (3) 0
"IOassistant-Force Mode:"	release	Bit (4) 0

## 5 Steuerungskonfiguration

### 5.5.2.5

#### Gruppenzuordnung



➔ Freeze- und Sync-Mode werden nicht unterstützt.

Dieser Dialog dient der Zuordnung des Slaves zu einer oder mehrerer der möglichen acht Gruppen. Die allgemeingültigen Gruppeneigenschaften (Sync-Mode und/oder Freeze-Mode) hingegen werden bei der Konfiguration der Master-Eigenschaften definiert. Über die Schaltfläche **Globale Gruppeneigenschaften** gelangt man ebenfalls zu diesem Dialog.

Die Gruppe(n), denen der Slave zugeordnet wurde, werden mit einem Pluszeichen markiert. Das Zuordnen bzw. Entfernen des Slaves zu/aus einer Gruppe erreicht man, indem man den Gruppennamen in der Spalte Gruppenzugehörigkeit selektiert und mit der rechten Maustaste 'Slave zu Gruppe hinzufügen' bzw. 'Slave aus Gruppe entfernen' wählt oder nochmals mit der Maus links neben den Gruppennamen klickt.

Ein Slave-Gerät kann nur solchen Gruppen zugeordnet werden, deren Eigenschaften es unterstützt. Die diesbezüglichen Eigenschaften des jeweiligen Slaves (Sync-Mode / Freeze-Mode) werden oberhalb der Tabelle angezeigt. Vom dem Gerät unterstützte Übertragungsarten sind mit einem Haken versehen.



### 5.5.3

#### **Busdiagnose**

Für die Projektierung der Busdiagnose im SPS-Programm stehen Ihnen diverse Funktionsbibliotheken zur Verfügung.

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

### 5.6

#### SmartWire-DT Konfiguration

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

Bedingung zum Einhängen von SmartWire-DT Teilnehmern ist eine mit den Teilnehmern enthaltene Konfigurationsdatei 'SWD-Master.gsd'. Diese Datei befindet sich im Ordner 'SWDConf'.

- ➔ Eine allgemeine Beschreibung des SmartWire-DT Systems ist im Dokument "SmartWire-DT System", MN05006002Z-DE nachzulesen.
- ➔ Funktionsbeschreibungen sowie Angaben zum Umfang der Ein-/Ausgangsdaten der einzelnen SmartWire-DT Teilnehmer finden Sie im Dokument "SmartWire-DT Teilnehmer", MN05006001Z-DE.

#### Systemvoraussetzung

SPS-Zielsystem	SPS-Laufzeitsystem	Betriebssystem (OS)
≥ XV-1xx-V2.3.9 SP3	≥ PLCWinCE V 2.4.10 (xxx)	Windows CE 5.0 ≥ Image Release 2.26.0 (xxx)

- ➔ 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.

## 5.6.1

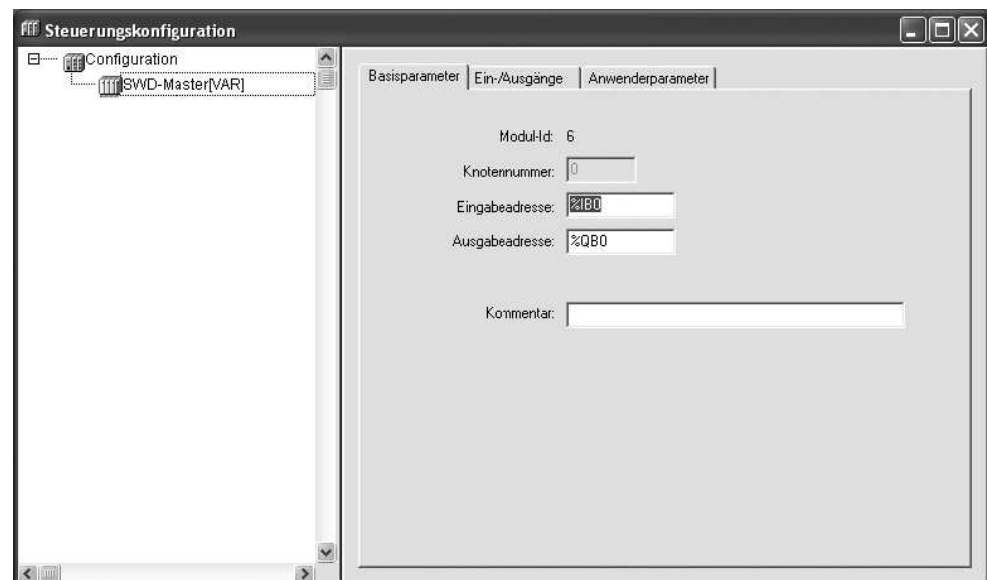
**Konfiguration des SWD-Masters und SWD-Modulauswahl**

In der Steuerungskonfiguration wird folgender SWD-Master bereitgestellt:

Bezeichnung	Beschreibung
SWD-Master	SWD-Koordinator

Wählen Sie in der Steuerungskonfiguration Unterelement anhängen → SWD-Master... aus.

## 5.6.1.1

**Basisparameter****Knotennummer**

Die Knotennummer ergibt sich aus einem Eintrag in der Konfigurationsdatei bzw. aus der Position im Konfigurationsbaum und kann vom Anwender nicht geändert werden.

**Eingabeadresse, Ausgabeadresse**

Hier geben Sie die IEC-Adressen ein, ab welchen die Eingänge bzw. Ausgänge im Projekt angesprochen werden können. Welche Adressen bereits vorgegeben sind, welcher Adressmodus in der Konfiguration verwendet wird und ob an dieser Stelle noch editiert werden kann, hängt von den allgemeinen Einstellungen und den Definitionen in der Konfigurationsdatei ab.

## 5 Steuerungskonfiguration

### 5.6.1.2

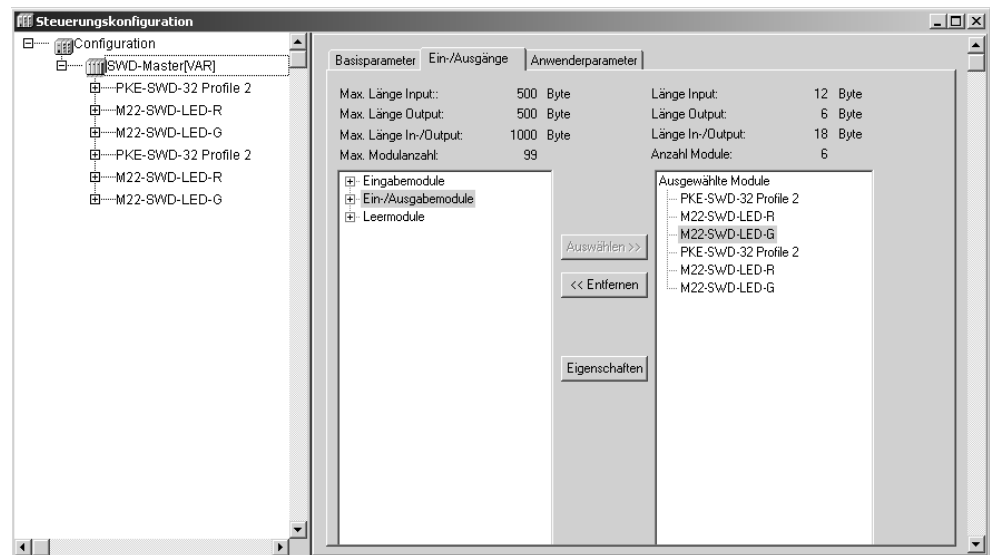
#### Teilnehmerauswahl

Wechseln Sie zum Register **Ein-/Ausgänge**

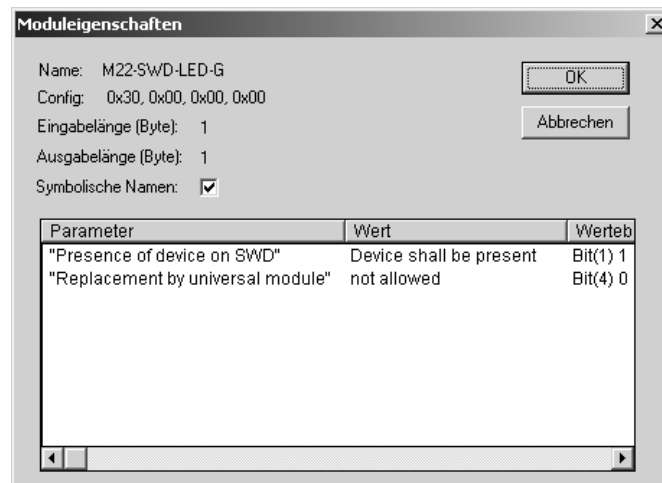
Dem eingefügten SmartWire-DT Master lassen sich nun Teilnehmer (Module) anhängen. Die bekannten Teilnehmer erscheinen im linken Fenster der Registerkarte Ein-/Ausgänge. Die Konfiguration des entsprechenden Aufbaus kann durch **Auswählen** oder **Entfernen** der SmartWire-DT Teilnehmer vorgenommen werden.

Beachten Sie bitte hierbei die korrekte Reihenfolge. Die Module müssen so konfiguriert werden, wie sie hinterher in Ihrer Anwendung angeordnet sind.

Insgesamt können 99 SmartWire-DT Teilnehmer an einen SmartWire-DT Strang angeschlossen werden. Die Ausdehnung des SmartWire-DT Netzwerkes kann bis zu 600 Meter betragen.



