

Dokumentation | DE

# BX3100

Busklemmen-Controller für PROFIBUS





# Inhaltsverzeichnis

<b>1</b>	<b>Vorwort</b> .....	<b>7</b>
1.1	Hinweise zur Dokumentation .....	7
1.2	Sicherheitshinweise .....	8
1.3	Ausgabestände der Dokumentation .....	9
<b>2</b>	<b>Produktübersicht</b> .....	<b>11</b>
2.1	Busklemmen-Controller der BX-Serie .....	11
2.2	BX3100 - Einführung .....	13
2.3	Technische Daten .....	14
2.3.1	Technische Daten - BX .....	14
2.3.2	Technische Daten - PROFIBUS-DP .....	15
2.3.3	Technische Daten - SSB-Interface .....	15
2.3.4	Technische Daten - SPS .....	16
2.4	Prinzip der Busklemme .....	16
2.5	Das Beckhoff Busklemmensystem .....	17
<b>3</b>	<b>Montage und Verdrahtung</b> .....	<b>19</b>
3.1	Hinweise zum ESD-Schutz .....	19
3.2	Montage .....	19
3.2.1	Abmessungen .....	19
3.2.2	Tragschienenmontage .....	21
3.3	Entsorgung .....	21
3.4	Verdrahtung .....	22
3.4.1	Potentialgruppen, Isolationsprüfung und PE .....	22
3.4.2	Spannungsversorgung .....	23
3.4.3	Programmierkabel für COM1 .....	25
3.4.4	SSB- und COM-Schnittstellen .....	26
3.4.5	PROFIBUS-Anschluss .....	28
3.4.6	PROFIBUS-Verkabelung .....	29
<b>4</b>	<b>Parametrierung und Inbetriebnahme</b> .....	<b>32</b>
4.1	Anlaufverhalten des Busklemmen-Controllers .....	32
4.2	PROFIBUS-Adresse .....	32
4.3	Konfiguration .....	33
4.3.1	Überblick .....	33
4.3.2	Anlegen einer TwinCAT-Konfiguration .....	35
4.3.3	Download einer TwinCAT-Konfiguration .....	36
4.3.4	Upload einer TwinCAT-Konfiguration .....	37
4.3.5	Ressourcen im Busklemmen-Controller .....	39
4.3.6	ADS-Verbindung über die serielle Schnittstelle .....	41
4.3.7	PROFIBUS .....	43
4.3.8	K-Bus .....	52
4.3.9	PLC .....	54
4.3.10	SSB .....	57
4.3.11	Echtzeit-Uhr (RTC) .....	87
4.3.12	COM-Port .....	88

4.4	Menü.....	90
4.4.1	BX-Menü-Settings.....	90
4.4.2	Erstellen eigener Menüs.....	94
4.5	Konfigurations-Software KS2000.....	95
<b>5</b>	<b>Programmierung.....</b>	<b>96</b>
5.1	PLC-Eigenschaften der BX-Controller.....	96
5.2	TwinCAT PLC.....	96
5.3	TwinCAT PLC - Fehler-Codes.....	97
5.4	Lokierte Merker.....	99
5.5	Mapping der Busklemmen.....	100
5.6	Lokales Prozessabbild im Auslieferungszustand (Default Config).....	100
5.7	Persistente Daten.....	101
5.8	Remanente Daten.....	102
5.9	Lokales Prozessabbild in der TwinCAT-Konfiguration.....	103
5.10	Erzeugen eines Boot-Projekts.....	104
5.11	Kommunikation zwischen TwinCAT und BX/BCxx50.....	105
5.12	Up- und Download von Programmen.....	106
5.13	Bibliotheken.....	110
5.13.1	Bibliotheken - Übersicht.....	110
5.13.2	TcBaseBX.....	113
5.13.3	TcSystemBX.....	124
5.13.4	TcComPortBX.....	127
5.13.5	TcTwinSAFE.....	138
5.14	Programmübertragung.....	145
5.14.1	Programmübertragung über die serielle Schnittstelle.....	145
5.14.2	Programmübertragung über PROFIBUS.....	147
5.15	Prozessabbild.....	149
5.15.1	PROFIBUS-Prozessabbild.....	149
<b>6</b>	<b>PROFIBUS.....</b>	<b>150</b>
6.1	Systemvorstellung.....	150
6.1.1	Feldbusübersicht.....	150
6.1.2	PROFIBUS DP.....	154
6.1.3	PROFIBUS DPV1.....	155
6.1.4	Stecker, Kabel und Schalter.....	156
6.1.5	Topologie.....	158
<b>7</b>	<b>Fehlerbehandlung und Diagnose.....</b>	<b>160</b>
7.1	Diagnose.....	160
7.2	Diagnose-LEDs.....	161
7.3	Diagnose-Display.....	165
7.4	PROFIBUS Diagnose-Daten.....	165
<b>8</b>	<b>Anhang.....</b>	<b>167</b>
8.1	Erste Schritte mit dem BX3100.....	167
8.2	Umstieg zwischen den Controllern.....	169
8.3	Firmware-Update.....	171
8.4	CFC-Client*.....	173

8.5	Beispielprogramme - Übersicht .....	176
8.6	Allgemeine Betriebsbedingungen .....	176
8.7	Prüfnormen für Geräteprüfung .....	178
8.8	Literaturverzeichnis .....	179
8.9	Abkürzungsverzeichnis .....	179
8.10	Support und Service .....	181



# 1 Vorwort

## 1.1 Hinweise zur Dokumentation

### Zielgruppe

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs- und Automatisierungstechnik, das mit den geltenden nationalen Normen vertraut ist.

Zur Installation und Inbetriebnahme der Komponenten ist die Beachtung der Dokumentation und der nachfolgenden Hinweise und Erklärungen unbedingt notwendig.

Das Fachpersonal ist verpflichtet, für jede Installation und Inbetriebnahme die zu dem betreffenden Zeitpunkt veröffentlichte Dokumentation zu verwenden.

Das Fachpersonal hat sicherzustellen, dass die Anwendung bzw. der Einsatz der beschriebenen Produkte alle Sicherheitsanforderungen, einschließlich sämtlicher anwendbaren Gesetze, Vorschriften, Bestimmungen und Normen erfüllt.

### Disclaimer

Diese Dokumentation wurde sorgfältig erstellt. Die beschriebenen Produkte werden jedoch ständig weiter entwickelt.

Wir behalten uns das Recht vor, die Dokumentation jederzeit und ohne Ankündigung zu überarbeiten und zu ändern.

Aus den Angaben, Abbildungen und Beschreibungen in dieser Dokumentation können keine Ansprüche auf Änderung bereits gelieferter Produkte geltend gemacht werden.

### Marken

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® und XPlanar® sind eingetragene und lizenzierte Marken der Beckhoff Automation GmbH. Die Verwendung anderer in dieser Dokumentation enthaltenen Marken oder Kennzeichen durch Dritte kann zu einer Verletzung von Rechten der Inhaber der entsprechenden Bezeichnungen führen.

### Patente

Die EtherCAT-Technologie ist patentrechtlich geschützt, insbesondere durch folgende Anmeldungen und Patente: EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702 mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.



EtherCAT® ist eine eingetragene Marke und patentierte Technologie lizenziert durch die Beckhoff Automation GmbH, Deutschland.

### Copyright

© Beckhoff Automation GmbH & Co. KG, Deutschland.

Weitergabe sowie Vervielfältigung dieses Dokuments, Verwertung und Mitteilung seines Inhalts sind verboten, soweit nicht ausdrücklich gestattet.

Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patent-, Gebrauchsmuster- oder Geschmacksmustereintragung vorbehalten.

## 1.2 Sicherheitshinweise

### Sicherheitsbestimmungen

Beachten Sie die folgenden Sicherheitshinweise und Erklärungen!  
Produktspezifische Sicherheitshinweise finden Sie auf den folgenden Seiten oder in den Bereichen Montage, Verdrahtung, Inbetriebnahme usw.

### Haftungsausschluss

Die gesamten Komponenten werden je nach Anwendungsbestimmungen in bestimmten Hard- und Software-Konfigurationen ausgeliefert. Änderungen der Hard- oder Software-Konfiguration, die über die dokumentierten Möglichkeiten hinausgehen, sind unzulässig und bewirken den Haftungsausschluss der Beckhoff Automation GmbH & Co. KG.

### Qualifikation des Personals

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs-, Automatisierungs- und Antriebstechnik, das mit den geltenden Normen vertraut ist.

### Erklärung der Hinweise

In der vorliegenden Dokumentation werden die folgenden Hinweise verwendet.  
Diese Hinweise sind aufmerksam zu lesen und unbedingt zu befolgen!

#### **GEFAHR**

##### **Akute Verletzungsgefahr!**

Wenn dieser Sicherheitshinweis nicht beachtet wird, besteht unmittelbare Gefahr für Leben und Gesundheit von Personen!

#### **WARNUNG**

##### **Verletzungsgefahr!**

Wenn dieser Sicherheitshinweis nicht beachtet wird, besteht Gefahr für Leben und Gesundheit von Personen!

#### **VORSICHT**

##### **Schädigung von Personen!**

Wenn dieser Sicherheitshinweis nicht beachtet wird, können Personen geschädigt werden!

#### **HINWEIS**

##### **Schädigung von Umwelt/Geräten oder Datenverlust**

Wenn dieser Hinweis nicht beachtet wird, können Umweltschäden, Gerätebeschädigungen oder Datenverlust entstehen.



##### **Tipp oder Fingerzeig**

Dieses Symbol kennzeichnet Informationen, die zum besseren Verständnis beitragen.



## 1.3 Ausgabestände der Dokumentation

Version	Änderungen
2.4.0	<ul style="list-style-type: none"> <li>• Kapitel <i>Technische Daten - BX</i> aktualisiert</li> <li>• Kapitel <i>Hinweise zum ESD-Schutz</i> hinzugefügt</li> <li>• Kapitel <i>Entsorgung</i> hinzugefügt</li> <li>• Neue Titelseite</li> </ul>
2.3.0	<ul style="list-style-type: none"> <li>• Kapitel <i>Technische Daten - BX</i> aktualisiert</li> </ul>
2.2.0	<ul style="list-style-type: none"> <li>• Download-Links aktualisiert</li> <li>• Gestaltung der Sicherheitshinweise an IEC 82079-1 angepasst.</li> </ul>
2.1.0	PROFIBUS-Anschluss korrigiert
2.0.0	Migration
1.2.0	Aktualisierung zu Firmware-Version 1.20
1.1.7	Aktualisierung zu Firmware-Version 1.17
1.1.6	Aktualisierung zu Firmware-Version 1.16
1.1.4	Aktualisierung zu Firmware-Version 1.14
1.1.2	Aktualisierung zu Firmware-Version 1.12
1.1.0	Hinweise zur Einhaltung der UL-Anforderungen hinzugefügt.
1.0.9	Aktualisierung zu Firmware-Version 1.09
1.0.8	Aktualisierung zu Firmware-Version 1.08
1.0.7	Aktualisierung zu Firmware-Version 1.07
1.0.6	Aktualisierung zu Firmware-Version 1.06
1.0.5	Aktualisierung zu Firmware-Version 1.05
1.0.4	Aktualisierung zu Firmware-Version 1.04
1.0.1	Aktualisierung zu Firmware-Version 1.01
1.0.0	Aktualisierung zu Firmware-Version 1.00

### Firmware-Versionen des BX3100

Nach dem Einschalten zeigt der BX-Controller seine Firmware-Version für ca. drei Sekunden auf dem Display an.

Zum Update Ihrer Firmware benötigen Sie ein serielles Kabel, die Konfigurations-Software KS2000 oder das [Firmware Update Programm](#) [► 171].

Firmware	Beschreibung
1.27	<ul style="list-style-type: none"> <li>BX ohne K-Bus, wenn man im System Manager keinen K-Bus angelegt hat wird auch kein Fehler erzeugt (nur bei TC Config möglich)</li> </ul>
1.25	<ul style="list-style-type: none"> <li>SSB-Reset-Problem bei kurzgeschlossener CAN-Leitung gelöst</li> </ul>
1.24	<ul style="list-style-type: none"> <li>SSB fix wenn der CAN beim starten des BX nicht angeschlossen ist</li> </ul>
1.22	<ul style="list-style-type: none"> <li>Large Model implementiert</li> </ul>
1.20	<ul style="list-style-type: none"> <li>Firmware für BX-Controller ab Hardware-Version 3.5</li> <li>Umschaltung der COM 2 Schnittstelle zwischen RS232 und RS485 geändert</li> <li>Neu: Unterstützung der TwinSAFE-Busklemmen: maximal eine Logik-Klemme am K-Bus mit maximal 7 Verbindungen erlaubt (weitere Informationen)</li> </ul> <p><b>Achtung</b></p> <p>Die Firmware-Version 1.20 läuft nicht auf älteren Hardware-Versionen (kleiner als 3.5). Die Hardware-Version Ihres BX-Controllers entnehmen Sie bitte dessen Aufkleber.</p>
1.17	<ul style="list-style-type: none"> <li>Neu: Unterstützung von persistenten Daten</li> <li>Forcen und Lesen in der TwinCAT-Konfiguration korrigiert</li> <li>CFC-Tool wird unterstützt wie in Firmware 1.14 und kleiner</li> </ul>
1.16	<ul style="list-style-type: none"> <li>Online Change Problem behoben bei großen Projekten</li> <li>New: Display Beleuchtung kann abgeschaltet werden</li> <li>New: Anzeige der CRC des Bootprojektes kann beim Booten aktiviert werden</li> <li>REAL_TO_STRING: Bei Real Variablen mit zum Beispiel 9.0 wird diese jetzt auch so als String ausgegeben '9.0' vorher '9.'</li> <li>TC-Config 4 und 8-fach analog Klemmen gefixet</li> </ul>
1.14	<ul style="list-style-type: none"> <li>K-Bus aktualisiert</li> <li>SSB - Reset des SSB implementiert</li> <li>Funktion Generic implementiert</li> <li>Nicht implementierte Eigenschaften werden mit dem System Manager deaktiviert, Build 1303.</li> <li>Schreiben von Ausgängen sind mit TwinCAT Build 1302 möglich</li> <li>Optimierung der Klemmenüberprüfung bei einer TwinCAT Konfiguration</li> </ul>
1.12	<ul style="list-style-type: none"> <li>KL1212 wird in der TC-Config unterstützt</li> <li>KM-Module werden in der TC-Config unterstützt</li> </ul>
1.09 / 0.99i*	<ul style="list-style-type: none"> <li>Optimierung Code, dadurch ein Performance Steigerung von ca. 30%</li> <li>Optimierung der TcComPortBX.lbx - Fb_BX_BK8x00_Slave verbesserte Diagnose</li> <li>NEU: <u>CFC-Client</u> [► 173] Unterstützung implementiert (Controller Flash Copy - Client)</li> <li>NEU: <u>PROFIBUS-DP Diagnose Daten</u> [► 165] aus der PLC verschicken</li> </ul>
1.08 / 0.99h*	<ul style="list-style-type: none"> <li>Optimierung SSB</li> </ul>
1.07 / 0.99g*	<ul style="list-style-type: none"> <li>Optimierung serielles ADS</li> <li>Optimierung Online Change</li> </ul>
1.06 / 0.99f*	<ul style="list-style-type: none"> <li>Optimierung COM 1 serielles ADS</li> </ul>
1.05 / 0.99e*	<ul style="list-style-type: none"> <li>Wenn nach einem Online Change zu schnell ein PLC Reset ausgelöst wird, kommt eine Fehlermeldung, dass der Dienst noch nicht bereit ist.</li> <li>Optimierung SSB</li> <li>NEU: Disable der roten PLC-Zyklusüberwachungs-LED</li> </ul>
1.04 / 0.99d*	<ul style="list-style-type: none"> <li>NEU: Laden und Speichern von Rezepturen, neue Funktion in TcSystemBX fb_ReadWriteFile</li> <li>Optimierung: TcComPortBX, FB_BX_COM_64ex</li> </ul>
1.01 / 0.99a*	<ul style="list-style-type: none"> <li>Wenn nach einem Online Change zu schnell das Bootprojekt erzeugt wird, kommt eine Fehlermeldung, dass der Dienst noch nicht bereit ist.</li> </ul>
1.00 / 0.99*	<ul style="list-style-type: none"> <li>NEU: ADS Bausteine implementiert</li> <li>NEU: SSB: SDO Kommunikation aus der SPS heraus</li> <li>BugFix: Mapping-Probleme digitaler Signale in der TwinCAT-Config behoben</li> <li>NEU: Serielles ADS implementiert</li> <li>Genauigkeit der Taskzeit verbessert (0.95)</li> </ul>

\*) Die Versionen 1.xx unterstützten nur neue (blaue) Displays! Die Versionen 0.99x unterstützten nur alte (grüne) Displays!

## 2 Produktübersicht

### 2.1 Busklemmen-Controller der BX-Serie

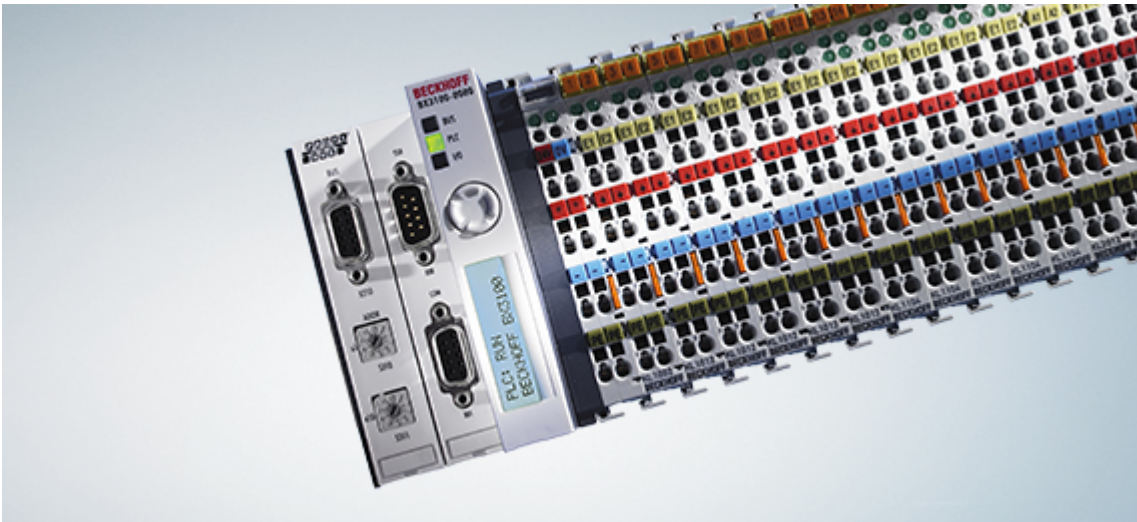


Abb. 1: Busklemmen-Controller der BX-Serie

Die Busklemmen-Controller der BX-Serie (BX-Controller) sind eine Steuerungsfamilie mit hohem Flexibilitätsgrad. Vom Ausstattungs- und Leistungsspektrum ist die BX-Serie zwischen den Busklemmen-Controller der BC-Serie und dem Embedded-PC CX1000 positioniert. Von der BC-Serie wurde das Konzept der autarken Steuerung in Kombination dem Anschluss für ein übergeordnetes Feldbusssystem übernommen. Das Gehäusekonzept stammt vom CX1000. Die Hauptunterscheidungsmerkmale zwischen BC- und BX-Serie sind die größere Speicherausstattung und die erweiterten Schnittstellen der BX-Serie.

Die BX-Controller bestehen aus einem programmierbaren IEC 61131-3 Controller, einem Anschluss für das übergeordnete Feldbusssystem und dem K-Bus-Interface zum Anschluss der Beckhoff Busklemmen. Zusätzlich verfügen die BX-Controller über zwei serielle Schnittstellen: eine für die Programmierung, die andere zur freien Nutzung. Im Gerät selbst enthalten sind ein beleuchtetes LC-Display (2 Zeilen mit je 16 Zeichen) mit Joystickschalter sowie eine Echtzeit-Uhr. Über den integrierten Beckhoff Smart System Bus (SSB) können weitere Peripheriegeräte, z. B. Displays, angeschlossen werden.

Die Busklemmen werden wie gewohnt auf der rechten Seite des BX-Controllers angesteckt. Durch das umfangreiche Spektrum an verschiedenen E/As kann jedes Eingangssignal gelesen und jedes benötigte Ausgangssignal erzeugt werden. Dadurch sind mit den BX-Controllern vielfältige Automatisierungsaufgaben lösbar, von der Garagentorsteuerung bis hin zur autarken Temperaturregelung an einer Spritzgussmaschine. Auch im Hinblick auf ein modulares Maschinenkonzept sind die BX-Controller hervorragend einzusetzen. Im Verbund kann der BX-Controller über die Feldbusschnittstelle mit anderen Maschinenteilen Daten austauschen. Die Echtzeit-Uhr ermöglicht auch einen dezentralen Einsatz, bei dem der Wochentag oder die Uhrzeit eine wichtige Rolle spielen.

Die Einsatzgebiete der BX-Serie sind denen der BC-Serie ähnlich, jedoch lassen sich mit dem BX, aufgrund der großen Speicherausstattung, wesentlich komplexere, größere Programme abarbeiten und lokal mehr Daten verwalten (z. B. Historie- und Trenddatenaufzeichnung), die dann sukzessive über den Feldbus abgeholt werden.

#### ● Busklemme und Endklemme erforderlich

**i** Zum Betrieb eines BX-Controllers müssen an dessen K-Bus mindestens eine Busklemme mit Prozessabbild und die Endklemme gesteckt sein!

## Feldbus-Interface

Die Varianten der Busklemmen-Controller der BX-Serie unterscheiden sich durch die unterschiedlichen Feldbusschnittstellen. Zusätzlich sind zwei serielle Schnittstellen für die Programmierung und für den Anschluss weiterer serieller Geräte integriert. Fünf verschiedene Ausführungen decken die wichtigsten Feldbussysteme ab:

- BX3100: PROFIBUS DP
- BX5100: CANopen
- BX5200: DeviceNet
- BX8000: RS232 oder RS485 (ohne Feldbusinterface)
- BX9000: Ethernet, ModbusTCP/ADS-TCP/UDP

## Programmierung

Programmiert werden die BX-Controller nach der leistungsfähigen IEC 61131-3 Norm. Wie auch bei allen anderen Beckhoff Steuerungen ist die Automatisierungssoftware TwinCAT Grundlage für die Parametrierung und Programmierung. Dem Anwender stehen also die gewohnten TwinCAT Werkzeuge, wie z. B. SPS-Programmieroberfläche, System Manager und TwinCAT Scope zur Verfügung. Der Datenaustausch erfolgt wahlweise über die serielle Schnittstelle (COM1) oder über den Feldbus via Beckhoff PC-Feldbuskarten FCxxxx.

## Konfiguration

Die Konfiguration erfolgt ebenfalls mit TwinCAT. Über den System Manager können das Feldbusinterface, der SSB-Bus und die Echtzeit-Uhr konfiguriert und parametrierung werden. Alle angeschlossenen Geräte und Busklemmen können vom System Manager ausgelesen werden. Die Konfiguration wird nach der Parametrierung über die serielle Schnittstelle auf den BX gespeichert. Diese erstellte Konfiguration kann auch wieder ausgelesen werden.

## 2.2 BX3100 - Einführung

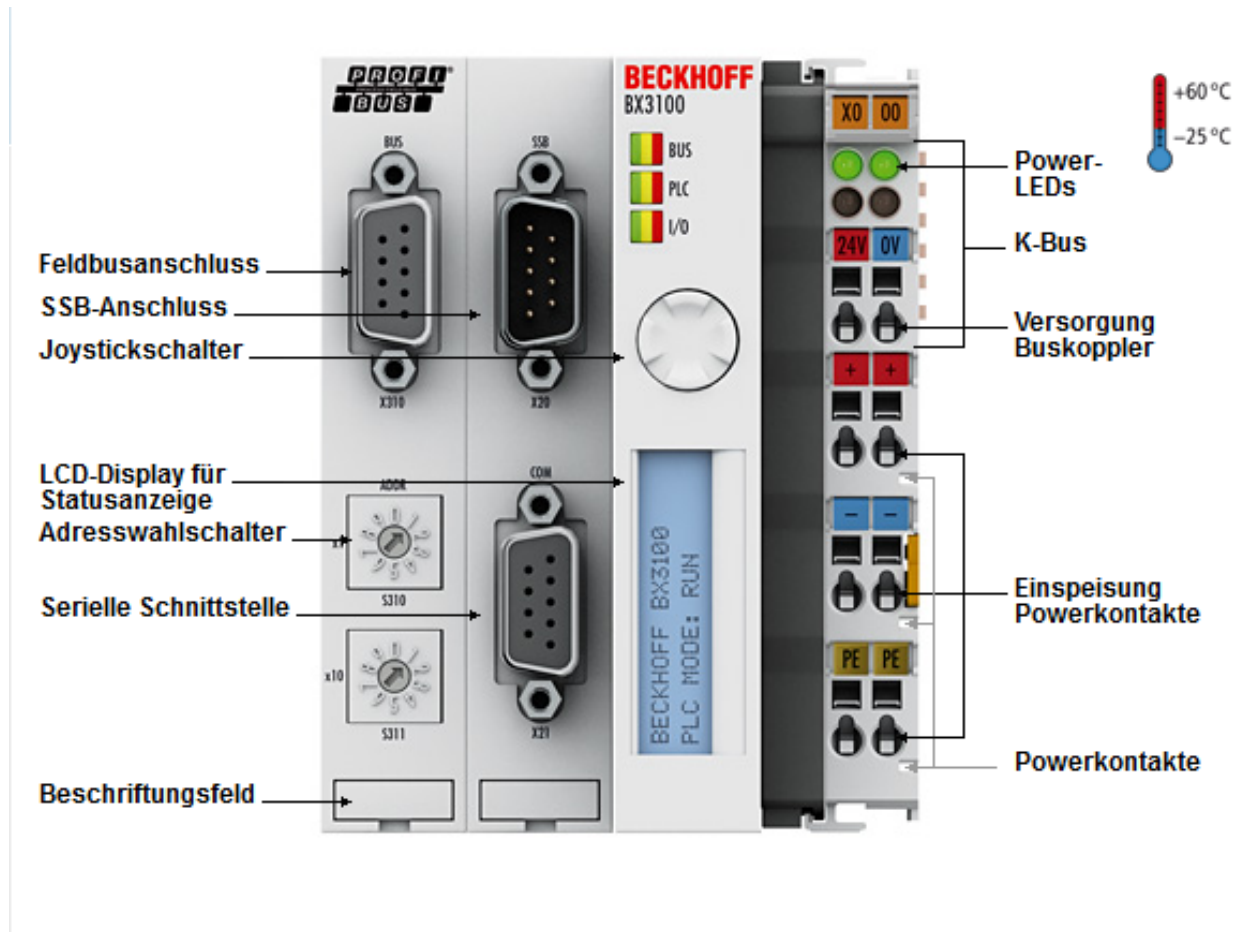


Abb. 2: BX3100

Der Busklemmen Controller BX3100 besitzt eine PROFIBUS-Slaveschnittstelle mit automatischer Baudratenerkennung bis 12 Mbaud sowie einen Adresswahlschalter für die Adressvergabe. Es können bis zu 244-Byte-Eingänge und 244-Byte-Ausgänge mit der Steuerung ausgetauscht werden.

## 2.3 Technische Daten

### 2.3.1 Technische Daten - BX

Technische Daten	BX3100	BX5100	BX5200	BX8000	BX9000
Prozessor	16 Bit Mikrocontroller				
Diagnose LEDs	2 x Spannungsversorgung, 2 x K-Bus				
Display	FSTN 2 x 16 Zeilen Display für Diagnose oder eigene Texte, beleuchtet				
Schalter	Joystickschalter für Parametrierung und Diagnose				
Uhr	interne akkugepufferte Uhr für Zeit und Datum				
Konfigurations- und Programmiersoftware	TwinCAT PLC				
Feldbus-Interface	PROFIBUS-DP	CANopen	DeviceNet	-	Ethernet
Feldbus-Anschluss	D-Sub, 9-pin	Open Style Connector, 5-pin		-	RJ45
Serielle Schnittstellen	COM1 (RS232 für Konfiguration und Programmierung, automatische Baudratenerkennung 9600/19200/38400/57600 Baud) COM2 (RS232 oder RS485) für den Anschluss serieller Geräte				
SSB	CANopen basierende Sub-Bus Schnittstelle				
Klemmenbus (K-Bus)	64 (255 mit K-Bus-Verlängerung)				
Digitale Peripheriesignale	2040 Ein-/Ausgänge				
Analoge Peripheriesignale	1024 Ein-/Ausgänge				
Konfigurationsmöglichkeit	über TwinCAT oder die Steuerung				
maximale Byte-Anzahl Feldbus	feldbusabhängig				
maximale Byte-Anzahl SPS	2048 Byte Eingangsdaten, 2048 Byte Ausgangsdaten				

Versorgung	BX3100	BX5100	BX5200	BX8000	BX9000
Spannungsversorgung (Us)	24 V <sub>DC</sub> (-15%/+20%), siehe UL-Anforderungen				
Eingangsstrom (Us)	180 mA + (ges. K-Bus Strom)/4				
Einschaltstrom (Us)	ca. 2,5 x Dauerstrom				
K-Bus-Strom (5 V)	maximal 1450 mA				
Spannung Powerkontakt (Up)	maximal 24 V <sub>DC</sub>				
Stromlast Powerkontakt (Up)	maximal 10 A, siehe UL-Anforderungen				
Spannungsfestigkeit	500 V (Powerkontakt/Versorgungsspannung/Feldbus)				

#### ⚠ VORSICHT



#### UL-Anforderungen

Für die Spannungsversorgungen des BX-Controllers (Us) und der Powerkontakte (Up) benutzen Sie eine 4 A Sicherung oder eine Spannungsversorgung, die *NEC Class 2* entspricht um die UL-Anforderungen zu erfüllen!

Technische Daten	BX3100	BX5100	BX5200	BX8000	BX9000
zulässiger Umgebungstemperaturbereich im Betrieb	0°C ... +55°C (vor Hardware-Version 4.4) -25°C ... +60°C (ab Hardware-Version 4.4)			0°C ... +55°C	
zulässiger Umgebungstemperaturbereich bei Lagerung	-20°C ... +85°C (vor Hardware-Version 4.4) -40°C ... +85°C (ab Hardware-Version 4.4)			-20°C ... +85°C	
Relative Feuchte	95% ohne Betauung				
Vibrations-/ Schockfestigkeit	gemäß EN 60068-2-6 / EN 60068-2-27				
EMV-Festigkeit / Aussendung	gemäß EN 61000-6-2 / EN 61000-6-4				
Schutzart	IP20				
Zulassungen/Kennzeichnungen*	CE, UKCA, cULus, EAC				

\*) Real zutreffende Zulassungen/Kennzeichnungen siehe seitliches Typenschild (Produktbeschriftung).

Mechanische Daten	BX3100	BX5100	BX5200	BX8000	BX9000
Gewicht	ca. 170 g				
Abmessungen (B x H x T)	ca. 83 mm x 100 mm x 90 mm (BX8000: ca. 65 mm x 100 mm x 90 mm)				
Montage	mit Verriegelung, auf Tragschiene (35 mm Hutschiene)				
Einbaulage	Beliebig				
Anschlussquerschnitt	0,08 mm <sup>2</sup> ... 2,5 mm <sup>2</sup> AWG 28 ... 14 8 ... 9 mm Abisolierlänge				

### 2.3.2 Technische Daten - PROFIBUS-DP

Systemdaten	PROFIBUS (BX3100)
Anzahl der E/A-Module	126 (BX3100 max. 99 Teilnehmer)
Anzahl der E/A-Punkte	steuerungsabhängig
Übertragungsmedium	abgeschirmtes Kupferkabel 2 x 0,25 mm <sup>2</sup> Leitungstyp A nach EN 50 170
Segment Länge	bis 1200 m
Anzahl der Segmente	3
Übertragungsrate	bis 12 MBaud
Topologie	RS485 Linie, LWL Ring
Übertragungszeit	ca. 0,5 ms bei 10 Stationen für je 32 Bit Ein/ und Ausgängen bei 12 MBaud

### 2.3.3 Technische Daten - SSB-Interface

Systemdaten	SSB-Interface
Max. Anzahl an Slaves	8
Max. Anzahl an PDOs	32 RxPODs / 32 TxPODs
Baudrate	10 k ... 1 MBaud
Erlaubte Slave-Adressen	1 bis 64

### 2.3.4 Technische Daten - SPS

SPS- Daten	BX3100	BX5100	BX5200	BX8000	BX9000
Programmiermöglichkeit	über Programmierschnittstelle (COM1 oder COM2) oder über Feldbus				
Programmspeicher	256 kByte				
Source Code Speicher	256 kByte				
Datenspeicher	256 kByte				
Remanente Merker	2 kByte				
SPS-Zykluszeit	ca. 0,85 ms für 1000 AWL Befehle (ohne E/A Zyklus)				
Programmiersprachen	IEC 6-1131-3 (AWL, KOP, FUP, ST, AS)				
Laufzeit	1 SPS Task (2. Task in Vorbereitung)				
Online Change	Ja				
Up/Down Load Code	Ja/Ja				

## 2.4 Prinzip der Busklemme

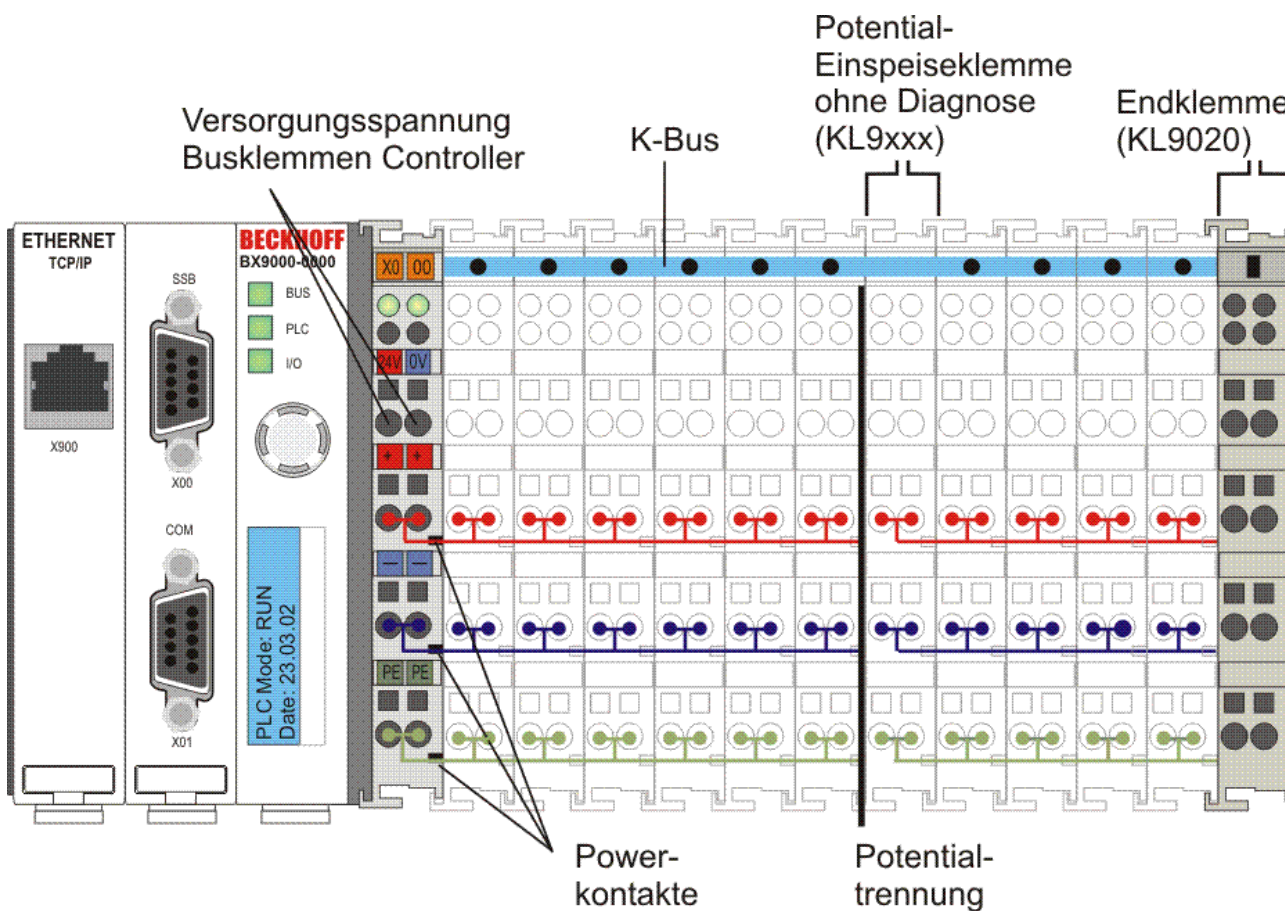


Abb. 3: Prinzip der Busklemme



## 2.5 Das Beckhoff Busklemmensystem

### Bis zu 256 Busklemmen mit ein bis 16 E/A-Kanälen für jede Signalform

Das Busklemmen-System ist das universelle Bindeglied zwischen einem Feldbus-System und der Sensor / Aktuator - Ebene. Eine Einheit besteht aus einem Buskoppler als Kopfstation und bis zu 64 elektronischen Reihenklemmen, wovon die letzte eine Endklemme ist. Mit der K-Bus Erweiterung können bis zu 255 Busklemmen angeschlossen werden. Für jede technische Signalform stehen Klemmen mit ein, zwei, vier oder acht E/A-Kanälen zur Verfügung, die beliebig gemischt werden können. Dabei haben alle Klemmentypen die gleiche Bauform, wodurch der Projektierungsaufwand sehr gering gehalten wird. Bauhöhe und Tiefe sind auf kompakte Klemmenkästen abgestimmt.

### Dezentrale Verdrahtung der E/A-Ebene

Die Feldbustechnik erlaubt den Einsatz kompakter Steuerungsbauformen. Die E/A-Ebene muss nicht bis zur Steuerung geführt werden. Die Verdrahtung der Sensoren und Aktuatoren ist dezentral mit minimalen Kabellängen durchführbar. Der Installationsstandort der Steuerung kann im Bereich der Anlage beliebig gewählt werden.

### Industrie-PCs als Steuerung

Durch den Einsatz eines Industrie-PCs als Steuerung lässt sich das Bedien- und Beobachtungselement in der Hardware der Steuerung realisieren. Der Standort der Steuerung kann deshalb ein Bedienpult, eine Leitwarte oder ähnliches sein. Die Busklemmen stellen die dezentrale Ein-/Ausgabebene der Steuerung im Schaltschrank und untergeordneten Klemmenkästen dar. Neben der Sensor/Aktuator-Ebene wird auch der Leistungsteil der Anlage über das Bussystem gesteuert. Die Busklemme ersetzt die konventionelle Reihenklemme als Verdrahtungsebene im Schaltschrank. Der Schaltschrank kann kleiner dimensioniert werden.

### Buskoppler für alle gängigen Bussysteme

Das Beckhoff Busklemmen-System vereint die Vorteile eines Bussystems mit den Möglichkeiten der kompakten Reihenklemme. Busklemmen können an allen gängigen Bussystemen betrieben werden und verringern so die Teilevielfalt in der Steuerung. Dabei verhalten sich Busklemmen wie herkömmliche Anschaltungen dieses Bussystems. Alle Leistungsmerkmale des jeweiligen Bussystems werden unterstützt.

### Montage auf genormten Tragschienen

Die einfache und platzsparende Montage auf einer genormten Tragschiene (EN 60715, 35 mm) und die direkte Verdrahtung von Aktoren und Sensoren ohne Querverbindungen zwischen den Klemmen standardisiert die Installation. Dazu trägt auch das einheitliche Beschriftungskonzept bei.

Die geringe Baugröße und die große Flexibilität des Busklemmen-Systems ermöglichen den Einsatz überall dort, wo auch eine Reihenklemme zur Anwendung kommt. Jede Art von Ankopplung, wie analoge, digitale, serielle oder der Direktanschluss von Sensoren kann realisiert werden.

### Modularität

Die modulare Zusammenstellung der Klemmleiste mit Busklemmen verschiedener Funktionen begrenzt die Zahl der ungenutzten Kanäle auf maximal einen pro Funktion. Die Anzahl von zwei Kanälen in einer Klemme trifft das Optimum zwischen der Zahl der ungenutzten Kanäle und den Kosten pro Kanal. Auch die Möglichkeit der Potentialtrennung durch Einspeiseklemmen hilft, die Anzahl der ungenutzten Kanäle gering zu halten.

### Anzeige des Kanalzustands

Die integrierten Leuchtdioden zeigen in Sensor/Aktuator-Nähe den Zustand des entsprechenden Kanals an.

## **K-Bus**

Der K-Bus ist der Datenweg innerhalb der Klemmleiste. Über sechs Kontakte an den Seitenwänden der Klemmen wird der K-Bus vom Buskoppler durch alle Klemmen geführt. Die Endklemme schließt den K-Bus ab. Der Benutzer muss sich keinerlei Wissen über die Funktion des K-Bus oder die interne Arbeitsweise von Klemmen und Buskoppler aneignen. Viele lieferbare Software-Tools erlauben eine komfortable Projektierung, Konfiguration und Bedienung.

## **Potential-Einspeiseklemmen für potentialgetrennte Gruppen**

Über drei Powerkontakte wird die Betriebsspannung an die nachfolgenden Klemmen weitergegeben. Durch den Einsatz von Potential-Einspeiseklemmen, können Sie die Klemmleiste in beliebige potentialgetrennte Gruppen gliedern. Die Potential-Einspeiseklemmen werden bei der Ansteuerung der Klemmen nicht berücksichtigt, sie dürfen an beliebiger Stelle in die Klemmleiste eingereiht werden.

In einem Klemmenblock können Sie bis zu 64 Busklemmen einsetzen und diesen über die K-Busverlängerung auf bis zu 256 Busklemmen erweitern. Dabei werden Potential-Einspeiseklemmen mitgezählt, die Endklemme nicht.

## **Buskoppler für verschiedene Feldbus-Systeme**

Verschiedene Buskoppler lassen sich einsetzen, um die elektronische Klemmleiste schnell und einfach an unterschiedliche Feldbus-Systeme anzukoppeln. Auch eine nachträgliche Umrüstung auf ein anderes Feldbus-System ist möglich. Der Buskoppler übernimmt alle Kontroll- und Steuerungsaufgaben, die für den Betrieb der angeschlossenen Busklemmen notwendig sind. Die Bedienung und Konfiguration der Busklemmen wird ausschließlich über den Buskoppler durchgeführt. Die eingestellten Parameter werden jedoch spannungsausfallsicher in den jeweiligen Busklemmen gespeichert. Feldbus, K-Bus und E/A-Ebene sind galvanisch getrennt.

Wenn der Datenaustausch über den Feldbus zeitweise gestört ist oder ausfällt, bleiben Registerinhalte (wie z. B. Zählerstände) erhalten, digitale Ausgänge werden gelöscht und analoge Ausgänge nehmen einen Wert an, der bei der Inbetriebnahme für jeden Ausgang konfigurierbar ist. Die Default-Einstellung der analogen Ausgänge ist 0 V bzw. 0 mA. Digitale Ausgänge fallen in einen inaktiven Zustand zurück. Die Timeout-Zeiten der Buskoppler entsprechen den für das Feldbus-System üblichen Zeiten. Bei der Umstellung auf ein anderes Bussystem beachten Sie im Falle großer Zykluszeiten des Bussystems die Änderung der Timeout-Zeiten.

## **Die Schnittstellen**

Ein Buskoppler besitzt sechs unterschiedliche Anschlussmöglichkeiten. Diese Schnittstellen sind als Steckverbindungen und Federkraftklemmen ausgelegt.

## 3 Montage und Verdrahtung

### 3.1 Hinweise zum ESD-Schutz

#### HINWEIS

##### Zerstörung der Geräte durch elektrostatische Aufladung möglich!

Die Geräte enthalten elektrostatisch gefährdete Bauelemente, die durch unsachgemäße Behandlung beschädigt werden können.

- Sie müssen beim Umgang mit den Komponenten elektrostatisch entladen sein; vermeiden Sie außerdem die Federkontakte (s. Abb.) direkt zu berühren.
- Vermeiden Sie den Kontakt mit hoch isolierenden Stoffen (Kunstfaser, Kunststofffolien etc.)
- Beim Umgang mit den Komponenten ist auf gute Erdung der Umgebung zu achten (Arbeitsplatz, Verpackung und Personen)
- Jede Busstation muss auf der rechten Seite mit der Endklemme KL9010 abgeschlossen werden, um Schutzart und ESD-Schutz sicher zu stellen.

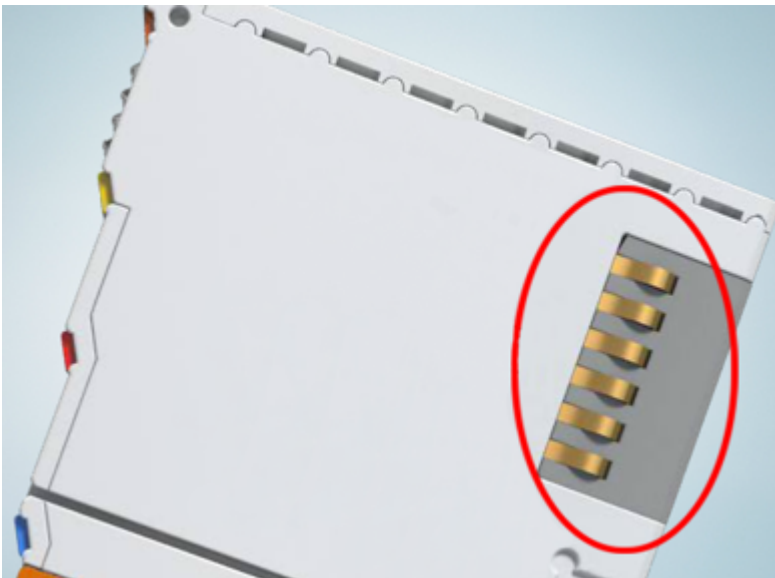


Abb. 4: Federkontakte der Beckhoff I/O-Komponenten

### 3.2 Montage

#### ⚠️ WARNUNG

##### Verletzungsgefahr durch Stromschlag und Beschädigung des Gerätes möglich!

Setzen Sie das Busklemmen-System in einen sicheren, spannungslosen Zustand, bevor Sie mit der Montage, Demontage oder Verdrahtung der Komponenten beginnen!

#### 3.2.1 Abmessungen

Das Beckhoff Busklemmen-System zeichnet sich durch geringes Bauvolumen und hohe Modularität aus. Für die Projektierung muss ein Buskoppler und eine Anzahl von Busklemmen vorgesehen werden. Die Abmessungen der Busklemmen-Controller sind unabhängig vom Feldbus-System.

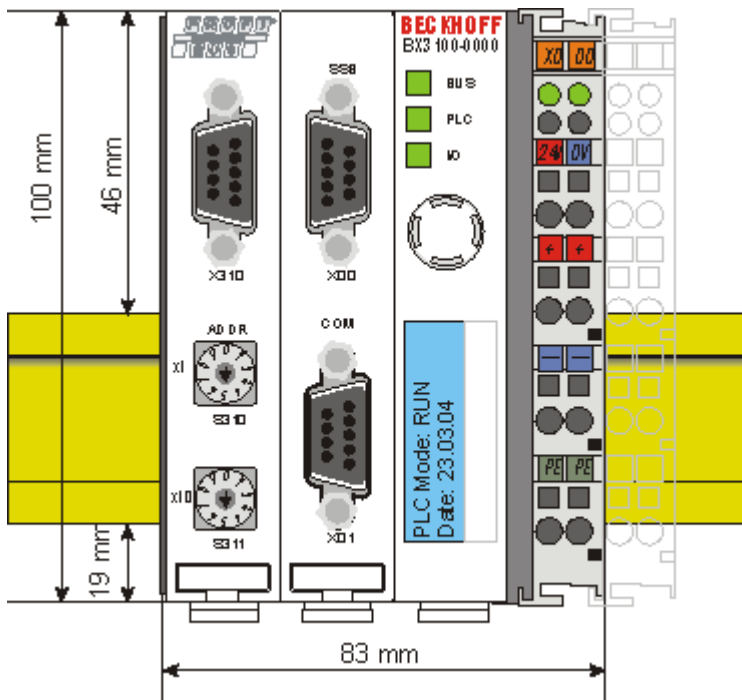


Abb. 5: BX3100, BX5100, BX5200, BX9000

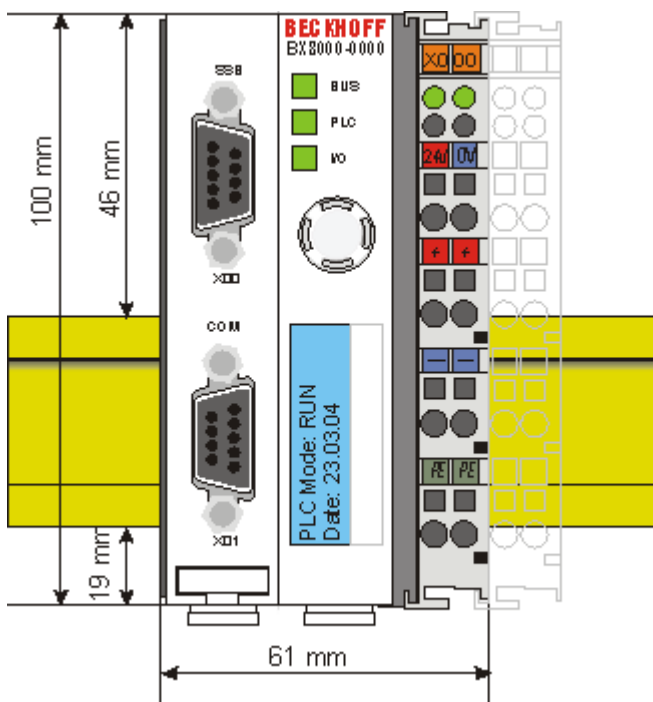


Abb. 6: BX8000

Die Gesamtbreite der Feldbusstation setzt sich aus der Breite des Busklemmen-Controllers und der Breite der verwendeten Busklemmen (incl. Busendklemme KL9010) zusammen. Die Busklemmen sind je nach Bauform 12 mm oder 24 mm breit. Die Die Höhe beträgt 100 mm.

Die Busklemmen-Controller der BX-Serie sind bis zu 83 mm breit und 91 mm tief.

**● Gesamttiefe beachten**

**i** Beachten sie, dass ein Busklemmen-Controller mit Hutschiene und angeschlossenen Steckern meist tiefer aufbaut als die angegebenen 91 mm. Beispiel:  
 BX3100 + ZB3100 + Hutschiene = 105 mm

### 3.2.2 Tragschienenmontage

#### Montage

1. Auf der Unterseite der BX-Controller befinden sich weiße Zuglaschen, die mit einem Rastmechanismus verbunden sind.  
Ziehen Sie diese Zuglaschen nach unten, bevor Sie den BX-Controller auf die Tragschiene drücken.

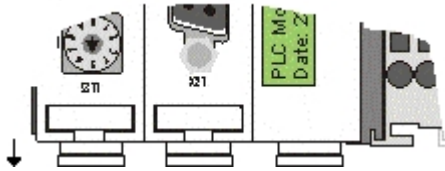


Abb. 7: Entriegelter BX-Controller

#### HINWEIS

#### Display bei der Montage nicht beschädigen!

Achten Sie darauf, nicht auf das Display zu drücken, wenn Sie den BX-Controller auf die Tragschiene drücken. Andernfalls kann das Display beschädigt werden.

2. Drücken Sie den BX-Controller nun auf die Tragschiene.
3. Nach dem Aufrasten auf die Tragschiene, schieben Sie die Zuglaschen wieder in die Ausgangsstellung zurück.

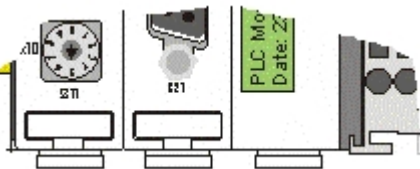


Abb. 8: Verriegelter BX-Controller

#### Demontage

1. Entriegeln Sie zuerst alle Zuglaschen an der Unterseite des BX-Controllers.
2. Ziehen Sie dann an der orangenen Lasche neben der Einspeisung für die Powerkontakte.

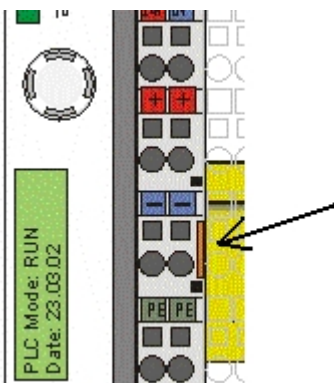


Abb. 9: Demontage

## 3.3 Entsorgung



Mit einer durchgestrichenen Abfalltonne gekennzeichnete Produkte dürfen nicht in den Hausmüll. Das Gerät gilt bei der Entsorgung als Elektro- und Elektronik-Altgerät. Die nationalen Vorgaben zur Entsorgung von Elektro- und Elektronik-Altgeräten sind zu beachten.

## 3.4 Verdrahtung

### ⚠️ WARNUNG

**Verletzungsgefahr durch Stromschlag und Beschädigung des Gerätes möglich!**

Setzen Sie das Busklemmen-System in einen sicheren, spannungslosen Zustand, bevor Sie mit der Montage, Demontage oder Verdrahtung der Komponenten beginnen!

### 3.4.1 Potentialgruppen, Isolationsprüfung und PE

#### Potentialgruppen

Ein Beckhoff Busklemmenblock verfügen in der Regel über drei verschiedene Potentialgruppen:

- Die Feldbusschnittstelle ist (außer bei einzelnen Low Cost Kopplern) galvanisch getrennt und bildet die erste Potentialgruppe.
- Buskoppler- / Busklemmen-Controller-Logik, K-Bus und Klemmenlogik bilden eine zweite galvanisch getrennte Potentialgruppe.
- Die Ein- und Ausgänge werden über die Powerkontakte gespeist und bilden weitere Potentialgruppen.

Gruppen von E/A-Klemmen lassen sich durch Potentialeinspeiseklemmen oder Trennklemmen zu weiteren Potentialgruppen zusammenfassen.

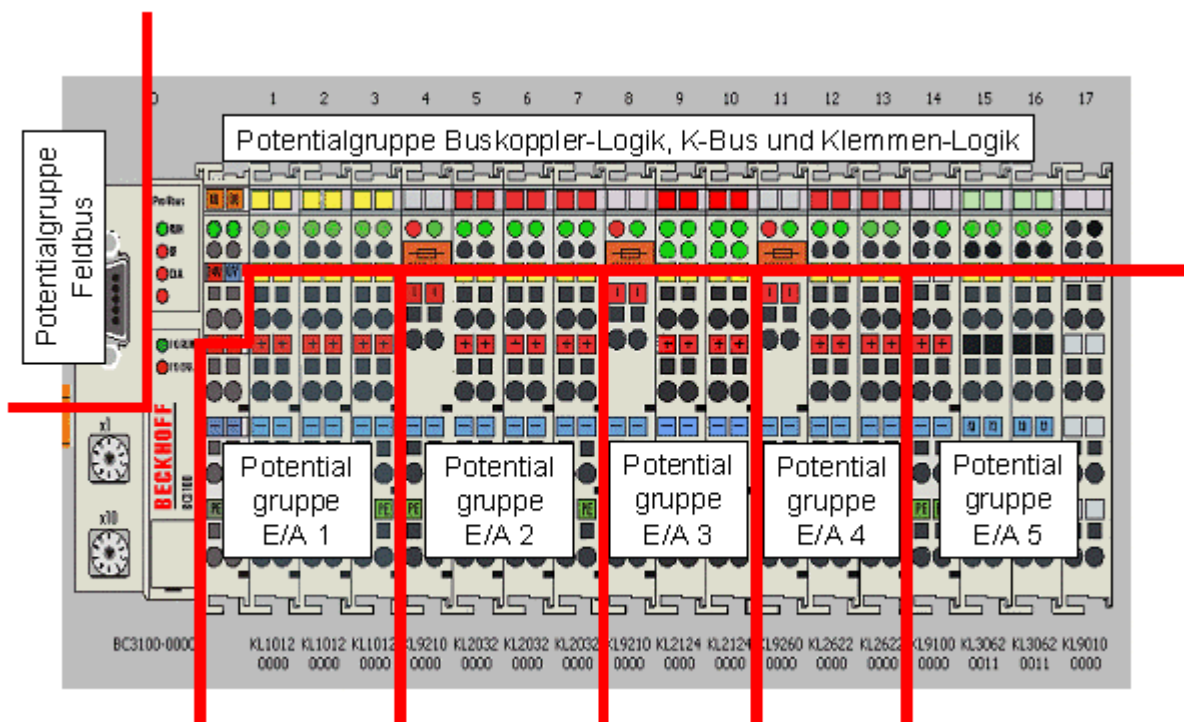


Abb. 10: Potentialgruppen eines Busklemmenblocks

#### Isolationsprüfung

Die Verbindung zwischen Buskoppler- / Busklemmen-Controller und Busklemmen wird durch das Zusammenstecken der Komponenten automatisch realisiert. Die Übertragung der Daten und die Versorgungsspannung der intelligenten Elektronik der Busklemmen übernimmt der K-Bus. Die Versorgung der Feldelektronik wird über die Powerkontakte durchgeführt. Die Powerkontakte stellen durch das Zusammenstecken eine Versorgungsschiene dar. Da einige Busklemmen (z. B. analoge Busklemmen oder digitale Vierkanal-Busklemmen) diese Powerkontakte nicht oder nicht vollständig durchschleifen, sind die Kontaktbelegungen der Busklemmen zu beachten.

Die Einspeiseklemmen unterbrechen die Powerkontakte und stellen den Anfang einer neuen Versorgungsschiene dar. Der Buskoppler- / Busklemmen-Controller kann auch zur Einspeisung der Powerkontakte eingesetzt werden.

**PE-Powerkontakte**

Der Powerkontakt mit der Bezeichnung PE kann als Schutz Erde eingesetzt werden. Der Kontakt ist aus Sicherheitsgründen beim Zusammenstecken voreilend und kann Kurzschlussströme bis 125 A ableiten.

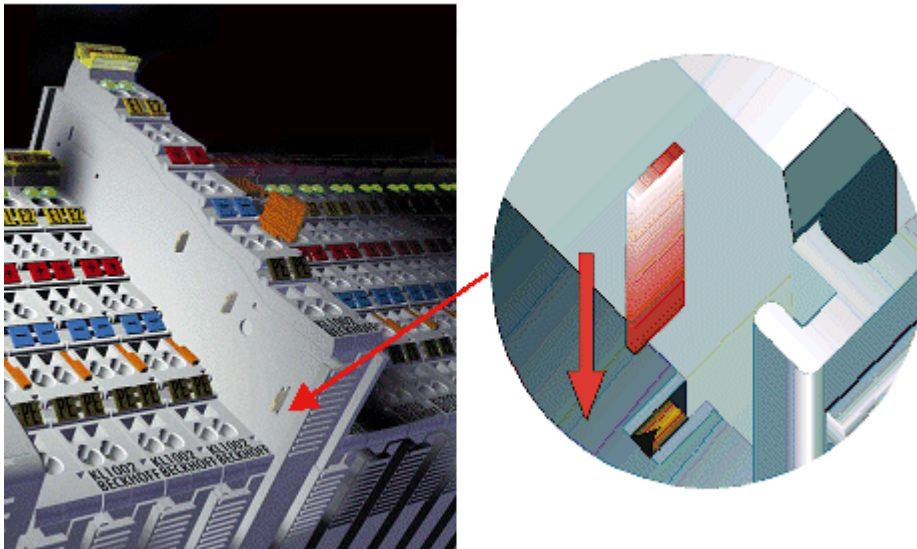




Abb. 11: Linksseitiger Powerkontakt

Es ist zu beachten, dass aus EMV-Gründen die PE-Kontakte kapazitiv mit der Tragschiene verbunden sind. Das kann zu falschen Ergebnissen und auch zur Beschädigung der Klemme bei der Isolationsprüfung führen (z. B. Isolationsdurchschlag an einem 230 V-Verbraucher zur PE-Leitung). Die PE-Zuleitung am Buskoppler- / Busklemmen-Controller muss zur Isolationsprüfung abgeklemmt werden. Um weitere Einspeisestellen für die Prüfung zu entkoppeln, können die Einspeiseklemmen aus dem Verbund der übrigen Klemmen mindestens 10 mm herausgezogen werden. Die PE-Zuleitungen müssen in diesem Fall nicht abgeklemmt werden.

Der Powerkontakt mit der Bezeichnung PE darf nicht für andere Potentiale verwendet werden.

**3.4.2 Spannungsversorgung**

<b>⚠ VORSICHT</b>	
	<p><b>Beachten Sie die UL-Anforderungen für die Spannungsversorgung!</b></p> <p>Diese UL-Anforderungen gelten für alle Versorgungsspannungen der BX-Controller (Us und Up)!</p> <p>Zur Einhaltung der UL-Anforderungen dürfen die BX-Controller nur mit Versorgungsspannungen (24 V<sub>DC</sub>) versorgt werden, die</p> <ul style="list-style-type: none"> <li>• von einer isolierten, mit einer Sicherung (entsprechend UL248) von maximal 4 A geschützten Quelle, oder</li> <li>• von einer Spannungsquelle die NEC class 2 entspricht stammen.</li> </ul> <p>Eine Spannungsquelle entsprechend NEC class 2 darf nicht seriell oder parallel mit einer anderen NEC class 2 entsprechenden Spannungsquelle verbunden werden!</p>
<b>⚠ VORSICHT</b>	
	<p><b>Keine unbegrenzten Spannungsquellen!</b></p> <p>Zur Einhaltung der UL-Anforderungen dürfen die BX-Controller nicht mit unbegrenzten Spannungsquellen verbunden werden!</p>

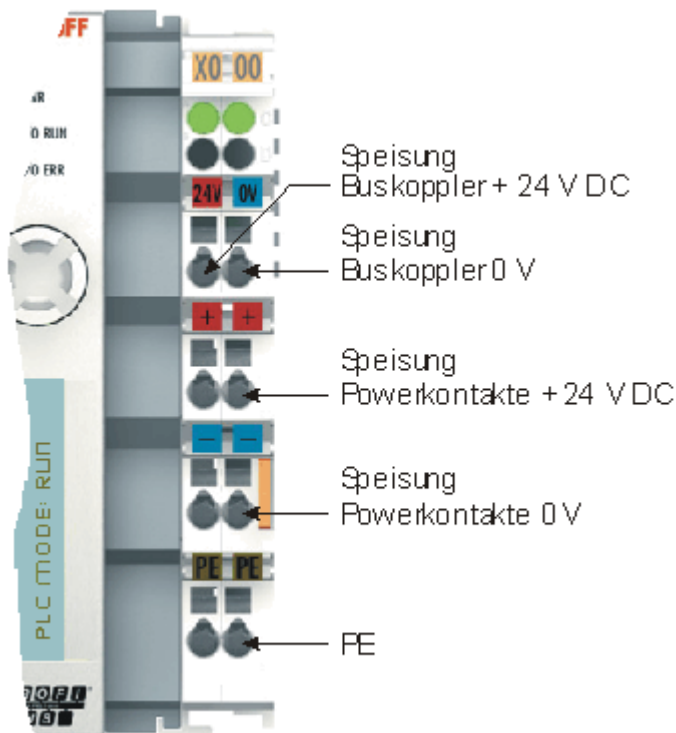


Abb. 12: Klemmstellen zur Versorgung des Busklemmen-Controllers

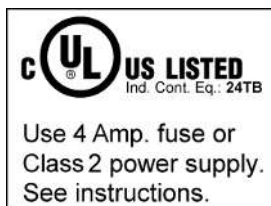


Abb. 13: UL-Kennzeichnung der BX-Controller

### Versorgung von Busklemmen-Controller und Busklemmen (Us)

Der Busklemmen-Controller benötigt zum Betrieb eine Versorgungsspannung von 24 V<sub>DC</sub>.

Der Anschluss findet beim BX-Controller über die oberen Klemmstellen mit der Bezeichnung 24 V und 0 V statt. Diese Versorgungsspannung versorgt die Elektronik des Busklemmen-Controllers sowie über den K-Bus die Elektronik der Busklemmen. Sie ist galvanisch von der Spannung der Feldebene getrennt.

### Versorgung der Powerkontakte (Up)

Die unteren sechs Anschlüsse mit Federkraft - Klemmen können zur Einspeisung der Peripherieversorgung benutzt werden. Die Federkraftklemmen sind paarweise mit einem Powerkontakt verbunden. Die Einspeisung zu den Powerkontakten besitzt keine Verbindung zur Spannungsversorgung der BX-Elektronik. Die Auslegung der Einspeisung lässt Spannungen bis zu 24 V zu.

Die Federkraftklemmen sind für Drähte von 0,08 mm<sup>2</sup> bis 2,5 mm<sup>2</sup> Querschnitt ausgelegt.

Die paarweise Anordnung und die elektrische Verbindung zwischen den Speiseklemmkontakten ermöglicht das Durchschleifen der Anschlussdrähte zu unterschiedlichen Klemmpunkten. Die Strombelastung über den Powerkontakt darf 10 A nicht dauerhaft überschreiten. Die Strombelastbarkeit zwischen zwei Federkraftklemmen ist mit der Belastbarkeit der Verbindungsdrähte identisch.



**Powerkontakte**

An der rechten Seitenfläche des Busklemmen Controllers befinden sich drei Federkontakte der Powerkontaktverbindungen. Die Federkontakte sind in Schlitzfenstern verborgen um einen Berührungsschutz sicher zu stellen. Durch das Anreihen einer Busklemme werden die Messerkontakte auf der linken Seite der Busklemme mit den Federkontakten verbunden. Die Nut/Federführung an der Ober- und Unterseite der Busklemmen Controller und Busklemmen garantiert eine sichere Führung der Powerkontakte.

**3.4.3 Programmierkabel für COM1**

Für die Programmierung der BX-Controller können Sie ein 1:1 Kabel nehmen (Buchse/Stecker und nur die unten angegebenen PINs verdrahten). Auf der BX-Seite benötigen Sie einen neunpoligen Stecker und auf der PC-Seite meist eine neunpolige Buchse. Die Verdrahtung ist 1:1 und die benötigten PINs sind aus der unten angegebenen Tabelle zu entnehmen. Die Länge des Kabels darf 5 Meter nicht überschreiten!

Beschreibung	BX COM Port 1	PC COM Port RS 232 Serielle Schnittstelle
Kabel	Stecker, Pin	Buchse, Pin
RS 232 RxD/TxD	2	2
RS 232 RxD/TxD	3	3
GND	5	5

**HINWEIS**

**Alle in der Tabelle nicht aufgeführten Pins sind reserviert**

Beachten Sie, dass an diesem COM Port [▶ [26](#)] die hier nicht aufgeführten Pins nicht frei verfügbar, sondern mit weiteren Signalen belegt sind.

**ZK1000-0030**

Das Programmierkabel bietet die Möglichkeit über die Schnittstelle COM 1 den BX-Controller zu programmieren und an der Schnittstelle COM 2 ein weiteres serielles Gerät anzuschließen. Achten Sie im eingebauten Zustand auf die maximale Bauhöhe des Steckers.

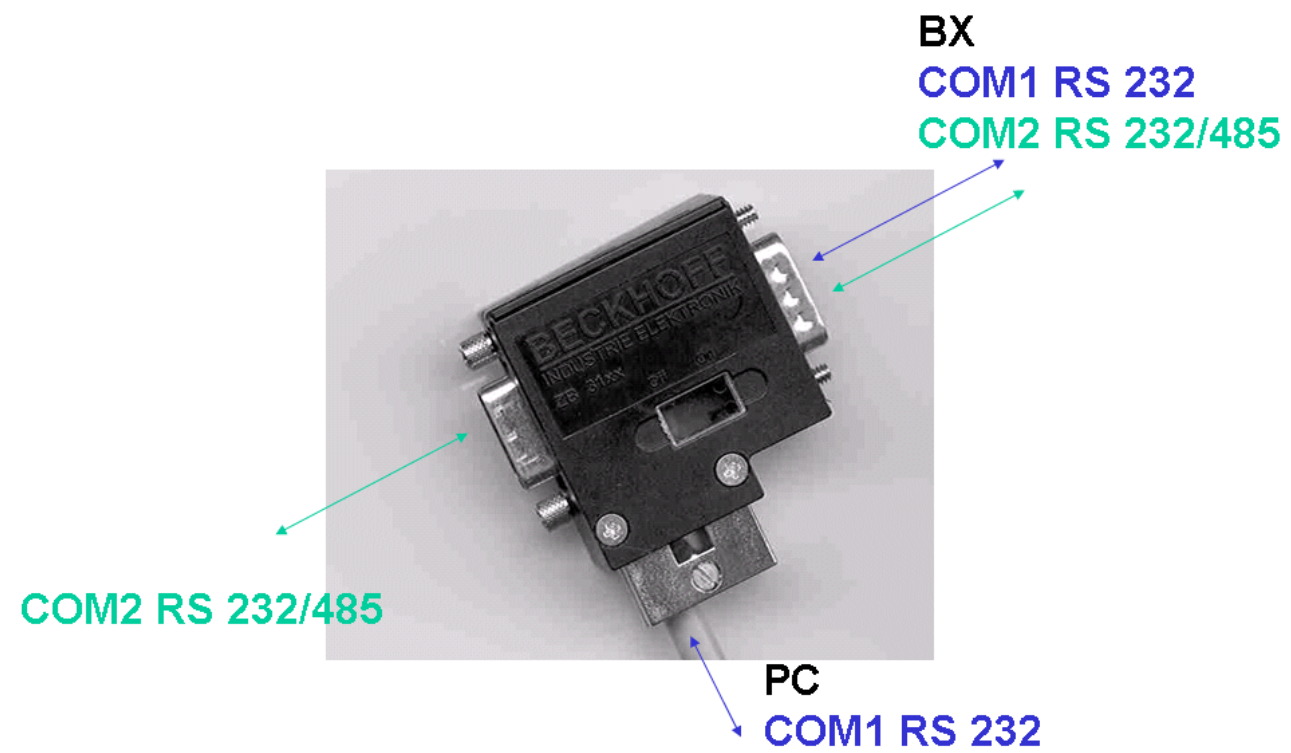


Abb. 14: Programmierkabel ZK1000-0030 - COM 1 und COM 2

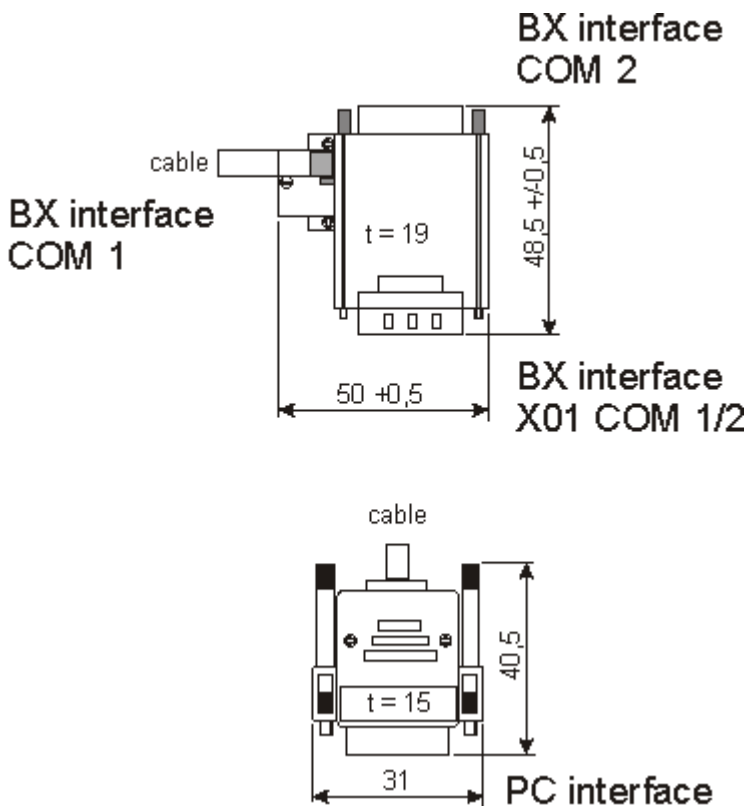


Abb. 15: Programmierkabel ZK1000-0030 - Abmessungen der Stecker

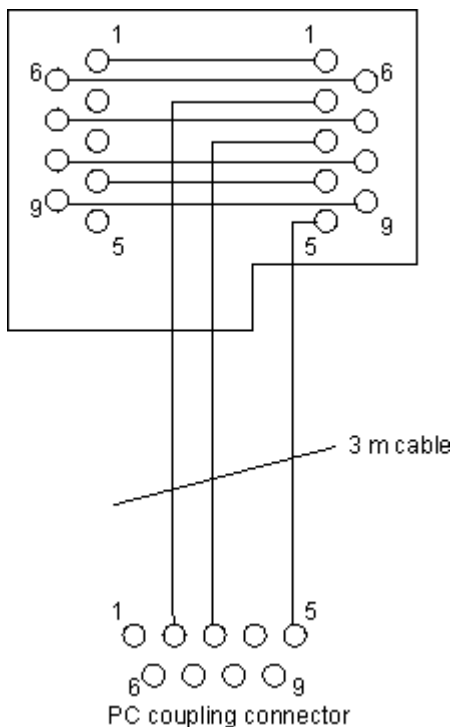


Abb. 16: Programmierkabel ZK1000-0030 - Pinning

### 3.4.4 SSB- und COM-Schnittstellen

Das Grundmodul der BX-Controller besitzt die Schnittstellen COM1, COM2 und SSB (Smart System. COM1 und COM2 sind auf einer D-Sub-Buchse untergebracht. Ein spezielles Programmierkabel (ZK1000-0030) mit dem Sie beide Schnittstellen verwenden können, können Sie bei Beckhoff bestellen. Die Schnittstelle COM2

ist für den Anschluss serieller Geräte vorgesehen. Sie können bei der Schnittstelle COM2 zwischen RS232 oder RS485 wählen.

Für die serielle Schnittstelle COM2 stehen [Bibliotheken \[► 116\]](#) zur Verfügung.

**SSB-Schnittstelle**

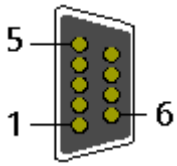


Abb. 17: SSB-Schnittstelle

**Belegung der SSB-Schnittstelle (Stecker X00)**

PIN	Signal
1	reserviert
2	CAN low
3	GND
4	reserviert
5	Shield
6	GND
7	CAN high
8	reserviert
9	reserviert

**COM1 (RS 232) und COM2 (RS 232/485) Schnittstelle**

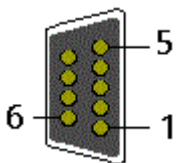


Abb. 18: COM1- (RS 232) und COM2-Schnittstelle (RS 232/485)

**Belegung der COM-Schnittstellen (Buchse X01)**

PIN	Schnittstelle	Signal
1	COM2	RS485 D+
2	COM1	RS232 TxD
3	COM1	RS232 RxD
4	VCC +5 V	VCC
5	GND	GND
6	COM2	RS485 D-
7	COM2	RS232 RxD
8	COM2	RS232 TxD
9	GND	GND

### 3.4.5 PROFIBUS-Anschluss

#### M12-Rundsteckverbinder

Die M12-Buchse ist invers codiert und besitzt 5 Stifte. Stift 1 überträgt 5 V<sub>DC</sub> und Stift 3 überträgt GND für den aktiven Abschlusswiderstand. Diese dürfen auf keinen Fall für andere Funktionen missbraucht werden, da dies zu Zerstörung des Gerätes führen kann.

Stift 2 und Stift 4 übertragen die Signale des PROFIBUS. Diese dürfen auf keinen Fall getauscht werden, da sonst die Kommunikation gestört ist. Stift 5 ist überträgt den Schirm (Shield) der kapazitiv mit der Grundfläche der Feldbus Box verbunden ist.

#### Pinbelegung M12 Buchse (-B310)

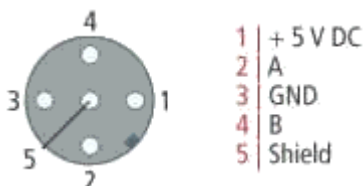


Abb. 19: Pinbelegung M12 Buchse (-B310)

#### Pinbelegung M12 Buchse/Stecker (-B318)

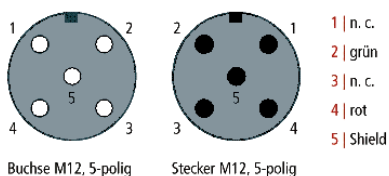


Abb. 20: Pinbelegung M12 Buchse/Stecker (-B318)

#### Neunpoliger D-Sub

Stift 6 überträgt 5 V<sub>DC</sub> und Stift 5 GND für den aktiven Abschlusswiderstand. Diese dürfen auf keinen Fall für andere Funktionen missbraucht werden, da dies zu Zerstörung des Gerätes führen kann.

Stift 3 und Stift 8 übertragen die Signale des PROFIBUS. Diese dürfen auf keinen Fall getauscht werden, da sonst die Kommunikation gestört ist.

#### Pinbelegung der PROFIBUS D-Sub Buchse

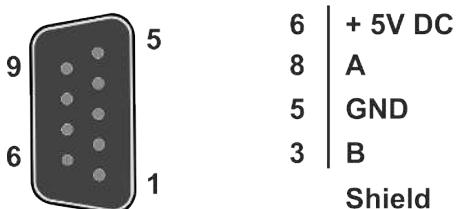


Abb. 21: Pinbelegung der PROFIBUS D-Sub Buchse

#### Leitungsfarben PROFIBUS

PROFIBUS Leitung	M12	D-Sub
B rot	Stift 4	Stift 3
A grün	Stift 2	Stift 8

**Anschluss der Feldbus Box Module**

Der Anschluss der Feldbus Box Module erfolgt entweder direkt oder mittels T-Stück (oder Y-Stück).

Die B318 Serie verfügt über jeweils eine Buchse und einen Stecker, d.h. hier wird der PROFIBUS in dem Modul weitergeleitet. Die Versorgungsspannung (+5 V<sub>DC</sub>) für den Abschluss-Widerstand liegt nur auf der Buchse an. Der Abschluss-Widerstand ZS1000-1610 steht nur als Stecker zur Verfügung!

Die ankommende PROFIBUS-Leitung sollte stets mit einer Buchse enden.

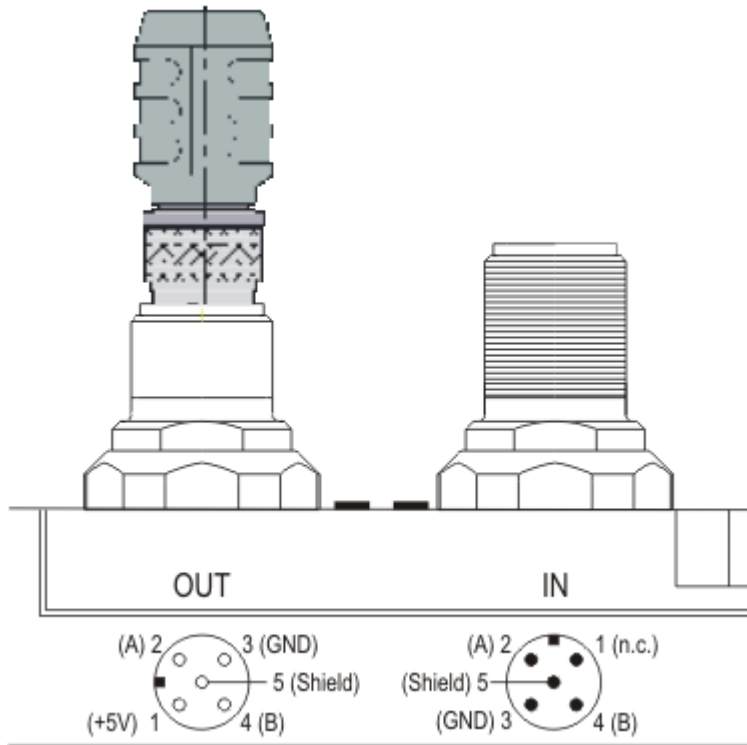


Abb. 22: Pin-Belegung Buchse/Stecker Feldbus Box Module

Es stehen zwei T-Stücke zur Verfügung:

- ZS1031-2600 mit +5 V<sub>DC</sub> Weiterleitung zur Versorgung des Abschluss-Widerstandes
- ZS1031-2610 ohne +5 V<sub>DC</sub> Weiterleitung

**3.4.6 PROFIBUS-Verkabelung**

Die physikalische Datenübertragung ist in der PROFIBUS-Norm definiert (siehe PROFIBUS Schicht 1: Physical Layer).

Der Einsatzbereich eines Feldbus-Systems wird wesentlich durch die Wahl des Übertragungs-Mediums und der physikalischen Busschnittstelle bestimmt. Neben den Anforderungen an die Übertragungssicherheit sind die Aufwendungen für Beschaffung und Installation des Buskabels von entscheidender Bedeutung. Die PROFIBUS-Norm sieht daher verschiedene Ausprägungen der Übertragungstechnik unter Beibehaltung eines einheitlichen Busprotokolls vor.

**Kabelgebundene Übertragung**

Diese Version, gemäß dem US-Standard EIA RS-485, wurde als Grundversion für Anwendungen im Bereich der Fertigungstechnik, Gebäudeleittechnik und Antriebstechnik festgelegt. Es wird ein verdrehtes Kupferkabel mit einem Leiterpaar verwendet. Die Abschirmung kann in Abhängigkeit des beabsichtigten Einsatzgebietes (EMV-Gesichtspunkte beachten) entfallen.

Es stehen zwei Leitungstypen mit unterschiedlichen Höchstleitungslängen zur Verfügung (siehe Tabelle RS485).

## RS485 - Grundlegende Eigenschaften

RS-485 Übertragungstechnik nach PROFIBUS-Norm	
Netzwerk Topologie	Linearer Bus, aktiver Busabschluss an beiden Enden, Stichleitungen sind möglich
Medium	Abgeschirmtes verdrehtes Kabel, Schirmung darf abhängig von den Umgebungsbedingungen (EMV) entfallen
Anzahl der Stationen	32 Stationen in jedem Segment ohne Repeater. Mit Repeater erweiterbar bis 127 Stationen
Max. Bus Länge ohne Repeater	100 m bei 12 MBit/s 200 m bei 1500 KBit/s, bis zu 1,2 km bei 93,75 KBit/s
Max. Bus Länge mit Repeater	Durch Leitungsverstärker (Repeater) kann die max. Buslänge bis zu 10 km vergrößert werden. Die Anzahl der möglichen Repeater ist mindestens 3 und kann je nach Hersteller bis zu 10 betragen.
Übertragungsgeschwindigkeit (in Stufen einstellbar)	9,6 kBit/s; 19,2 kBit/s; 93,75 kBit/s; 187,5 kBit/s; 500 kBit/s; 1500 kBit/s; 12 MBit/s
Steckverbinder	9-Pin D-Sub-Steckverbinder für IP20 M12 Rundsteckverbinder für IP65/67

### Verkabelung für PROFIBUS-DP und PROFIBUS-FMS

Beachten Sie die besonderen Anforderungen an das Datenkabel bei Baud-Raten von mehr als 1,5 Mbaud. Das richtige Kabel ist Grundvoraussetzung für den störungsfreien Betrieb des Bussystems. Bei der Verwendung des normalen 1,5 Mbaud-Kabels kann es durch Reflexionen und zu großer Dämpfung zu erstaunlichen Phänomenen kommen. Zum Beispiel bekommt eine angeschlossene PROFIBUS-Station keine Verbindung, kann diese aber nach Abziehen der benachbarten Station wieder aufnehmen. Oder es kommt zu Übertragungsfehlern, wenn ein bestimmtes Bitmuster übertragen wird. Das kann bedeuten, dass der PROFIBUS ohne Funktion der Anlage störungsfrei arbeitet und nach dem Hochlauf zufällig Busfehler meldet. Eine Reduzierung der Baud-Rate (< 93,75 kbaud) beseitigt das geschilderte Fehlverhalten.

Führt die Verringerung der Baud-Rate nicht zur Beseitigung des Fehlers, liegt in häufig ein Verdrahtungsfehler vor. Die beiden Datenleitungen sind an einem oder mehreren Steckern gedreht oder Abschlusswiderstände sind nicht oder an falschen Stellen aktiviert.

#### ● Empfohlene Kabel

**i** Mit den vorkonfektionierten Kabeln von Beckhoff vereinfacht sich die Installation erheblich! Verdrahtungsfehler werden vermieden und die Inbetriebnahme führt schneller zum Erfolg. Das Beckhoff-Programm umfasst Feldbuskabel, Stromversorgungskabel und Sensorkabel sowie Zubehör wie Abschlusswiderstände und T-Stücke. Ebenso sind jedoch auch feldkonfektionierbare Stecker und Kabel erhältlich.

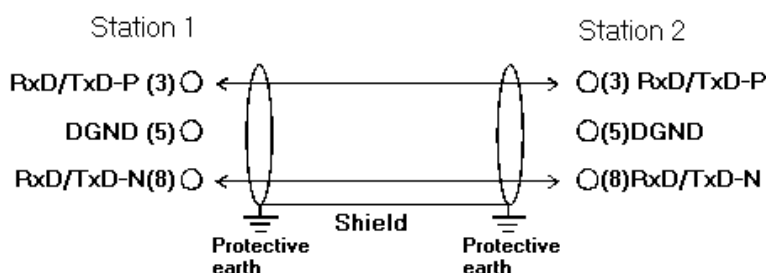


Abb. 23: Belegung des PROFIBUS-Kabels

#### ● Abschlusswiderstände

**i** In Systemen mit mehr als zwei Stationen werden alle Teilnehmer parallel verdrahtet. An den Leitungsenden muss das PROFIBUS-Kabel in jedem Fall mit Widerständen abgeschlossen werden, um Reflexionen und damit Übertragungsprobleme zu vermeiden.

**Längenausdehnung**

Die Busleitung ist in der EN 50170 spezifiziert. Daraus ergibt sich die nachfolgende Längenausdehnung eines Bussegment.

Baud-Rate in kBits/sec	9,6	19,2	93,75	187,5	500	1500	12000
Leitungslänge in m	1200	1200	1200	1000	400	200	100

Stichleitungen bis 1500 kBaud <6,6 m, bei 12 MBaud sollten keine Stichleitungen verwendet werden.

**Bussegment**

Ein Bussegment besteht aus maximal 32 Teilnehmern. An einem PROFIBUS-Netzwerk sind 126 Teilnehmer erlaubt. Um diese Anzahl zu erreichen sind Repeater erforderlich, die das Signal auffrischen. Dabei wird jeder Repeater wie ein Teilnehmer angesehen.

IP-Link ist das Sub-Bussystem der Feldbus Boxen dessen Topologie eine Ringstruktur ist. In den Koppler Modulen (IP230x-Bxxx oder IP230x-Cxxx) befindet sich ein IP-Link Master, an den bis zu 120 Erweiterungsmodule (IExxxx) angeschlossen werden dürfen. Der Abstand zwischen zwei Modulen darf dabei 5 m nicht überschreiten. Achten Sie bei der Planung und Installation der Module, dass aufgrund der Ringstruktur das letzte Modul wieder am IP-Link Master angeschlossen werden muss.

**Einbaurichtlinie**

Beachten Sie bei der Montage der Module und beim Verlegen der Leitung die technischen Richtlinien der PROFIBUS-Nutzerorganisation e.V. zu PROFIBUS-DP/FMS (siehe <https://www.profibus.de>).

**Überprüfung der PROFIBUS-Verkabelung**

Ein PROFIBUS-Kabel (bzw. ein Kabel-Segment bei Verwendung von Repeatern) kann mit ein paar einfachen Widerstandsmessungen überprüft werden. Dazu sollte das Kabel von allen Stationen abgezogen werden:

1. Widerstand zwischen A und B am Anfang der Leitung: ca. 110 Ohm
2. Widerstand zwischen A und B am Ende der Leitung: ca. 110 Ohm
3. Widerstand zwischen A am Anfang und A am Ende der Leitung: ca. 0 Ohm
4. Widerstand zwischen B am Anfang und B am Ende der Leitung: ca. 0 Ohm
5. Widerstand zwischen Schirm am Anfang und Schirm am Ende der Leitung: ca. 0 Ohm

Falls diese Messungen erfolgreich sind, ist das Kabel in Ordnung. Wenn trotzdem noch Bus-Störungen auftreten, liegt es meistens an EMV-Störungen. Beachten Sie die Installationshinweise der PROFIBUS-Nutzer-Organisation (<https://www.profibus.de>).

## 4 Parametrierung und Inbetriebnahme

### 4.1 Anlaufverhalten des Busklemmen-Controllers

Nach dem Einschalten prüft der Busklemmen-Controller seinen Zustand, konfiguriert den K-Bus, erstellt anhand der gesteckten Busklemmen eine Aufbaukarte und startet seine lokale SPS.

Beim Hochlaufen des Busklemmen-Controllers leuchten und blinken die I/O-LEDs. Im fehlerfreien Zustand sollte nach ca. 2 bis 3 Sekunden keine I/O-LED mehr Blinken. Im Fehlerfall hängt es von der Fehlerart ab, welche LED blinkt (siehe Kapitel *Diagnose-LEDs*).

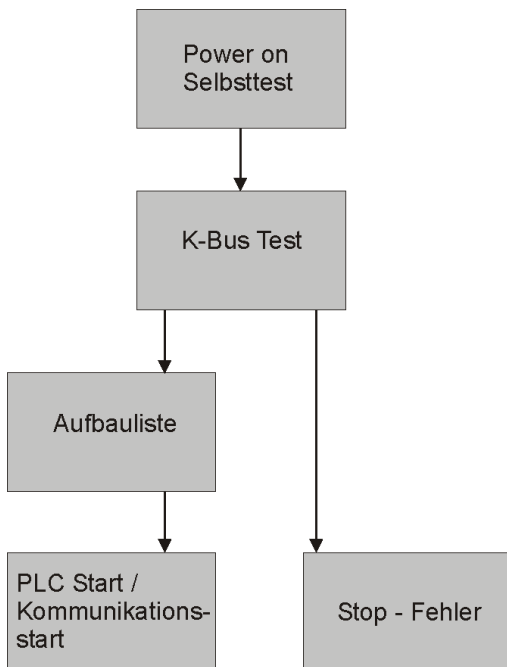


Abb. 24: Anlaufverhalten des Busklemmen-Controllers

### 4.2 PROFIBUS-Adresse

Die PROFIBUS-Adresse wird auf dem BX3100 über zwei Drehwahlschalter einzustellen. Die Default-Einstellung beträgt 11. Es sind alle Adressen erlaubt, allerdings darf jede Adresse im Netzwerk nur einmal vorkommen. Die Adresse wird im ausgeschalteten Zustand des Buskopplers geändert. Dafür verändern Sie mit Hilfe eines Schraubendrehers die Schalter auf die gewünschte Position. Beachten Sie dabei, dass die Schalter richtig einrasten. Der untere Schalter ist der Zehner-Multiplikator und der obere Schalter der Einer-Multiplikator. Die Adressänderung wird aktiv, sobald der BX3100 eingeschaltet wird.

#### Beispiel

Sie wollen Adresse 34 einstellen.

- Unterer Drehwahlschalter S311: 3
- Oberer Drehwahlschalter S310: 4



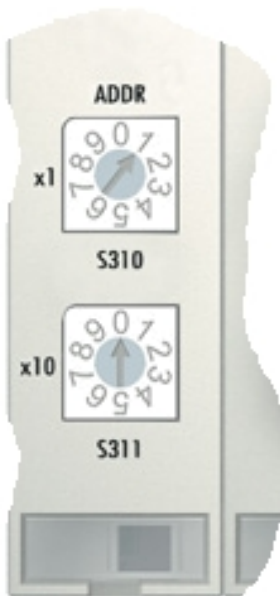


Abb. 25: Drehwahlschalter für die PROFIBUS-Adresse

## 4.3 Konfiguration

### 4.3.1 Überblick

#### Konfigurationsarten

Bei den Busklemmen-Controllern der Serien BCxx50, BCxx20 und BXxx00 gibt es zwei verschiedene Arten der Konfiguration, die DEFAULT CONFIG und die TWINCAT CONFIG.

#### DEFAULT-CONFIG

Die Busklemmen mappen sich in der Reihenfolge wie diese gesteckt sind, erst die komplexen Busklemmen, dann die digitalen Busklemmen.

Das Mapping der komplexen Busklemmen ist:

- Word-Alignment
- komplexe Darstellung

#### **⚠ VORSICHT**

#### **Prozessabbild ist abhängig von angesteckten Klemmen!**

Das Prozessabbild verändert sich, sobald eine Klemme dazu gesteckt wird oder entfernt wird!

Die Daten der Feldbus-Slave Schnittstelle werden SPS-Variablen genannt. Die SPS-Variablen befinden sich ab der Adresse %QB1000 und %IB1000.

Weiterhin kann die DEFAULT-CONFIG ohne SPS-Programm für das Schreiben und Testen der Angeschlossenen Busklemmen verwendet werden. Dafür muss im System Manager der Busklemmen-Controller gescannt werden und die Betriebsart FreeRun aktiviert werden (um diese Funktion zu nutzen darf kein SPS-Programm auf dem Busklemmen-Controller sein).

#### TWINCAT-CONFIG

In der TWINCAT-CONFIG können die Busklemmen und die SPS-Variablen frei verknüpft sein (TwinCAT System Manager-File notwendig). Die Konfiguration wird mit Hilfe des System Managers per ADS zum Koppler übertragen.

Für die TwinCAT Config (TC-File) benötigen Sie folgendes:

- Über der Feldbus (PROFIBUS, CANopen, Ethernet)  
PROFIBUS: (BC3150, BX3100)
  - PC mit FC310x ab Version 2.0 und TwinCAT 2.9 Build 1000
  - BX3100 mit CIF60 oder CP5412
  - TwinCAT 2.9 Build 946  
(**ACHTUNG:** bei den PROFIBUS Karten von Hilscher ist nur eine ADS-Kommunikation erlaubt, d.h. entweder System Manager oder PLC Control)
  - CANopen: (BC5150, BX5100)
  - PC mit FC510x ab Version 1.76 TwinCAT Build 1030  
DeviceNet: (BC5250, BX5200)
  - Auf Anfrage  
Ethernet: (BC9050, BC9020, BC9120, BX9000)
  - PC mit TwinCAT 2.10 Build 1322
- Über das serielle ADS TwinCAT 2.9 Build 1010
  - BX3100 Version 1.00
  - BX5100 Version 1.00
  - BX5200 Version 1.10
  - BX8000 Version 1.00
  - BC3150, BC5150, BC5250, BC9050, BC9020, BC9120 ab Firmware B0
  - Für BC8150 ab TwinCAT 2.10 Build 1243

BCxx50 und BXxx00 können über den System Manager des TwinCAT Programms parametriert werden.