Die Schaltfläche **Eigenschaften** führt zum Dialog 'Moduleigenschaften' des aktuell selektierten SmartWire-DT Teilnehmers. Er zeigt den Namen, die Config (Kodierung des IO-Abbilds nach PROFIBUS-Norm) und die Ein- und Ausgabelänge des Moduls in Byte.



Standardparameter aller SmartWire-DT Teilnehmer sind 'Presence of device on SWD' und 'Replacement by universal module'.

Der Eintrag 'Presence of device on SWD' legt fest, ob ein Teilnehmer im SmartWire-DT Netzwerk als notwendiger (Device shall be present) oder als optionaler (Device may not be present) Teilnehmer angesehen wird. Das Fehlen eines notwendigen Teilnehmers setzt das SmartWire-DT Netzwerk in den Zustand FAILSAFE zurück. Das Fehlen eines optionalen Teilnehmers hat keinen Einfluss auf den Zustand des SmartWire-DT Netzwerks. Defaulteinstellung ist 'Device shall be present'.

Der Eintrag 'Replacement by universal module' legt fest, ob ein Teilnehmer im SmartWire-DT Netzwerk durch einen Universalteilnehmer ersetzt werden darf. Defaulteinstellung ist 'not allowed'.

Enthält der Teilnehmer zusätzliche Parameter, werden auch diese hier mit Wert und Wertebereich aufgelistet. Ist die Option **Symbolische Namen** aktiviert, werden dabei die möglichen Einstellungen durch symbolischen Namen verdeutlicht.

## 5 Steuerungskonfiguration

### 5.6.1.3 Anwenderparameter

Wechseln Sie zum Register **Anwenderparameter**

Hier nehmen Sie allgemeine Einstellungen des SmartWire-DT Netzwerks vor.

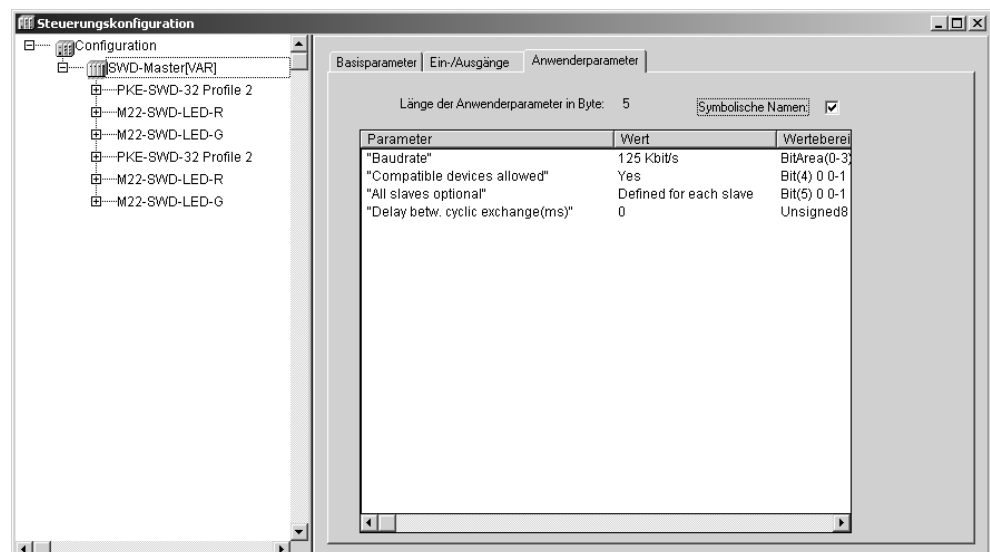
Über den Parameter **'Baudrate'** lässt sich die zu verwendende SmartWire-DT Baudrate einstellen.  
Mögliche Werte: 125kBaud / 250kBaud (Defaulteinstellung).

Der Parameter **'Compatible devices allowed'** legt fest, ob im SmartWire-DT Strang anstelle des projektierten Teilnehmers auch ein zu diesem kompatibler Teilnehmer akzeptiert wird.  
Defaulteinstellung = No

Z. B. anstelle eines projektierten M22-SWD-K11LED-W ist im Netzwerk ein M22-SWD-K22LED-W eingesetzt worden. → Als kompatibler Teilnehmer werden nur Teilnehmer mit identischem "Family-Code" und größerem "Device Code" angenommen.

Mit dem Parameter **'All slaves optional'** kann über einen Schalter festgelegt werden, dass alle projektierten SmartWire-DT Teilnehmer als optionale Teilnehmer behandelt werden.  
Defaulteinstellung = Defined for each slave

**'Delay betw. Cyclic exchange(ms)'** stellt eine Verzögerungszeit zwischen zwei zyklischen Datentelegrammen ein.  
Defaultwert = 0.



## 5.6.2

**Busdiagnose**

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

- Bit 0-3: Teilnehmerspezifische Nutzdaten.
- Bit 4: TRUE = Für den Teilnehmer liegt eine erweiterte Diagnosemeldung vor.
- Bit 5: Reserviert.
- Bit 6: TRUE = Der Teilnehmer ist vorhanden.
- Bit 7: TRUE = Der Teilnehmer wird durch ein Universalmodul ersetzt.

Weitere Statusmeldungen zum SmartWire-DT Koordinator und den angeschlossenen Teilnehmern können über Funktionen und Funktionsblöcke der SWD\_Util-Bibliothek abgerufen werden. Beschreibung im Dokument MN04802103Z-DE.

## 5.6.3

**Azyklische Kommunikation**

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

## 5.6.4

**SmartWire-DT LEDs**

Auf der XV100-SWD Hardware befinden sich drei LEDs. Sie sind mit POW, CONFIG und SWD bezeichnet.

**POW-LED**

Zustand	Beschreibung
aus	Spannungsversorgung des lokalen SmartWire-DT nicht vorhanden.
grünes Dauerlicht	Spannungsversorgung des lokalen SmartWire-DT ASIC OK.

**CONFIG-LED**

Voraussetzung für einen Datenaustausch ist, dass die erstellte Konfiguration (= projektierte Konfiguration) mit der tatsächlich im SWD-Master vorhandenen Konfiguration (Targetkonfiguration) übereinstimmt. Das Ergebnis dieser Prüfung wird über die SmartWire-DT Config-LED angezeigt.

Zustand	Beschreibung
aus	Keine Projektkonfiguration oder fehlerhafte Targetkonfiguration (Siehe SWD-LED).
grünes Dauerlicht	Projektkonfiguration identisch mit gespeicherter Targetkonfiguration.
blinkt grün	Projektkonfiguration kompatibel mit gespeicherter Targetkonfiguration.
rotes Dauerlicht	Projektkonfiguration und gespeicherte Targetkonfiguration sind nicht kompatibel.

## 5 Steuerungskonfiguration

### SWD-LED

Ist eine Konfiguration im SWD-Master gespeichert, so wird bei jedem Einschalten der Versorgungsspannung geprüft, ob die tatsächlich am Netzwerk gefundenen Teilnehmer mit der gespeicherten Targetkonfiguration übereinstimmen. Das Ergebnis der Prüfung wird über die SWD-LED angezeigt.

Zustand	Beschreibung
aus	Keine Targetkonfiguration vorhanden.
blinkt orange	Einlesen und Abspeichern der aktuellen Teilnehmerliste als neue Targetkonfiguration.
grünes Dauerlicht	Die am SmartWire-DT Strang gefundenen Teilnehmer entsprechen der Targetkonfiguration.
blinkt grün	Teilnehmersuche / Teilnehmeradressierung ist aktiv.
rotes Dauerlicht	Kurzschluss auf der 15-V-Spannungsversorgung, oder kein SmartWire-DT Teilnehmer gefunden.
blinkt rot	Die am SmartWire-DT Strang gefundenen Teilnehmer entsprechen nicht der Targetkonfiguration, oder ein als notwendig projektiertes SmartWire-DT Teilnehmer fehlt.

### 5.6.5

#### Config-Taster

Über den Config-Taster auf der XV100-SWD Hardware kann ein Speichern der aktuellen Konfiguration als Targetkonfiguration durchgeführt werden. Diese Funktionalität steht zur Verfügung, solange das SmartWire-DT Netzwerk sich nicht in den Zuständen "SWD\_RUN\_FAILSAFE" oder "SWD\_RUN\_NORMAL" befindet (die Config-LED leuchtet nicht grün).

Die Targetkonfiguration muss nach jedem Teilnehmertausch am SmartWire-DT Netzwerk durchgeführt werden.

Ein Einlesen und Speichern der Targetkonfiguration ist auch über den Browser-Befehl "swd\_setnew" möglich.



## 5.6.6

**Browserbefehle**

Der XV100 SmartWire-DT Master unterstützt folgende Browserbefehle:

Befehl	Beschreibung
?	Liste der implementierten und zur Verfügung stehenden Kommandos holen
swd_targetconfig	Zeigt eine Liste der in der aktuell gespeicherten Targetkonfiguration enthaltenen SmartWire-DT Teilnehmer an.
swd_projectconfig	Zeigt eine Liste der in der aktuell geladenen Projektkonfiguration enthaltenen SmartWire-DT Teilnehmer an.
swd_setnew	Speichert die aktuell am SmartWire-DT Strang gefundenen Teilnehmer als Targetkonfiguration ab.
swd_version [arg]	Zeigt Versionsinformationen für einen / alle SmartWire-DT Teilnehmer an.
swd_data [arg]	Zeigt den aktuellen Stand der Nutzdaten auf dem SmartWire-DT Strang an.
swd_state	Zeigt Statusinformationen des SmartWire-DT Netzwerks.
swd_diag [arg]	Zeigt erweiterte Diagnoseinformationen von SmartWire-DT Teilnehmern an.
swd_param [arg]	Zeigt erweiterte Parametereinstellungen von SmartWire-DT Teilnehmern an.
swd_umodules	Zeigt durch Universalmodule ersetzte SmartWire-DT Teilnehmer an.