- Variables I/O Mapping
- Typ gerechte PROFIBUS Daten (nur BC3150 und BX3100)
- RTC (Real Time Clock) (nur BX-Serie)
- SSB Bus (Smart System Bus) (nur BX-Serie)
- PLC Einstellungen
- K-Bus Einstellungen

Die Konfiguration kann per Feldbus ADS-Protokoll oder seriellem ADS-Protokoll zum BCxx50 oder BXxx00 übertragen werden.

Mit der TwinCAT Konfiguration kann man Variablen, I/Os und Daten verknüpfen. Folgendes ist möglich:

- PLC - K-BUS
- PLC - Feldbus (z. B. PROFIBUS Slave Schnittstelle zur PLC)
- K-Bus - Feldbus (nur bei den BX-Controllern)
- Unterstützung der TwinSAFE-Klemmen (nur BX-Controller ab Firmware 1.17)

Zusätzlich können mit der TwinCAT Konfiguration spezielle Verhalten parametriert werden, zum Beispiel ob bei einem Feldbus Fehler die Daten erhalten bleiben oder auf "0" gesetzt werden sollen.

Die Echtzeituhr kann man über einen Karteireiter im System Manager einstellen.

### Arbeitsschritte

1. Feldbus Adresse einstellen
2. System Manager öffnen und TC-File anlegen
3. Feldbus Daten in dem TC-File konfigurieren
4. TC-File speichern
5. Neuer System Manager öffnen und PC-File anlegen und gespeichertes TC-File einlesen
6. Verknüpfen zu einer SPS-Task herstellen
7. Speichern der Konfiguration
8. Starten des TwinCAT Systems

9. System Manager des TC-File öffnen, fertig konfigurieren und zum BCxx50, BCxx20 oder BXxx00 übertragen
10. Programm zum BCxx50, BCxx20 oder BXxx00 übertragen
11. Bootprojekt erzeugen

### 4.3.2 Anlegen einer TwinCAT-Konfiguration

Um einen Busklemmen-Controller der Serien BCxx50, BCxx20, BXxx00 oder BC9191 zu konfigurieren muss im System-Manager ein BX-File angelegt werden. Zur Vereinfachung sind die Grundgeräte schon als File vorbereitet. Dazu öffnen Sie mit *New from Template* den entsprechenden Busklemmen-Controller.

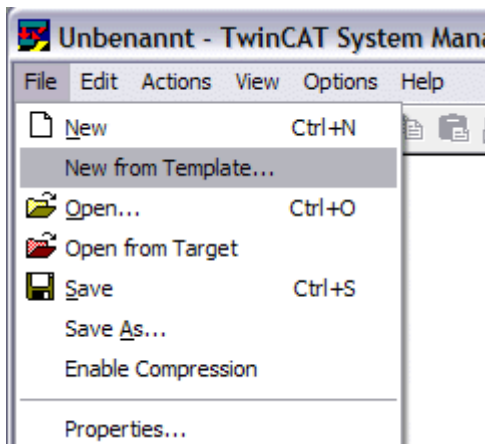


Abb. 26: Anlegen einer TwinCAT-Konfiguration

Wählen Sie den entsprechenden Busklemmen-Controller aus.

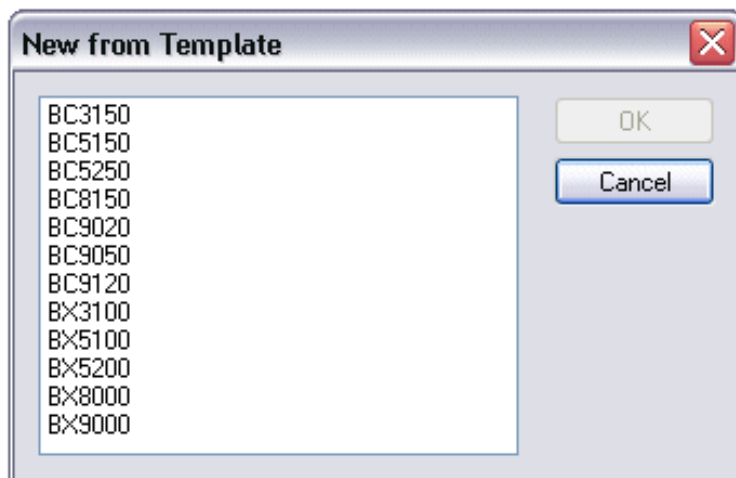


Abb. 27: Auswahl des Busklemmen-Controllers

Nun sind alle Komponenten des Busklemmen-Controllers vorhanden:

- Feldbusschnittstelle
- K-Bus Interface [[▶ 52](#)]
- PLC Programm [[▶ 54](#)]
- SSB [[▶ 57](#)] (nur Busklemmen-Controller der BX-Serie)

Die Konfiguration der Geräte entnehmen Sie den entsprechenden Kapiteln.

### 4.3.3 Download einer TwinCAT-Konfiguration

Die TwinCAT-Konfiguration wird per ADS-Protokoll zum Busklemmen-Controller geladen.

#### Seriellles ADS-Protokoll

(alle Busklemmen-Controller der Serien BXxx00 und BCxx50)

Tragen Sie die serielle ADS-Verbindung ein, wie unter dem Kapitel [Seriellles ADS \[► 41\]](#) beschrieben ist.

#### ADS-Protokoll über den Feldbus

(nur BC3150, BC5150, BC9x20, BC9050, BX3100, BX5100, BX9000, BC9191)

Vorraussetzung hierfür ist, dass TwinCAT als Master arbeitet und sich im Datenaustausch befindet, d.h. die physikalische, wie auch die Feldbus-Konfiguration muss abgeschlossen sein und der Datenaustausch vom Master (z. B. Feldbus-Master-Karte) zum Busklemmen-Controller stattfinden.

#### Auswahl des Zielsystems

Wählen Sie den Busklemmen-Controller aus, auf den Sie die Konfiguration laden wollen. Mit der Funktionstaste F8 öffnet sich der Dialog, mit dem Sie Ihr File auf das entsprechende Gerät herunterladen können.

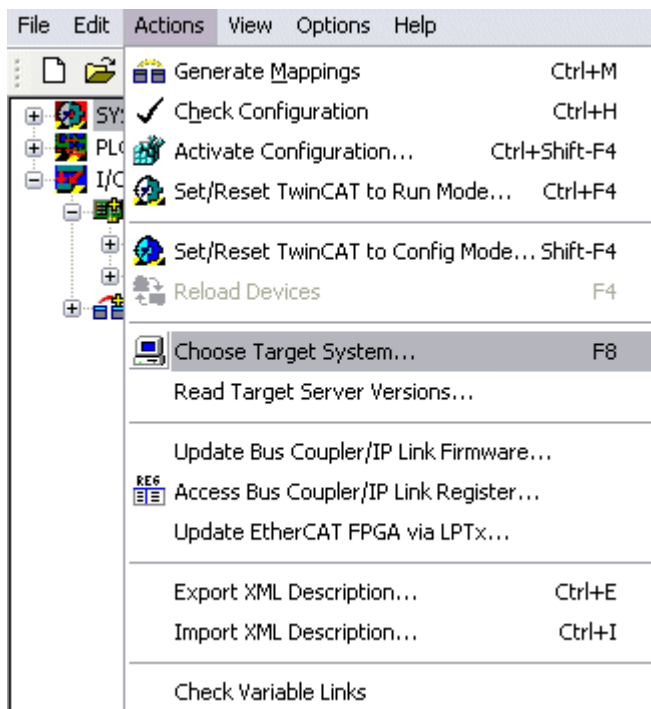


Abb. 28: Download einer TwinCAT-Konfiguration

Wählen Sie den entsprechenden Busklemmen-Controller aus.

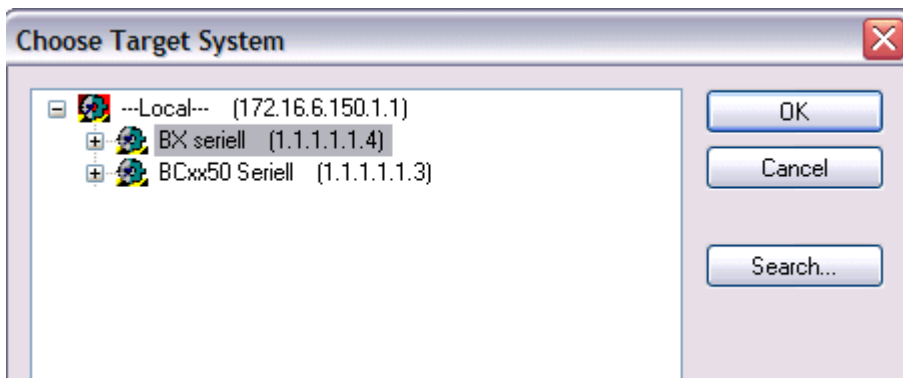


Abb. 29: Auswahl des Busklemmen-Controllers

Den Zustand des Busklemmen-Controllers wird unten rechts im System-Manager angezeigt.

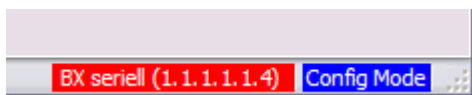


Abb. 30: Zustand des Busklemmen-Controllers

Im *Config Mode / FreeRun* kann man jetzt die Konfiguration zum Busklemmen-Controller herunterspielen. Wenn der Busklemmen-Controller im *Stop Mode* ist, ist die ADS-Kommunikation noch nicht aktiviert. Ein Download der Konfiguration ist dann nicht möglich.

Zum Aktivieren der TwinCAT-Konfiguration wählen Sie **Ctrl+Shift+F4** oder *Activate Configuration*.

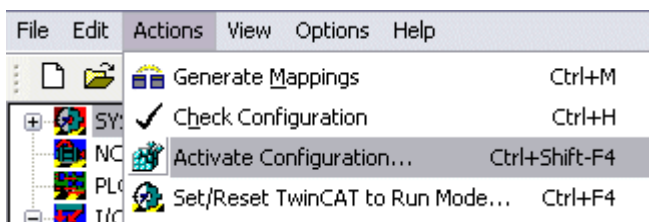


Abb. 31: Aktivieren der TwinCAT-Konfiguration

Die aktuelle Konfiguration wird in dem Busklemmen-Controller geladen. Im Display erscheint *Store Config* und die BUS und I/O LED blinken. Nachdem die Konfiguration erfolgreich im Busklemmen-Controller ist, sollte ein BXxx00 im Display *TwinCAT Config* anzeigen. Jetzt kann man das entsprechende Programm zum Busklemmen-Controller übertragen (Programm-Download über den Feldbus).

### 4.3.4 Upload einer TwinCAT-Konfiguration

Die TwinCAT-Konfiguration wird per ADS-Protokoll zum Busklemmen-Controller geladen.

#### Seriellles ADS-Protokoll

(alle Busklemmen-Controller der Serien BCxx50, BCxx20 und BXxx00)

Tragen Sie die serielle ADS-Verbindung ein, wie unter dem Kapitel [Seriellles ADS](#) [ 41] beschrieben ist.

#### ADS-Protokoll über den Feldbus

(nur BC3150, BC5150, BC9x20, BC9050, BX3100, BX5100, BX9000, BC9191)

Voraussetzung hierfür ist, dass TwinCAT als Master arbeitet und sich im Datenaustausch befindet, d.h. die physikalische, wie auch die Feldbus Konfiguration muss abgeschlossen sein und der Datenaustausch vom Master (z. B. Feldbus-Karte) zum Busklemmen-Controller stattfindet.

**Auswahl des Zielsystems**

Wählen Sie den Busklemmen-Controller aus, auf den Sie die Konfiguration laden wollen. Mit der Funktionstaste [F8] öffnet sich der Dialog, mit dem Sie Ihr File auf das entsprechende Gerät herunterladen können.

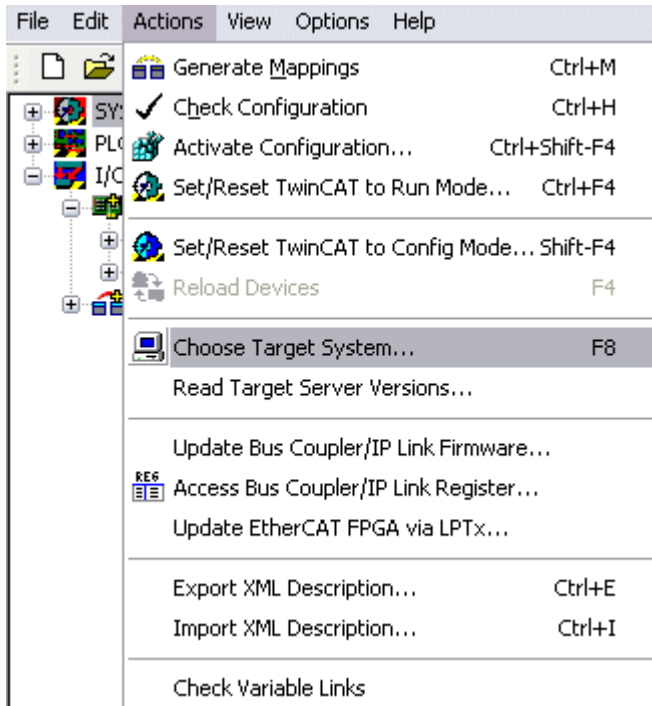


Abb. 32: Auswahl des Zielsystems

Wählen Sie den entsprechenden Busklemmen-Controller aus.

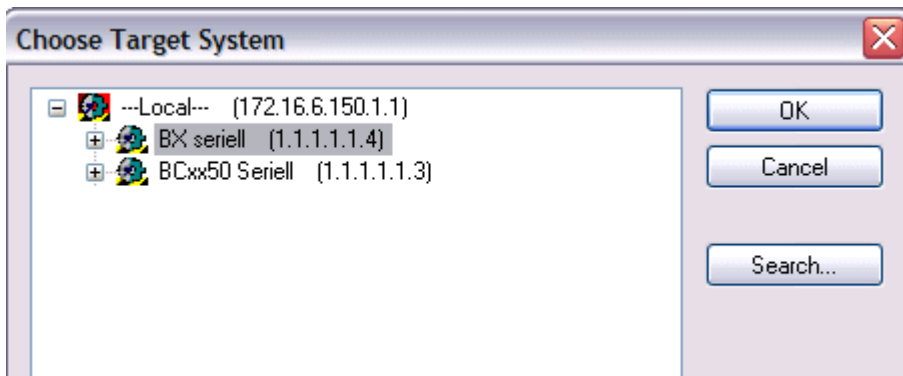


Abb. 33: Auswahl des Busklemmen-Controllers

Der Zustand des Busklemmen-Controllers wird unten rechts im System-Manager angezeigt.

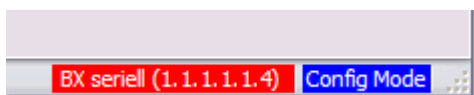


Abb. 34: Zustand des Busklemmen-Controllers

Klicken Sie den roten Ordner an. Die TwinCAT-Konfiguration wird jetzt hochgeladen.

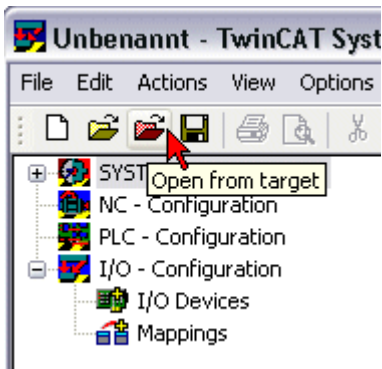


Abb. 35: Hochladen der TwinCAT-Konfiguration

### 4.3.5 Ressourcen im Busklemmen-Controller

Die im Busklemmen-Controller belegten Speicher-Ressourcen zeigt der System Manager auf dem Karteireiter *Resources* des Busklemmen-Controller an.

#### Mapping Code

Der Mapping Code wird für die Berechnung der TwinCAT Konfiguration benötigt (siehe Abb. *Speicher für das Code Mapping*). Die Prozentzahlen werden hier zusammen addiert, in dem Beispiel aus Abb. *Speicher für das Code Mapping* sind 8% des Speichers für die Mapping-Berechnung belegt.

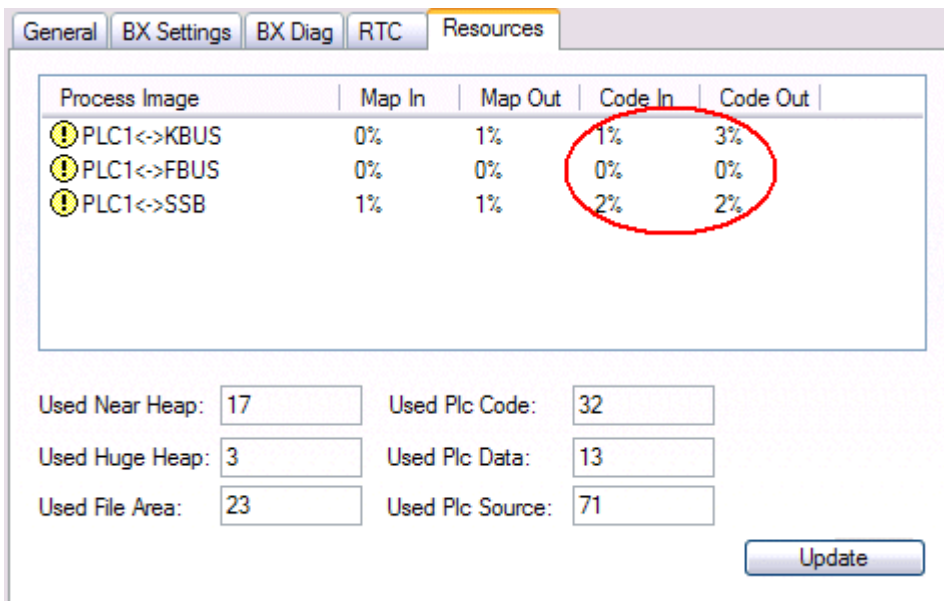


Abb. 36: Speicher für das Code Mapping

#### Daten Speicher Mapping

Daten Speicher für die Mappings. Die Werte sind einzeln zu betrachten, das bedeutet das jeder der Werte bis zu 100% betragen kann.

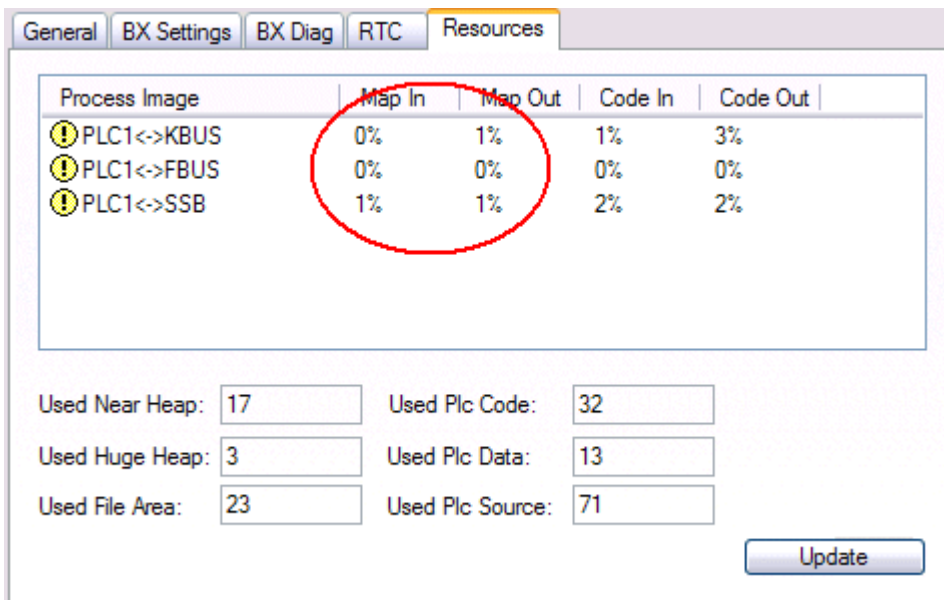


Abb. 37: Daten Speicher Mapping

**Verbraucher Code und Daten Speicher**

Abb. Code und Daten Speicher (1) "Used Plc Code" verbrauchter PLC Code, Angabe in %.

Abb. Code und Daten Speicher (2) "Used Plc Data" verbrauchter PLC Daten, Angabe Speicher in %.

Abb. Code und Daten Speicher (3) "Used Plc Source" verbrauchter Source Code, Angabe in %.

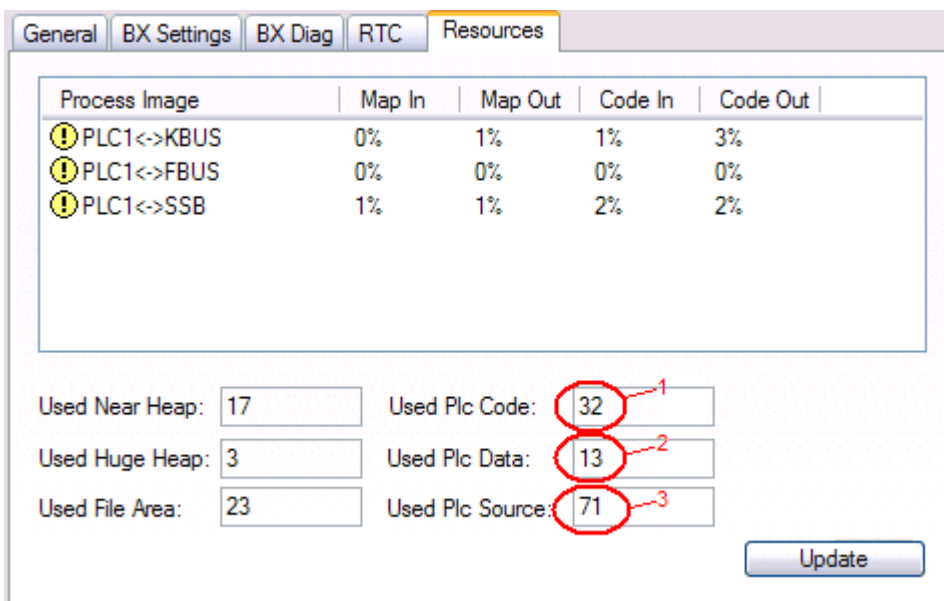


Abb. 38: Code und Daten Speicher

**Sonstiger Speicher**

Abb. Sonstiger Speicher (1) "Used Near Heap" wird für die COM Schnittstelle und SSB benötigt. Angabe in %.

Abb. Sonstiger Speicher (2) "Used Huge Heap" wird für die ADS Kommunikation benötigt. Angabe in %. Dieser Wert sollte kleiner 30 % betragen.

Abb. Sonstiger Speicher (3) "Used File Area" wird für die TwinCAT Konfiguration, dem TSM-File und dem 16 kByte Flash Zugriff benötigt. Angabe in %.



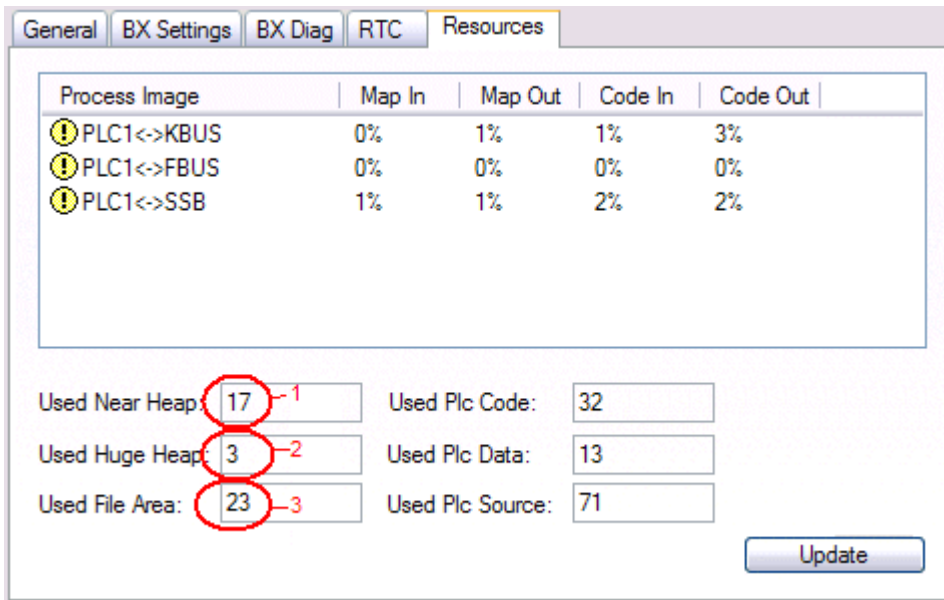


Abb. 39: Sonstiger Speicher

### 4.3.6 ADS-Verbindung über die serielle Schnittstelle

(ab Firmware-Version 1.xx oder 0.99x - Busklemmen-Controller der BX-Serie und bei allen BCxx50)

Ab TwinCAT 2.9 Build 1020 (TwinCAT Level PLC, NC oder NCI)

**i** **Verwenden Sie nur eine serielle Verbindung**

Um die problemlose Funktion der ADS-Verbindung über die serielle Schnittstelle zu garantieren, ist nur dabei eine serielle Verbindung zum BX-Controller erlaubt. Nach einer erfolgreichen Konfiguration mit dem System Manager schließen Sie diesen, bevor Sie die Programmierung starten.

**i** **AMS-Net-ID im Auslieferungszustand (Default)**

**Für BX9000**

Die Default AMS-Net-ID lautet 172.16.21.20.1.1. Wird die IP-Adresse des BX9000 verändert, so verändert sich auch die AMS-Net-ID des BX9000. Die aktuelle AMS-Net-ID können Sie mit Hilfe des Menüs sich auf dem Display anzeigen lassen.  
Beispiel: Sie ändern die IP-Adresse auf 10.2.3.7 so ändert sich die AMS-Net-ID auf 10.2.3.7.1.1.

**Für BC9050, BC9020, BC9120**

Die Default AMS-Net-ID lautet 172.16.xxx.[DIP-Switch].1.1. Wird die IP-Adresse des BC9xxx verändert, so verändert sich auch die AMS-Net-ID des BC9xxx.  
Beispiel: Sie ändern die IP-Adresse auf 10.2.3.7 so ändert sich die AMS-Net-ID auf 10.2.3.7.1.1.  
BC9050: DEFAULT 172.16.21.[DIP-Switch].1.1  
BC9020: DEFAULT 172.16.22.[DIP-Switch].1.1  
BC9120: DEFAULT 172.16.23.[DIP-Switch].1.1

**ADS-Verbindung initialisieren**

Tragen Sie den Busklemmen-Controller unter TwinCAT in die Remote-Verbindung ein. Klicken Sie dazu auf das TwinCAT-Icon und öffnen Sie das Eigenschafts-Menü. Unter dem Karteireiter >AMS Remote< können Sie dann folgende Einträge vornehmen.

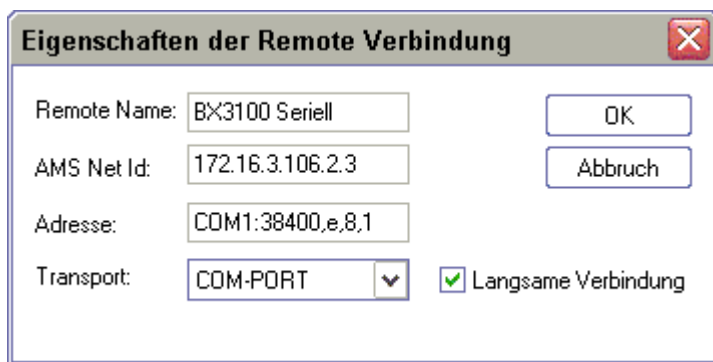


Abb. 40: Eigenschaften der Remote-Verbindung

Remote Name: Beliebig  
 AMS-Net-ID: 1.1.1.1.1.1 (Default)  
 Adresse: COM Port: Baudrate, Parity, Datenbits, Stop Bits  
 Transport: Hier ist "COM-PORT" anzuwählen

Wird der Busklemmen-Controller eingeschaltet, so hat dieser per Default immer die AMS-Net-ID "1.1.1.1.1.1" (außer alle Ethernet Controller).  
 Sie können diese AMS-Net-ID beliebig ändern. Die neue AMS-Net-ID kann dann allerdings nicht noch einmal geändert werden.

Wenn Sie die AMS-Net-ID erneut ändern möchten, müssen Sie den Busklemmen-Controller zuerst neu starten, so dass die AMS Net Id wieder auf die Default AMS-Net-ID "1.1.1.1.1.1" zurückgesetzt wird. Nun können sie die AMS-Net-ID wiederum einmal ändern.

**● Erst beim zweiten Aufruf können Sie Strings eingeben**

**i** Beim ersten Aufruf des Dialogs (siehe oben) können Sie unter Adresse keine Strings eingeben. Tragen Sie Namen, AMS-Net-ID und Transport-Typ ein und schließen Sie den Dialog. Beim zweiten Aufruf können Sie dann Ihre COM-Schnittstelle eintragen.

Die Kommunikation startet sobald TwinCAT im Config- (TwinCAT-Icon ist blau) oder RUN-Modus (TwinCAT-Icon ist grün) ist. Die COM-Schnittstelle wird erst mit einem TwinCAT Stopp (TwinCAT-Icon ist rot) geschlossen und ist dann wieder für andere Programme verfügbar. Sollte die COM-Schnittstelle bei einem TwinCAT-Neustart von einem anderen Programm (z. B. von der Konfigurations-Software KS2000) verwendet werden, so wird kein Fehlerhinweis ausgegeben.

**● AMS-Net-ID nach ADS-Verbindung über den Feldbus**

**i** Sollte Sie vor oder während der Nutzung der seriellen ADS-Verbindung den Busklemmen-Controller mit einer ADS-Verbindung über den Feldbus angesprochen haben, ist die AMS-Net-ID vom System Manager automatisch verändert worden. Eine erneute serielle ADS-Verbindung ist dann nur möglich, wenn die AMS-Net-ID angepasst wird.

**BX-Serie: auslesen der AMS-Net-ID**

Die aktuelle AMS-Net-ID kann aus dem Menü über das Display des Busklemmen-Controller der BX-Serie ausgelesen werden.

<b>AMS</b>	AMS-Net-ID
1.1.1.1.1.1	

## 4.3.7 PROFIBUS

### 4.3.7.1 PROFIBUS-Einstellungen

Im Karteireiter DP-Slaves werden die PROFIBUS-relevanten Einstellungen vorgenommen.

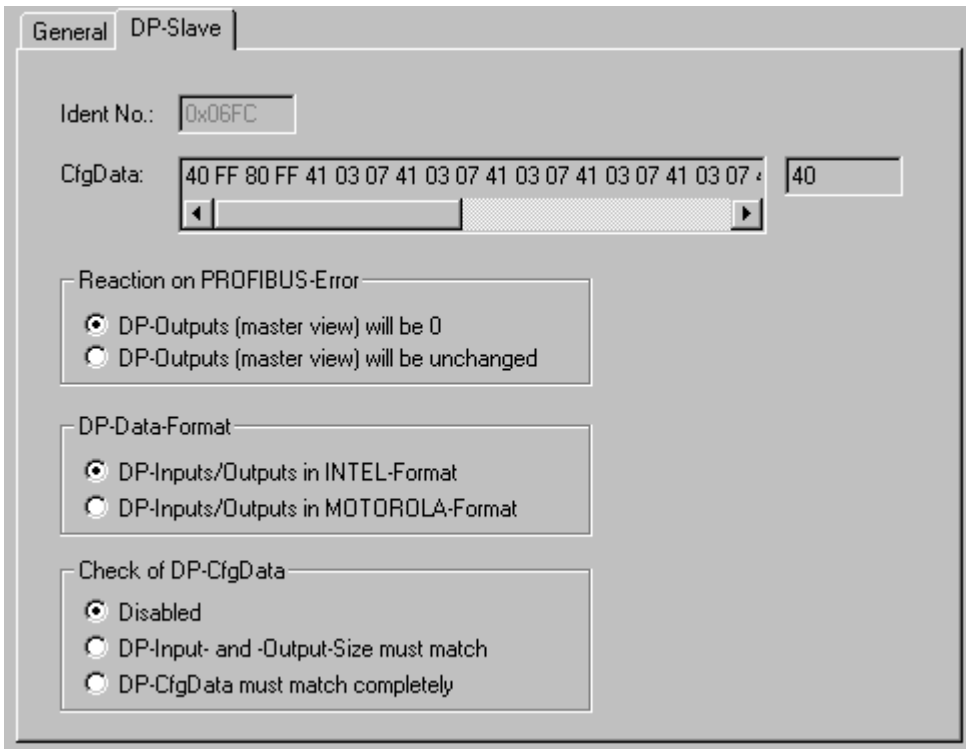


Abb. 41: Karteireiter DP-Slaves

#### Reaktion auf PROFIBUS-Fehler

Bei PROFIBUS-Fehler (z. B. abgezogenes PROFIBUS Kabel), werden die Ausgangsdaten des PROFIBUS-Masters unverändert beibehalten oder auf 0 gesetzt (je nach Parametrierung). Aus Sicht des Busklemmen-Controllers sind dies Eingangsdaten, die der PROFIBUS-Master zum Busklemmen-Controller überträgt.

#### DP-Daten-Format

Die PROFIBUS-Daten können im Intel- oder im Motorola-Format übertragen werden.

#### Kontrollieren der DP Konfigurationsdaten

Die Konfiguration-Daten können

- gar nicht geprüft werden
- auf richtige Länge geprüft werden
- auf Länge und Inhalt geprüft werden

## 4.3.7.2 Master Konfiguration

### 4.3.7.2.1 Geräte Stamm Datei (GSD)

Alle Feldgeräte mit PROFIBUS Slave-Schnittstelle werden anhand der GSD-Datei beschreiben. Der BX3100 ist ein PROFIBUS-DP Slave. Für den Datenaustausch muss die GSD-Datei in dem Master-Konfiguration-Tool eingefügt werden. Die maximale Länge der Ein- und Ausgangsdaten beträgt pro Richtung 244 Byte.

 Download GSD-File (deutsch) (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3238674315.zip>)

 Download GSD-File (englisch) (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3238672139.zip>)

Diese Dateien stehen Ihnen auch im Download-Bereich des BX3100 auf der Beckhoff-Homepage zur Verfügung: [www.beckhoff.de/BX3100](http://www.beckhoff.de/BX3100)

Die GSD-Datei kann folgende Datentypen beschreiben:

Variable	Länge
Integer 8	1 Byte
UnInteger 8	1 Byte
Integer 16	2 Byte
UnInteger 16	2 Byte
Integer 32	4 Byte
UnInteger 32	4 Byte
Float 32	4 Byte

Nicht aufgeführte Datentypen können auch einfach über die Länge beschrieben werden, wie zum Beispiel BOOL-Variablen. In der GSD-Datei gibt es dafür Konfigurations-Daten die bei einem Byte beginnen und bei 64 Worten enden.

### 4.3.7.2.2 TwinCAT-PC-File anlegen

#### DEFAULT CONFIG

In der DEFAULT-CONFIG befinden sich die PROFIBUS-Daten ab der Adresse %IB1000 Input und %QB1000. Die Länge der Daten ist abhängig von der Anzahl der konfigurierten PROFIBUS-Daten. Maximal können 244 Byte Ein- und 244 Byte Ausgangsdaten parametrisiert sein. In dieser Konfigurationsart sind keine weiteren Einstellungen notwendig.

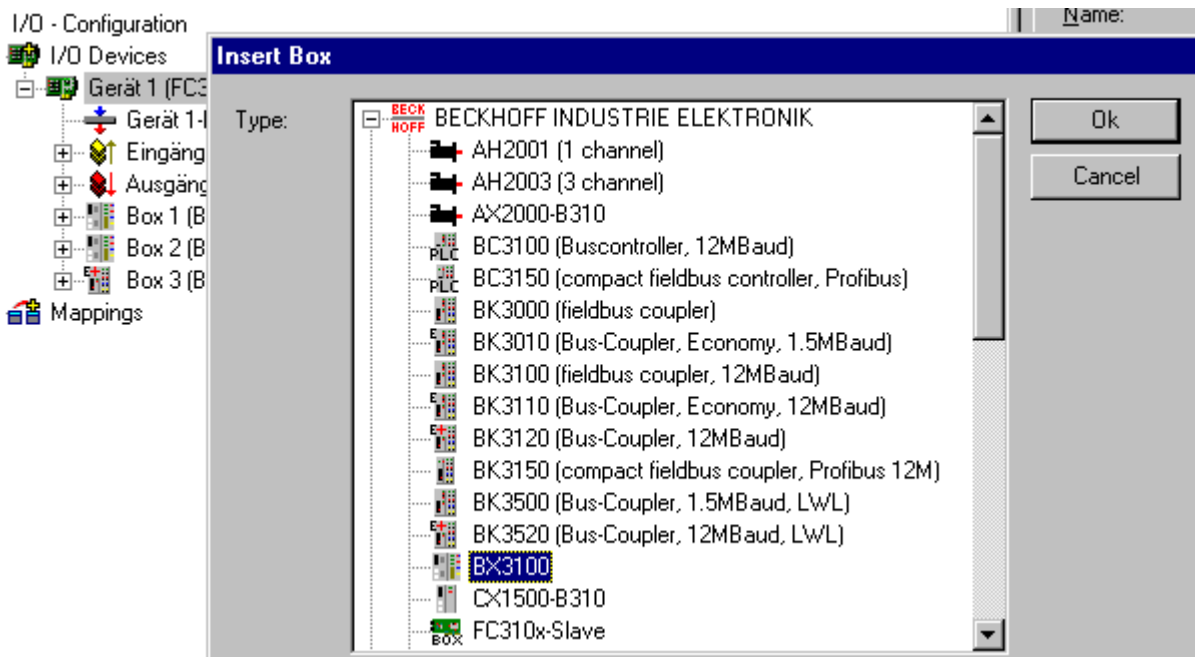


Abb. 42: BX3100 einfügen

**Einfügen des BX3100 in den System Manager an einen PROFIBUS-Master**

Wählen Sie im System Manager ihre PROFIBUS-Master PC-Karte aus und klicken Sie mit der rechten Maustaste auf Ihre PROFIBUS Master Karte. Ihnen werden dann die PROFIBUS-Geräte angeboten, unter anderem ein BX3100. Wählen Sie diesen aus und bestätigen Sie mit OK.

**Öffnen eines bestehendes BX-File**

Wenn Sie mit der DEFAULT-CONFIG arbeiten wollen brechen Sie an dieser Stelle ab (siehe Anlegen von PROFIBUS Daten in der DEFAULT-CONFIG). Sonst wählen Sie Ihr konfiguriertes BX-File aus. Sollten Sie noch keins haben, müssen Sie dies erst anlegen ([Anlegen eines BX-Files](#) [▶ 35]). Nachdem Sie Ihr BX-File angewählt haben, werden aus diesem File automatisch alle notwendigen PROFIBUS-Daten in Ihr Projekt kopiert. Sie müssen die Verknüpfung zu Ihrer Task durchführen und die PROFIBUS-Adresse einstellen. Dann kann die Konfiguration gesichert werden und das TwinCAT-System gestartet werden.

**Anlegen von PROFIBUS-Daten in der DEFAULT-CONFIG**

- Es können jetzt die PROFIBUS-Daten angelegt werden. Es stehen verschiedene Variablen zu Verfügung:
  - Integer 8 Bit
  - Integer 16 Bit
  - Integer 32 Bit
  - Unsigned 8 Bit
  - Unsigned 16 Bit
  - Unsigned 32 Bit
  - FLOAT 32 Bit

Diese müssen mit der im BX-projektierten Variablenarten übereinstimmen. Des Weiteren stehen Daten Größen zur Verfügung von 1 Byte bis 64 Worte.

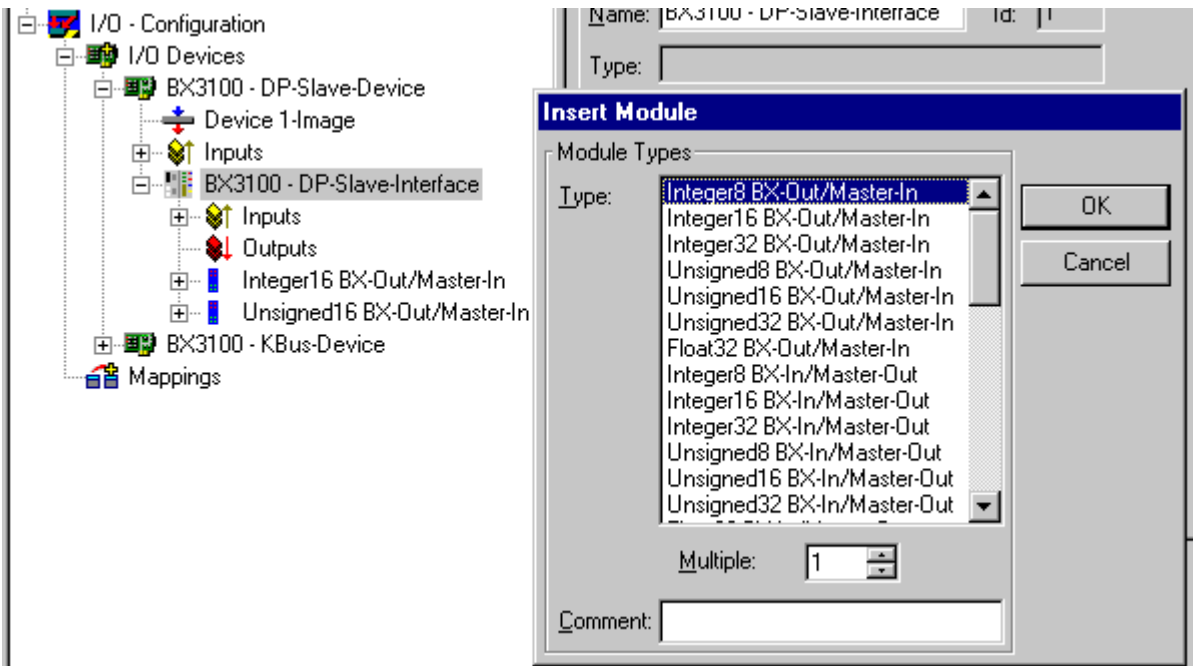


Abb. 43: Anlegen von PROFIBUS-Daten

Um die Konfiguration zu erstellen muss die ADS-Kommunikation angewählt sein. Sollen die Konfigurations-Daten über den Feldbus übertragen werden muss auch die Datenkommunikation laufen. D.h. der BX muss im Datenaustausch mit der übergeordneten Steuerung sein.

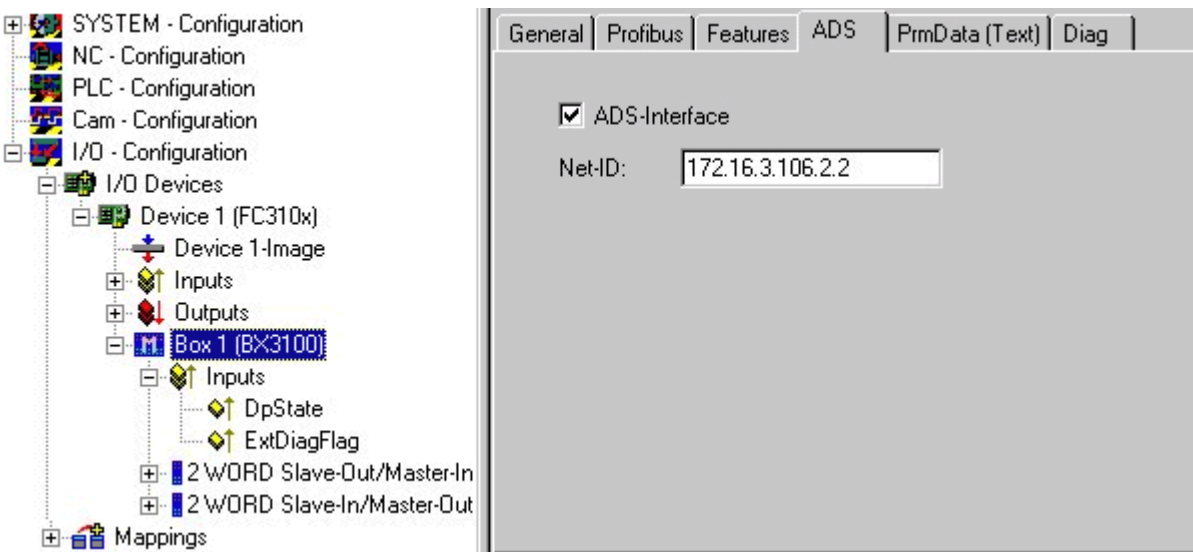


Abb. 44: ADS-Interface

### 4.3.7.2.3 Variables Mapping - PROFIBUS-DP

Anlegen der PROFIBUS Daten.

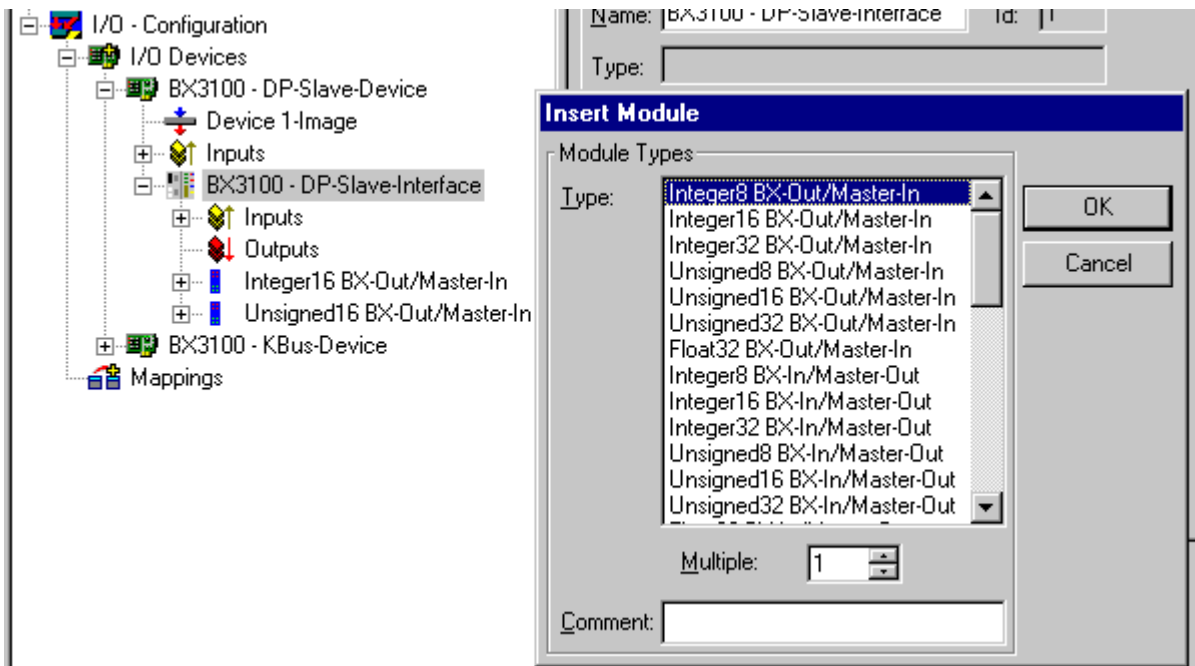


Abb. 45: Anlegen von PROFIBUS-Daten

Für die Verknüpfung mit dem PLC-Projekt müssen Sie das SPS-Programm einlesen. Fügen Sie dazu unter SPS-Konfiguration Ihr SPS-Programm ein. Nun können Sie die SPS-Variablen mit den Feldbusvariablen verbinden.

#### **i** Zulässige Anzahl von PROFIBUS-Daten

- Beachten Sie, dass die maximal zulässige Anzahl an PROFIBUS-Daten nicht überschritten wird.
- BX3100: max. 244 Byte Eingänge und 244 Byte Ausgänge
- BC3150: max. 128 Byte Eingänge und 128 Byte Ausgänge

#### 4.3.7.2.4 BX3100 an Siemens S7-Steuerungen

### Konfiguration: Siemens S7-Steuerung

#### Einfügen der Bilder

Um in der Siemens-Software den Geräten ein Bild zuzuweisen müssen die folgenden Grafiken in in das Verzeichnis *Step7\S7Data\ncbmp* kopiert werden.



<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3238676491.zip>



<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3238678667.zip>



<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3238680843.zip>

#### Einfügen der GSD-Dateien

- Gehen sie in den Hardwarekatalog ihrer Step7 unter *Extras\Neue GSD installieren*.
- Wählen sie das Verzeichnis aus, in dem sich die BECKHOFF GSD befinden und importieren Sie die Dateien.
- Sie finden diese dann im Hardware Katalog unter *Profibus-DP\Weiter Feldgeräte\I/O*.



## Konfiguration: Siemens S7-Steuerung mit BX3100

### Parameterdaten des BX3100

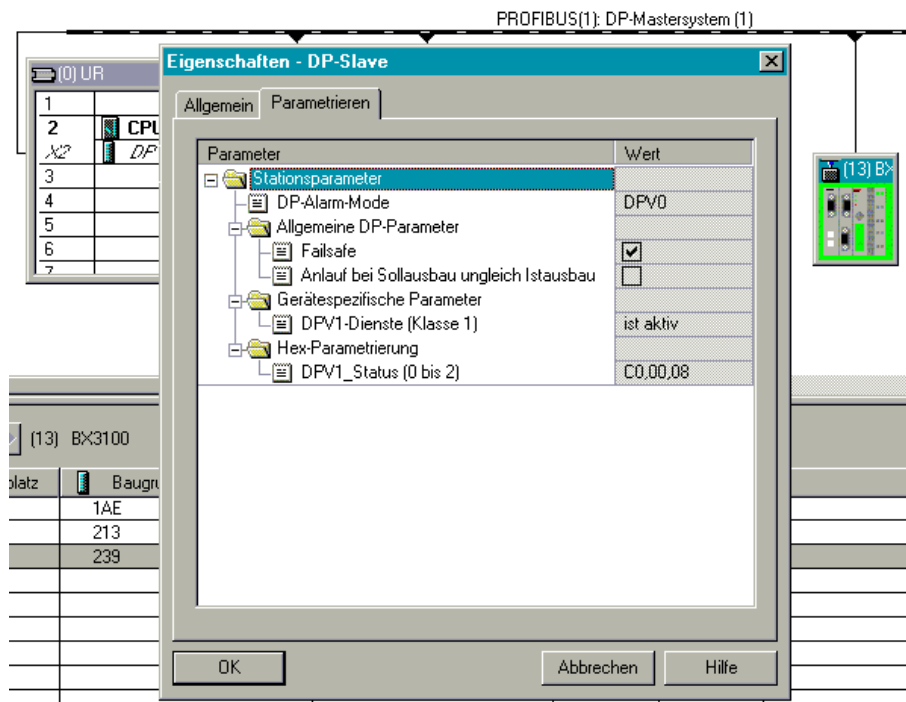


Abb. 46: Einstellungen

### Konfiguration BX3100 Modul

#### Beispiel 1

1 x BX3100

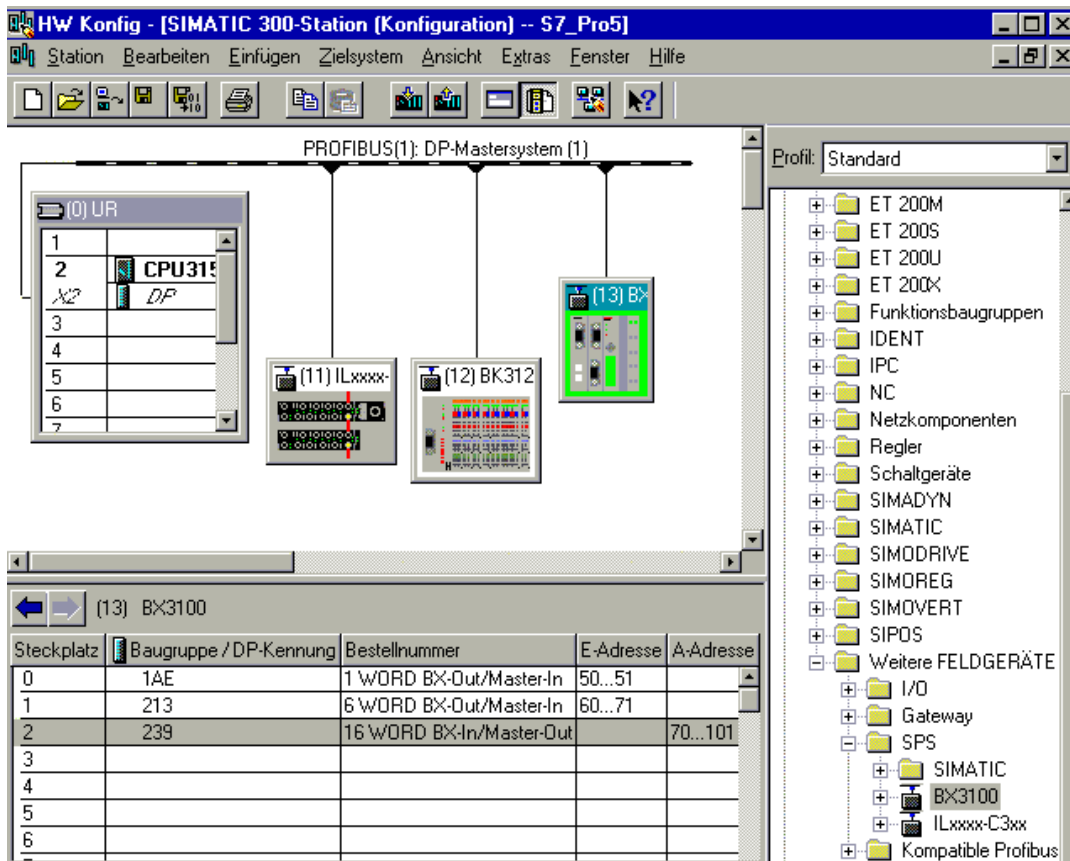


Abb. 47: Beispiel für das Eintragen einzelner Bytes.

**Byte für Byte**

**i** Jedes einzelne Byte erfordert 1 Byte ConfigData.

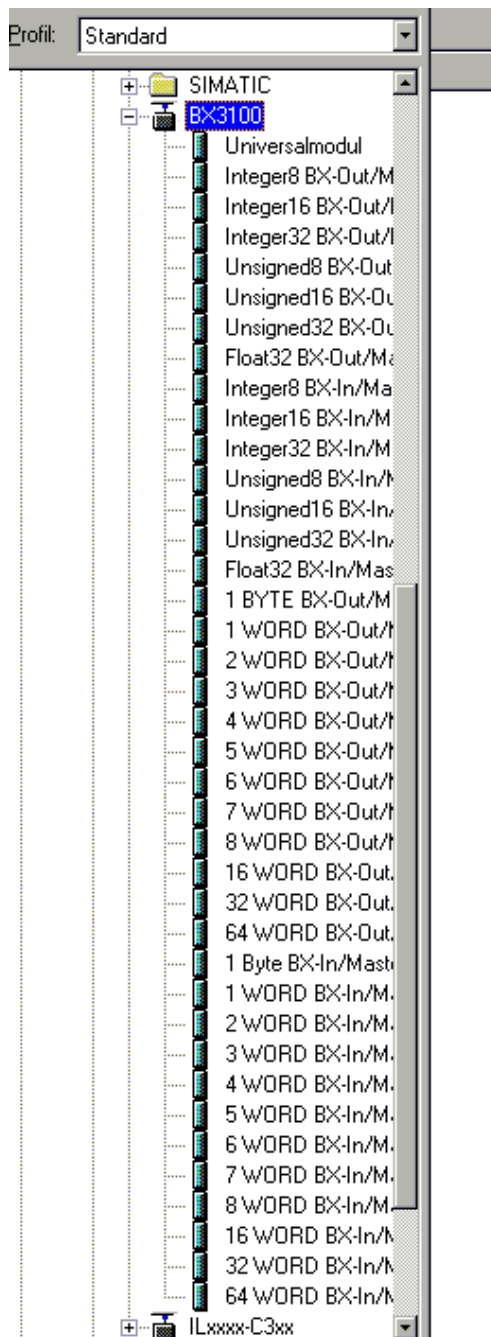


Abb. 48: Beispiel für das Eintragen zusammenhängender Bytes

### 4.3.8 K-Bus

**i Busklemme und Endklemme erforderlich**

Zum Betrieb eines Busklemmen-Controllers der BC- oder BX-Serie müssen an dessen K-Bus mindestens eine Busklemme mit Prozessabbild und die Endklemme gesteckt sein!

**Karteireiter BX Settings**

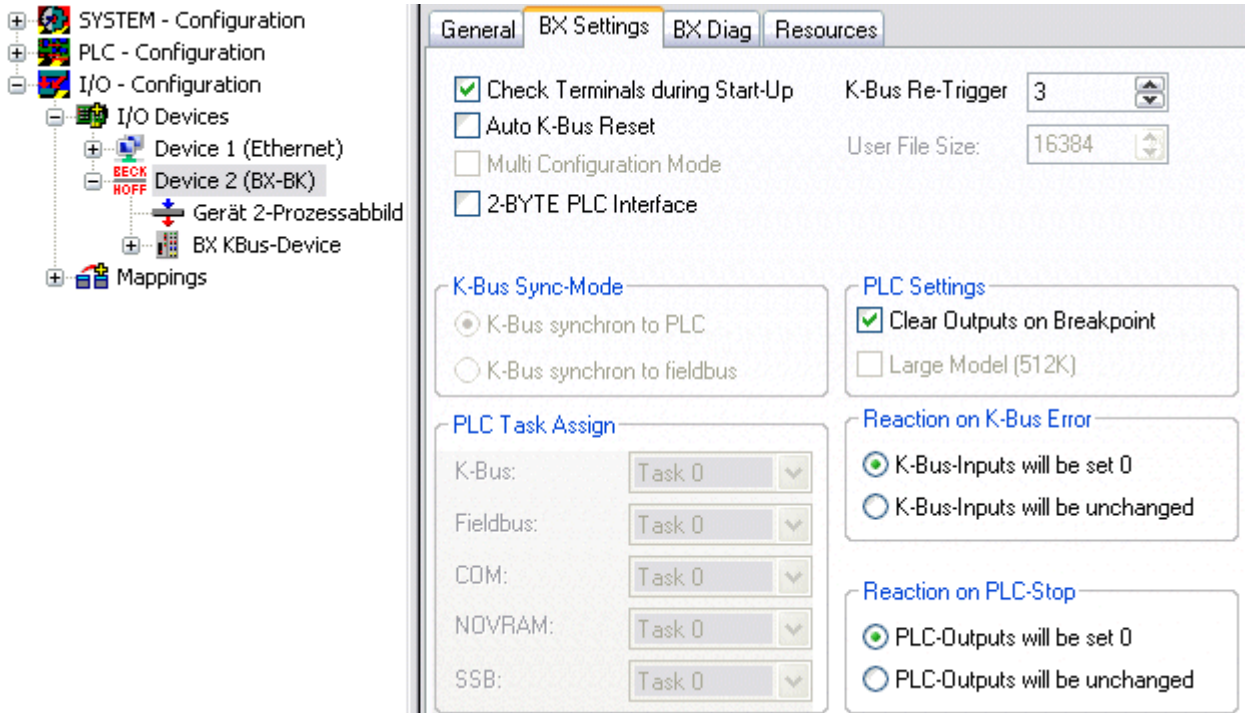


Abb. 49: Karteireiter BX Settings

**Check Terminals during Start up**

Beim Erzeugen eines Bootprojektes wird die aktuelle Busklemmenkonfiguration gespeichert. Beim Erneuten starten des Busklemmen-Controllers werden die angeschlossenen Busklemmen überprüft. Ist diese Option angewählt, geht der Busklemmen-Controller nicht in den Datenaustausch. Das PLC Projekt wird nicht gestartet.

**Auto K-Bus Reset**

Nach Behebung eines K-Bus-Fehlers geht der Busklemmen-Controller automatisch wieder in den Datenaustausch.

**⚠ VORSICHT**

**Nach Behebung eines K-Bus Fehlers werden die Ausgänge sofort wieder aktiv!**

Achten Sie darauf, dass die Ausgänge dann sofort wieder aktiv werden und analoge Ausgänge ihren programmierten Wert erhalten, wenn dies nicht in Ihrem PLC-Projekt beachtet wurde.

**Clear Outputs on Breakpoint**

Wenn Breakpoints im PLC Control gesetzt werden, wird der K-Bus nicht mehr bearbeitet, das heißt die Ausgänge werden in den sichern Zustand, sprich null, gesetzt.

**K-Bus Sync Mode**

Das Schreiben und lesen der Busklemmen kann synchron zur Task1 oder dem Feldbus stattfinden.

**K-Bus Re-Trigger**

Sollte das PLC Projekt oder der SSB länger den Prozessor belasten, kann der K-Bus eine Zeit nicht mehr bearbeitet werden. Das führt dazu, dass der Watchdog der Busklemmen zuschlägt und Ausgänge abfallen. Der Busklemmen-Controller ist so eingestellt, das nach 85 ms der K-Bus Watchdog nachgetriggert wird und das 3 mal. Danach würde der K-Bus Watchdog zuschlagen.

K-Bus Re-Trigger 0: 100 ms

K-Bus Re-Trigger 1: 2 x 85 ms = 170 ms

K-Bus Re-Trigger 2: 3 x 85 ms = 255 ms

K-Bus Re-Trigger 3: 4 x 85 ms = 340 ms

**Reaktion auf K-Bus Fehler**

Bei K-Bus Fehler werden die K-Bus Eingänge auf "0" geschrieben oder behalten Ihren letzten Stand.

**Reaktion auf PLC-Stop**

Wird das PLC Projekt gestoppt, kann man Einstellen, wie sich die PLC Feldbus Ausgangsdaten verhalten sollen. Im Master sind diese Daten dann Eingangsdaten. Die Daten können bei PLC Stop auf "0" geschrieben oder unverändert gelassen werden.

**Karteireiter BX Diag**

Anzeige der Zykluszeit für Task 1, K-Bus, Bearbeitung Feldbus und die Auslastung des SSB.

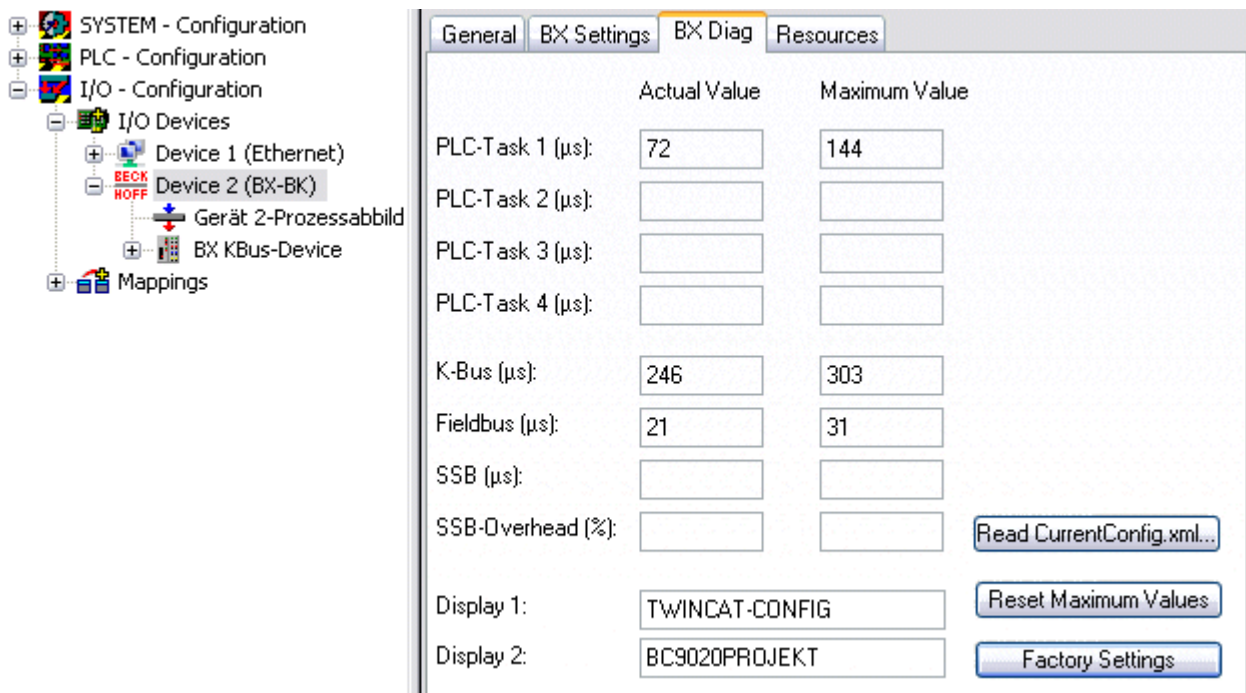
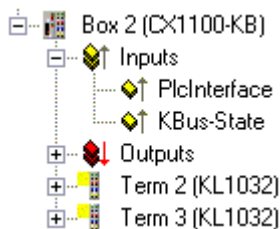


Abb. 50: Karteireiter BX Diag

*Factory Settings* - Der Busklemmen-Controller wird in seinen Auslieferungszustand gesetzt. Mit Restart System oder aus- und einschalten sind diese Einstellungen wieder gültig (Display DEFAULT-CONFIG).  
*Reset Maximum Values* - löscht die Maximalen Werte

**K-Bus Variablen**



**PLC-Interface:** Wird nicht unterstützt (nur enthalten um CX oder BX Projekte zu verschieben)

**K-Bus-State:** Siehe Diagnose

**4.3.9 PLC**

**4.3.9.1 Einfügen eines PLC-Projektes**

Für ein variables Mapping muss im System Manager die Konfiguration festgelegt sein. Hier vereinbart man die Verknüpfung zwischen PLC und der Hardware. Die Variablen können Bit, Byte, Wort oder Datenstrukturen verarbeiten und verknüpfen. Eine automatische Adressierung durch den System Manager ist möglich, sollte aber auf ihren Offset überprüft werden.

**i Word Alignment - Byte Orientierung**

Achten Sie bei Datenstrukturen darauf, dass der Busklemmen-Controller die Daten in Word Alignment speichert und der System Manager Byte orientiert arbeitet (siehe [Datenstrukturen \[► 105\]](#))

Im PLC-Control muss ein gültiges Projekt übersetzt und gespeichert sein. Diese Daten werden als \*.tpy Datei abgespeichert. Um ein PLC-Projekt einzufügen klicken Sie mit der rechten Maustaste auf *PLC-Configuration*. Wählen Sie Ihr aktuelles PLC Projekt aus.

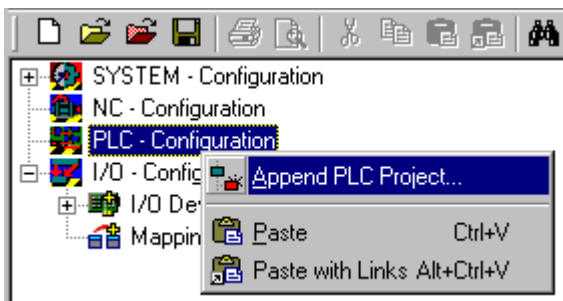


Abb. 51: Auswahl des PLC-Projekts

Verbinden Sie die PLC-Variable mit der Hardware (z. B. digitale Busklemme).

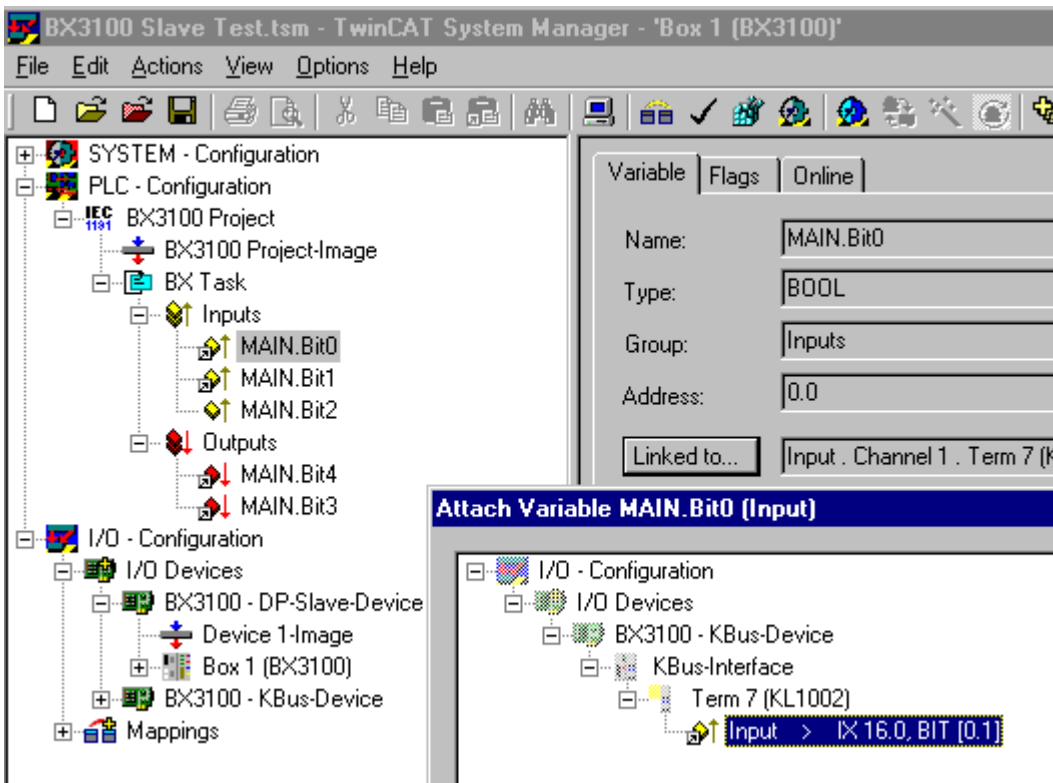


Abb. 52: Verbinden vom PLC-Variable und Hardware

Nachdem alle Verknüpfungen erstellt sind aktivieren Sie die Konfiguration *Actions/Activate Configuration* (Ctrl+Shift+F4) und Starten Sie TwinCAT *Set/Reset TwinCAT to Run Mode*. Achten Sie darauf, dass Sie das richtige Zielsystem angewählt haben (unten rechts im Fenster des System-Managers).

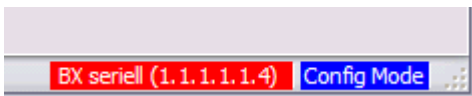


Abb. 53: Anzeige des Ziel-Systems

### 4.3.9.2 Messen der PLC-Zykluszeit

Die Task-Zeit wird im PLC Control eingestellt. Die Default Einstellung beträgt 20 ms.

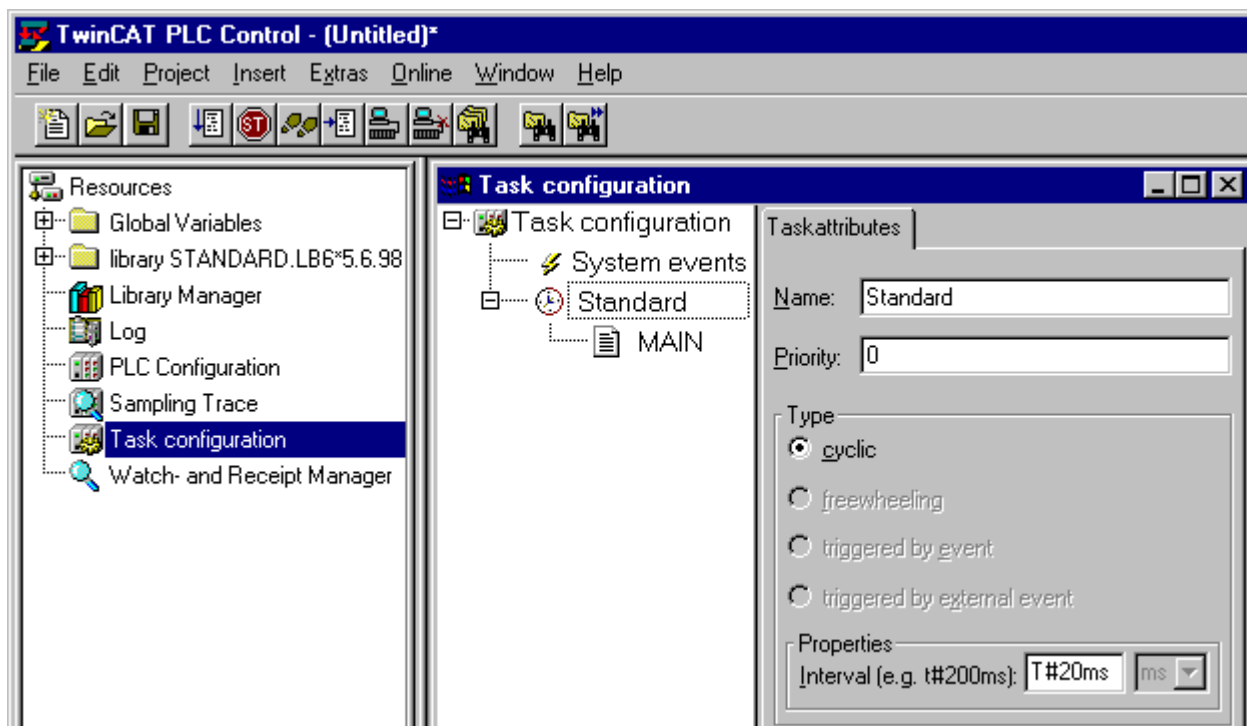


Abb. 54: Einstellen der Task-Zeit

Das PLC-Programm wird in der Default-Einstellung alle 20 ms erneut aufgerufen, solange die allgemeine Zykluszeit kleiner als 20 ms ist. Um die Auslastung ihres Systems zu ermitteln kann im System Manager die Zykluszeit der PLC gemessen werden. Um einen Problemlosen Betrieb zu garantieren, muss die eingestellte Task-Zeit 20 bis 30 % höher sein als die gemessene gesamt Zykluszeit. Eine genauere Aufschlüsselung der Zykluszeit finden Sie unter der Beschreibung [K-Bus-Reiter](#) [► 52]. Die gesamte Zykluszeit wird mit der TcBase Bibliothek angezeigt (siehe TcBase.lbx oder TcBaseBCxx50.lbx).



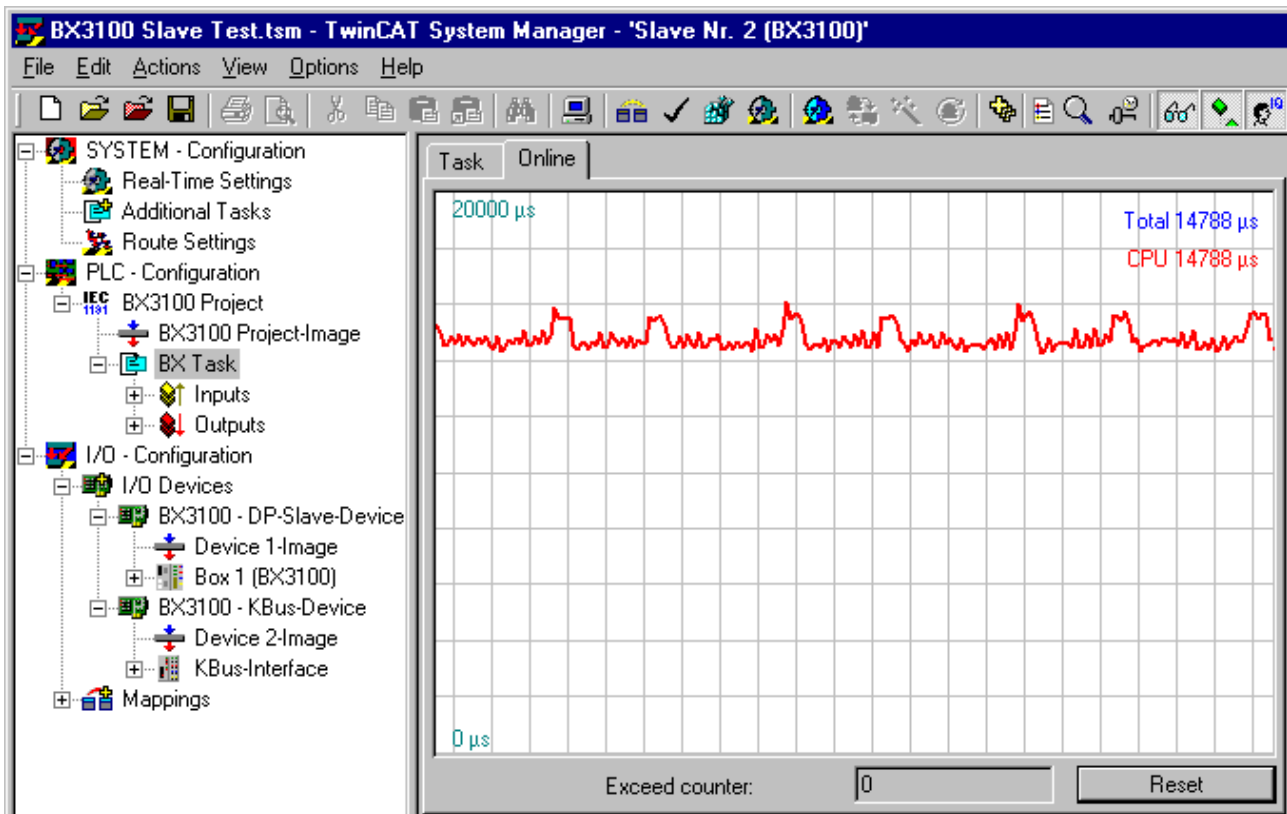


Abb. 55: Anzeige der PLC-Zykluszeit

### 4.3.10 SSB

#### 4.3.10.1 SSB - Überblick

Der SSB (Smart System Bus) ist ein auf CANopen basierendes Sub-Bussystem. Es handelt sich dabei um einen CANopen-Master mit eingeschränkter Funktionalität. CANopen-Slaves können an dieser Schnittstelle angeschlossen sein, um dezentrale I/Os einzulesen oder zu beschreiben. Über ein Startup-Fenster können dem Slave parametrierungs-SDOs (Service Data Objects) geschickt werden.

#### Konfiguration

Der SSB wird mit Hilfe des TwinCAT System Manager projiziert (siehe TwinCAT config). Die Konfiguration wird dann per ADS auf den BX-Controller gespielt.

#### Technische Daten

SSB	Daten
Max. Anzahl an Slaves	8
Max. Anzahl an PDOs	32 RxPODs / 32 TxPDOs
Baudrate	10 kBaud bis 1 MBaud
Erlaubte Slave Adressen	1 bis 64

#### Sync-Telegramm

Das Sync-Telegramm wird in Abhängigkeit zur PLC-Task-Zeit übertragen. Ist eine Taskzeit von 20 ms eingestellt wird auch das Sync-Telegramm alle 20 ms asynchron zur PLC Laufzeit versendet. Das Sync-Telegramm wird erst erzeugt, wenn ein Teilnehmer dieses Benötigt und dieser Konfiguriert wird. Das Sync-Telegramm wird ab der Firmware 1.12 unterstützt.

**Guarding**

Das Guarding wird unterstützt und wird nach einer konfigurierbaren Zeit versendet.

**4.3.10.2 CANopen Verkabelung**

Hinweise für die Überprüfung der CAN-Verdrahtung finden sich im Kapitel Fehlersuche / Trouble Shooting [► 71].

**4.3.10.2.1 CAN-Topologie**

CAN ist ein 2-Draht-Bussystem, an dem alle Teilnehmer parallel (d.h. mit kurzen Stichleitungen) angeschlossen werden. Der Bus muss an jedem Ende mit einem Abschlusswiderstand von 120 (bzw. 121) Ohm abgeschlossen werden, um Reflexionen zu vermeiden. Dies ist auch bei sehr kurzen Leitungslängen erforderlich!

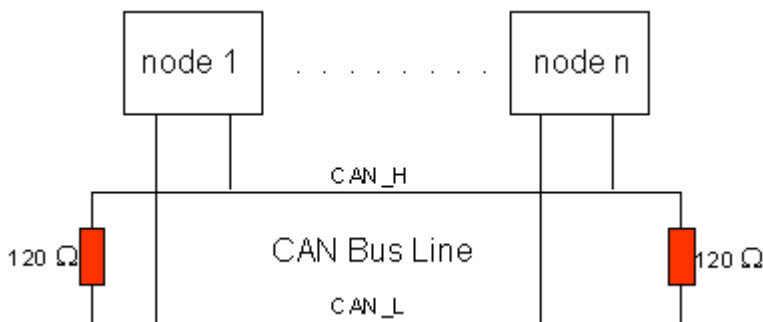


Abb. 56: Abschluss des Busses mit Abschlusswiderstand 120 Ohm

Da die CAN-Signale als Differenzpegel auf dem Bus dargestellt werden, ist die CAN-Leitung vergleichsweise unempfindlich gegen eingeprägte Störungen (EMI). Es sind jeweils beide Leitungen betroffen, somit verändert die Störung den Differenzpegel kaum.

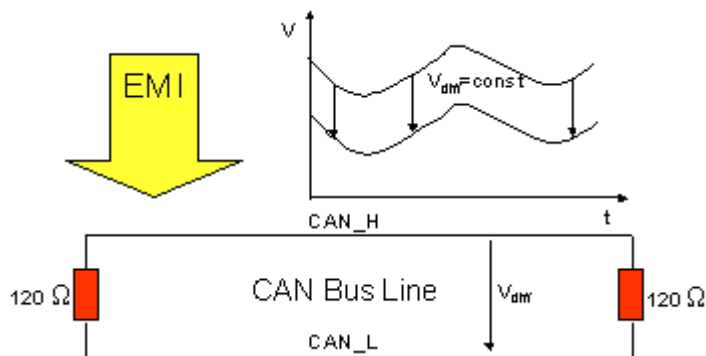


Abb. 57: Unempfindlichkeit gegen eingeprägte Störungen

**4.3.10.2.2 Buslänge**

Die maximale Buslänge wird bei CAN vorwiegend durch die Signallaufzeit beschränkt. Das Multi-Master-Buszugriffsverfahren (Arbitrierung) erfordert, dass die Signale quasi gleichzeitig (vor der Abtastung innerhalb einer Bitzeit) an allen Knoten anliegen. Da die Signallaufzeit in den CAN-Anschaltungen (Transceiver, Optokoppler, CAN-Controller) nahezu konstant sind, muss die Leitungslänge an die Bit-Rate angepasst werden.

Bit-Rate	Buslänge
1 MBit/s	< 20 m*
500 kBit/s	< 100 m
250 kBit/s	< 250 m
125 kBit/s	< 500 m
50 kBit/s	< 1000 m
20 kBit/s	< 2500 m
10 kBit/s	< 5000 m

\*) Häufig findet man in der Literatur für CAN die Angabe 40 m bei 1 MBit/s. Dies gilt jedoch nicht für Netze mit optoentkoppelten CAN-Controllern. Die worst case Berechnung mit Optokopplern ergibt bei 1 MBit/s eine maximale Buslänge von 5 m - erfahrungsgemäß sind jedoch 20 m problemlos erreichbar.

Bei Buslängen über 1000 m kann der Einsatz von Repeatern notwendig werden.

### 4.3.10.2.3 Stichleitungen

Stichleitungen ("drop lines") sind nach Möglichkeit zu vermeiden, da sie grundsätzlich zu Signalreflexionen führen. Die durch Stichleitungen hervorgerufenen Reflexionen sind jedoch in der Regel unkritisch, wenn sie vor dem Abtastzeitpunkt vollständig abgeklungen sind. Bei den in den Buskopplern gewählten Bit-Timing-Einstellungen kann dies angenommen werden, wenn folgende Stichleitungslängen nicht überschritten werden:

Bit-Rate	Länge Stichleitung	gesamte Länge aller Stichleitungen
1 MBit/s	< 1 m	< 5 m
500 kBit/s	< 5 m	< 25 m
250 kBit/s	< 10 m	< 50 m
125 kBit/s	< 20 m	< 100 m
50 kBit/s	< 50 m	< 250 m

Stichleitungen dürfen nicht mit Abschlusswiderständen versehen werden.

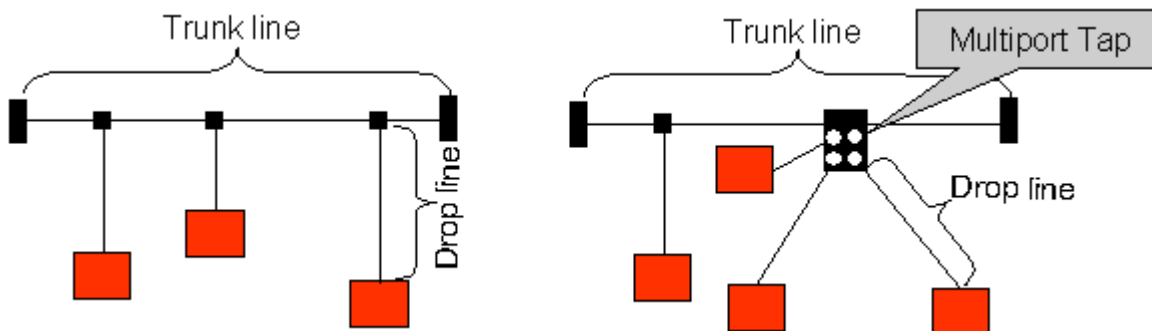


Abb. 58: Beispieltopologie Stichleitungen

### 4.3.10.2.4 Sternverteiler (Multiport Tap)

Beim Einsatz von passiven Verteilern ("Multiport Taps"), z. B. der Beckhoff Verteilerbox ZS5052-4500 sind kürzere Stichleitungslängen einzuhalten. Die folgende Tabelle gibt die maximalen Stichleitungslängen und die maximale Länge der Trunk Line (ohne Stichleitungen) an:

Bit-Rate	Länge Stichleitung bei Multi- port Topologie	Länge Trunk Line (ohne Stichleitungen)
1 MBit/s	< 0,3 m	< 25 m
500 kBit/s	< 1,2 m	< 66 m
250 kBit/s	< 2,4 m	< 120 m
125 kBit/s	< 4,8 m	< 310 m

#### 4.3.10.2.5 CAN-Kabel

Für die CAN-Verdrahtung wird die Verwendung von paarig verdrehten, geschirmten Kabeln (2x2) mit einem Wellenwiderstand von 108...132 Ohm empfohlen. Wenn das Bezugspotential der CAN-Transceiver (CAN-Ground) nicht verbunden werden soll, so kann auf das zweite Adernpaar verzichtet werden (nur bei kleinen Netzausdehnungen mit gemeinsamer Speisung aller Teilnehmer empfehlenswert).

##### ZB5100 CAN-Kabel

Beckhoff hat ein hochwertiges CAN-Kabel mit folgenden Eigenschaften im Programm:

- 2 x 2 x 0,25 mm<sup>2</sup> (AWG 24) paarig verseilt, Kabelfarben: rot/schwarz + weiß/schwarz
- doppelt geschirmt
- Schirmgeflecht mit Beilaufitze (kann direkt auf Pin3 der 5-pol Anschlussklemme aufgelegt werden)
- flexibel (Mindestbiegeradius 35 mm bei einmaligem Biegen, 70 mm bei mehrmaligem Biegen)
- Wellenwiderstand (60 kHz): 120 Ohm
- Leiterwiderstand < 80 Ohm/km
- Mantel: PVC grau, Außendurchmesser 7,3 +/- 0,4 mm
- Gewicht: 64 kg/km.
- Bedruckt mit "Beckhoff ZB5100 CAN-BUS 2x2x0.25" und Metermarkierung (Längenangaben, alle 20cm)

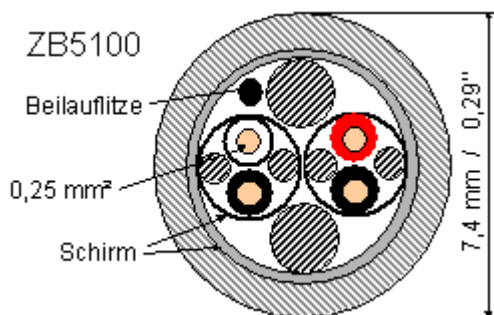


Abb. 59: Aufbau CAN-Kabel ZB5100

##### ZB5200 CAN/DeviceNet-Kabel

Das Kabelmaterial ZB5200 entspricht der DeviceNet Spezifikation und eignet sich ebenso für CANopen Systeme. Aus diesem Kabelmaterial sind auch die vorkonfektionierten Busleitungen ZK1052-xxxx-xxxx für die Feldbus Box Baugruppen gefertigt. Es hat folgende Spezifikation:

- 2 x 2 x 0,34 mm<sup>2</sup> (AWG 22) paarig verseilt
- doppelt geschirmt, Schirmgeflecht mit Beilaufitze
- Wellenwiderstand (1 MHz): 126 Ohm
- Leiterwiderstand 54 Ohm/km
- Mantel: PVC grau, Außendurchmesser 7,3 mm
- Bedruckt mit "InterlinkBT DeviceNet Type 572" sowie UL und CSA Ratings
- Litzenfarben entsprechen DeviceNet Spezifikation
- UL anerkanntes AWM Type 2476 Rating

- CSA AWM I/II A/B 80°C 300V FT1
- Entspricht DeviceNet "Thin Cable" Spezifikation

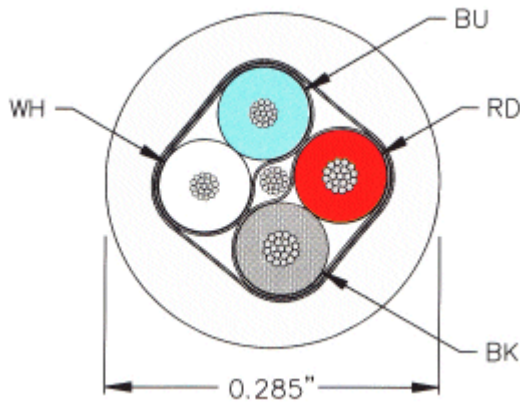


Abb. 60: Aufbau CAN-/DeviceNet-Kabel ZB5200

### 4.3.10.2.6 Schirmung

Der Schirm ist über das gesamte Buskabel zu verbinden und nur an einer Stelle galvanisch zu erden um Masseschleifen zu vermeiden.

Das Schirmungskonzept mit HF-Ableitung von Störungen über R/C-Glieder auf die Tragschiene geht davon aus, dass die Tragschiene entsprechend geerdet und störungsfrei ist. Sollte dies nicht der Fall sein, so kann es vorkommen, dass HF-Störpegel über die Tragschiene auf den Schirm des Buskabels übertragen werden. In diesem Fall sollte der Schirm an den Kopplern nicht aufgelegt werden - aber dennoch komplett durchverbunden sein.

Hinweise für die Überprüfung der CAN-Verdrahtung finden sich im Kapitel [Fehlersuche / Trouble Shooting](#) [► 71].

### 4.3.10.2.7 Kabelfarben

Vorschlag für die Verwendung der Beckhoff CAN-Kabel an Busklemme und Feldbus Box:

Pin BK51x0 PIN BX5100 (X510)	Pin BK5151 CX8050, CX8051, CXxxxx-B510/M510	Pin Feld- bus Box	Pin FC51xx	Funktion	Kabelfarbe ZB5100	Kabelfarbe ZB5200
1	3	3	3	CAN Ground	<b>schwarz</b> /(rot)	<b>schwarz</b>
2	2	5	2	CAN Low	<b>schwarz</b>	<b>blau</b>
3	5	1	5	Schirm	Beilaufnitze	Beilaufnitze
4	7	4	7	CAN high	<b>weiß</b>	<b>weiß</b>
5	9	2	9	nicht benutzt	<b>(rot)</b>	<b>(rot)</b>

**4.3.10.2.8 BK5151, FC51xx, CX mit CAN Interface und EL6751: D-Sub 9polig**

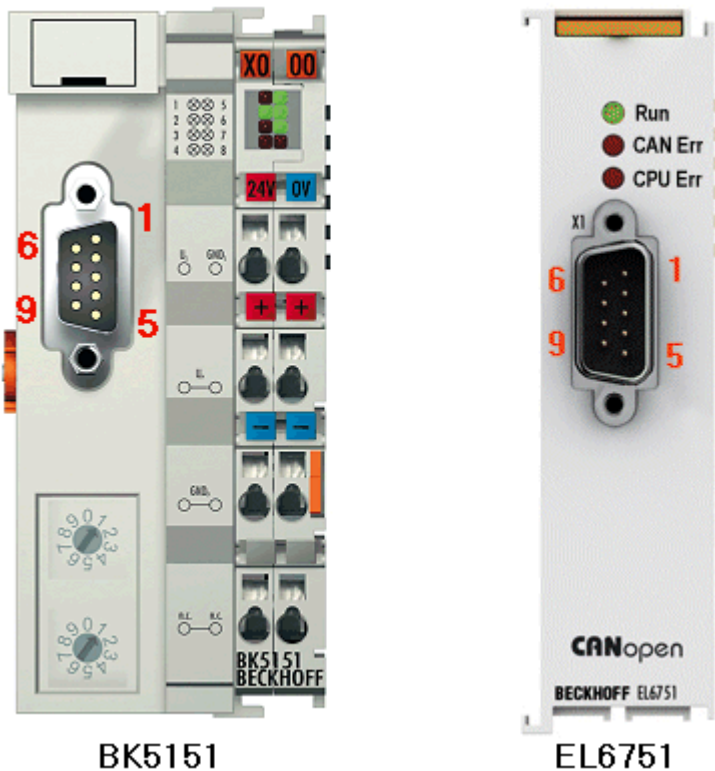
Die CAN Busleitung wird an die FC51x1, FC51x2 CANopen Karten und bei der EL6751 CANopen Master-/Slaveklemme über 9polige Sub-D-Buchsen mit folgender Steckerbelegung angeschlossen.

Pin	Belegung
2	CAN low (CAN-)
3	CAN Ground (intern verbunden mit Pin 6)
6	CAN Ground (intern verbunden mit Pin 3)
7	CAN high (CAN+)

Die nicht aufgeführten Pins sind nicht verbunden.

Die Tragschienenkontaktfeder und der Steckerschirm sind durchverbunden.

Hinweis: an Pin 9 darf eine Hilfsspannung bis 30 V<sub>DC</sub> angeschlossen sein (wird von manchen CAN Geräten zur Versorgung der Transceiver genutzt).



**BK5151**

**EL6751**

Abb. 61: Pinbelegung BK5151, EL6751

**FC51x2:**

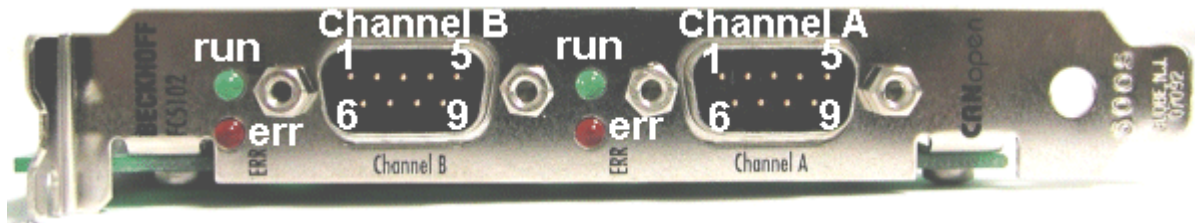


Abb. 62: FC51x2

**4.3.10.2.9 BK51x0/BX5100: 5poliger Open Style Connector**

Bei den BK51x0/BX5100 (X510) Buskopplern befindet sich auf der linken Seite eine abgesenkte Frontfläche mit einem 5poligen Stecker.