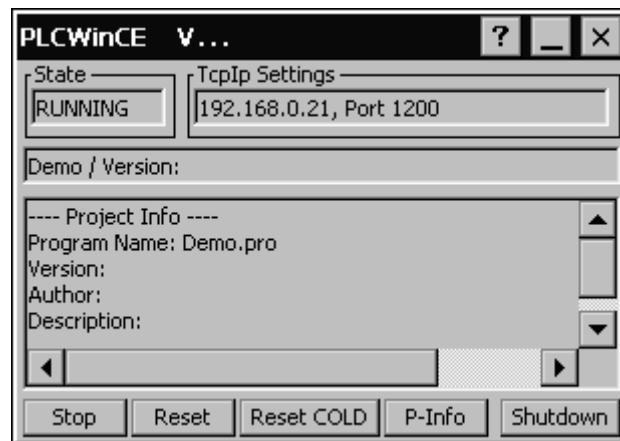
## 6 Betrieb

### 6.1 Einschaltverhalten

Nach dem Einschalten führt die Steuerung einen Systemtest durch. Die Steuerung nimmt den Zustand "Stop" oder "Run" erst an, wenn sie keinen Hardwarefehler festgestellt hat. Der Systemtest beinhaltet folgende Tests:

- Speichertest
- SPS-Programmtest

Das Ergebnis des Tests wird im PLCWinCE-Ausgabefenster visualisiert.



Im Weiteren hängt der Zustand der Steuerung vom parametrisierten Startverhalten ab (→ Kap. 6.4)

### 6.2 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.

### 6.3

#### 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 parametrimtem Startverhalten "Stop" (→ Kap. 6.4)
- Über die SPS-Programmierungsumgebung im PC
- Nach einer Zykluszeitüberschreitung

##### Run

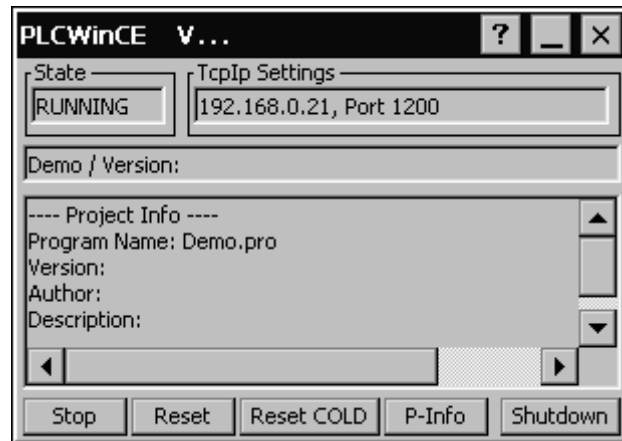
Im Zustand RUN wird das SPS-Programm zyklisch abgearbeitet.

Der Betriebszustand RUN wird eingenommen:

- Nach Einschalten der Spannung mit parametrimtem Startverhalten "Warm"
- Über die SPS-Programmierungsumgebung im PC

## 6.4 Betriebsartenumschaltung

Die Betriebsartenumschaltung erfolgt über das PLCWinCE-Ausgabefenster oder über die SPS-Programmierungsumgebung.



Im Weiteren kann das Start-Verhalten nach Einschalten der Spannung über die Datei **PicRts.reg** im Verzeichnis **PicRts** parametrieren werden.

### **PicRts.reg**

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Eaton\PLC Runtimesystem]

...  
...

**"Startup"="WARM"**

...  
...

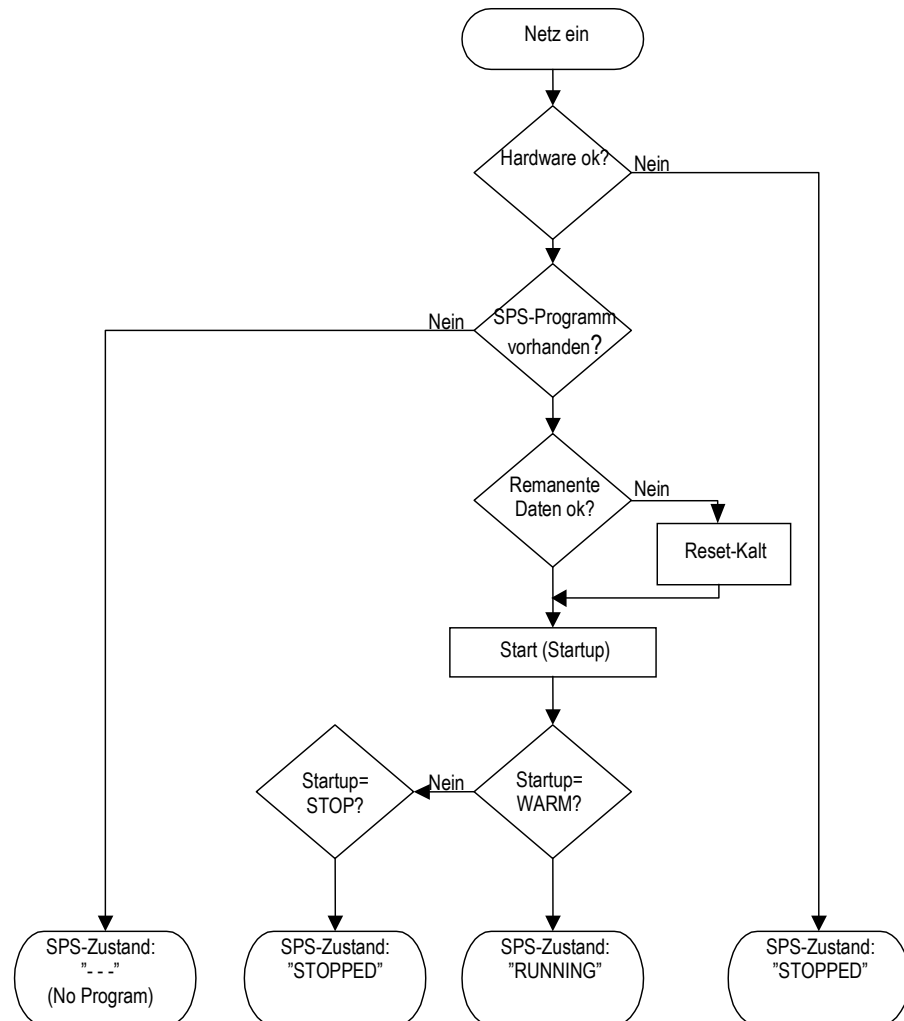
Es sind folgende Einstellungen möglich:

- Stop "Startup"="STOP"
- Warmstart (Standardeinstellung) "Startup"="WARM"

## 6.5

## Start-Verhalten

## Ablauf

**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, werden mit ihren Initialisierungswerten initialisiert und das SPS-Programm wird gestartet.

**Warmstart**

Bei sämtlichen weiteren Starts des geladenen SPS-Programms bzw. nach jedem Reset / Reset-Warm erfolgt ein Warmstart. Die remanenten (RETAIN, RETAIN PERSISITENT) Variablen behalten

ihre Werte, die übrigen Variablen werden mit ihren Initialisierungswerten initialisiert. Variablen ohne explizite Initialisierungswerte werden mit ihren Standardinitialisierungswerten initialisiert.

### 6.6 **Stop-Verhalten**

Die Abarbeitung des SPS-Programms wird am Ende des Programmzyklus gestoppt. Anschließend erfolgt ein Reset / Reset-Warm.

### 6.7 **Reset-Verhalten**

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

Entspricht der Initialisierung eines Warmstarts.

#### **Reset-Kalt**

Entspricht der Initialisierung eines Kaltstarts. Dieser Befehl setzt alle Variablen, auch die remanenten (RETAIN, RETAIN PERSISTENT) Variablen auf den Initialisierungswert zurück.

#### **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.

## 6.8

**Test und Inbetriebnahme**

Die Steuerung unterstützt folgende Tests- und Inbetriebnahmemöglichkeiten:

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

➔ Die Ablaufkontrolle (Power Flow) ist nicht implementiert und wird nicht unterstützt.

**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.

➔ Das SPS-Programm läuft jeweils nur auf Breakpoints der in der Taskkonfiguration als Debug-Task festgelegten Task auf.

**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.

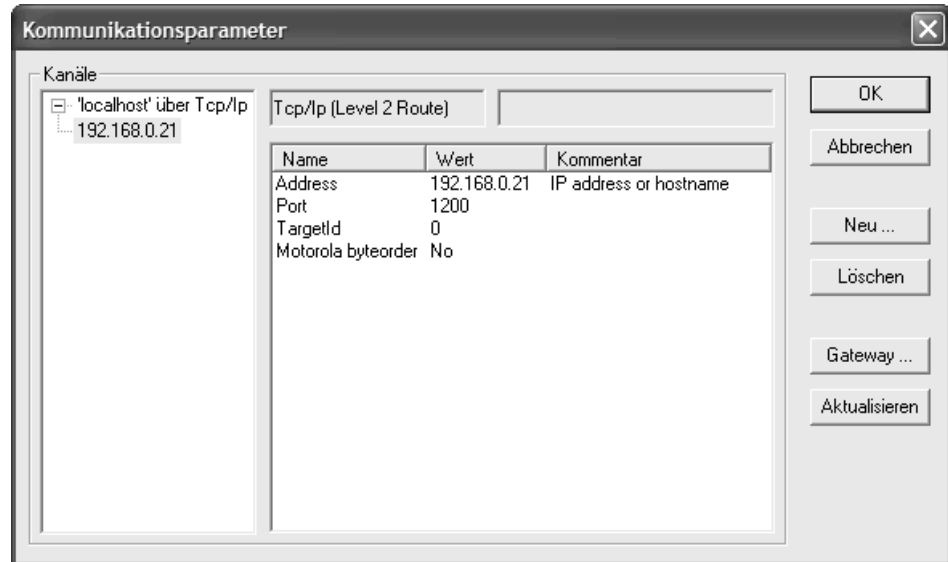
**Forcen**

Alle Variablen eines SPS-Programms können zwangsgesetzt werden.