Hier kann die mitgelieferte CANopen- Verbindungsbuchse eingesteckt werden.

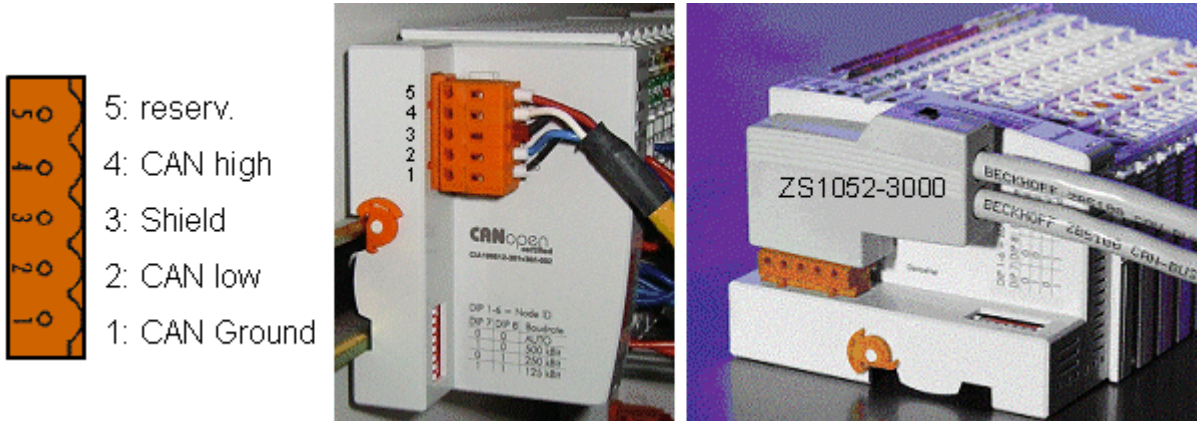


Abb. 63: Belegung Verbindungsbuchse BK51x0/BX5100

Das linke Bild zeigt die Buchse im Buskoppler BK51x0/BX5100. Pin 5 ist dabei der oberste Pin auf der Steckerleiste. Pin 5 ist nicht benutzt. Pin 4 ist die CAN-High-Leitung, Pin 2 die CAN-Low-Leitung und an Pin 3 wird der Schirm aufgelegt (ist über eine R/C-Schaltung mit der Tragschiene verbunden). Optional kann am Pin 1 CAN-GND angeschlossen werden. Wenn alle CAN-Ground Pins verbunden sind ergibt dies ein gemeinsames Bezugspotential für die CAN Transceiver im Netz. Es empfiehlt sich, CAN-GND an einer Stelle zu erden, damit das gemeinsame CAN Bezugspotential nahe beim Versorgungs-Potential liegt. Da die CANopen Buskoppler BK51X0/BX5100 über eine vollständige galvanische Trennung des Busanschlusses verfügen, kann u.U. auf die Verdrahtung von CAN-Ground verzichtet werden.

**Businterface Connector ZS1052-3000**

Alternativ zum mitgelieferten Stecker kann der CAN Interface Connector ZS1052-3000 eingesetzt werden. Dieser vereinfacht die Verdrahtung erheblich. Für die ankommenden und abgehenden Leitungen stehen separate Klemmen zur Verfügung, der Schirm wird durch die Zugentlastung großflächig angeschlossen. Der integrierte Abschlusswiderstand kann von außen zugeschaltet werden. Ist er eingeschaltet, so wird die abgehende Busleitung elektrisch abgetrennt - damit lassen sich Verdrahtungsfehler schnell lokalisieren, und es ist gewährleistet, dass nicht mehr als zwei Widerstände im Netz aktiv sind.

**4.3.10.2.10 LC5100: Busanschluss über Federkraftklemmen**

Beim Low-Cost-Koppler LC5100 wird die CAN-Leitung direkt auf die Klemmstellen 1 (CAN-H, gekennzeichnet mit C+) bzw. 5 (CAN-L, gekennzeichnet mit C-) aufgelegt. Der Schirm kann optional auf die Klemmstellen 4 bzw. 8 aufgelegt werden, diese sind über eine R/C-Schaltung mit der Tragschiene verbunden.

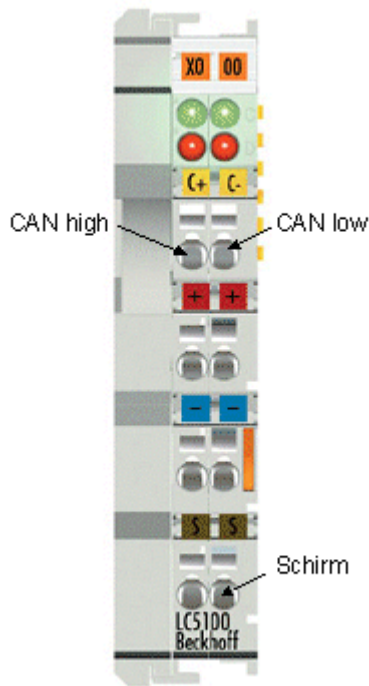


Abb. 64: LC5100

**HINWEIS**

**Beschädigung der Geräte möglich!**

Durch die nicht vorhandene galvanische Trennung kann durch unsachgemäße Verkabelung der CAN Treiber zerstört oder geschädigt werden. Verkabeln sie immer im ausgeschalteten Zustand. Verkabeln Sie erst die Spannungsversorgung und legen sie erst dann den CAN auf. Überprüfen Sie die Verkabelung und schalten dann erst die Spannung ein.

**4.3.10.2.11 Feldbus Box: M12 CAN Buchse**

Bei der Feldbus Box IPxxxx-B510, IL230x-B510 und IL230x-C510 wird der Busanschluss mit 5poligen M12 Steckverbindern ausgeführt.

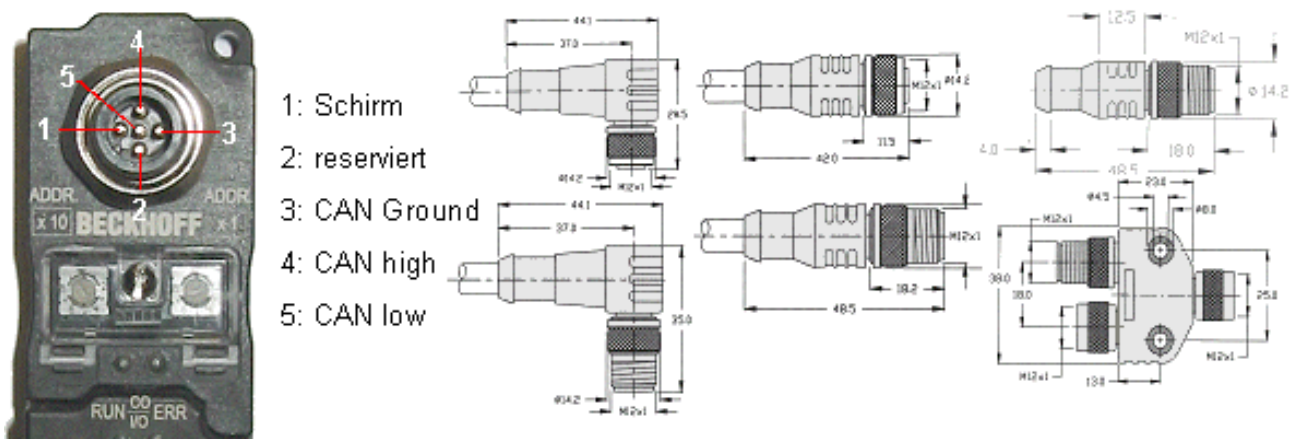


Abb. 65: Pinbelegung M12 Stecker Feldbus Box

Für das Feldbus Box System bietet Beckhoff feldkonfektionierbare Stecker, Passivverteiler, Abschlusswiderstände sowie eine große Auswahl an vorkonfektionierten Kabeln an. Details finden sich im Katalog oder unter [www.beckhoff.de](http://www.beckhoff.de).



### 4.3.10.3 SSB - Konfiguration

Der SSB wird im System Manager konfiguriert. Öffnen Sie Ihre bestehende Konfiguration oder öffnen Sie eine neue Konfiguration in der Sie das PLC-Projekt, K-Bus und den übergeordneten Feldbus bereits konfiguriert haben.

Sie können nun unter I/O Devices (linke Maustaste) ein weiteres Gerät anfügen.

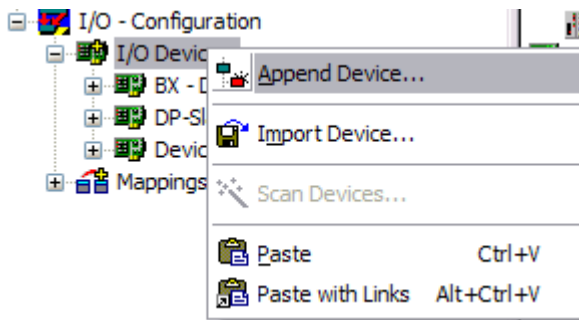


Abb. 66: Anfügen eines weiteren Gerätes

Wählen Sie den CANopen Master SSB aus und bestätigen Sie mit OK.

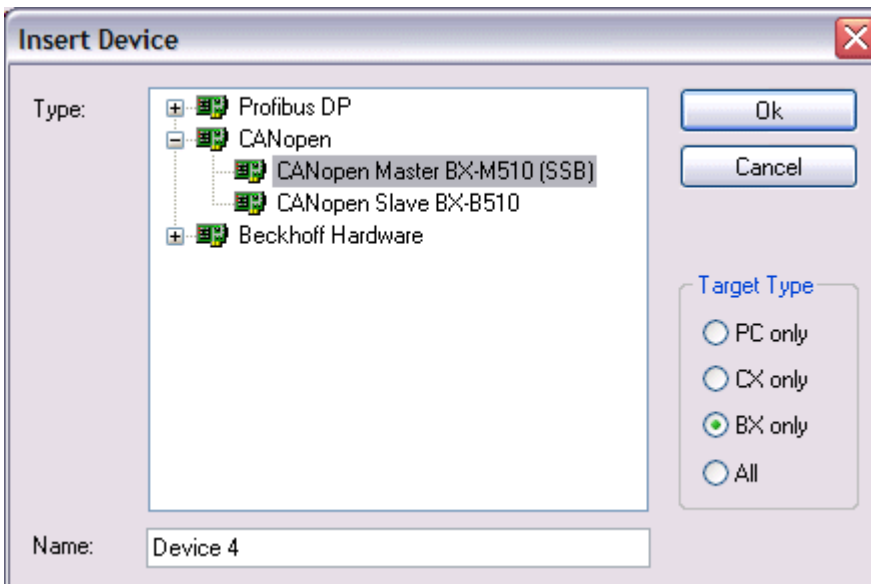


Abb. 67: Auswahl des CANopen Masters SSB

Als nächstes kann mit der linken Maustaste auf dem SSB-Device ein CANopen-Knoten angewählt werden.

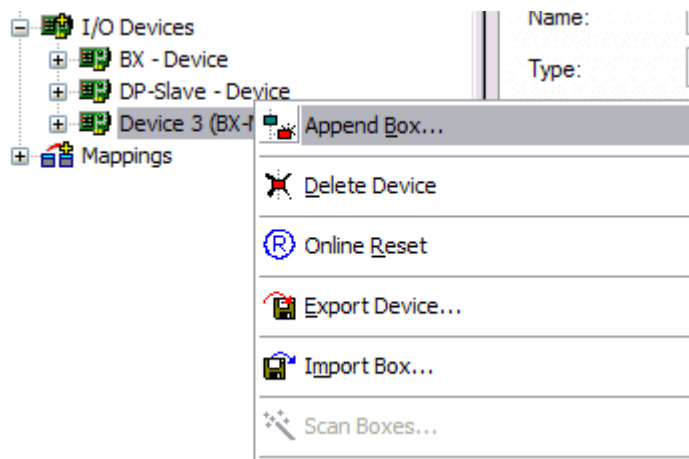


Abb. 68: Anfügen eines CANopen Geräts

Es stehen hier alle Beckhoff CAN-Knoten zur Verfügung sowie ein allgemeiner CANopen-Node für CANopen-Geräte anderer Hersteller.

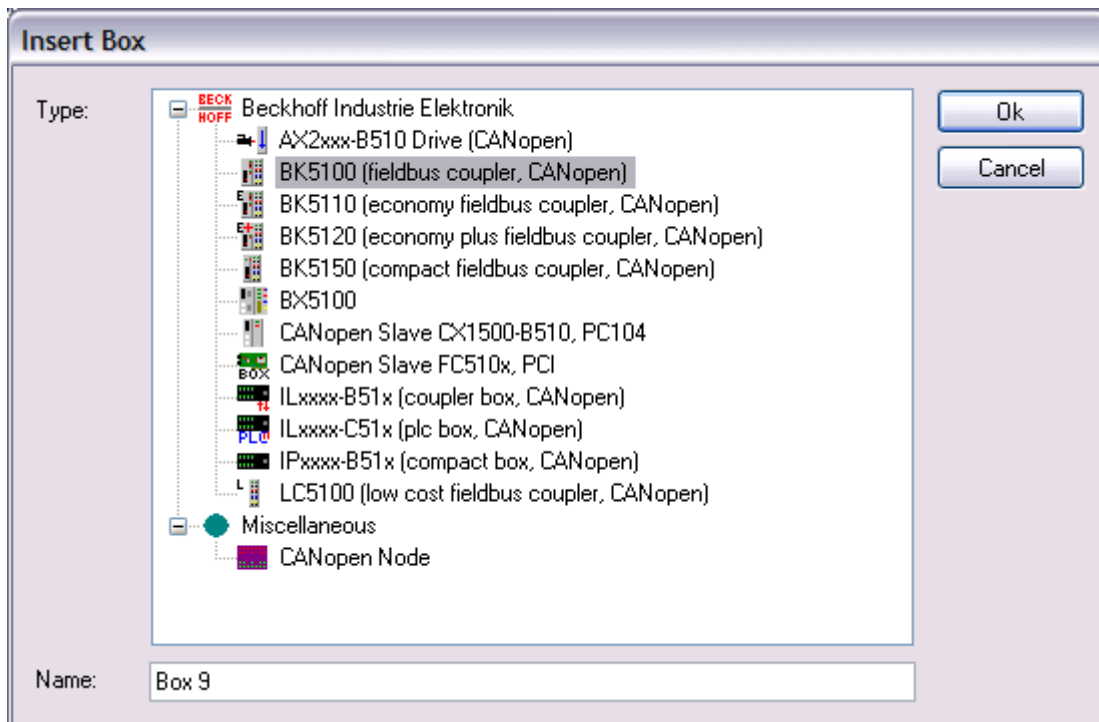


Abb. 69: Auswahl eines CANopen-Knotens

Verknüpfen Sie nun die PLC Variablen mit Ihren CAN-Knoten. Nach erfolgreicher Konfiguration laden Sie diese in den BX.

#### 4.3.10.4 SSB - SDO-Kommunikation

CANopen SDO-Kommunikation (Service Daten Objekt) dient zum Auslesen bzw. Beschreiben beliebiger Parameter im Objektverzeichnis des CANopen Busknotens. Der SSB benutzt die SDO-Kommunikation zur Konfiguration der Kommunikationsparameter beim Aufstarten.

##### Download von anwendungsspezifischen Parametern beim Aufstarten

Hierzu sind die entsprechenden Parameter im System Manager bei dem entsprechenden Knoten im Karteireiter SDO einzugeben. In eckigen Klammern erscheinen die Objekte, die sich aus den Konfigurationen im Griff CAN Node ergeben. Anschließend können beliebige Objektverzeichniseinträge angefügt werden.

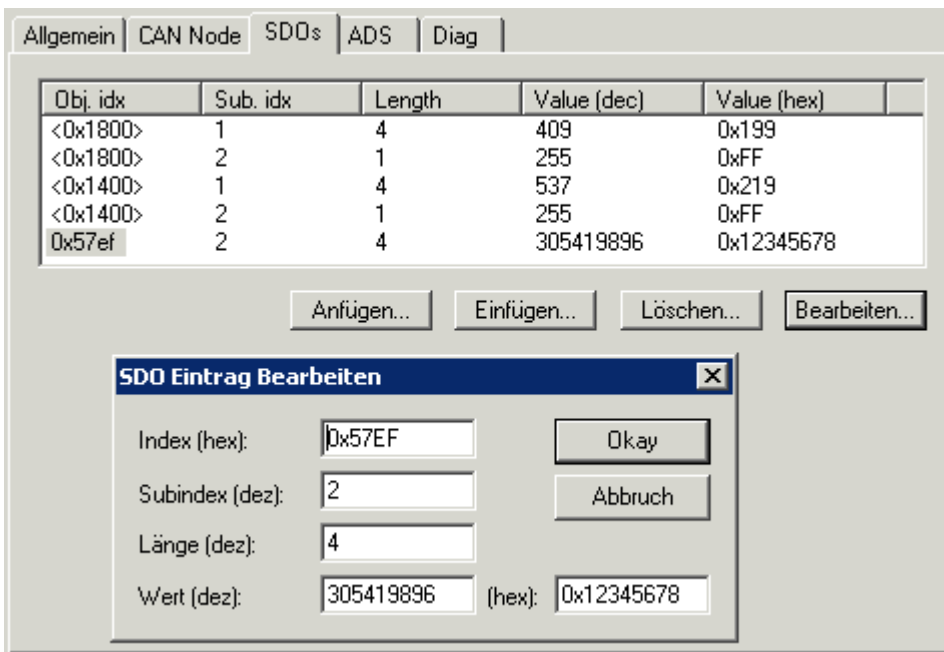


Abb. 70: Anfügen/Bearbeiten von Objektverzeichniseinträgen

Der SSB erwartet eine positive Quittierung des Parameterdownloads vom jeweiligen Busteilnehmer. Falls ein Parameter nicht geschrieben werden konnte (SDO-Abbruch durch Busteilnehmer), so versucht die Karte den entsprechenden Wert auszulesen und mit dem zu schreibenden Wert zu vergleichen - es könnte sich ja z. B. um einen schreibgeschützten Wert (Read only) handeln, der bereits korrekt im Busteilnehmer konfiguriert ist. Bei Übereinstimmung geht die Karte zum nächsten Parametereintrag.

### 4.3.10.5 SDO-Kommunikation aus der PLC

Für die SDO-Kommunikation aus der PLC heraus verwendet man die ADS-Bausteine. Mit diesen Bausteinen ist es möglich SDO-Telegramme zu versenden und die Antwort des Slaves zu empfangen (ADSWRITE/ADSREAD).

Eingangsparameter	Beschreibung
NETID	lokale NetId des BX oder leer lassen zum Beispiel mit "
Port Nummer	0x1000 <sub>hex</sub> + NodeId (Slave Nummer)
IDXGRP	SDO Index
IDXOFFS	SDO Subindex
LEN	Länge der SDO Daten (1...4)

 Download BX (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207257611.zip>)

### Setzen einzelner oder aller Knoten in den Pre-Operational oder Operational Zustand

Mit dem ADSWRTCTL Baustein können Sie einzelne CANopen Knoten oder alle Slaves in den Pre-Operational oder Operational Zustand versetzen.

Eingangsparameter	Beschreibung
NETID	lokale NetId des BX oder leer lassen zum Beispiel mit "
Port Nummer	0x1000 <sub>hex</sub> + NodeId (Slave Nummer) / 153 <sub>dez</sub> (alle Knoten)
ADSSTATE	ADSSTATE_RUN
DEVSTATE	1 - Pre / 0 - Operational
LEN	0
SRCADDR	0

 Download BX (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207259787.zip>)

**SSB Interface neu Starten**

Mit dem ADSWRTCTL Baustein kann der SSB gestoppt und neu gestartet werden. Führen Sie als erstes ein Stopp aus und als nächstes einen Start aus.

Eingangsparameter	Beschreibung
NETID	lokale NetId des BX oder leer lassen zum Beispiel mit "
Port Nummer	153 <sub>dez</sub>
ADSSTATE	ADSSTATE_STOP, ADSSTATE_RUN
DEVSTATE	0
LEN	0
SRCADDR	0

oder

Eingangsparameter	Beschreibung (ab Software Version 1.16 bei allen BX Controllern)
NETID	lokale NetId des BX oder leer lassen zum Beispiel mit "
Port Nummer	300 <sub>dez</sub>
ADSSTATE	ADSSTATE_RESET
DEVSTATE	0
LEN	4
SRCADDR	ADR auf eine DWORD Variable mit der ID des SSB Device (die ID ist aus dem System Manger File zu entnehmen und ist typischerweise ein Wert von 1...3)

**4.3.10.6 Emergency-Telegramme und Diagnose**

Über den NodeState wird der Status des CAN-Slaves angezeigt. Das DiagFlag wird gesetzt wenn ein Emergency-Telegramm empfangen wurde. Der EmergencyCounter zählt bei jedem Emergency-Telegramm eins hoch.

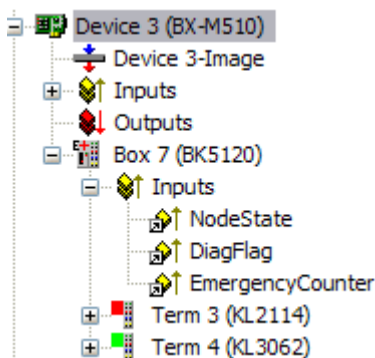


Abb. 71: NodeState, DiagFlag und EmergencyCounter

Wert NodeState	Beschreibung
0	No error
1	Node deactivated
2	Node not found
4	SDO syntax error at Start Up
5	SDO data mismatch at Start Up
8	Node start up in progress
11	SSB Bus off
12	Pre-Operational
13	Severe bus fault
14	Guarding: toggle error
20	TxPDO too short
22	Expected TxPDO is missing
23	Node is Operational but not all TxPDOs were received

**ADS Port 153**


**Auslesen der Emergency Telegramme mit AdsRead**

Eingangsparameter	Beschreibung
NETID	lokale NetId des BX
Port Nummer	153
IDXGRP	16#xxxxF180 (xxxx) NodeId, das Diag Flag wird nur beim Auslesen von mindestens 106 Byte zurückgesetzt 16#xxxxF181 (xxxx) NodeId, das Diag Flag wird sofort zurückgesetzt
IDXOFFS	Byte Offset

**Beschreibung des Arrays**

Offset	Bit	Wert / Beschreibung
0 - 1	Bit 0	reserviert
	Bit 1	Boot-Up-Message nicht empfangen oder fehlerhaft
	Bit 2	Emergency-Overflow
	Bit 3 - 15	reserviert
2 - 3	Bit 0 - 14	TX-PDO (i+1) empfangen
	Bit 15	alle TX-PDOs 16-n empfangen
4 - 5	Bit 0 - 4	1: falsche TX-PDO-Länge
		2: synchrone TX-PDO fehlt
		3: Node meldet PRE-OPERATIONAL
		4: Event-Timer bei einer TX-PDO abgelaufen
		5: keine Antwort beim Guarden
		6: mehrmals kein Toggeln beim Guarden
	Bit 5 - 15	zugehörige COB-ID
6	Bit 0 - 7	1: falscher Wert bei einem SDO-Upload
		2: falsche Länge bei einem SDO-Upload
		3: Abort bei einem SDO-Up-/Download
		4: falsches Datum bei einer Boot-Up-Message
		5: Timeout beim Warten auf Boot-Up-Message
7	Bit 0 - 7	2: falscher SDO-Command specifier
		3: SDO-Toggle-Bit hat sich nicht geändert
		4: SDO-Länge zu groß
		5: SDO-Abort
		6: SDO-Timeout
8 - 9	Bit 0 - 7	Index des SDO-Up/Downloads
10	Bit 0 - 7	Subindex des SDO-Up/Downloads
11	Bit 0 - 7	reserviert
12	Bit 0 - 7	errorClass des Aborts
13	Bit 0 - 7	errorCode des Aborts
14 - 15	Bit 0 - 15	additionalCode des Aborts
16 - 19		gelesener Wert (falls Offset 6 = 1)
20 - 23		erwarteter Wert (falls Offset 6 = 1)
24 - 25		Anzahl der folgenden Emergencies
26 - n		Emergencies (jeweils 8 Byte)

 Download BX (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207261963.zip>)

 Download Beispiel System-Manager File BX (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207264139.zip>)

**Auslesen der Anzahl der PDO Telegramme mit AdsRead**

Eingangsparameter	Beschreibung
NETID	lokale NetId des BX
Port Nummer	153
IDXGRP	16#xxxxF930 (xxxx) NodeId
IDXOFFS	0

 Download BX (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207266315.zip>)

**● Konfiguration der Node-ID erforderlich**



Die Node-ID muss vor dem Aufruf des ADS-Baustein in der TwinCAT-Konfiguration konfiguriert worden sein.

**Beliebige CAN Nachricht verschicken**

Mit diesem ADSWRITE Befehl ist es möglich eine beliebige CAN Nachricht zu versenden.

Eingangsparameter	Beschreibung
NETID	lokale NetId des BX
Port Nummer	153
IDXGRP	16#0000F921
IDXOFFS	0
LEN	11 Bytes
SRCADDR	Pointer auf ein ARRAY von 11 Byte

**Aufbau der 11 Byte CAN Daten**

Byte	Beschreibung	Beispiel Node 7 SDO 0x607 Len 8 Download Request 0x2100 (Index) Sub Index 1 - Value "1"
1	COB-ID LowByte	0x06 (SDO Low Byte)
2	COB-ID HighByte	0x07 (SDO High Byte)
3	LEN (Länge)	0x08 (Len, kann hier auch 5 sein)
4	Daten[1]	0x22 (Download Request)
5	Daten[2]	0x00 (Index Low Byte)
6	Daten[3]	0x21 (Index High Byte)
7	Daten[4]	0x01 (Sub Index)
8	Daten[5]	0x01 (Value "1")
9	Daten[6]	0x00
10	Daten[7]	0x00
11	Daten[8]	0x00

**4.3.10.7 CANopen Trouble Shooting**

**Error Frames**

Fehler in der CAN-Verkabelung, der Adressvergabe und der Baud-Rateneinstellung zeigen sich u.a. durch eine erhöhte Anzahl an Error Frames: die Diagnose LEDs zeigen dann *Warning Limit wird überschritten* oder *Bus-Off-Zustand erreicht*.

**● Error Frames**



Überschrittenes Warning Limit, Error Passive oder Bus-Off Zustand werden zunächst bei demjenigen Knoten angezeigt, der die meisten Fehler entdeckt hat. Dieser Knoten muss nicht unbedingt die Ursache für das Auftreten dieser Error Frames sein!

Wenn z. B. ein Knoten überdurchschnittlich stark zum Busverkehr beiträgt (z. B. weil er als einziger über analoge Eingänge verfügt, deren Daten in kurzen Abständen ereignisgesteuerte PDOs auslösen), so werden auch seine Telegramme mit großer Wahrscheinlichkeit zunächst gestört - entsprechend erreicht sein Fehlerzähler als erster kritische Zustände.

**Node-ID / Baud Rate Einstellung**

Es muss sorgfältig darauf geachtet werden, dass keine Knotenadresse doppelt vergeben ist: für jedes CAN-Datentelegramm darf es nur einen Sender geben.

**Test 1**

Knotenadressen überprüfen. Falls die CAN Kommunikation wenigstens zeitweise funktioniert und alle Geräte die Boot-Up-Nachricht unterstützen, so kann die Adressvergabe auch durch Aufzeichnen der Boot-Up-Nachrichten nach dem Einschalten der Geräte überprüft werden - hierdurch wird aber kein Vertauschen von Knotenadressen erkannt.

**Test 2**

Überprüfen, ob überall die gleiche Baud-Rate eingestellt ist. Bei Sondergeräten: Wenn Bittiming Parameter zugänglich, stimmen diese mit den CANopen-Definitionen überein (Abtastzeitpunkt, SJW, Oszillator).

**Test der CAN-Verkabelung**

Diese Tests nicht ausführen, wenn das Netzwerk aktiv ist: Während der Tests sollte keine Kommunikation stattfinden. Die folgenden Tests sollten in der angegebenen Reihenfolge ausgeführt werden, da manche Tests davon ausgehen, dass der vorhergehende Test erfolgreich war. In der Regel sind nicht alle Tests notwendig.

**Netzwerkabschluss und Signalleitungen**

Für diesen Test sollten die Knoten ausgeschaltet oder die CAN-Leitung abgesteckt sein, da die Messergebnisse sonst durch die aktiven CAN-Transceiver verfälscht werden können.

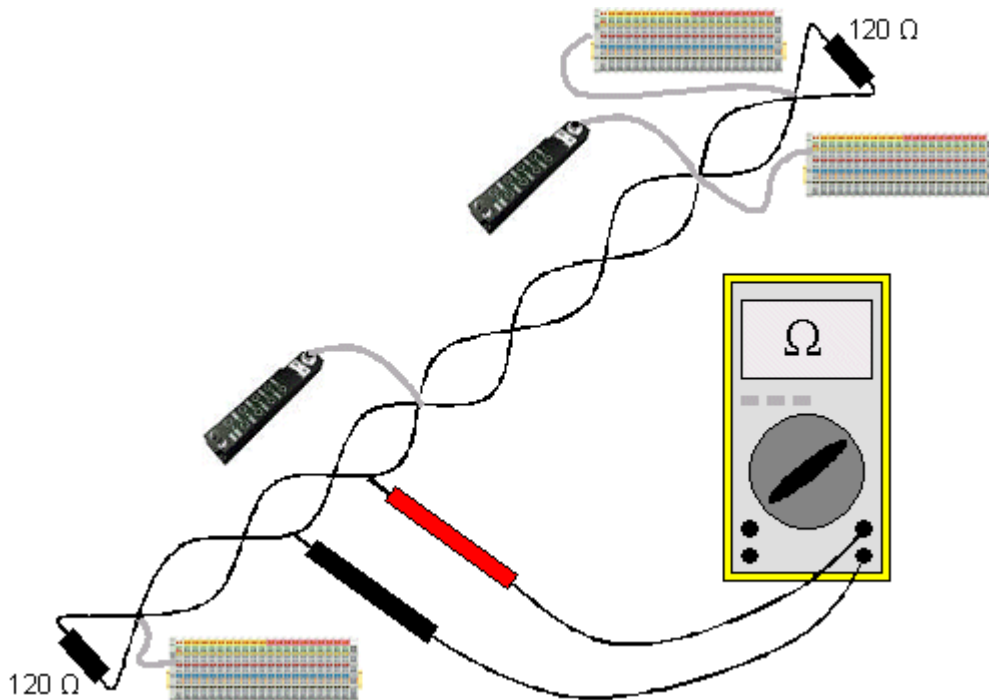


Abb. 72: Verdrahtungsplan für Testaufbau

**Test 3**

Widerstand zwischen CAN-high und CAN-low ermitteln - ggf. bei jedem Gerät.

Wenn der Messwert über 65 Ohm liegt, deutet dies auf fehlende Abschlusswiderstände oder den Bruch einer Signalleitung hin. Wenn der Messwert kleiner 50 Ohm ist, nach Kurzschluss zwischen CAN-Leitung, überzähligen Abschlusswiderständen oder fehlerhaften Transceivern suchen.

**Test 4**

Auf Kurzschluss zwischen CAN-Ground und den Signalleitungen sowie zwischen Schirm und Signalleitungen prüfen.



**Test 5**

Erdung von CAN-Ground und Schirm auftrennen. Auf Kurzschluss zwischen CAN-Ground und Schirm prüfen.

**Topologie**

Die Leitungslänge bei CAN Netzwerken hängt stark von der gewählten Baud-Rate ab. CAN toleriert dabei kurze Stichleitungen - ebenfalls in Abhängigkeit von der Baud-Rate. Die erlaubte Stichleitungslänge sollte nicht überschritten werden. Häufig wird die verlegte Leitungslänge unterschätzt - die Schätzung liegt teilweise Faktor 10 unter der tatsächlichen Länge. Deshalb empfiehlt sich folgender Test:

**Test 6**

Die Stichleitungslängen sowie die Busgesamtlänge nachmessen (nicht nur grob schätzen!) und mit den Topologieregeln (Baud-Ratenabhängig) vergleichen.

**Schirmung und Erdung**

Stromversorgung und Schirm sollten sorgfältig, einmalig und niederohmig beim Netzteil geerdet werden. Alle Verbindungsstellen, Abzweige etc. im CAN-Kabel müssen neben den Signalleitungen (und evtl. CAN-GND) auch den Schirm durchverbinden. In den Beckhoff IP20 Buskopplern wird der Schirm über ein R/C-Glied hochfrequenzmäßig geerdet.

**Test 7**

Mit DC-Strommessgerät (16 Amp max.) Strom zwischen Spannungsversorgungs-Masse und Schirm am vom Netzteil entfernten Ende des Netzes messen. Es sollte ein Ausgleichsstrom vorhanden sein. Wenn kein Strom vorhanden ist, so ist der Schirm nicht durchgängig verbunden oder das Netzteil ist nicht richtig geerdet. Wenn das Netzteil in der Mitte des Netzwerkes angeordnet ist, so sollte an beiden Enden gemessen werden. Dieser Test kann u.U. auch an den Stichleitungsenden durchgeführt werden.

**Test 8**

Den Schirm an mehreren Stellen auftrennen und den Verbindungsstrom messen. Wenn ein Stromfluss vorhanden ist, so ist der Schirm an mehreren Stellen geerdet (Erdschleife)

**Potentialunterschiede**

Der Schirm muss für diesen Test durchgängig sein und darf keinen Strom führen (vorher getestet).

**Test 9**

Spannung zwischen Schirm und Spannungsversorgungs-Erde an jedem Knoten ermitteln und notieren. Der maximale Potentialunterschied zwischen zwei beliebigen Geräten sollte kleiner als 5 Volt sein.

**Fehler erkennen und lokalisieren**

Am besten funktioniert in der Regel der "Low-tech-Ansatz": Teile des Netzes abhängen und beobachten, wann der Fehler verschwindet.

Aber: Dies funktioniert nicht gut bei Problemen wie zu großen Potentialunterschieden, Masseschleifen, EMV und Signalverfälschung da die Verkleinerung des Netzes häufig das Problem löst, ohne dass der „fehlende“ Teil ursächlich war. Auch die Buslast ändert sich beim Verkleinern des Netzes - damit können externe Störungen seltener CAN-Telegramme "treffen".

Die Diagnose mittels Oszilloskop führt meist nicht zum Erfolg: CAN Signale sehen auch im ungestörten Zustand teilweise recht wirr aus. Unter Umständen kann mit einem Speicheroszilloskop auf Error Frames getriggert werden - diese Art der Diagnose ist aber Messtechnik-Experten vorbehalten.

## Protokollprobleme

In seltenen Fällen sind auch Protokollprobleme (z. B. fehlerhafte oder unvollständige CANopen-Implementierung, unglückliches Timing im Boot-Up etc.) Ursache von Störungen. Hier ist dann ein Mitschrieb (Trace) des Busverkehrs mit anschließender Auswertung durch CANopen Experten erforderlich - das Beckhoff Support Team kann hier helfen.


Für solch einen Trace eignet sich ein freier Kanal einer Beckhoff FC5102 CANopen PCI-Karte - die erforderliche Trace-Software stellt Beckhoff im Internet zur Verfügung. Alternativ kann selbstverständlich auch ein handelsübliches CAN Analysetool eingesetzt werden.

Protokollprobleme lassen sich vermeiden, indem auf den Einsatz von Geräten verzichtet wird, die nicht Conformance getestet sind. Der offizielle CANopen Conformance Test und das entsprechende Zertifikat sind beim CAN in Automation Verband (<http://www.can-cia.de>) erhältlich.

## 4.3.10.8 Beispiele

### 4.3.10.8.1 BK5120 am SSB

Notwendiges Material:

- TwinCAT 2.9 Build 953 oder Größer
- BX3100 Version 0.80 oder Größer, BX5100 Version 0.13, BX8000 Version 0.04
- 1 x KL1xx4
- 1 x KL2xx4
- 1 x KL9010
- 1 x BK5120
- 1 x KL1xx4
- 1 x KL2xx4
- 1 x KL9010
- Verkabelungsmaterial sowie Spannungsversorgung
- TwinCAT System Manager File (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207268491.zip>)
  -  (Das System Manager File muss per ADS auf den BX-Controller geladen werden).
- BX-Programm File (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207270667.zip>)



Für den Konfigurations-Download über ADS benötigt man entweder einen BECKHOFF Feldbus-Master oder eine freie serielle Schnittstelle.

### 4.3.10.8.2 Kommunikation von BX-Controller zu BX-Controller (über SSB)

Über den SSB können 2 oder mehrere BX-Controller untereinander Daten austauschen. Verwenden Sie im System Manager bei der Projektierung dieses Datenaustausches CAN-Schicht 2 Telegramme.

Über die COB Id wird die Kommunikation des CAN-Telegramms festgelegt. Um welchen BX-Typ es sich dabei handelt spielt keine Rolle, da der SSB auf jedem BX-Controller vorhanden ist, sich auch gleich verhält und projektiert wird.



Abb. 73: Kommunikation von BX-Controller zu BX-Controller (über SSB)

**Beispielkonfiguration**

BX\_ONE:  
 Node Id 2  
 CAN\_Out AT %QB100: ARRAY[0..7] OF BYTE  
 COD Id 514 0x202  
 CAN\_In AT %IB100: ARRAY[0..7] OF BYTE  
 COD Id 386 0x182

BX\_TWO:  
 Node Id 2  
 CAN\_Out AT %QB100: ARRAY[0..7] OF BYTE  
 COD Id 386 0x182  
 CAN\_In AT %IB100: ARRAY[0..7] OF BYTE  
 COD Id 514 0x202

Beispiel Konfiguration und Programm:

**Notwendiges Material**

- TwinCAT 2.9 Build 959 oder Größer
- 2 x BXxx00
- Verkabelungsmaterial sowie Spannungsversorgung
- TwinCAT System Manager File BX\_ONE (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207272843.zip>)



- Programm File BX\_ONE (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207275019.zip>)



- TwinCAT System Manager File BX\_TWO (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207277195.zip>)



- Programm File BX\_ONE (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207279371.zip>)



Für den Konfigurations-Download über ADS benötigt man entweder einen BECKHOFF Master (FC310x, FC510x, FC520x) oder eine freie serielle Schnittstelle.

### 4.3.10.8.3 AX2000 am SSB



Abb. 74: AX2000

#### Notwendiges Material:

- TwinCAT 2.9 Build 953 oder Größer
- BX3100 Version 0.80 oder Größer
- 1 x KL1xx4
- 1 x KL2xx4
- 1 x KL9010
- 1 x AX2000 mit folgende Einstellungen: Slave Adresse 4, Baudrate 500 kByte
- Verkabelungsmaterial sowie Spannungsversorgung
- Beispiel Programm und Konfiguration auf dem BX-Controller
  - TwinCAT System Manager File (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207281547.zip>)



(Das System Manager File muss per ADS zum BX-Controller geladen werden).

- BX-Programm-File (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207283723.zip>)



Für den Konfigurations-Download über ADS benötigt man entweder einen BECKHOFF Feldbus-Master oder eine freie serielle Schnittstelle.

**AX2000 Beschreibung**

Folgende Abschnitte sind Auszüge aus der Dokumentation des AX2000 Drive Handbuch. Weitere Informationen finden Sie unter der Internet Adresse <http://www.Beckhoff.de>.

**Hardware und Schnittstellen**

**Einstellen der Stationsadresse**

Die Stationsadresse (Geräteadresse am CAN-Bus) des Servoverstärkers können Sie auf drei Arten einstellen:

- Mit der Tastatur in der Frontplatte (siehe Installationsanleitung AX2000)
- In der Inbetriebnahme-Software DRIVE.EXE auf der Bildschirmseite "Basiseinstellungen"
- Über die serielle Schnittstelle mit der Abfolge der ASCII-Kommandos:

ADDR nn > SAVE > COLDSTART (mit nn = Adresse)

Der Adressbereich kann mit Hilfe des ASCII – Objektes MDRV von 1..63 auf 1..127 expandiert werden.

**Einstellen der Baudrate**

Die CAN-Übertragungsgeschwindigkeit (Baudrate) können Sie auf drei Arten einstellen:

- Mit der Tastatur in der Frontplatte (siehe Installationsanleitung AX2000)
- In der Inbetriebnahme-Software DRIVE.EXE auf der Bildschirmseite "Basiseinstellungen"
- Über die serielle Schnittstelle mit der Abfolge der ASCII - Kommandos:  
CBAUD bb > SAVE > COLDSTART (mit bb = Baudrate in kB)

Mögliche Baudraten sind 10, 20, 50, 100, 125, 250, 333, 500 (default), 666, 800 und 1000 kBaud.

**CANopen Interface (X6)**

Interface zum Anschluss an den CAN Bus (default 500 kBaud). Das integrierte Profil basiert auf dem Kommunikationsprofil CANopen DS301 und dem Antriebsprofil DSP402. Im Zusammenhang mit dem Lageregler werden u.a. folgende Funktionen bereitgestellt:

Tippen mit variabler Geschwindigkeit, Referenzfahren, Fahrauftrag starten, Direktfahrauftrag starten, digitale Sollwertvorgabe, Datentransferfunktionen und viele andere.

Detaillierte Informationen finden Sie im CANopen-Handbuch. Die Schnittstelle ist über Optokoppler galvanisch getrennt und liegt auf dem gleichen Potential wie das RS232-Interface. Die analogen Sollwerteingänge sind weiterhin nutzbar. Mit der optionalen Erweiterungskarte -2CAN- werden die beiden Schnittstellen RS232 und CAN, die denselben Stecker X6 belegen, auf zwei Stecker verteilt.

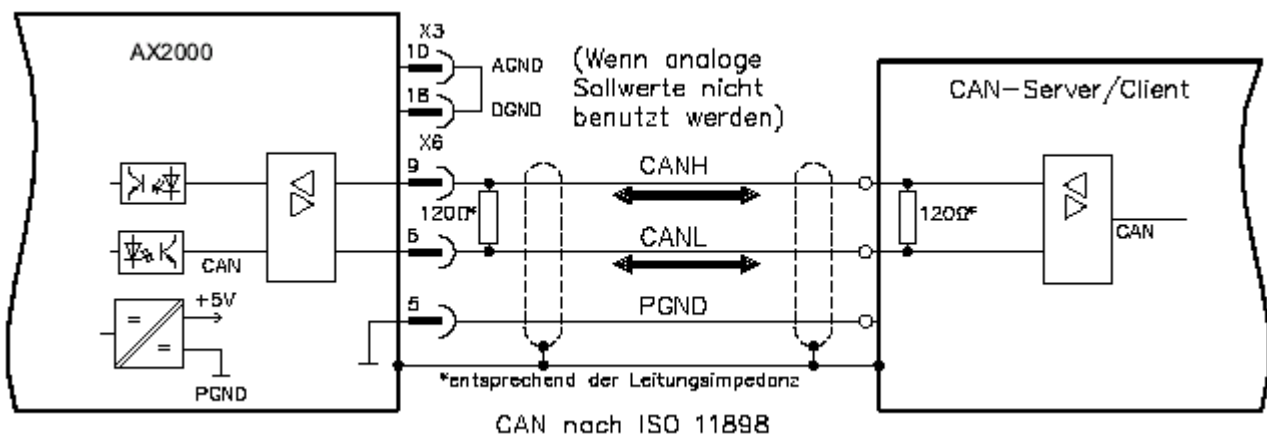


Abb. 75: CANopen Interface (X6)

#### 4.3.10.8.4 Cimrex-Panel am SSB des BX-Controllers

Die CAN-Schnittstelle des BX-Controllers kann auch dazu benutzt werden, ein Bedien-Panel anzuschließen. In diesem Beispiel wird ein Panel der Firma Beijers angeschlossen. Weitere Informationen zu dem Panel finden Sie unter <http://www.beijerelectronics.de>.



Abb. 76: Cimrex-Panel am SSB des BX-Controllers

#### Benötigte Komponenten

- 1 x BX3100
- Einige Busklemmen für den K-Bus (hier 3 x KL2114, kann aber in der System-Manager-Datei angepasst werden)
- 1 x Cimrex 41
- 1 x CAB 15 CAN Adapter
- Beispielprogramm in ST für den BX: (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207285899.zip>)



- Beispielkonfiguration für den BX: (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207288075.zip>)



- Beispiel für Cimrex 41: (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207290251.zip>)



- Baudrate 500 kBaud
- CAN Slave Adresse 10

#### 4.3.10.8.5 IclA-Drive am SSB



# IclA<sup>®</sup>

Abb. 77: IclA-Drive am SSB

Notwendiges Material:

- TwinCAT 2.9 Build 953 oder Größer
- BX3100 Version 0.80 oder Größer
- 1 x KL1xx4
- 1 x KL2xx4
- 1 x KL9010
- 1 x IclA D065 Folgende Einstellungen Slave Adresse 10, Baudrate 500 kByte (Achtung: Dies sind nicht die Default Parameter des Antriebs)
- Verkabelungsmaterial sowie Spannungsversorgung

Für den Konfigurations-Download über ADS benötigt man entweder einen BECKHOFF Feldbus-Master oder eine freie serielle Schnittstelle.

#### Umkonfigurierungsbeispiel für TwinCAT mit der CANopen-Masterkarte FC510x

Um den Antrieb umzustellen, kann dies mit folgendem Beispiel erfolgen.

- TwinCAT-System-Manager-File (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207292427.zip>)



- TwinCAT-PLC-File (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207294603.zip>)



#### Beispiel Programm und Konfiguration auf dem BX-Controller

- TwinCAT-System-Manager-File (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207296779.zip>)



(Das System-Manager-File muss per ADS zum BX-Controller geladen werden).

- BX-Programm-File (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207298955.zip>)



**IcIA D065 - Beschreibung**

Folgende Abschnitte sind Auszüge aus der Dokumentation des IcIA-Drive-Handbuchs. Diese wurden uns von der Firma SIG Positec Automation GmbH für die Beschreibung der grundlegenden Parameter zur Verfügung gestellt. Weitere Informationen finden Sie unter der Internet Adresse <http://www.sig-positec.de>.

**Hardware und Schnittstellen**

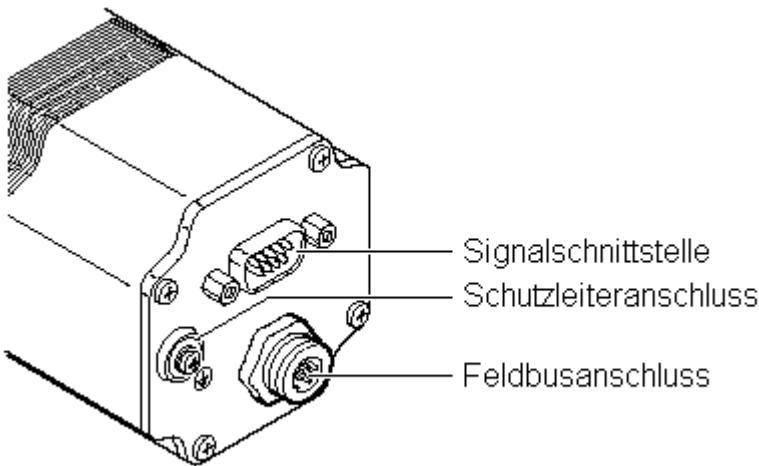


Abb. 78: Anschlüsse des IcIA-Drives

- Signalschnittstelle für
  - Versorgungsspannung
  - Steuersignale für Manuellbetrieb
  - Anschluss für Not-Aus-Signal
- Schutzleiteranschluss für die Erdung über PE-Sammelschiene
- Feldbusanschluss für den Anschluss des Feldbuskabels.