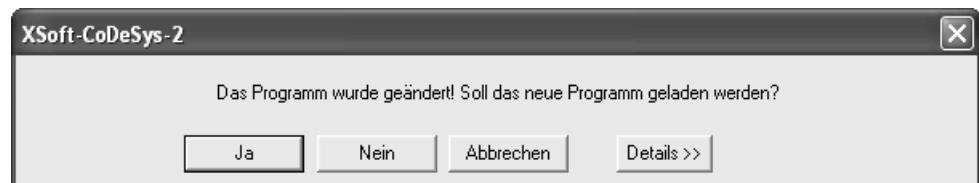
6.9

**Programmtransfer**

Wenn das SPS-Programm in der SPS-Programmierungsumgebung (PC) fehlerfrei übersetzt wurde, kann es in den Arbeitsspeicher der Steuerung geladen und anschließend gestartet werden. (→ Kap. 8)

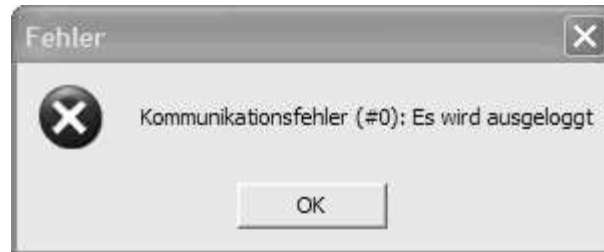


Bei einem Programmtransfer vom PC in die Steuerung wird das Programm in der Steuerung mit dem Programm auf dem PC verglichen. Bei Ungleichheit erfolgt eine Abfrage ob das Programm überschrieben werden soll. Wird diese Abfrage bestätigt, so geht die Steuerung in den Zustand "Stop" und das neue Programm wird in den Arbeitsspeicher geladen.





### Kommunikationsfehler



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

- Physikalische Verbindung
- Kommunikationsparameter in der SPS-Programmierungsumgebung
- TCP/IP Einstellungen in den Systemeinstellungen PC-seitig
- TCP/IP Einstellungen in den Systemeinstellungen steuerungsseitig

#### 6.9.1

### Bootprojekt erzeugen

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

➔ Das Bootprojekt kann nur "online" erzeugt werden.

### 7 Programmabarbeitung und Systemzeiten

#### 7.1 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
- Berarbeitung des Prozessabbilds (IO-Update)

#### Programmabarbeitung ohne Taskkonfiguration

Der erste Baustein, der in einem neuen SPS-Programm angelegt wird, trägt automatisch den Namen **PLC\_PRG**. Dort startet die Ausführung. Somit arbeitet die Steuerung das PLC\_PRG zyklisch freilaufend mit einem Minimalintervall von 10 ms ab.

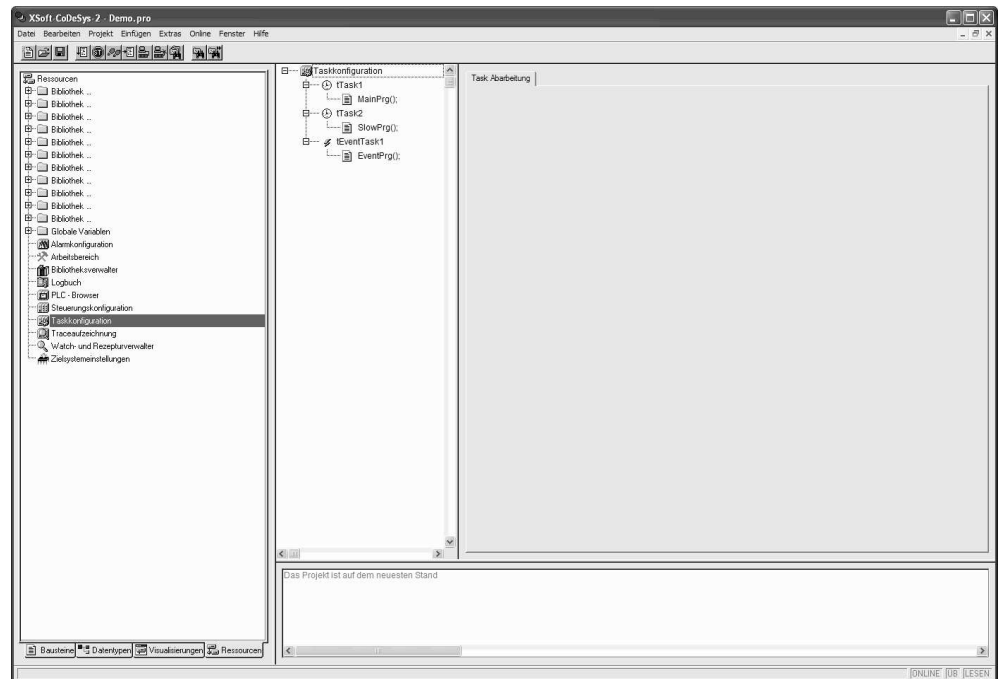
#### Programmabarbeitung mit Taskkonfiguration

Neben dem Deklarieren des speziellen Programmbausteins PLC\_PRG kann die Abarbeitung des SPS-Programms auch über die Taskkonfiguration gesteuert werden. Wenn eine Taskkonfiguration definiert wurde, muss kein Programmbaustein mit dem Namen PLC\_PRG angelegt werden.

## 7.2

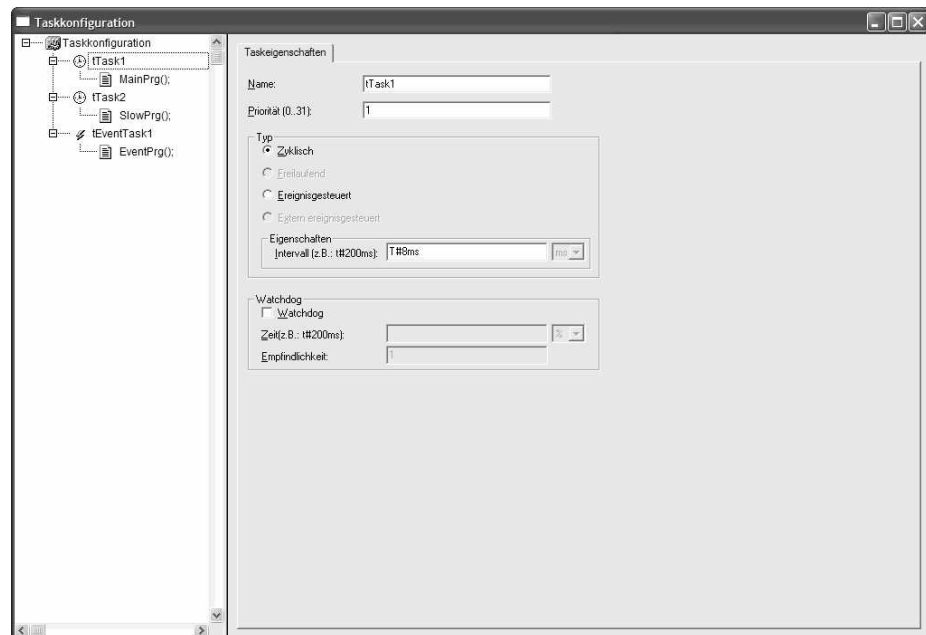
## Taskkonfiguration

Die Taskkonfiguration wird über die Funktionalität <Taskkonfiguration> im Registerblatt 'Ressourcen' erreicht.



Jeder Task kann eine Folge von Programmen zugeordnet werden, die beim Aufruf der Task ausgeführt werden sollen. Die Taskpriorität und Taskbedingung legen fest, in welcher zeitlichen Abfolge die Tasks abgearbeitet 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.

## 7 Programmabarbeitung und Systemzeiten



Die korrekte Definition einer Task besteht aus folgenden Parametern:

- Taskname
- Taskpriorität
- Tasktyp bzw. Taskbedingung
- Tasküberwachung / Watchdog

### Taskname

Eindeutiger Taskname

### Taskpriorität

Um die Tasks zu priorisieren, können unterschiedliche Taskprioritäten vergeben werden. Die Taskprioritäten können zwischen 0 und 31 parametrisiert werden. 0 entspricht der höchsten und 31 der niedrigsten Priorität.

➔ Tasks mit gleicher Priorität unterbrechen sich nicht gegenseitig.

### Tasktyp bzw. Taskbedingung

Die Taskbedingung kann entweder ein Zeitintervall sein, nach dem die Task ausgeführt werden soll (zyklisch), oder eine globale Variable, die im Falle einer steigenden Flanke deren Ausführung bewirkt (ereignisgesteuert).

Eine ereignisgesteuerte Task setzt eine zyklisch ablaufende Task voraus. Im Programm der zyklischen Task ist das Ereignis zu programmieren. Jede steigende Flanke des Ereignisses führt anschließend zur Ausführung des Programms der ereignisgesteuerten Task.