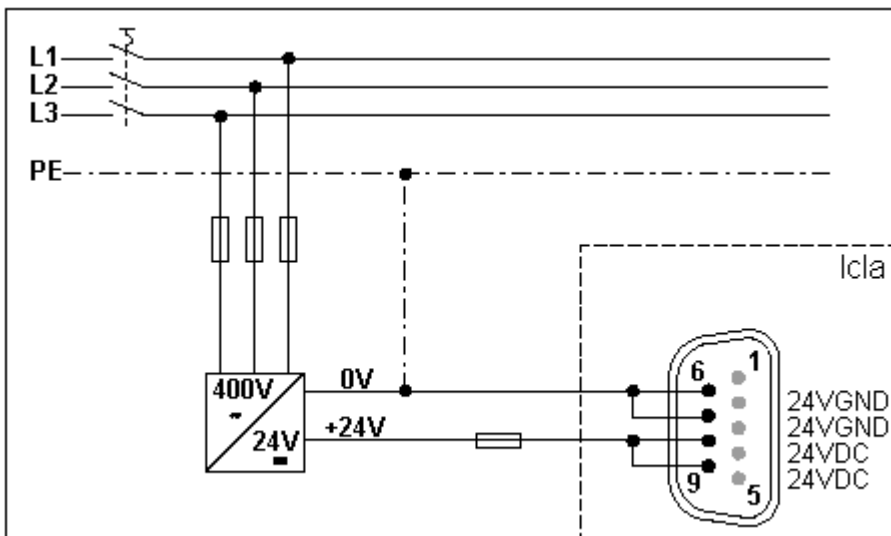


Abb. 79: Signalschnittstelle

Wird die Not-Aus-Funktion nicht benötigt, verbinden Sie Pin 2 mit Pin 8 oder 9 (24 V<sub>DC</sub>).



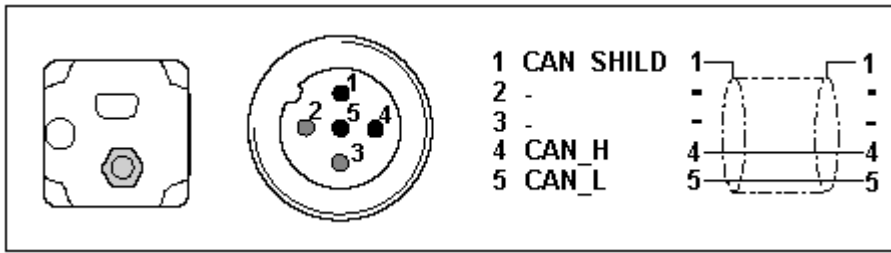


Abb. 80: Feldbusanschluss

**Control-Wort 0x6040**

Das Objekt stellt das Steuerungswort des Geräts dar. Mit dem Control-Wort werden mehrere Steuerungsaufgaben erfüllt:

- Wechsel zwischen den verschiedenen Betriebszuständen. Die möglichen Zustände und Übergänge finden Sie unter dem Index-Schlagwort „Zustandsmaschine“. Relevant für einen Zustandswechsel sind die Bits 0 bis 3 und Bit 7.
- Starten und Unterbrechen von betriebsartenspezifischen Funktionen, z. B. Starten eines Fahrauftrags über das Bit 4. Die Bits 4 bis 6 werden für betriebsartenspezifische Einstellungen benutzt. Einzelheiten finden Sie unter den Schlagworten „Betriebsart, starten“, „Betriebsart, überwachen“ und bei der Beschreibung der jeweiligen Betriebsarten in den Kapiteln „Manuellbetrieb“ und „Positionierbetrieb“.
- Stoppen des Positionierantriebs aus einem laufenden Fahrbetrieb. Zum Anhalten wird das Bit 8 „Halt“ benutzt Einzelheiten finden Sie unter den Schlagworten „Betriebsart, starten“ und „Betriebsart, überwachen“.

Objektbeschreibung	Wertebeschreibung
Index	6040h
Objektname	Control-Wort
Datentyp	Integned16
Subindex	00h, Control-Wort
Zugriff	read-write
PDO-Mapping	R_PDO

Bit	Bezeichnung	Bedeutung
11..15	Manufacturer specific	nicht benutzt
9, 10	-	reserviert
8	Halt	Motor stoppen
7	Reset fault	Fehler rücksetzen
4..6	-	betriebsartenabhängig,
3	Enable operation	Betriebsart ausführen
2	Quick Stop (low aktiv)	Abbremsen mit Quick Stop-Rampe
1	Disable voltage (low aktiv)	Spannung ausschalten
0	Switch on	betriebsbereit schalten

**Statusword 0x6041**

Das Objekt beschreibt den aktuellen Betriebszustand des Geräts. Mit dem Status-Wort führen Sie folgende Überwachungsfunktionen durch:

- Prüfen des Betriebszustands der Positioniersteuerung. Dazu sind die Bits 0 bis 3, 5 und 6 relevant.
- Das Bit 4 zeigt an, ob die Endstufe bereit ist, einen Fahrauftrag zu bearbeiten.
- Die Bits 7 bis 15 werden zur Überwachung des Fahrbetriebs und zur Statusüberwachung gerätespezifischer Zustände benutzt.

Einzelheiten zur Überwachung des Fahrbetriebs finden Sie unter den Schlagworten „Betriebsart, starten“, „Betriebsart, überwachen“ und bei der Beschreibung der jeweiligen Betriebsarten in den Kapiteln „Manuellbetrieb“ und „Positionierbetrieb“. Die Bits zur Statusüberwachung des Geräts sind im Kapitel „Diagnose und Fehlerbehebung“ beschrieben.  
Das Steuerungswort ist in den ersten beiden Byte der R\_PDOs abgebildet.

Objektbeschreibung	Wertebeschreibung
Index	6041h
Objektname	Status-Wort
Datentyp	Unsigned16
Subindex	00h, Status-Wort
Zugriff	read-only
PDO-Mapping	T_PDO

Bit	Bezeichnung	Bedeutung
15	Out of security area	Sicherheitsbereich verlassen 0->1: Endschalterposition S0 oder S1 überfahren
14	Out of drive area	Fahrbereich verlassen 0->1: Endschalterposition D0 oder D1 überfahren
12..13	-	betriebsartenabhängige Bedeutung
11	Internal limit active	Arbeitsbereich verlassen
10	Target reached	Ziel erreicht 1->0: Neue Zielposition übergeben 0->1: Angeforderte Zielposition erreicht oder Motorstillstand nach Halt-Anforderung
9	Remote	0: Manuellbetrieb 1: kein Manuellbetrieb
8	Right out of drive area	Nur gültig, wenn Bit 11 = 1 - 0: Endschalterposition W1 überfahren - 1: Endschalterposition W0 überfahren
7	Warning	Warnung
6	Switch on disabled	nicht betriebsbereit
5	Quick Stop	Quick Stop aktiv
4	Voltage disabled	Spannung ausgeschaltet
3	Fault	Fehler aufgetreten
2	Operation enabled	Betriebsart aktiviert
1	Switched on	betriebsbereit
0	Ready to switch on	einschaltbereit

**Referenzierungsbereiche**

Eine gültige Referenzierung wird über drei Endschalterzonen definiert, die im möglichen Verfahrbereich des Antriebs liegen müssen. Die Endschalter schützen Antrieb und Anlage vor einer Beschädigung.

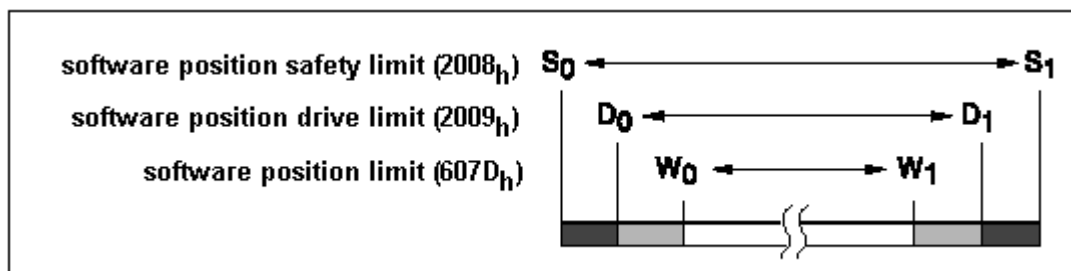


Abb. 81: Referenzierungsbereiche

- Arbeitsbereich W0 - W1 für den Positionierbetrieb.

- Fahrbereich D0 - D1. Aus den Bereichen D0 - W0 und D1 - W1 kann
- der Antrieb nur in Richtung Arbeitsbereich zurückgefahren werden.
- Sicherheitsbereich S0 - S1. Aus den Bereichen S0 - D0 und S1 - D1 kann der Antrieb nur manuell zurückbewegt werden.
- CANopen-Objekte Für die Einrichtung der Endschalter werden drei CANopen-Objekte eingesetzt, in denen die Positionswerte für die obere und untere Bereichsgrenze eingetragen werden.
- Grenzen des Arbeitsbereichs in software position limit (607D<sub>hex</sub>)
- Grenzen des Fahrbereichs in software position drive limit (2009<sub>hex</sub>)
- Grenzen des Sicherheitsbereichs in software position safety limit (2008<sub>hex</sub>)

**Beispiel für eine Referenzierung**

Das folgende Listing zeigt die Eingabe der Referenzierungswerte. Die Knotenadresse des Positionierantriebs ist auf 01<sub>hex</sub> eingestellt.

COB-ID	Daten	Bedeutung
601	2F 60 60 00 06	R_SDO: Umschalten in Homing Mode
581	60 60 60 00 xx	T_SDO: OK
601	23 08 20 02 0C 7B 00 00	R_SDO: max. Wert Sicherheitsbereich S <sub>1</sub> : 7B0Ch
581	60 08 20 02 xx xx xx xx	T_SDO: OK
601	23 08 20 01 00 00 00 00	R_SDO: min. Wert Sicherheitsbereich S <sub>0</sub> : 0000h
581	60 08 20 01 xx xx xx xx	T_SDO: OK
601	23 09 20 02 42 72 00 00	R_SDO: max. Wert Fahrbereich D <sub>1</sub> : 7242h
581	60 09 20 02 xx xx xx xx	T_SDO: OK
601	23 09 20 01 CA 08 00 00	R_SDO: min. Wert Fahrbereich D <sub>0</sub> : 8CAh
581	60 09 20 01 xx xx xx xx	T_SDO: OK
601	23 7D 60 02 AE 60 00 00	R_SDO: max. Wert Arbeitsbereich W <sub>1</sub> : 60AEh
581	60 7D 60 02 xx xx xx xx	T_SDO: OK
601	23 7D 60 01 5E 1A 00 00	R_SDO: min. Wert Arbeitsbereich W <sub>0</sub> : 1A5Eh
581	60 7D 60 01 xx xx xx xx	T_SDO: OK
601	23 10 10 03 73 61 76 65	R_SDO: Applikationsparameter speichern: "save"
581	60 10 10 03 xx xx xx xx	T_SDO: OK
601	2F 98 60 00 FF	R_SDO: Auswahl der Referenzierungsart
581	60 98 60 00 xx	T_SDO: OK
601	23 0B 20 00 BC 34 00 00	R_SDO: Maßsetzen, Istposition bis S <sub>0</sub> : 34BCh
581	60 0B 20 00 xx xx xx xx	T_SDO: OK
601	2B 40 60 00 1F 00	R_SDO: Homing Operation Start (steigende Flanke, Bit 4)
581	60 40 60 00 xx xx	T_SDO: OK

Abb. 82: Listing der Referenzierungswerte

#### 4.3.10.8.6 Lenze Frequenzumrichter am SSB

# Lenze



Abb. 83: Frequenzumrichter der Fa. Lenze

#### Notwendiges Material

- TwinCAT 2.9 Build 953 oder Größer
- BXxx00
- 1 x KL1xx4
- 1 x KL2xx4
- 1 x KL9010
- 1 x Lenze 8200 vector + Motor
- 1 x Lenze CANopen Interface 2175
- Verkabelungsmaterial sowie Spannungsversorgung

Für den Konfigurations-Download über ADS benötigen Sie eine BECKHOFF Feldbus-Masterkarte oder eine freie serielle Schnittstelle.

#### Lenze Beschreibung

Folgende Abschnitte sind Auszüge aus der Dokumentation des Lenze 2175-Handbuchs. Diese wurden uns von der Firma Lenze Drive Systems GmbH für die Beschreibung der grundlegenden Parameter zur Verfügung gestellt. Weitere Informationen finden Sie unter der Internet Adresse <http://www.Lenze.com>.

#### Erstinbetriebnahme

Stellen Sie die Spannungsversorgung für das Busmodul auf Interne Spannungsversorgung.

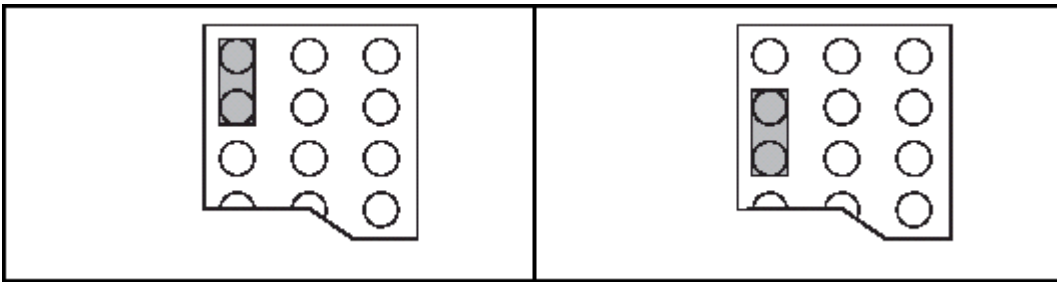


Abb. 84: Externe Spannungsversorgung - Interne Spannungsversorgung (Auslieferungszustand)

Für die CANopen-Kommunikation stellen Sie den DIP-Schalter 10 auf "ON".

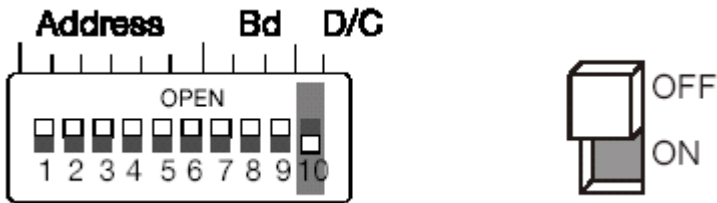


Abb. 85: DIP-Schalter

**Baudrate DIP-Schalter 7-9**

Übertragungsrate [kBit/s]	S7	S8	S9
10	ON	ON	OFF
20	ON	OFF	ON
50	OFF	ON	ON
125	OFF	ON	OFF
250	OFF	OFF	ON
500 (default)	OFF	OFF	OFF
1000	ON	OFF	OFF

**● Wertigkeit der DIP-Schalter**

**i** DIP Schalter 6 hat die kleinste Wertigkeit.  
Beispiel: Adresse 3 Schalter 5 und 6 auf "ON".

**Freigabe des Kommunikationsmoduls**

Um das Kommunikationsmodul freizuschalten muss die Bedienungsart auf 3 gestellt werden. Dies kann über den SSB erfolgen mit folgendem Eintrag:

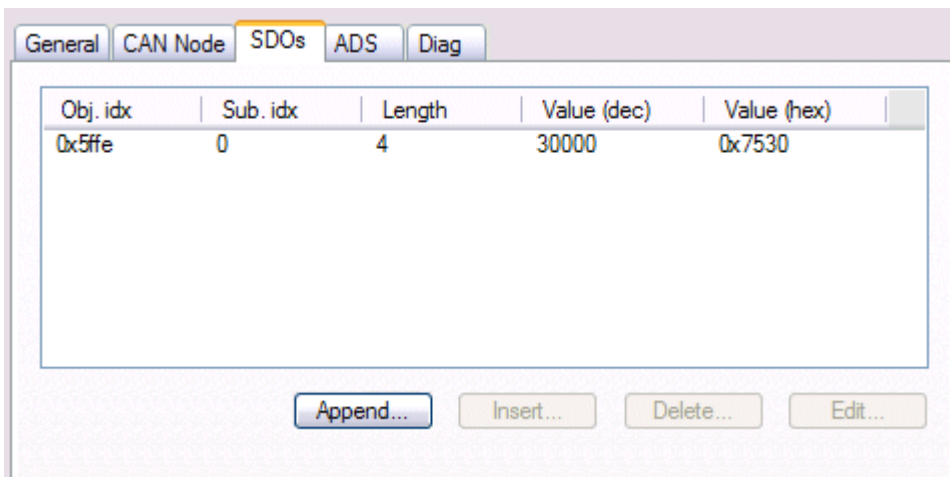
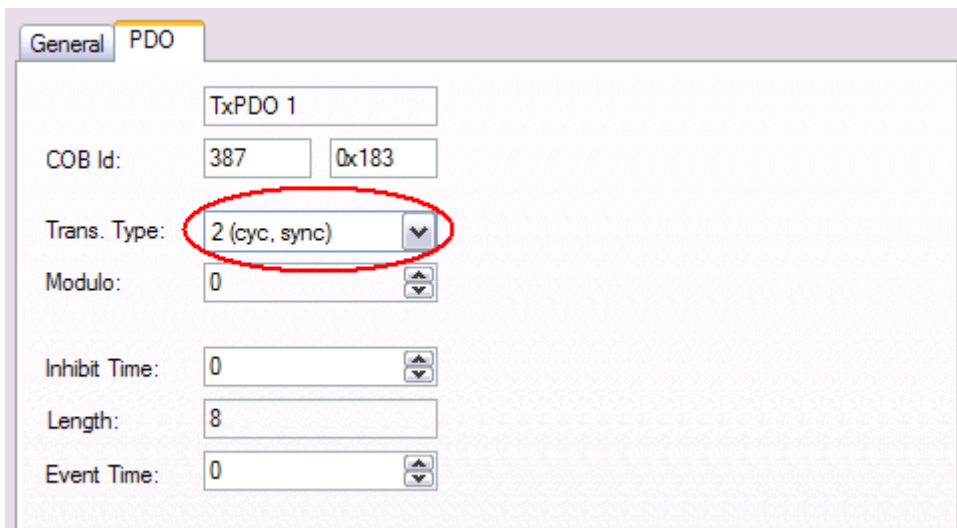


Abb. 86: Freigabe des Kommunikationsmoduls

### Sync-Telegramm

In der Default-Einstellung sendet der Lenze-Drive seine Ausgangs-PDOs nur, nachdem er vom CAN-Master ein Sync-Telegramm empfangen hat. Wenn Sie den Trans. Type zum Beispiel auf 2 stellen, sendet der Lenze-Drive nach jedem 2. empfangenen Sync-Telegramm ein Ausgangs-PDO.



### Beispiel Projekt

- TwinCAT-System-Manager-File: (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207301131.zip>)



- TwinCAT-PLC-File: (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207303307.zip>)



### 4.3.11 Echtzeit-Uhr (RTC)

Auf dem BX-Controller ist eine Echtzeit-Uhr (Real Time Clock - RTC) implementiert. Die Uhrzeit ist akkugepuffert.

#### Einstellen der Echtzeit-Uhr

Die einfachste Art die Uhr einzustellen ist über den System Manager. Sofern die ADS-Kommunikation steht wird die momentan aktuelle Uhrzeit auf dem BX-Controller angezeigt. Um die Uhrzeit einzustellen editieren Sie einfach die Zeit und mit der Drop Down Taste können Sie Tage, Monate und Jahre einstellen. Für die Einstellungen der Jahre klicken Sie auf die Jahresanzeige und stellen Sie das gewünschte Jahr ein. Ebenso verfahren Sie mit dem Monat. Nach dem Sie alle Einstellungen vorgenommen haben klicken Sie auf den Button *Update RTC on BX*.

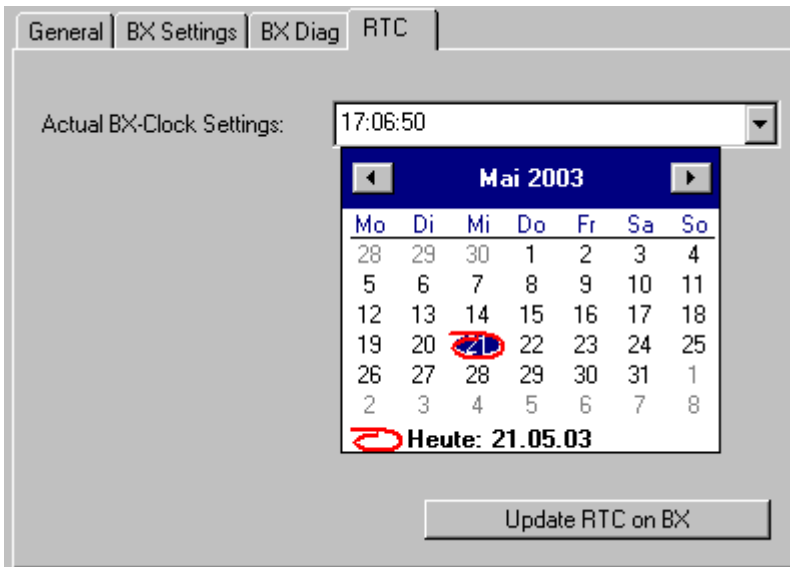


Abb. 87: Einstellen der Echtzeit-Uhr (RTC)



#### Lebensdauer des Akkus

Je nach Belastung des Akkus kann die Lebensdauer des Akkus variieren.

#### Lesen der RTC auf dem BX-Controller (siehe Beispiel [▶ 124] Programmierung\Bibliothek)

Die RTC kann per Funktionsbaustein ausgelesen werden. Notwendige Bibliotheken:

- TcSystemBX.lbx
- TCBaseBX.lbx

#### Schreiben der RTC auf dem BX-Controller

Die RTC kann per Funktionsbaustein eingestellt werden. Notwendige Bibliotheken:

- TcSystemBX.lbx
- TCBaseBX.lbx

**Lesen der RTC per ADS**

Beschreibung	Bedeutung	Wert
NETID	Ziel-Gerät	siehe System Manager "ADS"
Port	ADS Port Nummer	150 <sub>dez</sub>
IDXGRP	IDX Gruppe	0x0000_F100 <sub>hex</sub>
IDXOFFS	IDX Offset	0x0000_0000 <sub>hex</sub>
Länge	Länge der Daten	16 Byte
Variablen Typ	Art der Variabel	TIMESTRUCT

**Schreiben der RTC per ADS**

Beschreibung	Bedeutung	Wert
NETID	Ziel-Gerät	siehe System Manager "ADS"
Port	ADS Port Nummer	150 <sub>dez</sub>
IDXGRP	IDX Gruppe	0x0000_F100 <sub>hex</sub>
IDXOFFS	IDX Offset	0x0000_0000 <sub>hex</sub>
Länge	Länge der Daten	16 Byte
Variablen Typ	Art der Variabel	TIMESTRUCT

**Einstellung über den Navigationsschalter**

Siehe Menü. [[▶ 90](#)]

**Technische Daten zur RTC**

Genauigkeit: ca. 1 Sekunde/Tag  
 Speicherung der Uhrzeit: ca. 3 Monate bei aufgeladenem Akku  
 Lebensdauer des Akkus: ca. 10 Jahre bei 10 Zyklen pro Tag (1000 Zyklen bei einem kompletten Lade- und Entladezyklus)

**4.3.12 COM-Port**

Auf dem BX-Controller befinden sich zwei serielle Schnittstellen. Die PIN-Belegung entnehmen Sie bitte der [Hardware-Beschreibung](#) [[▶ 26](#)].

**Einstellmöglichkeiten:**

Beschreibung	Auswahl
<a href="#">Baudrate</a> [ <a href="#">▶ 116</a> ]	9600 Baud 19200 Baud 38400 Baud (beginnt bei der Autobaudratenerkennung) 57600 Baud 115200 Baud (nur COM 2)
Datenbits	7 8 (Default)
Parity	NONE ODD EVEN (Default)
Stoppbits	1 (Default) 2

**COM 1**

Die Schnittstelle COM 1 für die Kommunikation mit der KS2000 Software oder dem TwinCAT PLC Control (einloggen über die serielle Schnittstelle).



**COM 2**

Die Schnittstelle COM 2 (mit RS 232 oder RS 485 Physik) für die Nutzung eigener Protokolle oder der Protokoll-Bibliotheken (wie ModbusRTU, RK512, etc.) zum Anschluss andere serieller Geräte.

**Bibliothek**

Für die Kommunikation mit der seriellen Schnittstelle stehen Funktionsbausteine zur Verfügung.

- [Dokumentation](#) [▶ 116]
- [Beispiel](#) [▶ 121]
- [Bibliothek](#) [▶ 116]

## 4.4 Menü

### 4.4.1 BX-Menü-Settings

Um in das Menü zu wechseln, drei Sekunden den Navigationsschalter gedrückt halten. Als erstes erscheint das Verzeichnis *Menue*.

- Zwischen den Einstellungen eines Menüs, können Sie man mit den Tasten RIGHT/LEFT wechseln (es ist immer das in in der 1. Zeile angezeigte Menü aktiv).
- Drücken Sie zum Wechseln in ein Untermenü, die Taste DOWN.
- Um wieder in das Hauptmenü zu wechseln drücken Sie die Taste UP.

Im Untermenü wird in der ersten Zeile der Menüpunkt angezeigt und in der 2. Zeile die aktuelle Einstellung zu diesem Menüpunkt.

Einige Einstellungen sind nicht veränderbar (*read only*). Diese Punkte sind rein für die Kontrolle und Information für den Anwender gedacht. Um das Menü zu beenden muss man sich im Hauptmenü befinden und dann den Navigationsschalter drei Sekunden gedrückt halten.

Bevor Einstellungen geändert werden können, muss Passwort gesetzt werden. Das Passwort bleibt auch beim Firmware-Update oder beim zurücksetzen auf die Herstellerkonfiguration gespeichert. Sollten Sie das Passwort vergessen haben, so müssen Sie den BX-Controller einschicken.



Abb. 88: Navigationsschalter der BX-Controller

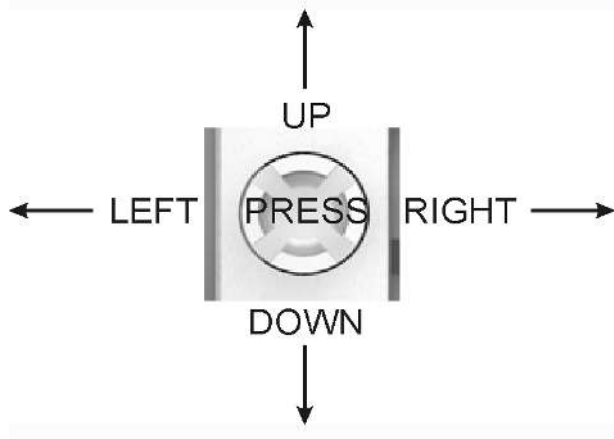
**Schalterbelegung**

Abb. 89: Schalterbelegung

Hauptmenü	Untermenü 1.Zeile	Untermenü 2.Zeile	Read/Write
MENUE	Passwort	**** nicht gesetzt ???? gesetzt	siehe unten
	Factory Settings	Aktivieren?	wenn die Taste gedrückt wird, werden die Herstellereinstellungen gesetzt und der Controller automatisch reboootet
	Reboot	Aktivieren?	wenn die Taste gedrückt wird, reboootet der Controller neu
AMS	AMS	>AMS Net-ID<	read only
PLC	NAME	>aktueller NAME<	read only
	Curr. Exex. Time	>aktueller Wert<	[ms]
	Task Time	>aktueller Wert<	wenn die Taste gedrückt wird, kann die Zyklus Zeit eingestellt werden
	Status	>Boot-Prj< >PLC Status<	Bootprojekt vorhanden PLC Status
Config	NAME	>aktueller NAME<	read only
	Config löschen	Aktivieren?	wenn die rechte Taste gedrückt wird, löscht man die aktuelle Konfiguration
Real Time Clock	Date and Time	>aktuelle Zeit<	read only
	Year	Setting	2003-2xxx
	Month	Setting	1-12
	Day	Setting	1-31
	Day of week	Setting	Mon, ... Fri
	Hour	Setting	0-23
	Minute	Setting	0-59
	Second	Setting	0-59
COM 1 <i>read only</i>	Baudrate	>aktueller Wert<	9600/19200/38400/56800
COM 2 <i>read only</i>	Baudrate	>aktueller Wert<	9600/19200/38400/56800/115k
SSB <i>read only</i>	Baudrate	>aktueller Wert<	1MBaud, 500k, 250k, 125k, 100k, 50k
	Cycle Time	>aktueller Wert< [in µs]	read only
	Utilization	>aktueller Wert< [in %]	read only
K-Bus <i>read only</i>	Diagnose	>aktuelle Diagnose<	read only
	Anzahl der Busklemmen	>aktueller Wert<	read only

**Buspezifische Menü Punkte**

**BX3100**

F-Bus PROFIBUS <i>read only</i>	Adresse*	>aktueller Wert<	1-126
	Baudrate*	>aktueller Wert<	read only
	Status	>aktueller Wert<	read only
	Diagnose*	>aktueller Wert<	read only

**BX5100**

F-Bus CANopen <i>read only</i>	Adresse*	>aktueller Wert<	1-126
	Baudrate*	>aktueller Wert<	read only
	Status	>aktueller Wert<	read only
	Diagnose*	>aktueller Wert<	read only

\*) in Vorbereitung

**BX9000**

Ethernet	MAC ID	>aktueller Wert<	000105-xx-xx-xx, read only
	ADDR.STATE	>aktueller Wert<	read only
	ADDRESSING MODE	FIXED IP (Default) DHCP BOOTP BOOTP & SAVE	read / write
	NAME	>aktueller Wert<	BX_xxxxxx (xxxxxxx last 3 Bytes from the MAC ID) read / write
	DEFAULT GATEWAY	0.0.0.0	read / write
	IP MASK	255.255.0.0	read / write
	IP ADDRESS	172.16.21.20	read / write

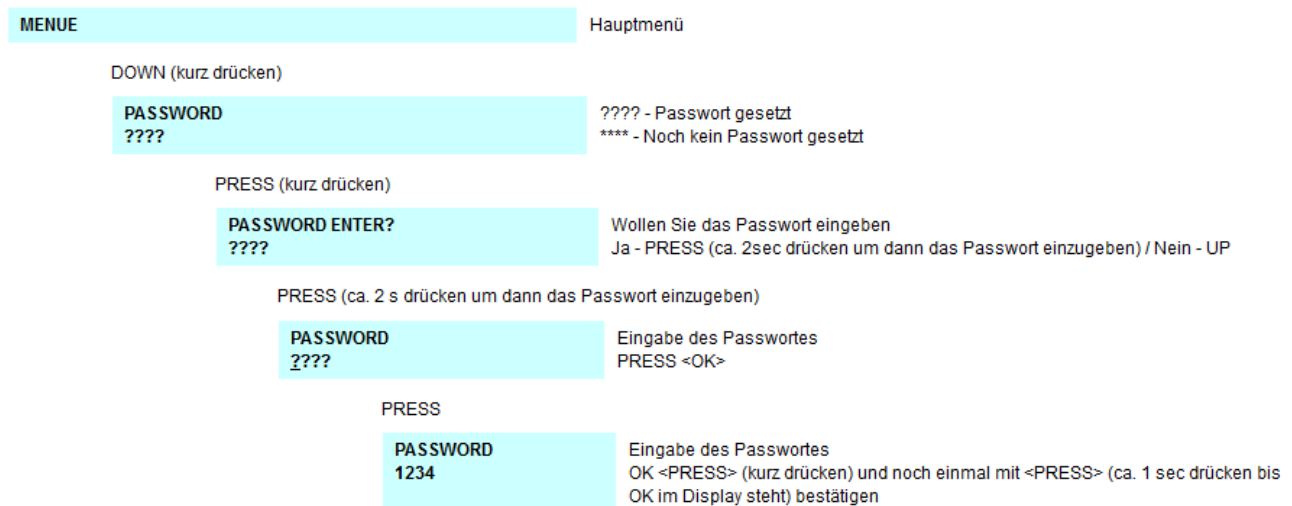
**Code**

Die Default Einstellung ist "\*\*\*\*", das heißt das kein Passwort aktive ist. Um Einstellungen vorzunehmen ist ein Passwort zu setzen.

**Menü Navigation**

Wenn Sie den Navigationsschalter (Press) drei Sekunden drücken wechseln Sie in das Menü - Verzeichnis. Hier sind einige Menüpunkte beschrieben.

**MENÜ**



**F-Bus (nur BX3100)**

**F-BUS**  
**WAIT FOR SETPRM**      Feldbus Status (only read)  
 WAIT FOR SETPRM - Wartet auf die Parameter Daten vom PROFIBUS

**SSB**

**SSB**      Smart System Bus

**COM2**

**COM2**      Serielle Schnittstelle COM2 (only read)

DOWN

**Baudrate**  
**xxx**      Aktuelle Baudrate (only read)

**COM1**

**COM1**      Serielle Schnittstelle COM2 (only read)

DOWN

**Baudrate**  
**xxx**      Aktuelle Baudrate (only read)

**K-Bus**

**KBUS OK**  
**10 TERMINALS**      K-Bus Diagnose (only read)

DOWN

**KBUS RESET**      Reset K-Bus

PRESS (short)

**KBUS RESET EXCT?**      PRESS 1sec - K-Bus Reset wird durchgeführt

**PLC**

**PLC**      PLC Status (only read)

DOWN

**PROJECT**  
**>NAME<**      PLC Projekt Name

RIGHT

**CURR. EXEC. TIME**  
**xxx ms**      Gesamte Bearbeitungszeit im [ms]

RIGHT

### 4.4.2 Erstellen eigener Menüs

Man kann den das Display und Navigations-Schalter auch für eigene Zwecke gebrauchen, z. B. um Diagnoseinformationen anzuzeigen und oder Parameter zu verändern. Für den einfachen Einstieg ist ein Beispiel angefügt, das Sie für Ihre ersten Schritte verwenden und anpassen können.

 Download (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207305483.zip>)

## 4.5 Konfigurations-Software KS2000

Eine Parametrierung und Konfigurierung von Busklemmen-Controllern der BCxx50-, BXxx20- und BXxx00-Serien ist mit der Konfigurations-Software KS2000 nicht möglich. Deren Konfiguration muss mit dem TwinCAT System Manager vorgenommen werden.

Die Konfigurations-Software KS2000 kann Ihnen aber bei der Konfiguration oder Diagnose der an die Busklemmen-Controller angereichten Busklemmen helfen.

Es empfiehlt sich hierzu die Baudrate in Konfigurations-Software KS2000, BCxx50 BCxx20 und BXxx00 auf 38400 Baud zu stellen (8 Datenbits, Even, 1 Stoppbit).

---

### ● COM1 - automatische Baudraten Erkennung



Die Schnittstelle COM 1 der BXxx00 hat eine automatische Baudraten Erkennung von 9,6 kBaud bis 56,4 kBaud.

---

### ● Erforderliche KS2000-Version



Die Konfiguration oder Diagnose von Busklemmen an BXxx00 wird ab KS2000 Version 4.3.14 unterstützt.

---

Um bei einigen Busklemmen die Konfigurationsdialoge nutzen zu können (z. B. KL25xx, KL6811, KL6201, KL6401) müssen folgende Punkte eingestellt werden:

- Ein SPS-Projekt oder ein Boot-Projekt muss deaktiviert sein.
- Der BX-Controller muss in der Default Configuration sein. Setzen Sie die Herstellereinstellung oder schalten Sie im TwinCAT System Manager auf Config Mode (blaues TwinCAT Icon).
- Der BX-Controller muss im FreeRun Modus sein. Aktivieren Sie dies mit dem TwinCAT System Manager.

Nun können Sie sich mit der Konfigurations-Software KS2000 über ADS (Port 100) oder das serielle Kabel einloggen und die KS2000-Dialoge der Busklemmen nutzen.

## 5 Programmierung

### 5.1 PLC-Eigenschaften der BX-Controller

Beschreibung	Wert
Datenspeicher	256 kByte
Programmspeicher	256 kByte minus Task-Konfiguration minus POU's beim Online Change
Source Code Speicher	256 kByte
RETAIN	2 kByte
INPUT	2 kByte
OUTPUT	2 kByte
MERKER	4 kByte
max. grÖÙe einer Variable	16 kByte
max. POU's	Beschränkung durch Speicher

### 5.2 TwinCAT PLC

Das Beckhoff TwinCAT Software-System verwandelt jeden kompatiblen PC in eine Echtzeitsteuerung mit Multi-SPS-System, NC-Achsregelung, Programmierumgebung und Bedienstation. Die Programmierumgebung von TwinCAT wird auch für die Programmierung der BC/BX genutzt. Wenn Sie TwinCAT PLC (Windows NT4/2000/XP) installiert haben, können Sie die Feldbus-Verbindung oder die serielle Schnittstelle für Software-Download und Debugging verwenden.

TwinCAT I/O oder TwinCAT PLC können auch als Ethernet-Master (Host) genutzt werden, um Prozessdaten mit dem Busklemmen-Controller auszutauschen. TwinCAT stellt ihnen hierzu den System Manager als Konfigurationstool sowie die Treiber und das ADS-Protokoll zu Verfügung.

#### Busklemmen-Controller der Serien BCxx50, BCxx20 und BXxx00

Diese Busklemmen Controller der 2. Generation werden mit dem TwinCAT System Manager konfiguriert und mit TwinCAT PLC-Control programmiert. Sie müssen für diese Koppler TwinCAT PLC installieren (Windows NT4, Windows 2000, Windows XP).

#### Programmierung und Programmübertragung

- über die serielle Schnittstelle [► 145]
- über die Feldbus-Schnittstelle (nur bei Busklemmen-Controllern für PROFIBUS, CANopen und Ethernet)

#### Online Change

Die Busklemmen-Controller Busklemmen-Controller der BX-Serie und die BCxx50 unterstützen Online Change. Das bedeutet, dass das PLC-Programm durch ein neues Programm ersetzt wird, ohne dass dadurch das Programm unterbrochen wird. Nach Beendigung der Task wird auf das neue Programm umgeschaltet. Damit verbunden ist die doppelte Haltung des SPS Programms. Es stehen 512 kByte zu Verfügung, die für die doppelte Haltung des PLC Programm durch zwei dividiert werden muss, also 256 kByte. Zusätzlich müssen noch für Task-Konfiguration etc. einige kByte abgezogen werden. Beim Online Change werden noch dynamische Daten in den Speicher abgelegt. Sollte ein Programm an die Speichergrenze stoßen (größer 240 kByte) kann es passieren das ein Online Change nicht mehr funktioniert auch wenn das Programm nach einem "Rebuild all" wieder in den BX geschrieben werden kann.

#### Wann geht kein Online Change?

Es gibt einige Punkte, nachdem ist ein Online Change nicht mehr möglich.



- einfügen einer neuen Bibliothek
- Änderung der Task-Einstellung
- ein "Rebuild all"
- Grenzbereich in der Speicherauslastung des Controllers (PLC Programm größer 90%)

### 5.3 TwinCAT PLC - Fehler-Codes

Fehlerart	Beschreibung
PLC-Kompilierfehler	Maximum number of POU's (...) exceeded
PLC-Kompilierfehler	Out of global data memory ...

#### Fehler POU's

Für jeden Baustein wird eine POU (Process Object Unit) angelegt. Per Default sind 256 Bausteine möglich.

**Error 3612: Maximum number of POU's (100) exceeded! Compile is aborted.**

Data allocation

1 Error(s), 0 Warning(s).

Abb. 90: Maximale Anzahl der POU's überschritten

Wenn man Bibliotheken einbindet, kann dieser Wert nicht mehr ausreichen. Erhöhen Sie dann die Anzahl an POU's.

Öffnen Sie dazu im PLC Control unter Projekte/Optionen...

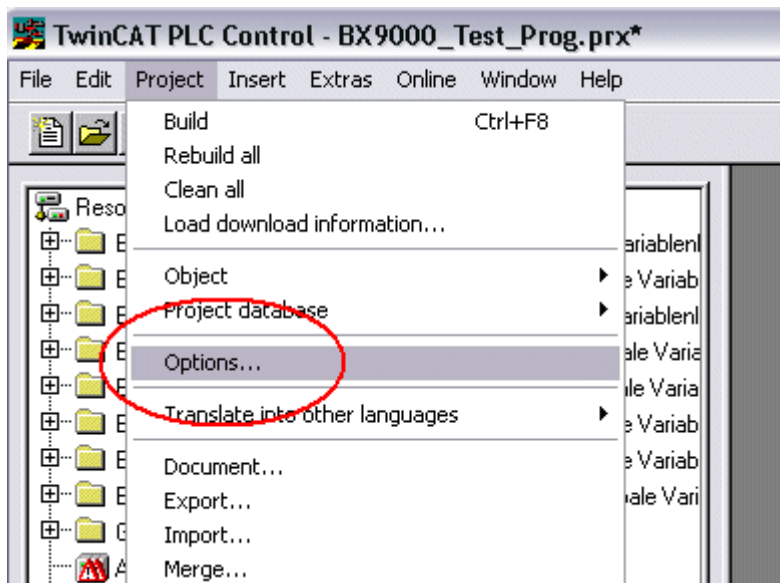


Abb. 91: Menüpfad Projekte / Optionen / Controller Settings

...die Controller Settings.

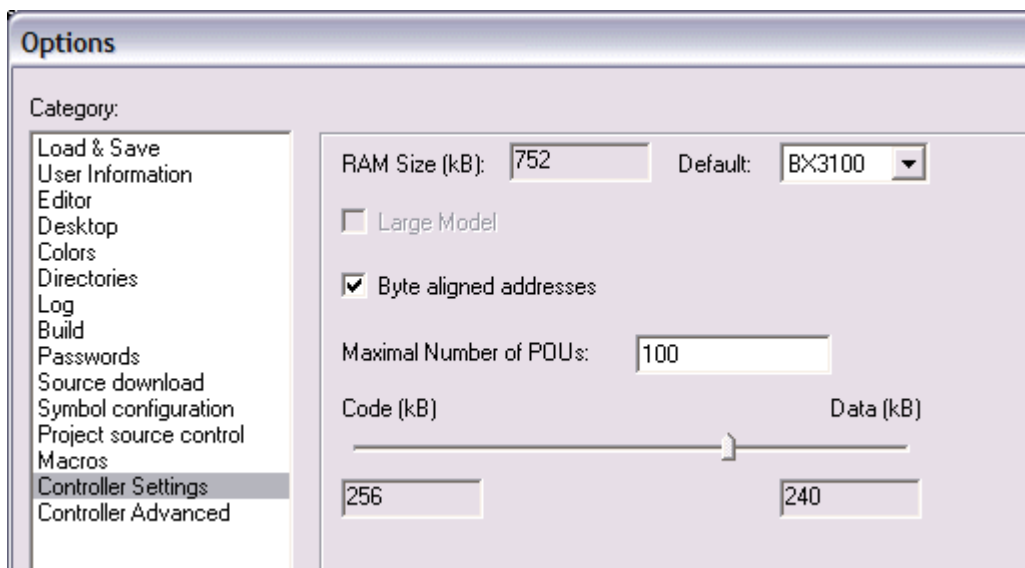


Abb. 92: Controller Settings

Eine Änderung dieser Einstellungen hat zur Folge, dass kein Online Change mehr geht.

### Fehler Globaler Speicher

```
Interface of POU 'MAIN'
Data allocation
Error 3803: MAIN (7): Out of global data memory. Variable 'Test_', 16002 bytes.
1 Error(s), 0 Warning(s).
```

Abb. 93: Globaler Speicher nicht ausreichend

Per Default sind 2 x 16 kByte Daten angelegt. Wenn viele Daten benutzt werden sollen, müssen Sie diesen Bereich vergrößern. Beim BX sind maximal 14 Datensegmente möglich.

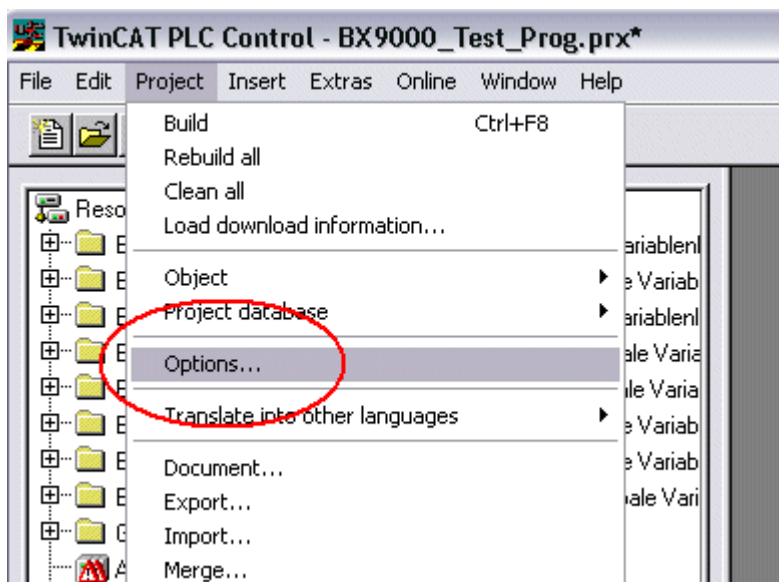


Abb. 94: Menüpfad Projekte / Optionen / Build

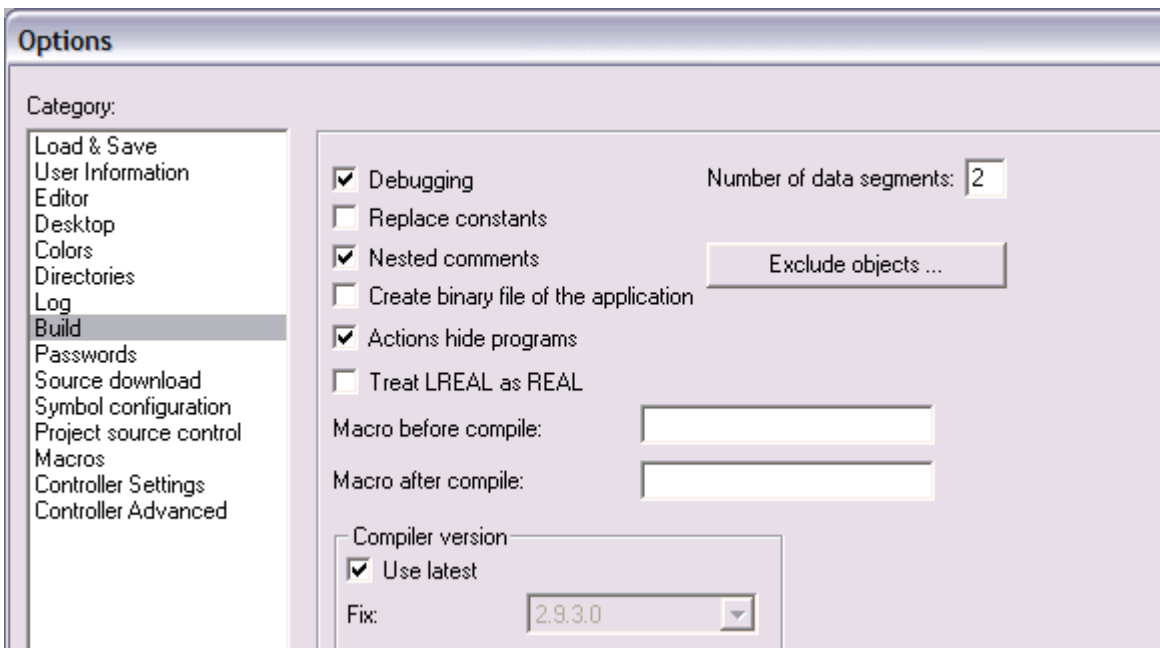


Abb. 95: Build

## 5.4 Lokierte Merker

Es stehen 4 kByte an lokierte Merker zur Verfügung. Diese können genutzt werden, um unterschiedliche Variabelentypen auf die gleiche Adresse zu legen, zum Beispiel um aus Strings Bytes zu machen. Weiter können Daten hinterlegt werden, die per ADS von der Steuerung gelesen und/oder geschrieben werden können.