### **Tasküberwachung / Watchdog**

Jede Task kann mit Hilfe des Watchdogs überwacht werden (→ Kap. 7.4)

## **7.3**

### **Multitasking**

Das SPS-Laufzeitsystem ist ein Multitaskingsystem. Das bedeutet, mehrere Tasks werden quasi zeitgleich abgearbeitet. In einem Multitaskingsystem können sich einzelne Task mit unterschiedlicher Priorität unterbrechen.

Die Folge daraus ist, dass die Konsistenz des Ein-/Ausgangsabbildes nur innerhalb der Task gewährleistet ist, die als einzige auf das Ein-/Ausgangsabbild zugreift. Greifen mehrere Tasks unterschiedlicher Priorität auf die gleichen Ein-/Ausgänge zu, ist die Konsistenz nur bei der Task mit der höchsten Priorität gegeben. (→ Kap. 7.9)



Vermeiden Sie grundsätzlich einen Zugriff auf physikalische Ein-/Ausgänge aus mehreren Tasks heraus, um so einen eindeutigen Steuerungsablauf sicher zu stellen.

## **7.4**

### **Tasküberwachung / Zykluszeitüberwachung**

Die Zyklusüberwachung überwacht die zyklischen Tasks des SPS-Programms. Die Zyklusüberwachung unterbricht die Programmbearbeitung, wenn die Task eine definierte Zeit in einer definierten Häufigkeit überschreitet. Standardmäßig werden die Ausgänge der Steuerung beim Auslösen der Zykluszeitüberwachung abgeschaltet und die Steuerung wird in den Zustand "Stop" gesetzt. Danach muss das SPS-Programm mit "Reset" zurückgesetzt werden.

Die korrekte Definition der Tasküberwachung besteht aus folgenden Parametern:

- Watchdog ein/aus
- Watchdogzeit
- Watchdog-Empfindlichkeit



Ist in der Taskkonfiguration der Watchdog der entsprechenden Task deaktiviert, erfolgt keine Zykluszeitüberwachung!



In einem SPS-Programm ohne Taskkonfiguration ist die Zykluszeitüberwachung standardmäßig mit einer Zykluszeitüberwachung von 10ms parametrisiert und aktiviert.

## 7 Programmabarbeitung und Systemzeiten

### Funktionsweise

Der Watchdog wird zu Beginn eines jeden Abarbeitungszyklus gestartet und nach erfolgreichem Abarbeiten beendet. Dauert die Tasklaufzeit länger als das definierte Taskintervall wird die Zykluszeitüberwachung ausgelöst.

Das Auslösen der Zyklusüberwachung ist weiterhin abhängig von der Watchdog-Empfindlichkeit. Die Watchdog-Empfindlichkeit bestimmt, nach wie vielen unmittelbar aufeinander folgenden Tasklaufzeit-Überschreitungen die Zyklusüberwachung ausgelöst wird.

Der Watchdog wird ausgelöst:

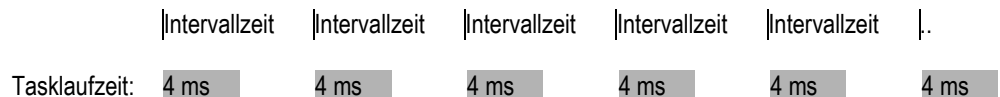
- bei einer Watchdog-Empfindlichkeit von 1 gleich bei der ersten Überschreitung der Watchdogzeit.
- bei einer Watchdog-Empfindlichkeit von x erst beim x-ten unmittelbar aufeinander folgendem Überschreiten der Watchdogzeit.
- Darüber hinaus wird der Watchdog bei einer Endlosschleife ausgelöst, wenn die Ausführungszeit der Task länger dauert als das Ergebnis aus Watchdogzeit \* Watchdog-Empfindlichkeit. Dieses Kriterium dient dem Erkennen und Reagieren auf Endlosschleifen bei vorgewählter Watchdog-Funktionalität.

Wenn Sie z. B. bei einer Task die Watchdogzeit von 10ms und die Watchdog-Empfindlichkeit von 5 eintragen, wird die Task spätestens nach  $10\text{ms} * 5 = 50\text{ms}$  beendet.

Folgende Beispiele sollten das Zusammenspiel von Taskintervall, Tasklaufzeit und Watchdogzeit verdeutlichen.

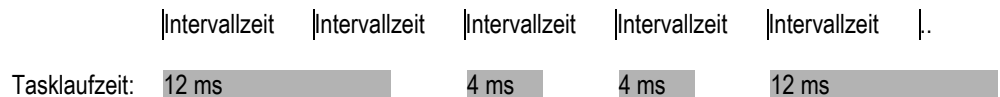
- parametrisiertes Taskintervall = 8ms
- parametrisierte Watchdogzeit = 8ms
- parametrisierte Empfindlichkeit = 2

### Beispiel 1



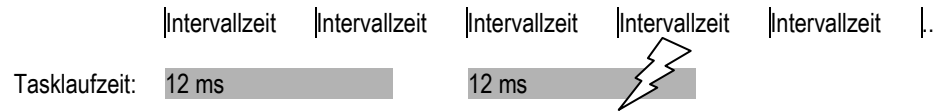
Die Zykluszeitüberwachung wird nicht ausgelöst, da die Tasklaufzeit immer unter dem parametrisierten Taskintervall bleibt.

### Beispiel 2



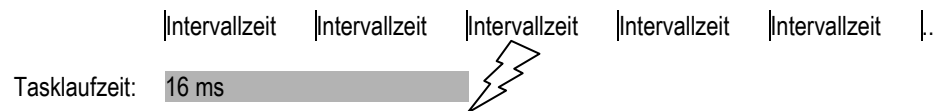
Die Zykluszeitüberwachung wird nicht ausgelöst, da nicht 2 aufeinander folgende Tasklaufzeit-Überschreitungen auftreten.

### Beispiel 3



Die Zykluszeitüberwachung wird 8ms nach Beginn der zweiten Taskabarbeitung ausgelöst, da beide Tasklaufzeiten länger als die angegebene Watchdogzeit sind und unmittelbar aufeinander folgen.

### Beispiel 4 (Endlos-Schleufe)



Die Zykluszeitüberwachung wird ausgelöst, da die Tasklaufzeit länger dauert als die Multiplikation von Watchdogzeit und Empfindlichkeit."

## 7 Programmabarbeitung und Systemzeiten

### 7.5

#### 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

Aktion	VAR	VAR RETAIN	VAR PERSISTENT	VAR RETAIN PERSISTENT
Stop → Start	Initialisierungswert	Wert bleibt erhalten	Initialisierungswert	Wert bleibt erhalten
Warmstart bzw. Reset / Reset-Warm	Initialisierungswert	Wert bleibt erhalten	Initialisierungswert	Wert bleibt erhalten
Kaltstart bzw. Reset-Kalt	Initialisierungswert	Initialisierungswert	Initialisierungswert	Initialisierungswert
Reset-Ursprung	Initialisierungswert	Initialisierungswert	Initialisierungswert	Initialisierungswert
'Alles Bereinigen' und anschließendem Programmdownload	Initialisierungswert	Initialisierungswert	Wert bleibt erhalten	Wert bleibt erhalten

➔ Das Variablenverhalten globaler Variablen entspricht auch dem Variablenverhalten lokaler Variablen.

### 7.6

#### Direkter Peripheriezugriff

➔ Direkte Peripheriezugriffe (z.B. Prozessabbild) sind zielsystemspezifisch implementiert.

### 7.7

#### Interruptverarbeitung

➔ Eine direkte Interruptverarbeitung wird nicht unterstützt.



### 7.8 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.

### 7.9 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-Programmierungsumgebung Warnungen ausgegeben.

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

## 7 Programmabarbeitung und Systemzeiten

### 7.9.1

#### 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.

### 7.9.2

#### Profibus

##### Profibus-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.

### 7.9.3

#### 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.

### 8 Verbindungsaufbau Programmier-PC – Steuerung

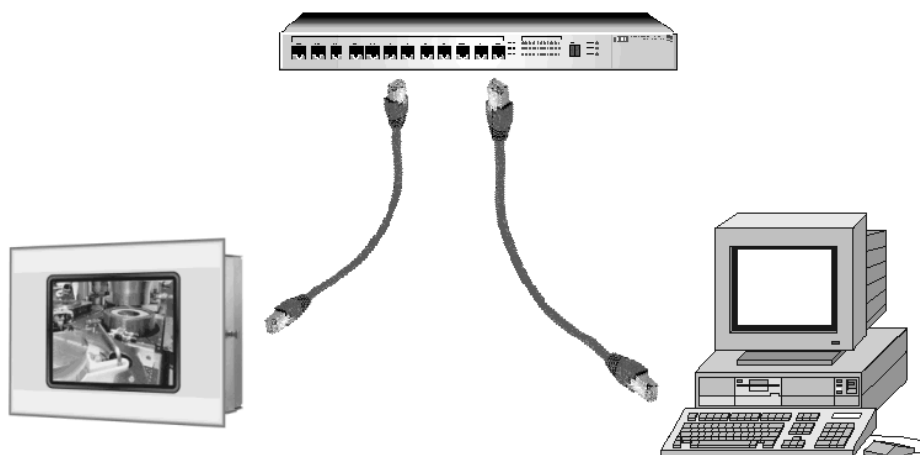
#### 8.1 Verbindungsaufbau über Ethernet

Die Kommunikation vom Programmier-PC auf die Steuerung erfolgt mit Ethernet und dem TCP/IP Protokoll. Bei einer direkten Verbindung, ohne Ethernet-Hub oder Switch, ist ein gekreuztes "Crossover" Kabel zu verwenden. Steuerungsseitig ist eine standardisierte RJ45 Buchse angebracht.