### ● Lokierte Variablen sind keine remanenten Daten

**i** Bei den Busklemmen-Controller der BX-Serie und den BCxx50 werden die lokierte Variablen **nicht** als remanente Daten gespeichert.

### Read / Write lokierter Merker per ADS

Die Merker können auch über die Steuerung per ADS ausgelesen werden. Bei PROFIBUS werden dazu die DPV-1 Dienste genutzt, bei CANopen die SDO Kommunikation.

Die AmsNetId ist aus dem System Manager zu entnehmen oder man kann Sie sich in dem Menü des Busklemmen-Controllers Anzeigen lassen.

Die Port Nummer ist 800 für die PLC.

Index Group	Bedeutung	Index Offset (Wertebereich)
0x4020	Merker (nur BXxxx0)	0..4096

### Beispiel

#### BX Programm

```
VAR
    Flag_01 AT %MB0: WORD;
END_VAR
```

#### TwinCAT PC/CX Master Programm

```
VAR
    fbADRSREAD: ADSREAD;
    Flag_M: WORD;
END_VAR

fbADRSREAD (
    NETID:='172.16.3.0.2.3' , (* AMSNetId BX *)
    PORT:=800 , (* 800 - PLC *)
    IDXGRP:=16#4020 , (* 0x4020hex falgs *)
```

```

IDXOFFS:=0 ,          (* byte offset *)
LEN:=2 ,              (* Lenght byte *)
DESTADDR:=ADR(Merker) ,
READ:=TRUE ,
TMOUT:=t#1s );
IF NOT fbADRSREAD.BUSY THEN
  fbADRSREAD(READ:=FALSE);
END_IF
    
```

## 5.5 Mapping der Busklemmen

Die genaue Belegung der byteorientierten Busklemmen entnehmen Sie bitte der Konfigurations-Anleitung zur jeweiligen Busklemme. Diese Dokumentation finden Sie im Internet unter <http://www.beckhoff.de>.

byteorientierte Busklemmen	bitorientierte Busklemmen
KL15x1	KL10xx, KL11xx, KL12xx, KL17xx, KM1xxx
KL25xx	KL20xx, KL21xx, KL22xx, KL26xx, KL27xx, KM2xxx
KL3xxx	
KL4xxx	
KL5xxx	
KL6xxx	
KL7xxx	
KL8xxx	
	KL9110, KL9160, KL9210, KL9260

## 5.6 Lokales Prozessabbild im Auslieferungszustand (Default Config)

Das Prozessabbild der Busklemmen-Controller besteht aus Eingangs-, Ausgangs- und Merkerbereich. Daneben gibt es noch die unlokierten Daten. Diese Daten besitzen keine feste Adresse. Sie werden ohne Angabe einer Adresse angelegt. Für diese Art der Variablen stellt

- der BCxx50 48 kByte,
- der BC9x20, BC9191 128 kByte und
- der BXxx00 256 kByte Speicher zur Verfügung.

Eine Variable oder Struktur (Array) darf maximal 16 kByte groß sein. Für die lokierten Daten stehen 2048 Byte Ein- und 2048 Byte Ausgangsdaten bereit. Für den lokierten Merkerbereich verfügt der Busklemmen-Controller über 4 kByte Speicher.

Im Auslieferungszustand (Default Configuration) des BX/BCxx50 werden allen angeschlossenen Busklemmen feste Adressen zugewiesen. Die Daten für die Ethernet-Kommunikation beginnen ab dem Adress-Offset 1000<sub>dez</sub>. Die Länge der Ethernet-Daten hängt von der Anzahl der konfigurierten Daten ab und beträgt maximal 1000 Byte beim BX9000.

Eingänge	Ausgänge
Busklemme %IB0 ...	Busklemmen %QB0 ...
Ethernet DATEN (SPS- Variablen) %IB1000 ... (ModbusTCP/ADS-TCP/ADS-UDP)	Ethernet DATEN (SPS- Variablen) %QB1000 ... (ModbusTCP/ADS-TCP/ADS-UDP)
... %IB2047 Maximal	... %QB2047 Maximal

### Adressierung der Angeschlossenen Busklemmen

Alle angeschlossenen Busklemmen werden in der Default-Einstellung dem lokalen Prozessabbild zugewiesen. Das Mapping im Busklemmen Controller erfolgt nach folgender Gesetzmäßigkeit: Erst alle komplexen Busklemmen, in der Reihenfolge wie diese gesteckt sind und anschließend die digitalen Busklemmen, die zu einem Byte aufgefüllt werden. Das Default-Mapping der komplexen Busklemmen ist:

- komplette Auswertung
- Intel-Format
- Word Alignment

**Beispielaufbau**

Busklemmen Controller: 1 x BCxx50, BCxx20 oder BXxx00  
 Position 1: 1 x KL1012  
 Position 2: 1 x KL1104  
 Position 3: 1 x KL2012  
 Position 4: 1 x KL2034  
 Position 5: 1 x KL1501  
 Position 6: 1 x KL3002  
 Position 7: 1 x KL4002  
 Position 8: 1 x KL6001  
 Position 9: 1 x KL9010

Tab. 1: Prozessabbild

Busklemme	Position	Eingangsabbild	Ausgangsabbild	Größe
KL1501	5	%IB0...%IB5	%QB0...%QB5	6 Byte
KL3002	6	%IB6...%IB13	%QB6...%QB13	8 Byte
KL4002	7	%IB14...%IB21	%QB14...%QB21	8 Byte
KL6001	8	%IB22...%IB29	%QB22...%QB29	6 Byte
KL1012	1	%IX30.0...%IX30.1	-	2 Bit
KL1104	2	%IX30.1...%IX30.5	-	4 Bit
KL2012	3	-	%QX30.0...%IX30.1	2 Bit
KL2034	4	-	%QX30.2...%IX30.5	4 Bit
KL9010	9	-	-	-

## 5.7 Persistente Daten

Es stehen auf dem Busklemmen-Controller 1000 Byte an persistenten Daten zur Verfügung. Im Unterschied zu den Retain-Daten werden diese auch bei einem neuen Projekt, bei einem Reset der SPS sowie bei einem neuen Download nicht gelöscht.

Um die persistenten Daten zu nutzen müssen diese erst einmal mit einem Funktionsbaustein aus der SPS heraus aktiviert werden.

Zweitens müssen die Variablen auf dem lokierten Merkerbereich liegen. Hier steht es Ihnen frei wo ihre persistenten Daten liegen.

Es stehen 4 kByte an lokierten Merkern zur Verfügung und sie können davon 1000 Byte als persistente Daten deklarieren.

**Beispiel**

```
VAR
    Test AT %MX1000 :BOOL;
    Count AT %MB1002 :INT;
END_VAR
```

Mit dem Baustein **Persistent\_Data** legen sie die Anfangsadresse fest und die Länge in Byte ab der die Daten persistent sein sollen.

Mit der Eingangsvariable *WriteOffset* gibt mal den Byte Offset des Merkerbereichs an, mit *WriteSize* die Länge in Byte.

Den Baustein finden Sie in der TcSystemBX.lbx Bibliothek. Sollte diese nicht vorhanden sein, laden Sie sie sich aus dieser Dokumentation herunter (siehe [Bibliotheken](#) [► 110]).

## Beispielwerte

WriteOffset 1000  
WriteSize 10

Alle Daten die im Bereich %MB1000 - %MB1009 sind dann persistent. Es spielt keine Rolle um welchen Variablen Typ es sich handelt.

Die Daten werden wie bei den Retain Daten in NOVRAM kopiert und sind daher in jedem Zyklus beschreibbar.

### ● Persistenten Daten ab Firmware 1.17


**i** Die persistenten Daten werden bei allen BX-Controllern ab Firmware 1.17 oder höher unterstützt.

### ● Parameter sind sofort gültig

**i** Das Schreiben der Parameter muss nur einmal erfolgen und ist sofort gültig. Diese Daten werden dauerhaft gespeichert.

Das Aktivieren der Herstellereinstellung löscht dies wieder und auch die persistenten Daten werden gelöscht.

## Beispielprogramm

Klicken Sie auf den Link  (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207307659.zip>) um ein Beispielprogramm aus dieser Dokumentation herunterzuladen.

## 5.8 Remanente Daten

Auf dem BC9191 und den BX-Controller stehen 2000 Byte an remanenten Daten zur Verfügung. Diese Daten werden im PLC Control als VAR RETAIN deklariert.

### Beispiel

```
VAR RETAIN
    Test      :BOOL;
    Count     :INT;
END_VAR
```

Zwischen VAR RETAIN und END\_VAR stehen die Retain Daten. Diese Daten werden in einem NOVRAM gespeichert und sind über den ganzen 2 kByte großen Bereich konsistent. Die RETAIN Daten werden nach jedem Zyklus ins NOVRAM gespeichert. Für 2 kByte werden ca. 2 ms benötigt (für 1 kByte ca. 1 ms). Die Variablen können lokal oder global Konfiguriert sein. Variablen die lokiert sind (%MB, %QB, %IB) können nicht als Remanente Daten genutzt werden.

### ● VAR\_RETAIN nicht in Funktionsbausteinen benutzen

**i** VAR\_RETAIN sollte nicht in Funktionsbausteinen benutzt werden. Sämtliche Daten in einem FB werden in den Retain Speicher kopiert, damit erhöht sich unnötig die Zykluszeit und der Retain Speicher wird mit unnötigen Datenmengen gefüllt.

### ● Variablen mit Adresse nicht als remanente Daten verwenden

**i** Variablen die auf einer Adresse liegen (%MB, %QB, %IB) dürfen nicht als remanente Daten verwendet werden.

## Beispiel für remanente Daten im Funktionsbaustein

Da immer alle Daten eines Funktionsbausteins, in dem auch nur ein remanentes Bit zu finden ist, gespeichert wird, sollte dies möglichst vermieden werden. Im Anschluss finden Sie ein Programmbeispiel.

**Funktionsbaustein Test (Kein Programm Code notwendig - in ST reicht ein Semikolon)**

```
FUNCTION_BLOCK Test
VAR_INPUT
END_VAR
VAR_OUTPUT
END_VAR
VAR
END_VAR
VAR_IN_OUT
    Counter    :INT;
END_VAR
```

**Programm MAIN**

```
PROGRAM MAIN
VAR
    fb_Test:Test;
END_VAR
VAR_RETAIN
    iCounter1:INT;
END_VAR

fb_Test(Counter:=iCounter1);
```

**5.9 Lokales Prozessabbild in der TwinCAT-Konfiguration**

Die TwinCAT-Konfiguration (TwinCAT-CONFIG) ermöglicht das freie Mapping zwischen Feldbus, K-Bus und SPS-Variablen. Variablen können unabhängig von ihrer Adresse mit Hilfe des System Managers verknüpft werden.

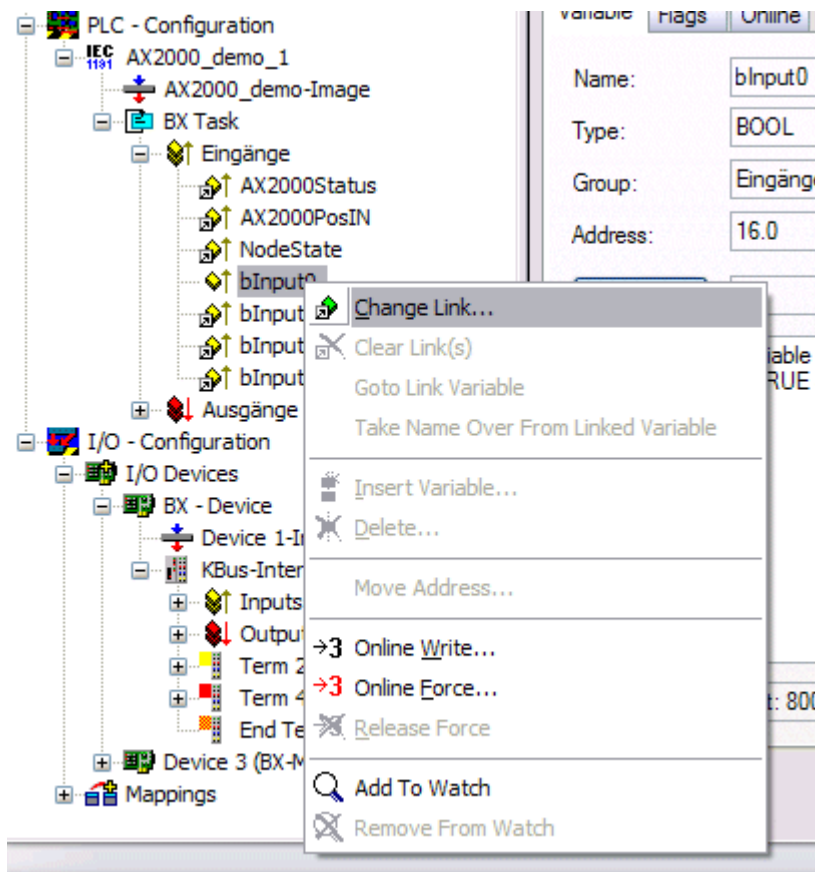


Abb. 96: Ändern der Verknüpfung von Variablen

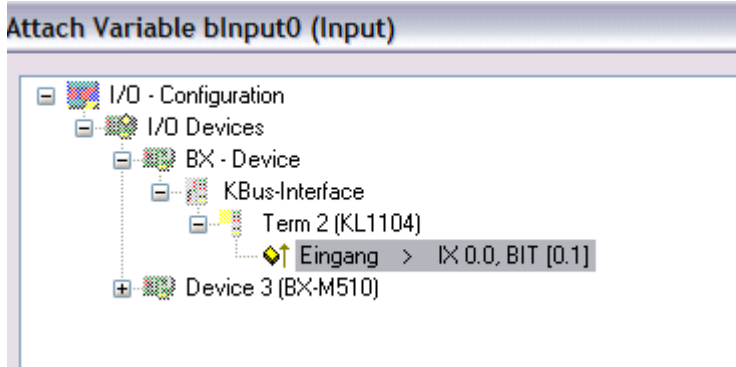


Abb. 97: Verknüpfen einer Variable mit einem Eingang

In der Default-Konfiguration liegen alle Busklemmen auf feste Adressen. Fügt man eine Busklemme ein, verschiebt sich eventuell der ganze Adressbereich. Mit der TwinCAT-Konfiguration hat man die Möglichkeit, seine lokierten Variablen, frei mit einer Busklemme zu verknüpfen. Man parametrisiert dies im System Manager und lädt dann diese erstellte Konfiguration auf den Busklemmen-Controller herunter (siehe [TwinCAT-Konfiguration \[► 33\]](#)). Ein Hochladen einer bestehenden TwinCAT-Konfiguration ist auch möglich.

## 5.10 Erzeugen eines Boot-Projekts

Für das Erzeugen des Boot-Projektes stehen

- auf den Busklemmen-Controllern der BX-Serie ca. 250 kByte Flash zu Verfügung.
- auf den Busklemmen-Controllern der BCxx50-Serie ca. 48 kByte Flash zu Verfügung.

### PLC Control

Im TwinCAT PLC Control kann man, wenn man eingeloggt ist ein Boot-Projekt erzeugen.

- Öffnen eines PLC Projektes
- Auswahl des Zielsystem (oder Auswahl der seriellen Schnittstelle)
- Einloggen auf den BX/BCxx50
- Erzeugen des Boot-Projektes (Online\Create Bootproject)

Ist ein gültiges Boot-Projekt auf dem BX/BCxx50 leuchtet die LED PLC grün.

Bei den Busklemmen-Controllern der BX-Serie blinkt während der Erzeugung des Boot-Projektes die die PLC LED orange. Ist auf dem BX kein Boot-Projekt vorhanden, leuchtet die PLC LED orange.

### Löschen eines Boot-Projektes

Sie können das Boot-Projekt auch vom Busklemmen-Controller löschen. Folgende Schritte sind einzuhalten:

- Öffnen des Projektes
- Einloggen auf den Busklemmen-Controller
- Löschen des Boot-Projektes (Online>Delete Boot Project)

Nach dem Löschen des Boot-Projektes ist die PLC LED orange.

### **i** Übernahme des aktuellen Projektes als Boot-Projekt

Nach einem Online Change ist als Boot-Projekt noch immer das alte Projekt eingetragen. Soll das aktuelle Projekt (nach dem Online-Change) als Boot-Projekt übernommen werden, muss dieses neu erzeugt werden.



**Umgehen eines Starten des Boot-Projekt\***

Bei den Busklemmen-Controllern der BX-Serie kann beim Booten das Starten des Boot-Projekts durch Drücken des Navi-Schalters verhindert werden. Das Boot-Projekt ist damit nicht gelöscht und wird beim erneuten Booten des Busklemmen-Controller wieder gestartet.

\* ab Version 0.85

**5.11 Kommunikation zwischen TwinCAT und BX/BCxx50**

Um von TwinCAT-Daten zum Busklemmen-Controller zu transportieren liegt es nahe, die Daten in einer Struktur anzulegen. Da die Datenhaltung auf beiden Systemen unterschiedlich ist sind folgende Hinweise zu beachten.

- Wenn zwei unterschiedliche Datentypen aufeinander folgen (zum Beispiel Byte und INT) wird die folgende Variable auf den nächsten graden Adress-Offset gelegt
- Boolsche Variablen sollten nie einzeln in eine Struktur gelegt werden, da sie so immer 1 Byte belegen würden. Boolsche Ausdrücke sollten immer in ein Byte oder Wort maskiert sein.

**Beispiel 1: Eine Struktur auf den BX/BCxx50 und auf dem PC**

Variable	Speicher des BX/BCxx50	Speicher des PC (TwinCAT)
Byte	%..B0	%..B0
INT (1)	%..B2	%..B1
INT (2)	%..B4	%..B3

Dadurch das hinter dem ersten Byte eine anderer Variable-Typ (INT) folgt ist dieser im BX/BCxx50 auf die nächste freie grade Adresse gelegt worden. Will man beide auf beiden Systemen die gleiche Datenstruktur haben, muss im PC-Projekt ein Dummy-Byte eingefügt werden (siehe Beispiel 2).

**Beispiel 2: Eine Struktur auf den BX/BCxx50 und auf dem PC mit gleicher Speicherbelegung**

Variable	Speicher des BX/BCxx50	Speicher des PC (TwinCAT)
Byte	%..B0	%..B0
Byte (Dummy)	%..B1 (nicht unbedingt notwendig, da dies das System selber macht, wenn diese Variabel nicht vorhanden ist)	%..B1
INT (1)	%..B2	%..B2
INT (2)	%..B4	%..B4

**Daten Struktur**

```
Type PB_Data
STRUCT
  wVar_1:WORD;
  iValue_1:INT;
  iValue_2:INT;
  iValue_3:INT;
END_STRUCT
END_TYPE
```

**Anlegen einer Struktur in den Variablen**

```
VAR_Global
  strData_Out AT %QB1000:PB_Data; (*PLC Variables *)
  bInput_01 AT %IX0.0:BOOL; (* Input from a terminal *)
END_VAR
```

**Kleines Programmbeispiel**

```
strData_Out.wVar_1.0:=bInput_01;
```

**● Keine Real-Werte in gemischter Datenstruktur verwenden**

**i** In einer gemischten Datenstruktur sollten keine Real-Werte enthalten sein. Wenn dies doch der Fall ist, muss zusätzlich im BX/BCxx50 oder im TwinCAT-Masterprojekt das High und Low Word vertauscht werden. Verwenden Sie besser ein Array von Real-Werten oder übertragen Sie die Real-Werte einzeln.

**● Größere Feldbusdatenblöcke**

**i** Sie können auch größere Feldbusdatenblöcke übertragen, um eine Reserve für Ihre Struktur zu haben. Nachteil: Diese Reserven werden dann mit jedem Feldbustelegramm übertragen, was eine Mehrbelastung der Feldbuskommunikation verursacht.

## 5.12 Up- und Download von Programmen

Der Busklemmen-Controller verfügt über einen Speicher für den Quell-Code. Hier können das Programm, die Task-Konfiguration und die Bibliotheken abgespeichert werden. Sollte Speicher für den Quell-Code nicht ausreichen, kann man auch nur den Quell-Code ohne Task-Konfiguration und die Bibliotheken ablegen. Dies benötigt wesentlich weniger Speicherplatz!

### Allgemeine Einstellungen

Unter Bearbeiten/Optionen kann eingestellt werden wann der Quell-Code zum Zielsystem heruntergeladen werden soll. Öffnen Sie das Optionsmenü.

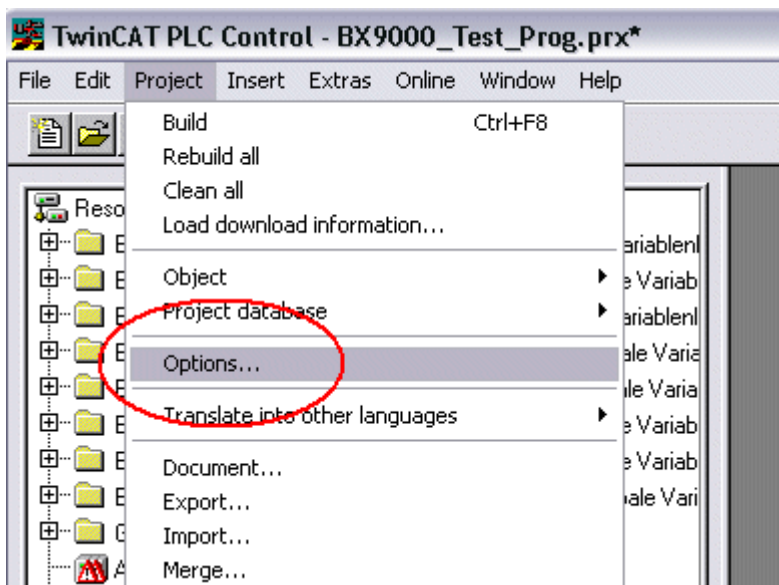


Abb. 98: Öffnen des Optionsmenüs

Wählen Sie nun den Source Download an.

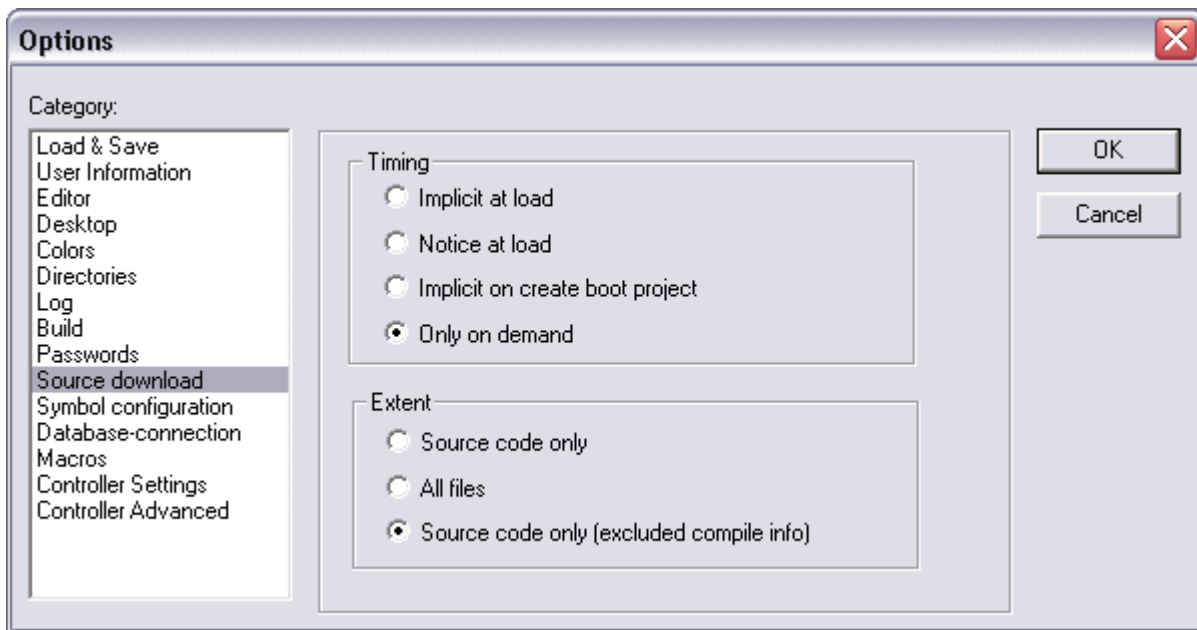


Abb. 99: Auswahl des Source Downloads

Hier können Sie einstellen, wann und was vom Source-Code zum Busklemmen-Controller heruntergeladen werden soll.

**Source code only:** es wird das prx-File mit Informationen zum online Change übertragen. Damit ist ein Einloggen per Online-Change möglich (die SPS stoppt nicht).

**All files:** Wie *Source code only* plus alle notwendigen Bibliotheken.

**Source code only (excluded compile info):** es wird nur das prx File übertragen. Ein Einloggen ist nur möglich, wenn die SPS stoppt.

Welche Option Sie verwenden können, hängt von der Größe Ihrer Projekte ab.

### Download eines Programms

Der Quell-Code kann man auf Anforderung zum Zielsystem übertragen. Dafür muss man mit seinem Programm eingeloggt sein. Unter Online/Quell-Code Download kann jetzt der Programm-Code zum Busklemmen-Controller übertragen werden.

Online	Window	Help
Login		F11
Logout		F12
Download		
Run		F5
Stop		Shift+F8
Reset		
Reset All		
Toggle Breakpoint		
Breakpoint Dialog		F9
Step over		F10
Step in		F8
Single Cycle		Ctrl+F5
Write Values		
Force Values		Ctrl+F7
Release Force		F7
Write/Force-Dialog		Shift+F7
Write/Force-Dialog		
Write/Force-Dialog		Ctrl+Shift+F7
Show Call Stack...		
Display Flow Control		Ctrl+F11
Simulation Mode		
Communication Parameters...		
<b>Sourcecode download</b>		
Choose Run-Time System...		
Create Bootproject		
Create Bootproject (offline)		
Delete Bootproject		

Abb. 100: Download des Programm Codes

Nach einer kurzen Zeit öffnet sich ein Fenster, das den Vorschritt des Downloads anzeigt.

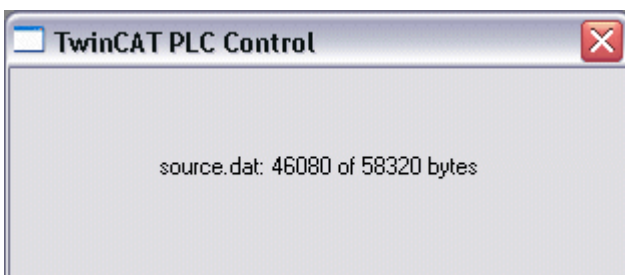


Abb. 101: Vorschritt des Downloads

### Upload eines Programms

Um den Programm-Code wieder hoch zu laden, öffnen Sie im PLC Control ein neues File. Als nächstes klicken Sie auf die Schaltfläche PLC.

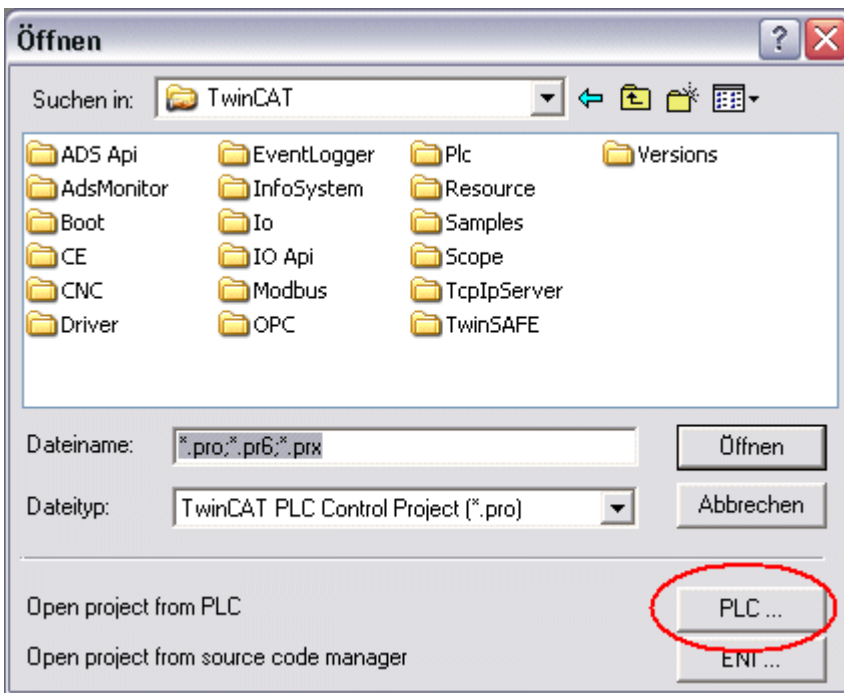


Abb. 102: Upload eines Programms

Wählen Sie den Datenübertragungsweg aus:

- *BCxx50 or BX via AMS*, wenn sie über den Feldbus mit dem Busklemmen-Controller verbunden sind oder
- *BCxx50 or BX via serial*, wenn sie über die serielle Schnittstelle mit dem Busklemmen-Controller verbunden sind.

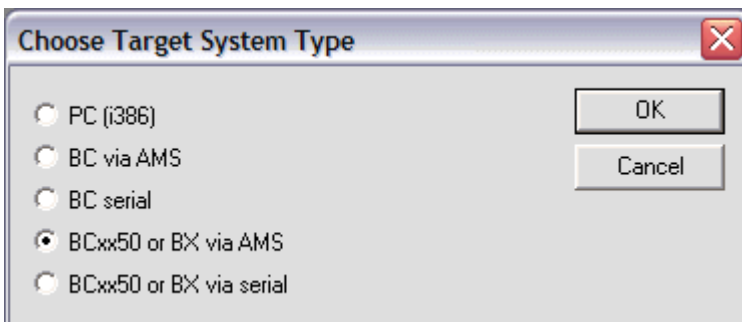


Abb. 103: Auswahl des Datenübertragungswegs

Als nächsten Schritt wählen Sie das Gerät aus und bestätigen Sie mit OK.

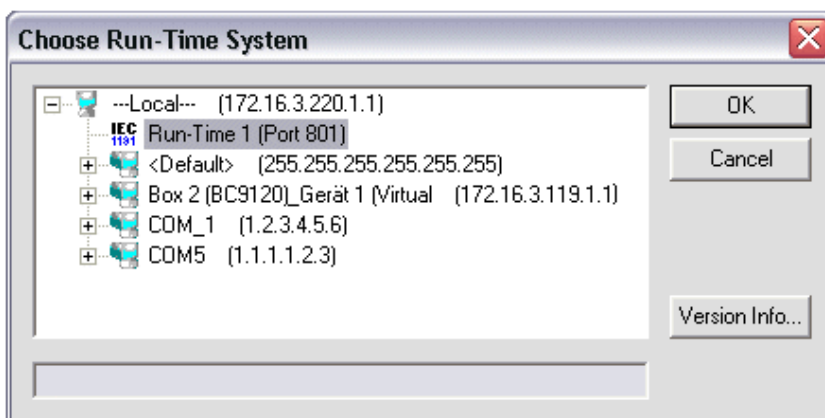


Abb. 104: Auswahl des Gerätes

Der Quell-Code wird nun hochgeladen.

**Password**

Mit einem Passwort können Sie Ihr Projekt schützen (im PLC Control Projekt/Optionen/Kennworte).






## 5.13 Bibliotheken

### 5.13.1 Bibliotheken - Übersicht

Für die Busklemmen Controller (Buskoppler mit SPS-Funktionalität, Bezeichnung BXxxxx) gibt es verschiedene Bibliotheken ([siehe Beckhoff Information System](#)).

**Download**

Zum Download der Bibliotheken bitte auf den Link mit der linken Maustaste klicken. Bitte die Bibliotheken in das Verzeichnis TwinCAT\PLC\LIB kopieren.

- Standard (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207309835.zip>)  

- TcSystemBX (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207312011.zip>)  
 (die TcSystemBX benötigt die TcBaseBX Bibliothek)
- TcBaseBX (Download)  
 (die TcDisplayBX, TcNaciSwitchBX und TcDebugBX sind nun hier enthalten)
- TcComPortBX (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207314187.zip>)  

- ChrAscBX.lbx (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207316363.zip>)  


**● Verwenden Sie die zur Firmware passende Bibliothek**

**i** Für die neuste Firmware ist auch die neuste Bibliothek zu benutzen. Sollten Sie Ihren BX-Controller Updaten tauschen Sie auch die Bibliotheken. Kopieren Sie diese Bibliotheken in das LIB-Verzeichnis, entfernen diese Bibliotheken aus Ihrem Projekt und fügen Sie sie erneut hinzu.

**TcSystemBX**

ADS	Version	Firmware				
		BX3100	BX5100	BX5200	BX8000	BX9000
ADSREAD	04.03.04	0.90	0.14	1.00	0.02	1.00
ADSWRITE	04.03.04	0.90	0.14	1.00	0.02	1.00
ADSRDWR	04.03.04	0.90	0.14	1.00	0.02	1.00
ADSWRTCTL	04.03.04	0.90	0.14	1.00	0.02	1.00
ADSRDSTATE	04.03.04	0.90	0.14	1.00	0.02	1.00
ADSRDDEVINFO	04.03.04	0.90	0.14	1.00	0.02	1.00

Bit Functions	Version	Firmware				
		BX3100	BX5100	BX5200	BX8000	BX9000
CLEARBIT32	07.03.03	0.28	0.01	1.00	0.01	1.00
CSETBIT32	07.03.03	0.28	0.01	1.00	0.01	1.00
GETBIT32	07.03.03	0.28	0.01	1.00	0.01	1.00
SETBIT32	07.03.03	0.28	0.01	1.00	0.01	1.00

Display Function	Version	Firmware				
		BX3100	BX5100	BX5200	BX8000	BX9000
FB_DisbWrite [▶ 123]	31.03.03	0.28	0.01	1.00	0.01	1.00

Diagnose	Version	Firmware				
		BX3100	BX5100	BX5200	BX8000	BX9000
BX_Security	15.08.06	1.12	1.14	-	1.12	1.14
DeviceTyp	15.08.06	1.12	1.14	-	1.12	1.14
FirmwareVersion	15.08.06	1.12	1.14	-	1.12	1.14
FirmwareVersionSt ring	15.08.06	1.12	1.14	-	1.12	1.14
DeviceTyp	15.08.06	1.12	1.14	-	1.12	1.14
Read_Diagnose	15.08.06	1.12	1.14	-	1.12	1.14
CRCBootproject	15.08.06	1.14	1.14	-	1.14	1.14

Read Address	Version	Firmware				
		BX3100	BX5100	BX5200	BX8000	BX9000
ReadSlaveAddress	15.08.06	1.12	1.12	1.10	1.12	-

Controller	Version	Firmware				
		BX3100	BX5100	BX5200	BX8000	BX9000
FB_BasicPID	04.03.04	0.64	0.01	1.00	0.01	1.00

Event Logger Functions	Version	Firmware				
		BX3100	BX5100	BX5200	BX8000	BX9000
-	-	-	-	-	-	-

File Access	Version	Firmware				
		BX3100	BX5100	BX5200	BX8000	BX9000
FB_ReadFromFile	03.08.04	1.04	1.04	1.00	1.04	1.00
FB_WriteToFile	03.08.04	1.04	1.04	1.00	1.04	1.00
FB_ReadWriteFile	03.08.04	1.04	1.04	1.00	1.04	1.00

Memory Functions	Version	Firmware				
		BX3100	BX5100	BX5200	BX8000	BX9000
MEMCMP	07.03.03	0.41	0.01	1.00	0.01	1.00
MEMCYP	07.03.03	0.41	0.01	1.00	0.01	1.00
MEMMOVE	07.03.03	0.41	0.01	1.00	0.01	1.00
MEMSET	07.03.03	0.41	0.01	1.00	0.01	1.00

NOVRAM Functions	Version	Firmware				
		BX3100	BX5100	BX5200	BX8000	BX9000
-	-	-	-	-	-	-

Serial Communication Interface	Version	Firmware				
		BX3100	BX5100	BX5200	BX8000	BX9000
FB_COMPortClose	14.07.03	0.49	0.01	1.00	0.01	1.00
FB_COMPortOpen	14.07.03	0.49	0.01	1.00	0.01	1.00
F_COMPortRead	14.07.03	0.49	0.01	1.00	0.01	1.00
F_COMPortWrite	14.07.03	0.49	0.01	1.00	0.01	1.00

SFC	Version	Firmware				
		BX3100	BX5100	BX5200	BX8000	BX9000
AnalyzeExpression	07.03.03	0.28	0.01	1.00	0.01	1.00
AppendErrorString	07.03.03	0.28	0.01	1.00	0.01	1.00
SFCActionControl	07.03.03	0.28	0.01	1.00	0.01	1.00

System / Time / TBus	Version	Firmware				
		BX3100	BX5100	BX5200	BX8000	BX9000
DRAND	07.03.03	0.28	0.01	1.00	0.01	1.00
RTC	07.03.03	0.28	0.01	1.00	0.01	1.00
SYSTEMTIME_TO_DT	07.03.03	0.28	0.01	1.00	0.01	1.00
DT_TO_SYSTEMTIME	07.03.03	0.28	0.01	1.00	0.01	1.00
GetSysTick	14.07.03	0.49	0.01	1.00	0.01	1.00
PresetSysTick	14.07.03	0.49	0.01	1.00	0.01	1.00
Reboot	21.07.03	0.59	0.14	1.00	0.02	1.00
Persistent_Data	21.08.07	1.17	1.17	-	1.17	1.17

Debug	Version	Firmware				
		BX3100	BX5100	BX5200	BX8000	BX9000
<a href="#">F_ReadDebugTimer</a> [▶ 122]	08.08.03	0.59	0.14	1.00	0.02	1.00
<a href="#">F_StartDebugTimer</a> [▶ 122]	08.08.03	0.59	0.14	1.00	0.02	1.00

NaviSwitch	Version	Firmware				
		BX3100	BX5100	BX5200	BX8000	BX9000
<a href="#">Alle Bausteine</a> [▶ 122]	10.10.03	0.64	0.14	1.00	0.02	1.00

**TcComPortBX**

Com FBs	Version	Firmware				
		BX3100	BX5100	BX5200	BX8000	BX9000
<a href="#">Alle Bausteine</a> [▶ 128]	20.08.03	0.60	0.02	1.00	0.01	1.00



## 5.13.2 TcBaseBX

### 5.13.2.1 System Task Info

```
VAR_GLOBAL
    SystemTaskInfoArr : ARRAY[1..2] OF SYSTEMTASKINFOTYPE;
END_VAR
```

Systemflags sind implizit deklarierte Variablen. Mit der Eingabehilfe finden Sie unter Systemvariablen eine Variable SystemTaskInfoArr. Diese Variable ist ein Feld von vier Strukturen des Types `SYSTEMTASKINFOTYPE` [► 113]. Die Strukturdefinition ist in der System-Bibliothek zu finden. Der Index in dieses Feld ist die Task-Id.

Entwicklungsumgebung	Zielformat	Einzubindende SPS Bibliotheken
TwinCAT v2.9.0	BX Controller	TcBaseBX.lbx

### 5.13.2.2 System Task Info Type

```
TYPE SYSTEMTASKINFOTYPE
STRUCT
    active : BOOL;
    taskName : STRING(16);
    firstCycle : BOOL;
    cycleTimeExceeded : BOOL;
    cycleTime : UDINT;
    lastExecTime : UDINT;
    priority : BYTE;
    cycleCount : UDINT;
END_STRUCT
END_TYPE
```

#### Legende

active: Diese Variable zeigt an, ob die Task aktiv ist.

taskName: Der Taskname.

firstCycle: Diese Variable hat im ersten Zyklus der SPS-Task den Wert: TRUE.

cycleTimeExceeded: In dieser Variablen wird ein Überschreiten der eingestellten Taskzykluszeit gemeldet.

cycleTime :Eingestellte Taskzykluszeit in Vielfachen von 100 ns.

lastExecTime: Benötigte Zykluszeit für den letzten Zyklus in Vielfachen von 100 ns.

priority: Eingestellte Priorität der Task.

cycleCount: Zykluszähler.

Entwicklungsumgebung	Zielformat	Einzubindende SPS Bibliotheken
TwinCAT v2.9.0	BX Controller	TcBaseBX.lbx

### 5.13.2.3 System Info

```
VAR_GLOBAL
    SystemInfo : SYSTEMINFOTYPE;
END_VAR
```

Systemflags sind implizit deklarierte Variablen. Mit der Eingabehilfe finden Sie unter Systemvariablen eine Variable SystemInfo. Der Typ `SYSTEMINFOTYPE` [► 113] ist in der System-Bibliothek deklariert. Um auf die Variable zugreifen zu können muss die System-Bibliothek in das Projekt eingebunden werden.

Entwicklungsumgebung	Zielformat	Einzubindende SPS Bibliotheken
TwinCAT v2.9.0	BX Controller	TcBaseBX.lbx

### 5.13.2.4 System Info Type

```
TYPE SYSTEMINFOTYPE
STRUCT
    runTimeNo : BYTE;
    projectName : STRING(32);
END_STRUCT
```

```

    numberOfTasks      :      BYTE;
    onlineChangeCount  :      UINT;
    bootDataFlags      :      BYTE;
    systemStateFlags   :      WORD;
END_STRUCT
END_TYPE

```

**Legende**

runTimeNo: Gibt die Nummer des Laufzeitsystems (1) an.  
 projectName: Name des Projekts als STRING.  
 numberOfTasks: Anzahl der im Laufzeitsystem befindlichen Tasks (max. 2).  
 onlineChangeCount: Anzahl der seit dem letzten Komplettdownload gemachten Online-Änderungen.  
 bootDataFlags: Reserviert  
 systemStateFlags: Reserviert.

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.9.0	BX Controller	TcBaseBX.lbx

**5.13.2.5 ADS**

**5.13.2.5.1 Lokale ADS-Port Nummern - Übersicht**

Port Nummer	Beschreibung
100 [ <a href="#">▶ 114</a> ] <sub>dez</sub>	Lesen und schreiben von Registern und Tabellen aus den Koppler und den komplexen Busklemmen
150 [ <a href="#">▶ 87</a> ] <sub>dez</sub>	Lesen und schreiben der RTC (Echt-Zeit-Uhr)
153 [ <a href="#">▶ 68</a> ] <sub>dez</sub>	SSB - auslesen der Emergency-Nachricht
800 [ <a href="#">▶ 114</a> ] <sub>dez</sub>	Lokales Prozessabbild der PLC, siehe auch Port 801
801 [ <a href="#">▶ 114</a> ] <sub>dez</sub>	Lokales Prozessabbild der PLC, siehe auch Port 800
0x1000 + Node ID <a href="#">▶ 67</a>	SSB - SDO Kommunikation mit CANopen-Knoten (Slave Nummer)

**5.13.2.5.2 ADS-Dienste**

**Lokales Prozessabbild PLC Task 1 Port 800/801**

Im lokalen Prozessabbild können Daten gelesen und geschrieben werden. Sollten Ausgänge geschrieben werden muss darauf geachtet werden, das diese von der lokalen SPS nicht verwendet werden, da die lokale Steuerung diese Werte überschreibt. Die Daten sind nicht an einen Watchdog gebunden und müssen und dürfen daher nicht für Ausgänge verwendet werden, die im Fehlerfall ausgeschaltet werden müssen.

Index Group	Bedeutung	Index Offset (Wertebereich)
0xF020	Input - Eingänge	0...2047
0xF021	Input - Eingänge Bit	0...16376
0xF030	Output - Ausgänge	0...2047
0xF031	Output - Ausgänge Bit	0...16376
0x4020	Merker	0...4095
0x4021	Merker Bit	0...32760

**Dienste des ADS**

**AdsServerAdsState**

Datentyp (only Read)	Bedeutung
String	Start - die lokale PLC läuft Stop - die lokale PLC ist im Stop

**AdsServerDeviceState**

Datentyp (only Read)	Bedeutung
INT	0 - Start - die lokale PLC läuft 1 - Stop - die lokale PLC ist im Stop

**AdsServerType**

Datentyp (only Read)	Bedeutung
String	BX PLC Server

**ADSWriteControl**

Datentyp (write only)	Bedeutung
NetID	Net ID des Ethernet Controllers*
Port	800
ADSSTATE	5 - RUN / 6 - STOP
DEVSTATE	0
LEN	0
SRCADDR	0
WRITE	positive Flanke startet den Baustein
TMOUT	zum Beispiel: t#1000 ms

\* BC9050, BC9020, BC9120, BX9000

**Registerzugriff Port 100**

Die ADS-Portnummer ist bei den Busklemmen-Controllern der BX-Serie und den BCxx50/xx20 für die Register-Kommunikation fest vorgegeben und beträgt 100.

Index Group	Index Offset (Wertebereich)		Bedeutung
	Hi-Word	Lo-Word	
0 [READ ONLY]	0...127	0...255	Register des Buskopplers Hi-Word Tabellenummer des Buskopplers Lo-Word Registernummer der Tabelle
1...255	0...3	1...255	Register der Busklemmen Hi-Word Kanalnummer Lo-Word Registernummer der Busklemme

● **Minimaler Time-Out**

**i** Beachten Sie beim Lesen der Register, dass der Time-Out beim ADS-Baustein auf eine Zeit größer eine Sekunde eingestellt wird.

● **Passwort setzen**

**i** Beachten Sie beim Schreiben auf die Register, dass das Passwort gesetzt wird (siehe Dokumentation zur entsprechenden Busklemme).

### 5.13.2.5.3 Deaktivieren der LED für Zykluszeit-Überschreitung

Der BX-Controller überwacht die eingestellte Task-Zykluszeit. Wenn diese überschritten wird, wird das Bit `cycleTimeExceeded` [► 113] und die rote LED *PLC* gesetzt. In einigen Applikationen kann es zu kurzzeitigen, tolerierbaren Überschreitungen kommen. Zum Beispiel, wenn bei der seriellen Kommunikation, viele Daten empfangen werden. Um das Flackern der roten LED *PLC* zu verhindern, kann diese mit einem `ADSWRITE` abgeschaltet werden.

#### Aufbau des Befehls `ADSWRITE`

Mit dem Befehl `ADSWRITE` ist es möglich, die rote LED *PLC* des BX-Controllers zu deaktivieren.

Eingangsparameter	Beschreibung
NETID	lokale NetId des BX
Port Nummer	800
IDXGRP	16#0000_4080
IDXOFFS	0
LEN	1 Bytes
SRCADDR	Pointer auf 1 Byte 0: rote LED ON 1: rote LED OFF

### 5.13.2.6 COM Port

#### 5.13.2.6.1 Com Port - Übersicht

Die Bibliothek beinhaltet Funktionsbausteine die einen Datenaustausch zwischen dem **BXxxxx**-Bus-Controller und einem Remote-Partner ermöglichen. Der maximale COM-Puffer beträgt 512 Byte für beide Richtungen.