**gekreuztes** RJ45 TwistedPair-Kabel

Erfolgt die Verbindung über einen Ethernet-Hub bzw. Switch, sind von Hub zur Steuerung wie zum Programmier-PC nicht gekreuzte Kabel einzusetzen.



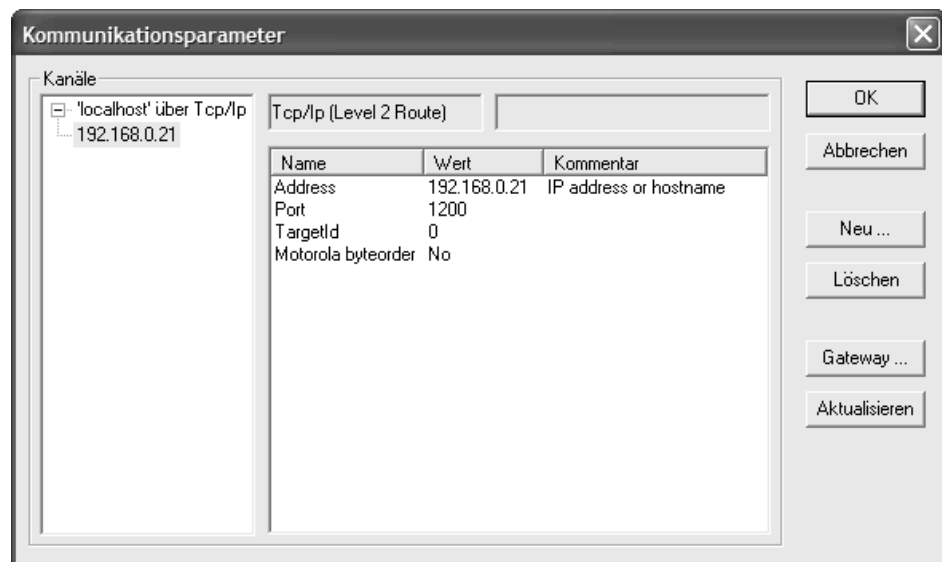
**nicht gekreuzte** RJ45 TwistedPair-Kabel

## 8 Verbindungsaufbau Programmier-PC – Steuerung

### Kommunikationsparameter

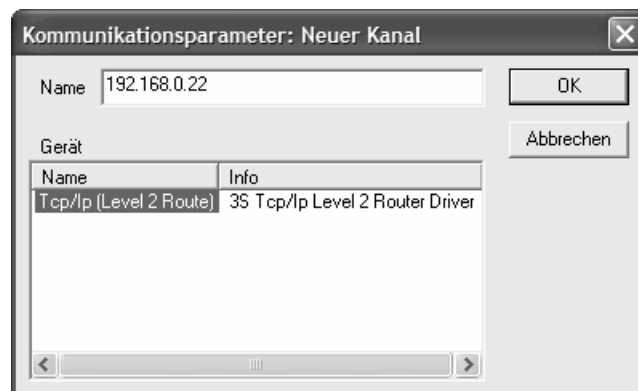
In der SPS-Programmierungsumgebung wird über den Menüpunkt <Online> <Kommunikationsparameter> der Dialog 'Kommunikationsparameter' erreicht. In diesem Dialog werden die Parameter eingestellt, welche für die Kommunikation zwischen dem Programmier-PC und der Steuerung gelten.

Im Dialog 'Kommunikationsparameter' sind unter "Kanäle" alle bereits erstellten Kommunikationswege aufgelistet. Wählen Sie nun einen der Kanäle aus, indem Sie mit der Maus auf einen Eintrag klicken. Die entsprechenden Parameter werden dann in der Tabelle angezeigt. Für eine Verbindung über Ethernet ist ein Kanal vom Typ Tcp/Ip (Level 2 Route) zu verwenden. Unter Address ist die IP-Adresse der Steuerung aufgeführt mit der durch diesen Kanal kommuniziert wird.



### Einrichten eines neuen Kanals

Drücken Sie die Schaltfläche **Neu**. Sie erhalten den Dialog 'Kommunikationsparameter: Neuer Kanal':

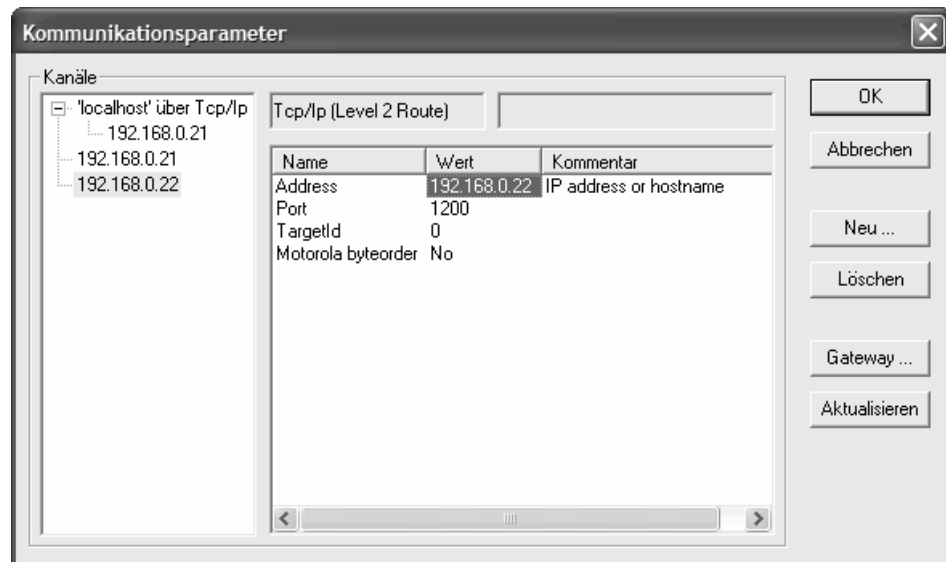


## 8 Verbindungsaufbau Programmier-PC – Steuerung

Im Eingabefeld **Name** wird automatisch der für den zuletzt eingetragenen Kanal verwendete Name angeboten. Sie können den Kanalnamen hier editieren. Die Kanalnamen sind rein informativ, Eindeutigkeit ist nicht zwingend, aber empfehlenswert.

In der Tabelle unter **Gerät** sind die verfügbaren Gerätetreiber aufgelistet. Aus der Spalte **Name** wählen Sie per Mausklick den Treiber **Tcp/Ip (Level 2 Route)** aus, in der Spalte **Info** steht der dazu vorhandene Kommentar.

Wenn Sie den Dialog 'Neuer Kanal' mit **OK** geschlossen haben, erscheint der neu definierte Kanal im Dialog 'Kommunikationsparameter' als weiterer Eintrag bei **Kanäle**. Er ist hiermit zunächst nur lokal im Projekt gespeichert. In diesem Stadium können Sie die Spalte **Wert** editieren. Geben Sie unter Wert zur Zeile Address die IP-Adresse der Steuerung ein, indem Sie auf die Adresse Doppelklicken, die Adresse editieren und wichtig, die Eingabe mit **Return** abschließen. Bestätigen Sie dann die eingestellten Parameter mit **OK** und verlassen damit den Dialog 'Kommunikationsparameter'.



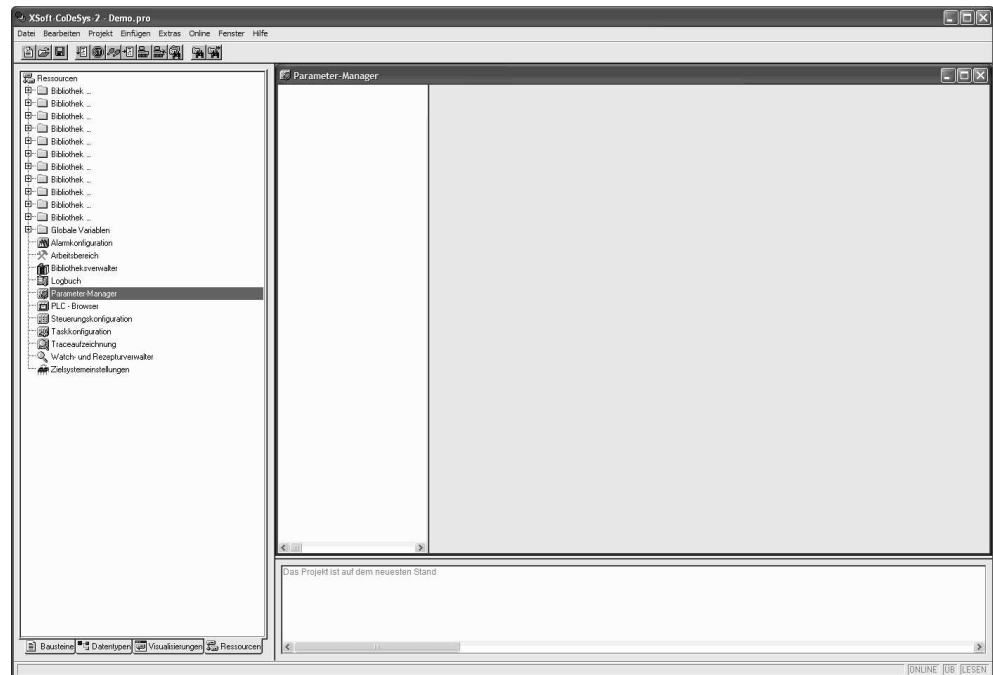
Die Parameter eines bereits verwendeten Kanals können im Konfigurationsdialog nicht mehr editiert werden. Die Parameterfelder erscheinen grau. Sie können die Verbindung allerdings löschen, solange sie nicht aktiv ist.

Beachten Sie, dass der Löschvorgang für einen Kanal nicht reversibel ist. Er erfolgt in dem Moment, in dem Sie auf die Schaltfläche **Löschen** drücken!

- ➔ Speichern Sie Ihr SPS-Programm mit den neuen Kommunikationsparametern, übersetzen Sie dies und loggen Sie sich anschließend auf die Steuerung ein.
- ➔ Achten Sie darauf, dass die IP-Adressen des Programmier-PCs und der Steuerung der gleichen Adressenfamilie angehören. Detaillierte Information entnehmen Sie bitte der Dokumentation "Netzwerk in Kürze".

### 9 Parameter-Manager / Objektverzeichnis

Bei aktivierter Option erscheint im Registerblatt 'Ressourcen' der Eintrag **Parameter-Manager**. Dieser ermöglicht das Erstellen eines Objektverzeichnisses für Variablen und Parameter, die dem gezielten, aktiven Datenaustausch mit anderen Steuerungen dienen.



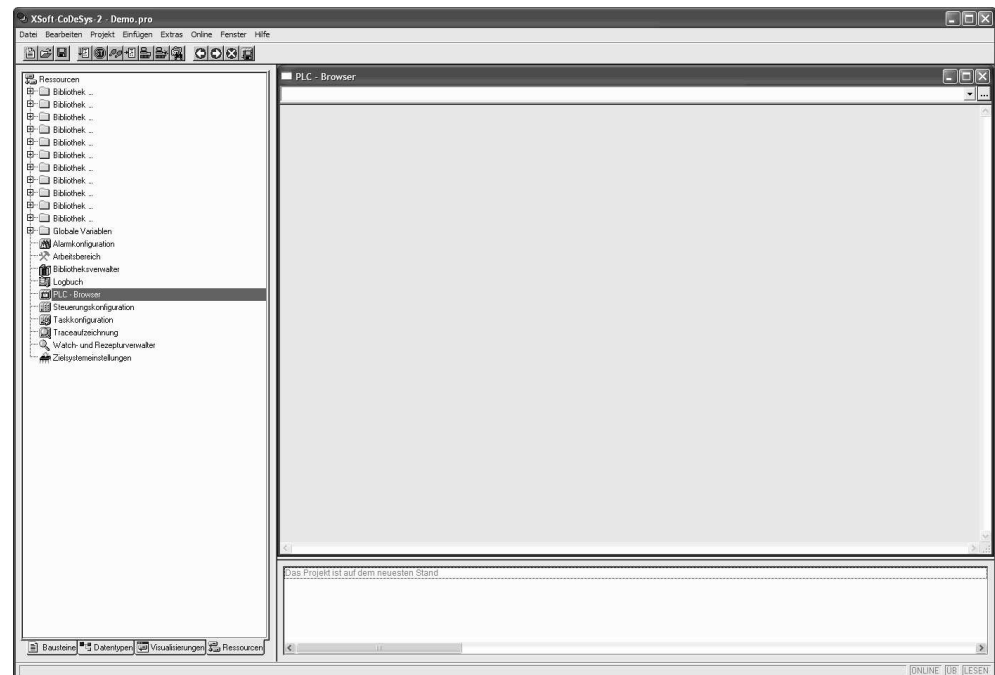
- ➔ Die Objektverzeichnis-Funktionalität wird in Verbindung mit einem in der Steuerungskonfiguration eingefügten CAN-Device unterstützt.
- ➔ Detaillierte Informationen sind dem Handbuch CoDeSys V2.3 oder der Online-Hilfe der SPS-Programmierungsumgebung zu entnehmen.

## 10

## SPS-Browser

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

Der SPS-Browser wird über die Funktionalität <PLC-Browser> im Registerblatt 'Ressourcen' erreicht.



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
?	Liste der implementierten und zur Verfügung stehenden Kommandos holen
...	...

➔ Um die SPS-Browser-Funktionalität zu nutzen, müssen Sie mit dem SPS-Zielsystem online verbunden sein.

### 11 Alarmkonfiguration

Sämtliche SPS-Zielsysteme mit aktivierter Target-Visualisierung unterstützen die CoDeSys Alarmkonfiguration (→ Kap. 4.5).

→ Detaillierte Informationen sind dem Handbuch CoDeSys V2.3, dem Handbuch zur CoDeSys Visualisierung oder der Online-Hilfe der SPS-Programmierungsumgebung zu entnehmen.



## 12

### Anbindung an die Visualisierung / Erstellen der Symboldatei

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 und beim anschließenden Programmdownload auf die Steuerung geladen.

### 12.1

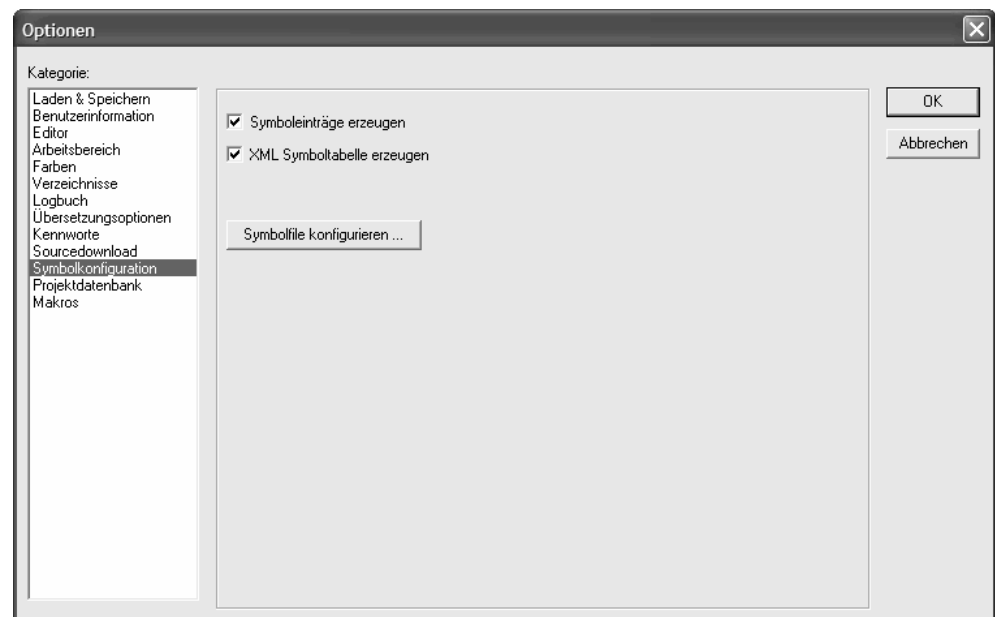
#### Symboldatei konfigurieren

Der Inhalt der Symboldatei wird in der Symbolkonfiguration konfiguriert. Die Symbolkonfiguration wird über den Menüpunkt <Projekt> <Optionen> erreicht.

➔ Ist im Menüpunkt 'Online' die Simulation aktiviert, erscheint im Dialog Option der Eintrag Symbolkonfiguration nicht.

Damit die Symboldatei beim Kompilieren erzeugt wird, muss die Option **Symboleinträge erzeugen** aktiviert werden.

Teilweise unterstützen Visualisierungen den Import der Symboldatei im XML-Format. Damit diese Symboldatei beim Kompilieren erzeugt wird, muss die Option **XML Symboltabelle erzeugen** aktiviert werden.



Über die Schaltfläche **Symbolfile konfigurieren** wird definiert, aus welchen Objekten die Variablen in die Symboldatei exportiert werden.

## 12 Anbindung an die Visualisierung / Erstellen der Symboldatei

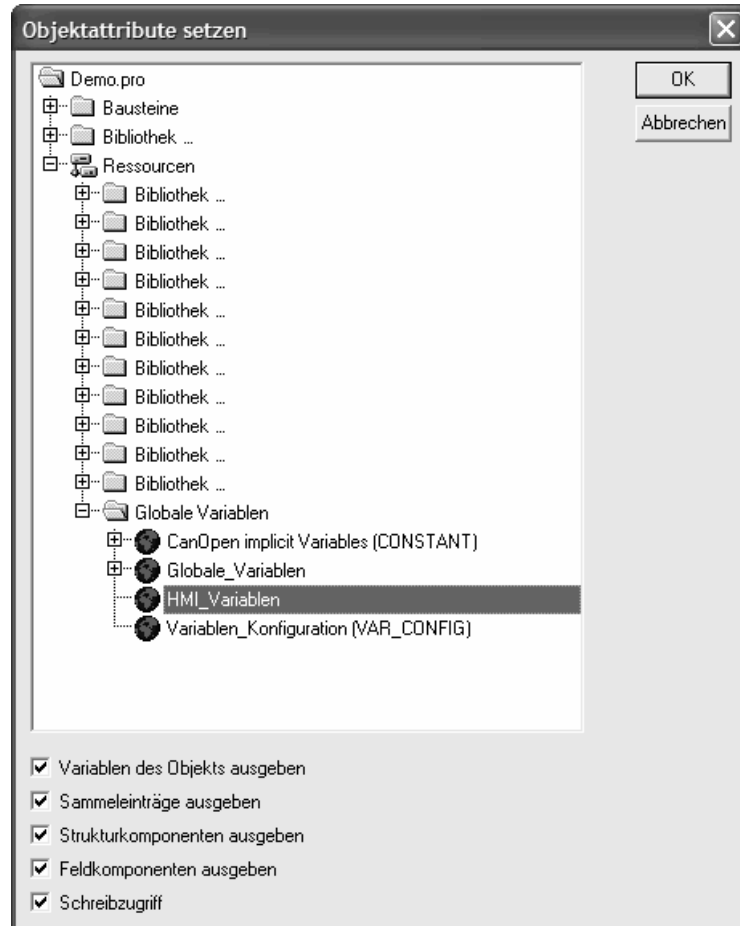
- ➔ Es wird empfohlen, nur die für den Variablen austausch mit der Visualisierung benötigten Variablen in die Symboldatei zu exportieren. Daher sollte der Bereich dieser Variablen strukturiert werden. Dies erfolgt über das Definieren von globalen Variablenarbeitsblätter oder über das Verwenden von Pragma-Anweisungen.
- ➔ Detaillierte Informationen sind dem Handbuch CoDeSys V2.3 oder der Online-Hilfe der SPS-Programmierungsumgebung zu entnehmen.

Standardmäßig sind sämtliche Objekte selektiert. Deaktivieren Sie deshalb beim erstmaligen Öffnen dieses Dialoges bei sämtlichen Objekten die Option 'Variablen des Objekts ausgeben'.



## 12 Anbindung an die Visualisierung / Erstellen der Symboldatei

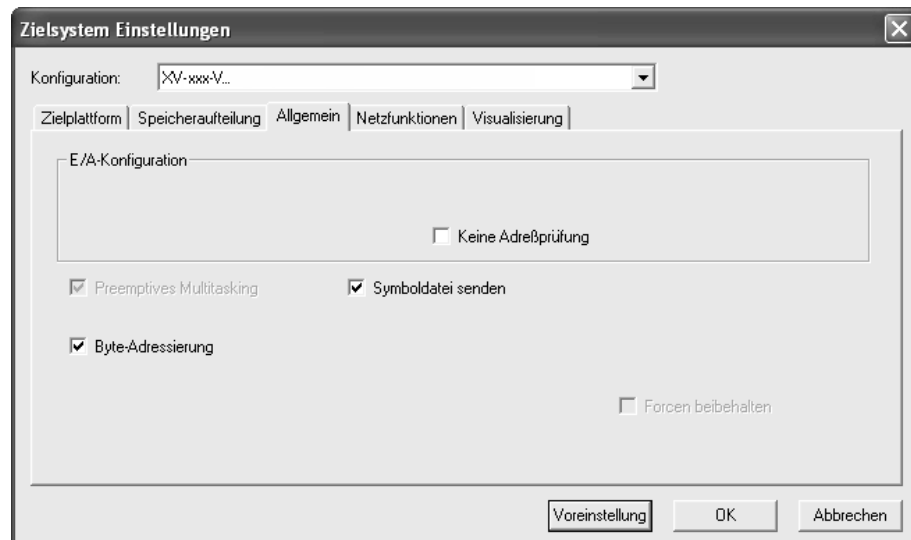
Aktivieren Sie nun selektiv die Option 'Variablen des Objekts ausgeben' zu den gewünschten Objekten.



### 12.2

#### Symboldatei senden

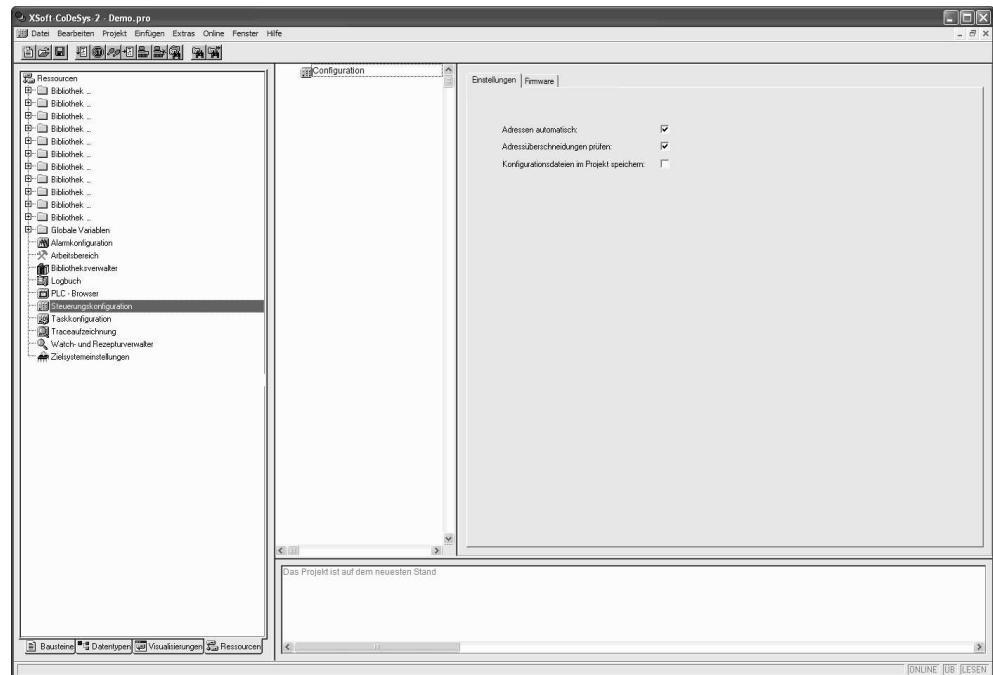
Damit die Symboldatei beim Projektdownload auf die Steuerung geladen wird, muss in der Zielsystemeinstellung im Registerblatt 'Allgemein' die Option **Symboldatei senden** aktiviert werden.



## 13

## Zielsysteminstallation und Firmware Update

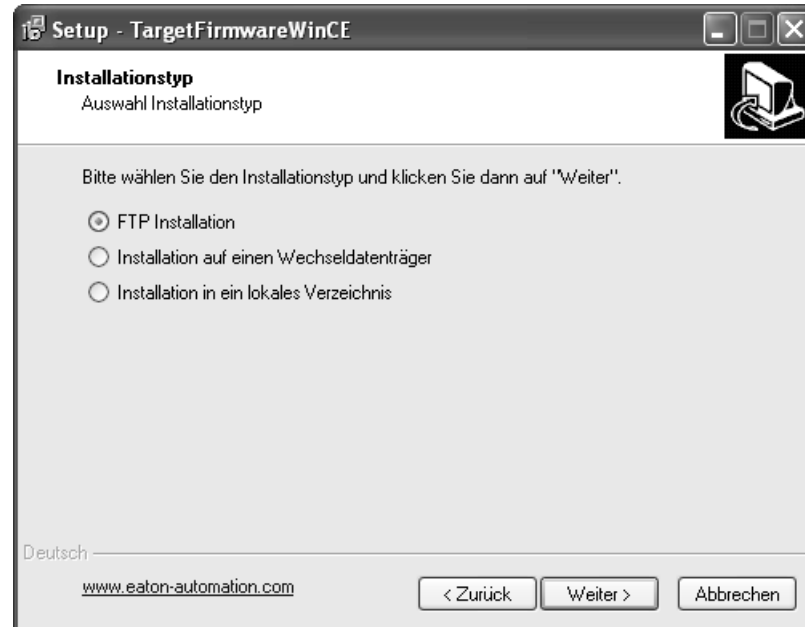
Die Zielsysteminstallation bzw. ein Firmware Update wird mit Hilfe des Programmaufrufs "TargetFirmwareWinCE" ausgeführt und erfolgt in der SPS-Programmierungsumgebung über die Funktionalität <Steuerungskonfiguration> im Registerblatt Firmware!



Betätigen Sie hierfür im Feld "Aktualisieren des Betriebssystems" die Schaltfläche **Start** und wählen Sie anschließend entsprechende Firmware.

## 13 Zielsysteminstallation und Firmware Update

Es stehen folgende Installationsvarianten zur Verfügung:



**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 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.

## 14 Lizenzierung

### 14.1 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.

### 14.2 SPS-Laufzeitsystem

Das SPS-Runtime-System ist lizenzpflichtig und benötigt auf dem SPS-Zielsystem **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 <Online> <Start> erfolgen.

### 14.3 Target-Visualisierung

Die Target-Visualisierung ist lizenzpflichtig und benötigt auf dem SPS-Zielsystem 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 <Online> <Start> erfolgen.

### 14.4 Web-Visualisierung

Die Web-Visualisierung ist nicht lizenzpflichtig und benötigt auf dem SPS-Zielsystem **keine Lizenzpunkte**.

## Änderungsindex

Index	Datum / Visum	Änderungen
01	21.12.10 / DL	Initialversion
02	05.05.11 / NM 25.05.11 / DL	Beschreibung zur SWD-Konfiguration hinzugefügt, Anpassungen und Korrekturen
03	18.05.12 / DL	Anpassungen und Korrekturen infolge XSOFTE-CODESYS V2.3.9 SP3, Hinweis zum korrekten Wert der HSA bei der Profibus Master Konfiguration hinzugefügt, Hinweis zum Symbolimport im XML-Format hinzugefügt, Hinweis zum SmartWire-DT IO Update hinzugefügt
04	12.05.14 / DL	Anpassungen und Korrekturen infolge XSOFTE-CODESYS V2.3.9 SP4, Kapitel Zielsysteminstallation und Firmware Update angepasst
05	02.04.15 / DL	Anpassungen und Korrekturen infolge XSOFTE-CODESYS V2.3.9 SP5, Änderung von Code- und Datengröße