#### Funktionsbausteine

Name	Beschreibung
<code>FB_ComPortOpen</code> [► 118]	Eine serielle Verbindung zu einem Partner öffnen.
<code>FB_ComPortClose</code> [► 118]	Eine serielle Verbindung zu einem Partner schließen.

#### Funktionen

Name	Beschreibung
<code>F_ComPortRead</code> [► 117]	Daten aus dem COM-Puffer lesen
<code>F_ComPortWrite</code> [► 117]	Daten in COM-Puffer schreiben

**Unterstützte Baudraten**

Baudrate [Baud]	COM 1	COM 2
300	NEIN	JA
600	NEIN	JA
1200	NEIN	JA
2400	NEIN	JA
4800	NEIN	JA
9600	JA	JA
19200	JA	JA
38400	JA	JA
57600	JA	JA
115200	NEIN	JA

Weitere hilfreiche Bausteine finden Sie in der [TcComPortBX.lbx](#) [► 127].

- Baustein für die Nutzung der ComLib, ModbusRTU etc. für die BX Com Ports
- Baustein für die Kommunikation mit den BK8x00 Buskopplern
- Baustein für die Emulation eines BK8x00 Slaves

**5.13.2.6.2 COM Port - Funktionen**

**COM Port Read**

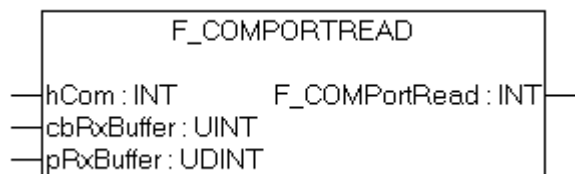


Abb. 105: Funktionsbaustein F\_COMPORTRREAD

**FUNCTION F\_COMPORTRREAD**

**VAR\_INPUT**

```

hCom      :INT;
cbRxBuffer :UINT;
pRxBuffer :UDINT;
    
```

**Legende**

hCom: wir mit dem iHandle des FB\_COMPORTOPEN verbunden

cbRxBuffer: Maximale Länge der Daten, die gelesen werden können.

pRxBuffer: Pointer auf die Daten, die mit dem COM Puffer Inhalt geschrieben werden sollen.

Rückgabewert	Bedeutung
> 0	Anzahl der Bytes, die vom COM Puffer in die PLC kopiert wurden
0x8000	Speicherüberlauf

**COM Port Write**

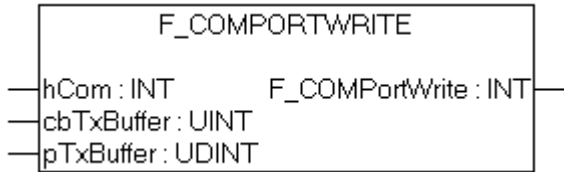


Abb. 106: Funktionsbaustein F\_COMPORTWRITE

**FUNCTION F\_COMPORTWRITE**

**VAR\_INPUT**

```
hCom      :INT;
cbTxBuffer :UINT;
pTxBuffer :UDINT;
```

**Legende**

hCom: wir mit dem iHandle des FB\_COMPORTOPEN verbunden

cbTxBuffer: Anzahl der Datenbytes, die in den COM Puffer kopiert wurden.

pTxBuffer: Pointer auf die Daten, aus denen der COM Puffer gefüllt werden soll.

Rückgabewert	Bedeutung
> 0	Anzahl der Bytes, die vom der PLC in den COM Puffer kopiert wurden
0x8000	Speicherüberlauf

**5.13.2.6.3 COM Port - Funktionsbaustein**

**COM Port Open**

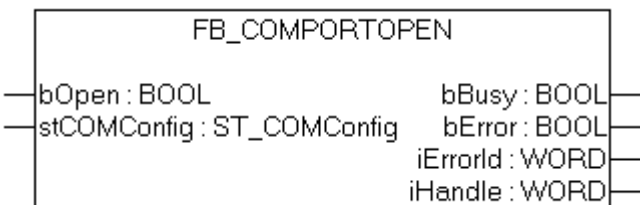


Abb. 107: Funktionsbaustein FB\_COMPORTOPEN

**FUNCTION\_BLOCK FB\_COMPORTOPEN**

**VAR\_INPUT**

```
bOpen      :BOOL;
stComConfig :ST_COMCONFIG;
```

**Legende**

bOpen: Positive Flanke startet den Baustein

stComConfig [▶ 120]: Datenstruktur COM Schnittstelle.

**VAR\_OUTPUT**

```
bBusy      :BOOL;
bErr       :BOOL;
iErrId     :WORD;
iHandle    :WORD;
```

**Legende**

bBusy: So lange der Baustein TRUE ist der Baustein aktiv.  
 bErr: Fehler Bit.  
 iErrId: Fehler Nummer.  
 iHandle: Pointerübergabe.

Rückgabeparameter iErrId	Bedeutung
0	kein Fehler
-1, 0xFFFF	Falscher COM-Port
-2, 0xFFFFE	Fehlerhafte oder nicht unterstützte Baudrate. Überprüfen Sie den Parameter stComConfig.BaudRate.
-3, 0xFFFFD	Fehlerhaftes oder falsches Datenformat. Überprüfen Sie den Parameter stComConfig.
-4, 0xFFFFC	Fehlerhaft Initialisierung der COM-Schnittstelle
-5, 0xFFFFB	Nicht unterstützter Instanz
-6, 0xFFFFA	Falsche Größe des RX-Buffers
-7, 0xFFFF9	Falsche Größe des TX-Buffers
-8, 0xFFFF8	COM-Port ist gesperrt

**COM Port Close**

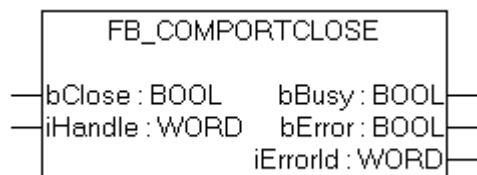


Abb. 108: Funktionsbaustein FB\_COMPORTCLOSE

**FUNCTION\_BLOCK FB\_COMPORTCLOSE**

**VAR\_INPUT**

```
bOpen      :BOOL;
iHandle    :WORD;
```

**Legende**

bClose: Positive Flanke startet den Baustein  
 iHandle: Pointerübergabe von FB\_COMPORTOPEN.

**VAR\_OUTPUT**

```
bBusy      :BOOL;
bErr       :BOOL;
iErrId     :WORD;
```

**Legende**

bBusy: So lange der Baustein TRUE ist der Baustein aktiv.  
 bErr: Fehler Bit.  
 iErrId: Fehler Nummer.

Rückgabeparameter iErrId	Bedeutung
0	kein Fehler
> 0	Fehler Nummer (#nicht dokumentiert#)

### 5.13.2.6.4 Datenstruktur ComConfig

Die Einstellungen für die seriellen Schnittstellen des BX werden mit der folgenden Datenstruktur übergeben.

```

TYPE ST_COMConfig:
STRUCT
    cbRxBufferLen    :WORD;
    cbTxBufferLen    :WORD;
    dwMode           :DWORD;
    BaudRate         :DWORD;
    eCommPort        :E_CommPort;
    eDataBits        :E_DataBits;
    eParity          :E_Parity;
    eStoppBits       :E_StoppBits;
END_STRUCT
END_TYPE
    
```

#### Legende

- cbRxBufferLen: hat keine Bedeutung (ist aus kompatibilitäts- Gründen beibehalten worden)
- cbTxBufferLen: hat keine Bedeutung (ist aus kompatibilitäts- Gründen beibehalten worden)
- dwMode: Daten Mode COM 1 nur "0" - COM 2 RS232 "0" und RS485 "1"
- BaudRate: Baudrate
- eCommPort: Com Port COM1/COM2
- eDataBits: Anzahl der Datenbits SEVEN\_DATABITS/EIGHT\_DATABITS
- eParity: EVEN/ODD/NONE
- eStoppBits: Anzahl der Stoppbits ONE\_STOPPBIT/TWO\_STOPPBITS



### 5.13.2.6.5 Beispiel

#### Beispiel Programm in ST



Download (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207320715.zip>)

```
PROGRAM MAIN
VAR
(* EXAMPLE - BRIDGE between PIN 7 and 8 from X01 COM 2 Port*)
fb_COMPortOpen_1      : FB_COMPortOpen;
stCOMConfig_1        : ST_COMConfig;
hCOM                  : WORD;
Result_R              : INT;
Result_W              : INT;
Var_M                 : ARRAY[0..9] OF BYTE:=11,22,0,33,0(6);
Var_R                 : ARRAY[0..9] OF BYTE;
Value                 : INT;
Counter_V             : BYTE; (* It is all OK, this value counts up *)
i                     : INT;
i_k                   : INT;
fbTimer               : TON;
END_VAR
```

```
stCOMConfig_1.cbRxBufferLen :=300;
stCOMConfig_1.cbTxBufferLen :=300;
stCOMConfig_1.dwMode :=0;
stCOMConfig_1.BaudRate :=19200;
stCOMConfig_1.eCommPort :=COM2;
stCOMConfig_1.eDataBits:=EIGHT_DATABITS;
stCOMConfig_1.eParity:=EVEN;
stCOMConfig_1.eStoppBits:=ONE_STOPPBIT;

CASE i OF
(* Open Port *)
0: fb_COMPortOpen_1(bOpen:=TRUE , stCOMConfig:=stCOMConfig_1);
   IF NOT fb_COMPortOpen_1.bBusy THEN
     IF NOTfb_COMPortOpen_1.bError THEN
       hCOM:=fb_COMPortOpen_1.iHandle ;
       i:=i+1;
     ELSE
i:=100;
       END_IF
     END_IF
   (* Write data*)
1: fbTimer(IN:=FALSE);
   Result_W:=F_COMPortWrite(hCom, 4,ADR(Var_M[0]));
   IF Result_W>0 THEN
     i:=i+1;
     Var_M[2]:=Var_M[2]+1;
   ELSE
     i:=101;
   END_IF
   (*Receive data*)
2: Result_R:=F_COMPortRead(hCom, 100,ADR(Var_R[Value]));
   IF Result_R<>0 THEN
     Value:=Result_R+Value;
   END_IF
   IF Value>=4 THEN
     FOR i_k:=0 TO Value DO(*Check protocol*)
       IF Var_R[i_k-4]=11 AND Var_R[i_k-3]=22 AND Var_R[i_k-1]=33 THEN
         Counter_V:=Var_R[i_k-2];
         i:=1;
         Value:=0;
       END_IF
     END_FOR
   END_IF
   fbTimer(IN:=TRUE,PT:=t#1s); (*Watchdog receive*)
   IF fbTimer.Q THEN
     fbTimer(IN:=FALSE);
     i:=102;
   END_IF
100: ; (*ERROR open port*)
```

```
101: ; (*ERROR send data*)
102: i:=1; (*WD ERROR no data receive*)
END_CASE
```

### 5.13.2.7 BX Debug-Funktion

Diese Funktionen können benutzt werden um Befehls-Ausführungszeiten in einem PLC-Projekt zu messen. Die Einheit ist ein Tick. Ein Tick entspricht 5.12 µs.

#### Funktion Start Debug Timer

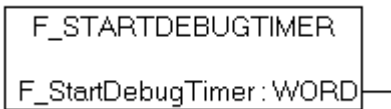


Abb. 109: Funktionsbaustein F\_STARTDEBUGTIMER

Der Aufruf dieser Funktion startet den Timer. Der Rückgabe Wert ist "0".

#### Funktion Read Debug Timer

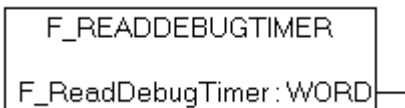


Abb. 110: Funktionsbaustein F\_READDEBUGTIMER

Mit dieser Funktion wird der Timer-Wert gelesen. Der Rückgabewert muss mit 5.12 µs multipliziert werden.

#### Beispiel

```
VAR
    Timer_BX      :WORD;
    i              :INT;
END_VAR
```

#### Programm

```
F_STARTDEBUGTIMER();
For i:=0 to 1000 do
;
END_FOR
Timer_BX:=F_READDEBUGTIMER();
```

### 5.13.2.8 Navigations-Schalter

#### 5.13.2.8.1 FUN GetNavSwitch

Dieser Funktionsbaustein ermöglicht es Ihnen das Auslesen des Navigationsschalters.

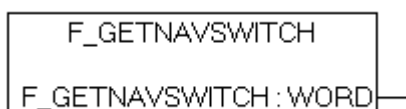


Abb. 111: Funktionsbaustein F\_GETNAVSWITCH

#### VAR\_Output

```
F_GETNAVSWITCH      :WORD;
```


**Legende**

F\_GETNAVSWITCH: Daten des Schalters

**Beschreibung des WORD**

<b>Bit</b>	15	14	...	5	4	3	2	1	0
<b>Name</b>	LOCKED	-	...	-	PRESS	RIGHT	LEFT	DOWN	UP

Wenn Bit 15 gesetzt ist befindet man sich im Menü des BX-Controllers. Dieses Bit ist solange gesetzt, solange der Anwender sich im Menü des BX3100 befindet. Sobald das Menü verlassen wurde, wird der Navigationsschalter sofort wieder für die PLC freigegeben, d.h. dass das Drücken des Press Buttons noch in dem Programm sichtbar ist. Bitte berücksichtigen Sie das in Ihrer Anwendung. Werten Sie zum Beispiel den Schalter erst einer kurzen warte Zeit aus, in dem Sie mit der fallenden Flanke von Bit 15 einen Timer starten.

 Download Beispiel Programm in ST <https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207322891.zip>

**5.13.2.9 Display**

**5.13.2.9.1 FB DISPWRITE**

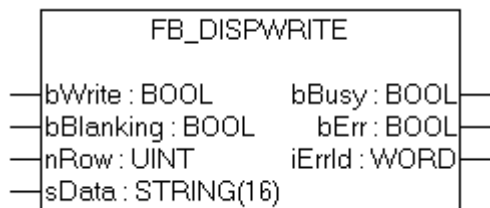


Abb. 112: Funktionsbaustein FB\_DISPWRITE

**VAR\_INPUT**

```
bWrite      :BOOL;
bBlanking   :BOOL;
nRow        :UINT;
sData       :STRING(16)
```

**Legende**

bWrite: Positive Flanke startet den Baustein

bBlanking: FALSE Hintergrundbeleuchtung an, TRUE Hintergrundbeleuchtung aus, Default ist diese an (wird FW 1.15 bei allen BX Controllern unterstützt).

nRow: Zeile im Display 1 oder 2.

sData: String der im Display angezeigt wird

**VAR\_OUTPUT**

```
bBusy       :BOOL;
bErr        :BOOL;
iErrId      :WORD;
```

**Legende**

bBusy: So lange der Baustein TRUE ist der Baustein aktiv.

bErr: Fehler Bit.

iErrId: Fehler Nummer.

Rückgabeparameter	Bedeutung
0	kein Fehler
> 0	Fehler Nummer

### Beispiel Programm in ST



Download <https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207325067.zip>

```
PROGRAM MAIN
VAR
    fb_DispWrite1: FB_DispWrite;

i:
    udiCounter:    UDINT;
    strCounter:    STRING;
    strLine:       STRING;
    k:             INT;
END_VAR

CASE i OF
0:  strCounter:=CONCAT('Counter :',UDINT_TO_STRING(udiCounter));
    fb_DispWrite1(bWrite:=TRUE , nRow:=1 , sData:=strCounter );
    IF NOT fb_DispWrite1.bBusy THEN
        IF NOTfb_DispWrite1.bErr THEN
            fb_DispWrite1(bWrite:=FALSE);
            udiCounter:=udiCounter+1;
            i:=1;
        END_IF
    END_IF
1:  fb_DispWrite1(bWrite:=TRUE , nRow:=2 , sData:=strLine);
    IF NOT fb_DispWrite1.bBusy THEN
        IF NOTfb_DispWrite1.bErr THEN
            fb_DispWrite1(bWrite:=FALSE);
            k:=k+1;
            strLine:=REPLACE(' ', '#',1,k);
            IF k=16 THEN
                k:=0;
            END_IF
            i:=0;
        END_IF
    END_IF
END_CASE
END_PROGRAM
```

### Display ASCII Tabelle

Beispiel für das Zeichen "&" (siehe Zeile 1 Spalte 7):  $00100110_{bin} = 38_{dez} = 26_{hex}$ . Dies entspricht in dem SPS-Wert '\$26' (String.)

<https://infosys.beckhoff.com/content/1031/bx3100/Resources/pdf/3207327243.pdf>

## 5.13.3 TcSystemBX

### 5.13.3.1 Echtzeit-Uhr - Beispiel

Auf dem BX-Controller ist eine Echtzeit-Uhr implementiert. Die aktuelle Uhrzeit kann per Funktionsbaustein ausgelesen werden. Das folgende Beispiel soll das verdeutlichen.

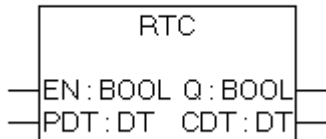


Abb. 113: Funktionsbaustein RTC

**FUNCTION\_BLOCK RTC****VAR\_INPUT**

```
EN      :BOOL;
PDT     :DT;
```

**Legende**

EN: Positive Flanke setzt die Uhrzeit auf den Wert der an den PDT-Eingang anliegt.

PDT: Datum und Uhrzeit die eingestellt werden sollen.

**VAR\_OUTPUT**


```
Q       :BOOL;
CDT     :BOOL;
```

**Legende**

CDT: Aktuelle Uhrzeit.

Notwendige Bibliotheken:

- TcSystemBX.lb6
- TcBaseBX.lb6

 Download Beispiel Programm in ST (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207329419.zip>)

```
PROGRAM MAIN
VAR
  fbTimer: TON;
  fbRTC: RTC;
END_VAR
```

```
fbTimer (PT:=t#60s, IN:=NOT fbTimer.Q);

IF fbTimer.Q THEN
  fbRTC;
END_IF
```

### **i** RTC-Baustein nicht in jedem SPS-Zyklus aufrufen

Der Aufruf des RTC-Bausteins erhöht die Zykluszeit um ca. 5 ms, verursacht durch die Datenkonvertierung in eine TIME AND DATE Variable. Deshalb sollte der Baustein nicht in jedem SPS-Zyklus aufgerufen werden!

Alternativ können Sie die Uhrzeit auch per ADS-Baustein auslesen. Der ADS-Baustein liefert Datum und die Uhrzeit als WORD-Variablen zurück.

Beispiel 19:30 Uhr - Hour: 19 / Minute: 30

### 5.13.3.2 Laden und Speichern von Rezepturen

Der Funktionsbaustein fb\_ReadWriteFile ermöglicht Daten (max. 16.000 Byte) dauerhaft im Flash-Speicher des BX-Controllers zu sichern. Ein neues Programm oder ein Reset des Projektes lässt diesen Speicherinhalt unberührt. Dieser Funktionsbaustein ist nicht für den Dauerhaften und ständigen Gebrauch geeignet. Es sind max. 10000 Schreibzyklen erlaubt. Es darf beliebig oft gelesen werden.

Anwendung: Sichern von Rezepten oder Einstellungen die sich selten oder gar nicht ändern, zum Beispiel Reglerparameter.

#### ● **Beachten Sie beim Schreiben der Daten**

**i**

- Während des Schreibens darf die Spannung nicht unterbrochen werden, es empfiehlt sich daher, das Schreiben von einem Bedien-Panel oder über die Navigationsschalter oder einfach über einen digitalen Eingang anzustoßen, um sicher zu gehen, dass während des Schreibens der BX-Controller nicht ausgeschaltet wird. Ein automatisches Schreiben empfiehlt sich nicht, da man nicht sicherstellen kann, ob nicht gerade in diesem Moment das Schreiben unterbrochen wird.
- Das Schreiben der Daten benötigt ca. zwei Sekunden, unabhängig von der Anzahl der Daten, die geschrieben werden.
- Wird der BX-Controller während des Schreibvorgangs ausgeschaltet gehen die Daten verloren.
- Es ist nur eine Instanz dieses Bausteins erlaubt.

#### Funktionsbaustein fb\_ReadWriteFile

Funktionsbaustein zum Lesen und Schreiben von Rezepturen

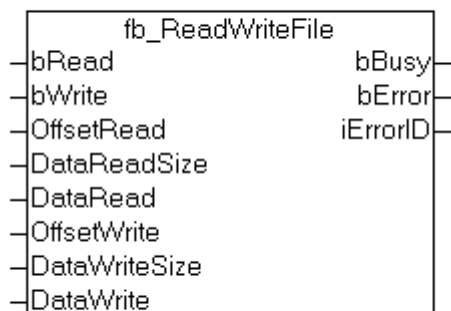


Abb. 114: Funktionsbaustein fb\_ReadWriteFile

#### VAR\_INPUT

```

bRead      :BOOL;
bWrite     :BOOL;
OffsetRead :WORD;
DataReadSize :WORD;
DataRead   :Pointer to Byte;
OffsetWrite :WORD;
DataWriteSize :WORD;
DataWrite  :Pointer to Byte;
    
```

#### Legende

- bRead*: Eine positive Flanke stößt das Lesen des Bausteins an (*bWrite* muss FALSE sein)
- bWrite*: Eine positive Flanke stößt das Schreiben des Bausteins an (*bRead* muss FALSE sein)
- OffsetRead*: Offset im Speicher max.16.000 Byte
- DataReadSize*: Größe der Daten in Byte, die gelesen werden sollen (max. 16.000 Byte)
- DataRead*: Pointer per ADR auf die Daten zeigen

*OffsetWrite*: Offset im Speicher max.16.000 Byte

*DataWriteSize*: Größe der Daten in Byte, die geschrieben werden sollen (max. 16.000 Byte)

*DataWrite*: Pointer per ADR auf die Daten zeigen

**VAR\_OUTPUT**

```
bBusy      :BOOL;
bError     :BOOL;
bErrorId   :UDINT
```

**Legende**

*bBusy*: Zeigt an des der Baustein noch aktiv ist


*bError*: Baustein hat einen Fehler

*bErrorId*: Fehlernummer

Rückgabeparameter iErrorId	Bedeutung
0	kein Fehler
1 <sub>dez</sub>	READ: Daten-Offset und Daten-Länge über 16.000 Byte
2 <sub>dez</sub>	WRITE: Daten-Offset und Daten-Länge über 16.000 Byte
0x31440708	CRC-Fehler im Datenspeicher
0x31470708	Das Schreiben der Daten ist noch nicht abgeschlossen

Notwendige Bibliotheken:

- TcSystemBX.lb6
- TcBaseBX.lb6

 Download Beispiel Programm in ST (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207331595.zip>)

**5.13.4 TcComPortBX**

**5.13.4.1 TcComPortBX - Übersicht**

Erforderliche Bibliotheken:

- TcBaseBX
- TcSystemBX

**Überblick**

Name	Beschreibung
<a href="#">fb_BX_BK8x00_Master</a> [► 128]	BK8x00 COM Port Funktionsbaustein, Kommunikation mit Buskoppler BK8x00 oder BC8x00
<a href="#">fb_BX_BK8x00_Slave</a> [► 128]	BK8x00 COM Port Funktionsbaustein, Kommunikation mit PC. Es wird ein BK8x00 simuliert.

Name	Beschreibung
FB_BX_COM_5 [▶ 132]	Baustein zum Emulieren einer KL60x1 (wenn COMlib.lib6 oder ModbusRTU.lib6 verwendet wird).
FB_BX_COM_64 [▶ 132]	Baustein zum Emulieren einer PC-Schnittstelle (wenn COMlib.lib oder ModbusRTU.lib verwendet wird).
FB_BX_COM_64ex [▶ 133]	Baustein zum Emulieren einer PC-Schnittstelle (wenn COMlib.lib oder ModbusRTU.lib verwendet wird). Hier kann die COM-Schnittstelle auch während des Betriebs geschlossen werden.

### 5.13.4.2 BK8x00 - FB COM-Port

Über diesen Funktionsbaustein kann über die serielle Schnittstelle des BXxxxx die seriellen Buskoppler BK8000 mit RS485 und BK8100 mit RS232-Anschluss verwendet werden. Die Maximale Baudrate beträgt 38400 Baud.

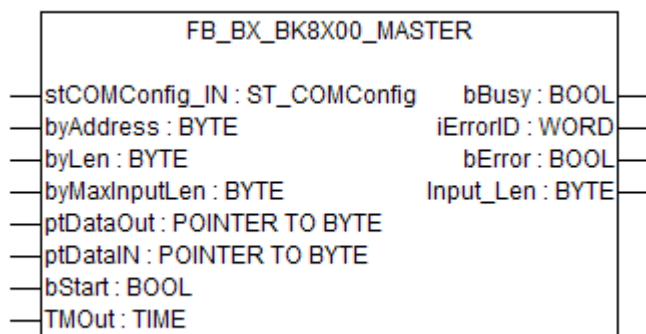


Abb. 115: Funktionsbaustein FB\_BX\_BK8X00\_MASTER

#### VAR\_INPUT

```
stCOMConfig      :ST_COMConfig;
byAddress        :BYTE;
byLen            :BYTE;
byMaxInputLen    :BYTE;
ptDataOut        :POINTER TO BYTE;
ptDataIN         :POINTER TO BYTE;
bStart           :BOOL;
TmOut            :TIME;
```

#### Legende

stComConfig: Struktur für die Auswahl der COM Parameter  
 byAddress: BX8x00 Adresse 1-98 (0 und 99 sind reserviert)  
 byLen: Daten Länge in [BYTE] (es sind nur grade Zahlen zugelassen 0, 2, 4, ...)  
 byMaxInputLen: wird mit SIZEOF und der Variable, die mit ptDataIN verbunden ist, angeschlossen  
 ptDataOut: wird mit ADR und den Daten Out verbunden  
 ptDataIn: wird mit ADR und den Daten In verbunden  
 bStart: positive Flanke startet den Baustein  
 TMOut: Wartezeit bis abgebrochen wird

#### VAR\_OUTPUT

```
bBusy           :BOOL;
bError          :BOOL;
iErrorId        :WORD;
Input_len       :WORD;
```



**Legende**

bBusy: So lange der Baustein TRUE ist der Baustein aktiv  
 bError: Fehler Bit  
 iErrorId: Fehler Nummer  
 Input\_Len: Anzahl der Daten die Empfangen wurden

Rückgabeparameter iErrorId	Bedeutung
0	kein Fehler
100 <sub>dez</sub>	Fehler beim Öffnen des COM-Ports
101 <sub>dez</sub>	Fehler beim Senden der Daten
102 <sub>dez</sub>	Watchdog Fehler, keine Antwort vom Slave innerhalb der WD-Zeit
105 <sub>dez</sub>	Der Input Puffer ist zu klein
200 <sub>dez</sub>	CRC-Fehler
0x80xx <sub>hex</sub>	Buskoppler Fehler xx Status Byte des Buskopplers (siehe BX8x00 Dokumentation)

**Hardware**

**RS 232 Kommunikation PIN Belegung**

BX COM1 RS232	BX COM2 RS232	BK8100
2	8	2
3	7	3
5	5	5

**RS485 Kommunikation PIN Belegung**

Einstellungen im FB: Wichtig ist das bei der Nutzung der RS485-Verbindung die Variable stCOMConfig:=1 ist und die COM2-Schnittstelle angewählt ist.

BX COM2 RS485	BK8000
1	3
6	8

**Beispiel Programm in ST**

 Download (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207333771.zip>)

Notwendiges Material:

- BX3100 + Busklemme
- BK8100, KL1xx8, KL2xx8, KL9010
- Serielles Kabel, PIN Belegung: siehe im Beispielprogramm

**5.13.4.3 BK8x00 - FB Slave COM-Port**

Mit diesem Funktionsbaustein kann der PC (TwinCAT oder KS8000) über die serielle Schnittstelle mit dem BXxxxx verbunden werden. Dabei ist der PC der serielle Master und der BXxxxx emuliert mit Hilfe des Funktionsbausteins eines BK8x00.

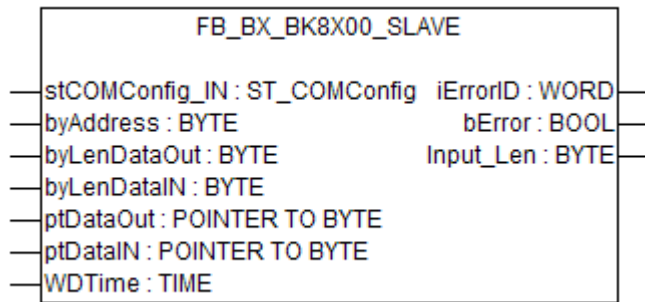


Abb. 116: Funktionsbaustein FB\_BX\_BK8X00\_SLAVE

**VAR\_INPUT**

```
stCOMConfig      :ST_COMConfig;
byAddress        :BYTE;
byLenDataOut     :BYTE;
byLenDataIN     :BYTE;
ptDataOut       :POINTER TO BYTE;
ptDataIN       :POINTER TO BYTE;
WDTime          :TIME;
```

**Legende**

stComConfig: Struktur für die Auswahl der COM Parameter  
 byAddress: BX8x00 Adresse 1-98 (0 und 99 sind reserviert)  
 byLenDataOut: Daten Länge in [BYTE] (es sind nur grade Zahlen zugelassen 0, 2, 4, ...)  
 byLenDataIn: Daten Länge in [BYTE] (es sind nur grade Zahlen zugelassen 0, 2, 4, ...)  
 ptDataOut: wird mit ADR und den Daten Out verbunden  
 ptDataIn: wird mit ADR und den Daten In verbunden  
 WDTime: Fehlermeldung wenn innerhalb der Watchdog - Zeit keine neuen Daten empfangen wurden (0 ms disable WD)

**VAR\_OUTPUT**

```
bError          :BOOL;
iErrorId        :WORD;
Input_Len      :BYTE;
```

**Legende**

**bError:** Fehler Bit  
**iErrorId:** Fehler Nummer  
**Input\_Len:** Anzahl der Daten die Empfangen wurden

Rückgabeparameter iErrorId	Bedeutung
0	kein Fehler
1	Watchdog Fehler wenn diese größer 0 ms ist (WD disable wenn 0 ms)
100 <sub>dez</sub>	Fehler beim Öffnen des COM Ports
101 <sub>dez</sub>	Fehler beim Senden der Daten
103 <sub>dez</sub>	Interner Recive Buffer übergelaufen
104 <sub>dez</sub>	Daten passen nicht in den PLC Buffer (größer 500 Byte)
105 <sub>dez</sub>	Daten können nicht in den PLC Buffer kopiert werden
200 <sub>dez</sub>	CRC - Fehler

Hardware

RS232-Kommunikation PIN-Belegung

BX COM 1 RS232	BX COM 2 RS232	PC COM Schnittstelle
2	8	2
3	7	3
5	5	5

RS485-Kommunikation PIN-Belegung

Einstellungen im FB: Wichtig ist das bei der Nutzung der RS485-Verbindung die Variable stCOMConfig:=1 ist und die COM2-Schnittstelle angewählt ist.

BX COM 2 RS485	PC COM Port (zum Beispiel: RS485 Karte W&T #13601, 2-Draht, ohne Echo, Automatik)
1	1 - 2 Brücken
6	6 - 7 Brücken

Beispiel Programm für den BXxxx in Strukturiertem Text

 Download (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207335947.zip>)

System Manager File für TwinCAT als Master. Wie man im Bild sieht, wird ein Buskoppler mit Busklemmen projektiert. Die Art und Anzahl der Busklemmen gibt dann die Datenlänge an. Welche Busklemmen das sind spielt im Prinzip keine Rolle. Zum Beispiel:

- 2 x KL3002 ergibt 4 Worte Eingänge
- 2 x KL4002 ergibt 4 Worte Ausgänge

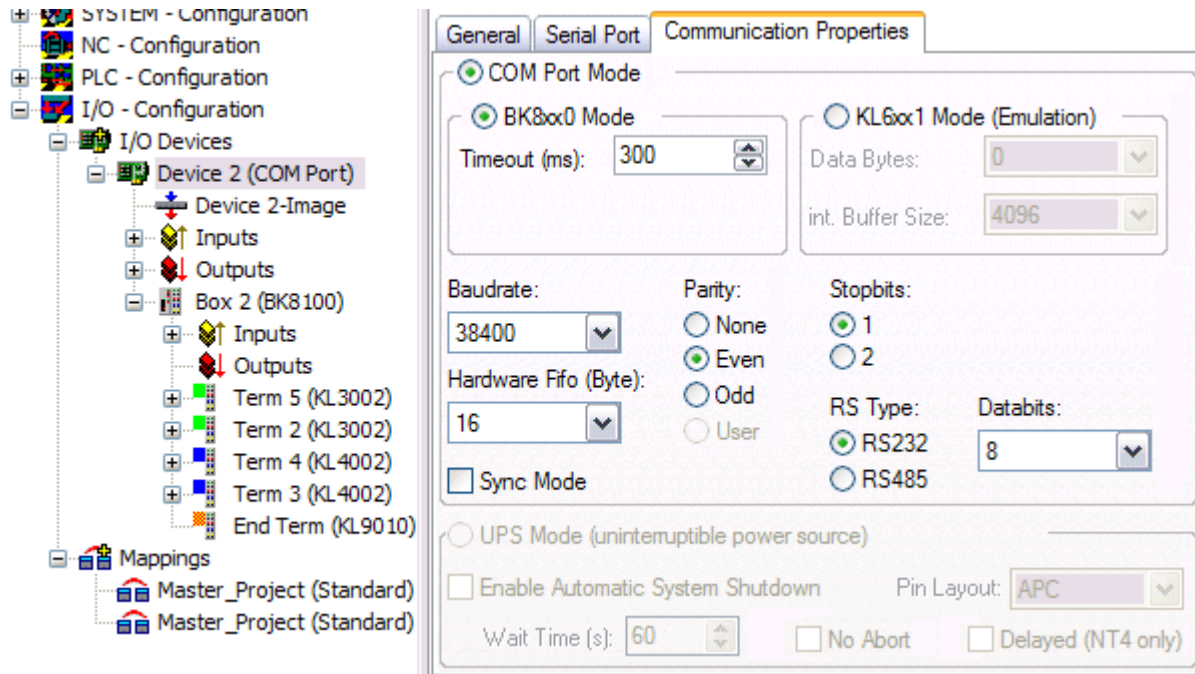


Abb. 117: Kommunikationseigenschaften

Notwendiges Material:

- BX3100 + Busklemme
- PC mit RS232-Schnittstelle und TwinCAT ab Version 2.9, serielles Kabel, PIN-Belegung: siehe oben

### 5.13.4.4 FB\_BX\_COM\_5

Dieser Baustein verbindet die ModbusRTU.lib6, ModbusRTU.lib oder die ComLib.lib6 mit der seriellen Schnittstelle des BX-Controllers. Es wird eine KL60x1 Emuliert, die Daten werden nicht auf eine Busklemme ausgegeben, sonder über eine der beiden auf dem BX befindlichen seriellen Schnittstellen.




Abb. 118: Funktionsbaustein FB\_BX\_COM\_5

#### VAR\_INPUT

```
pstrEmo_IN      :POINTER TO BYTE;
pstrEmo_OUT     :POINTER TO BYTE;
ComConfig       :ST_COMConfig;
```

#### Legende

pstrEmo\_IN: Pointer auf die KL6inData5B  
 pstrEmo\_OUT: Pointer auf die KL6outData5B  
 ComConfig [▶ 120]: Parametrierung der COM Schnittstelle

 Download Beispielprogramm in ST für die Verknüpfung COMLib zum BX: (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207338123.zip>)

### 5.13.4.5 FB\_BX\_COM\_64

Dieser Baustein verbindet die ModbusRTU.lib oder ComLib.lib mit der seriellen Schnittstelle des BX-Controllers. Es wird eine PC-Schnittstelle emuliert. Die Daten werden nicht auf einer PC-Schnittstelle ausgegeben, sondern über eine der beiden auf dem BX befindlichen seriellen Schnittstellen (COM1 oder COM2).




Abb. 119: Funktionsbaustein FB\_BX\_COM\_64


#### VAR\_INPUT

```
pstrEmo_IN      :POINTER TO BYTE;
pstrEmo_OUT     :POINTER TO BYTE;
ComConfig       :ST_COMConfig;
```

#### Legende

pstrEmo\_IN: Pointer auf die ModbusPCComInData  
 pstrEmo\_OUT: Pointer auf die ModbusPCComInData  
 ComConfig [▶ 120]: Parametrierung der COM Schnittstelle

 Download Beispielprogramm in ST für die Verknüpfung ModbusRTU zum BX: (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207340299.zip>)

 Download Beispielprogramm in ST für die Verknüpfung ModbusRTU Version 2 zum BX: (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207342475.zip>)

Für das Beispiel ist die ModbusRTU-Bibliothek notwendig!

### 5.13.4.6 FB\_BX\_COM\_64ex

Dieser Baustein verbindet die ModbusRTU.lib oder ComLib.lib mit der seriellen Schnittstelle des BX-Controllers. Es wird eine PC Schnittstelle mit 64 Byte Nutzdaten emuliert. Die Daten werden nicht auf eine PC-Schnittstelle ausgegeben, sondern über eine der beiden auf dem BX befindlichen seriellen Schnittstellen (COM1 oder COM2).

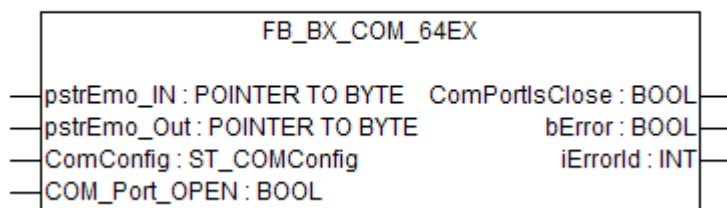


Abb. 120: Funktionsbaustein FB\_BX\_COM\_64EX

#### VAR\_INPUT

```
pstrEmo_IN      :POINTER TO BYTE;
pstrEmo_OUT     :POINTER TO BYTE;
ComConfig       :ST_COMConfig;
```

#### VAR\_OUTPUT

```
ComPortIsClose :BOOL;
bError          :BOOL;
iErrorId        :INT;
```

#### Legende

**pstrEmo\_IN:**

Pointer auf die ModbusPCComInData

**pstrEmo\_OUT:**

Pointer auf die ModbusPCComOutData

**ComConfig [▶ 120]:**

Parametrierung der COM Schnittstelle

**COM\_Port\_Open:**

Wird dieses Bit gesetzt, wird die Schnittstelle geöffnet. Wird dieses Bit zurückgesetzt wird die COM Schnittstelle geschlossen.

**ComPortIsClose:**


Wenn die Schnittstelle geschlossen ist, ist diese Bit gesetzt

**bError:**

Es liegt ein Fehler vor

**iErrorId:**

Fehlercode (siehe FB\_COMPortOpen) [▶ 118]

 Download Beispielprogramm in ST für die Verknüpfung ModbusRTU zum BX: (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207344651.zip>)

Für das Beispiel ist die ModbusRTU Bibliothek erforderlich!

### 5.13.4.7 Weitere Beispiele

#### 5.13.4.7.1 BX COM-Port als ModbusRTU-Master

Die serielle Schnittstelle des BX kann auch dazu benutzt werden, diese als Modbus-Master zu verwenden.

##### Benötigte Komponenten


- 1 x BX3100
- Busklemmen für den K-Bus (hier beliebig, da diese für das Beispiel nicht gebraucht werden)
- 1 x BK7300
- 2 x KL2xx4
- 2 x KL1xx4
- 1 x KL9010

##### Kabel RS 485\*

BX3100 COM 2 / RS 485	BK7300 / RS 485
1	3
6	8

\*) aktiver Abschlusswiderstand ist bei kurzen Leitungslängen (< 5 m) und kleinen Baudraten (<19200 Baud) nicht erforderlich

 **Download Beispiel Programm in ST für die Verknüpfung ModbusRTU Master zum BX:** (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207346827.zip>)

 **Download Beispiel Programm in ST für die Verknüpfung ModbusRTU Master Version 2 zum BX:** (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207349003.zip>)

Für das Beispiel ist die ModbusRTU, TcComPortBC, TcBaseBX Bibliothek notwendig.  
 Baudrate 9600, n, 8,1 Default BK7300,  
 BK7300 Adresse 11

##### Reaktionszeiten

Die Reaktionszeiten sind abhängig von der eingestellten Task-Zeit, die Anzahl an Slaves, die Länge der Modbus-Telegramme und der Antwortzeit der Slaves.

Für die Ermittlung folgender Tabelle wurden Beckhoff BK7300 Modbus Slaves verwendet. Da dies nicht auf jeden Slave übertragbar ist, sollte diese Tabelle nur als Richtwert verwendet werden.

Baudrate 38400 Baud (jeweils ein Read Reg. und ein Write Reg. Telegramm pro Slave)

Slave-Anzahl	Task-Zeit auf dem BX	Zeit für einen Durchlauf
1	5	100 ms* / 125 ms**
2	5	200 ms / 225 ms
1	10	180 ms / 220 ms
2	10	350 ms / 390 ms
1	20	350 ms / 350 ms
2	20	700 ms / 700 ms

\*) 2 Worte Ein- und 2 Worte Ausgänge

\*\*) 20 Worte Ein- und 20 Worte Ausgänge

### 5.13.4.7.2 BX COM-Port - ComLibV2

Beispiele für ComLibV2 senden von Strings über die interne COM Schnittstelle des BX Controllers. Um Daten zu empfangen kann man eine Brücke vom PIN 7 und 8 herstellen auf X01 (COM2).


#### Notwendiges Material

Hardware:

- BX-Controller

Software:

- TwinCAT ab 2.10
- COMlibV2.lib
- TcComPortBX.lbx
- Standard.lbx
- TcBase.lbx
- TcSystemBX.lbx

 Download Beispielprogramm BX: (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207351179.zip>)

### 5.13.4.7.3 BX COM-Port - Cimrex-Panel

Die serielle Schnittstelle des BX-Controllers kann auch als Modbus-Slave benutzt werden. In diesem Beispiel wird ein Panel der Firma Beijers angeschlossen. Weitere Informationen zu dem Panel finden Sie unter <http://www.beijerelectronics.de>.



Abb. 121: Cimrex-Panel am COM-Port des BX-Controllers

#### Benötigte Komponenten

- 1 x BX3100
- 1 x Cimrex 12
- beliebige Busklemmen (beliebig, da diese im Beispiel nicht benutzt werden)


**Kabel RS232**


BX3100 COM 2 / RS485	Cimrex 12 RS232
7	2
8	3
9	5

**Kabel RS485\***

BX3100 COM 2 / RS 485	Cimrex 12 RS485
1	2 -3
6	15 -16

\*) aktiver Abschlusswiderstand ist bei kurzen Leitungslängen (≤ 5 m) und kleinen Baudraten (≤19200 Baud) nicht erforderlich

 Download Beispielprogramm in ST für den BX: (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207340299.zip>)

 Download Beispiel mit Cimrex-Panel: (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207353355.zip>)

Für das Beispiel sind die Bibliotheken ModbusRTU, TcComPortBC und TcBaseBX erforderlich.  
 - Baudrate 9600,n,8,1 D  
 - Cimrex 12



#### 5.13.4.7.4 BX COM-Port - RK512-Protokoll

Über die Schnittstelle COM1 oder COM2 des BX-Controllers kann das RK512-Protokoll Daten mit einer Gegenstelle austauschen. Dokumentation zum RK512-Baustein finden Sie im Beckhoff Information System. Über die 64 Byte Emulation des BX-Controllers wird die serielle Schnittstelle des PC simuliert.

##### Notwendiges Material


Hardware:


- PC mit RS232 Schittstelle und TwinCAT PLC ab 2.9
- BX-Controller
- Serielles Kabel für die Verbindung BX - PC

Software:

- TwinCAT ab 2.9
- COMlib.lib
- COMlib3964R.lib
- COMlibRK512.lib
- TcComPortBX.lbx
- Standard.lbx
- TcBase.lbx
- TcSystemBX.lbx
- ChrAscBx.lbx

 Download Beispielprogramm BX3100: (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207355531.zip>)

 Download Beispielprogramm PC: (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207357707.zip>)

 Download Beispiel System Manager File PC: (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207359883.zip>)

### 5.13.4.7.5 BX COM-Port - SMS über Mobiltelefon

Die serielle Schnittstelle kann auch dafür verwendet werden, vom BX-Controller aus eine SMS zu versenden. Das folgende Beispiel benutzt die SMS-Bibliothek mit einem Siemens S35 Mobiltelefon.

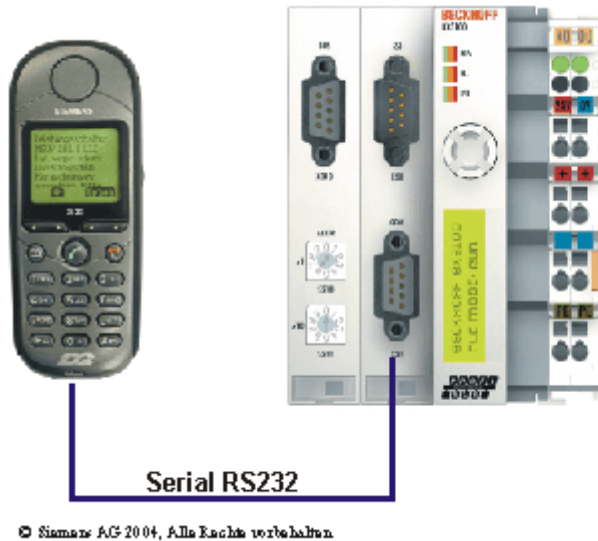


Abb. 122: Mobiltelefon am COM-Port des BX-Controllers

 Download: (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207362059.zip>)

#### Pinbelegung (Siemens-Kabel S30880-S4501 A801-2)

S35	COM 1	COM 2
2	3	7
3	2	8
5	5	9

## 5.13.5 TcTwinSAFE

### 5.13.5.1 Übersicht


Die Busklemmen-Controller der BX-Serie unterstützen die TwinSAFE-Busklemmen wenn folgende Voraussetzungen erfüllt sind:

- Am Busklemmen-Controller ist nur eine Logik-Klemme zugelassen und diese muss am K-Bus-Interface angeschlossen sein, nicht am SSB.
- An dieser Logik-Klemme sind maximal 7 Verbindungen zugelassen.
- TwinSAFE-Ein- und Ausgangsklemmen können an den K-Bus wie auch an den SSB angeschlossen werden, zum Beispiel über BK5120 oder BK515x.
- Falls Online-Changes verwendet werden soll, muss der Connection-Timeout auf 500 ms oder höher eingestellt werden.
- Für den Download der TwinSAFE-Projekte muss eine ADS-Verbindung bestehen. Diese kann seriell aber auch über den Feldbus hergestellt werden.
- Die Firmware-Version des Busklemmen-Controllers muss 1.17 oder höher sein.

#### TwinSAFE-Bibliothek

Die TwinSAFE-Bibliothek beinhaltet Funktionsbausteine, mit denen Dienste/Funktionen in Zusammenhang mit den TwinSAFE-Klemmen KL1904, KL2904 und KL6904 ausgeführt werden können.

Name	Beschreibung
F_GetVersionTcTwinSAFE [► 139]	Versionsnummer der Bibliothek
FB_TwinSAFE_KLx904_input [► 139]	Auswertung der TwinSAFE Daten, welche von einer KL1904 oder KL2904 zu einer KL6904 gesendet werden
FB_TwinSAFE_KLx904_output [► 142]	Auswertung der TwinSAFE Daten, welche von einer KL6904 zu einer KL1904 oder KL2904 gesendet werden

 Download der TwinSAFE-Bibliothek: (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3207364235.zip>)

### 5.13.5.2 FUNCTION F\_GetVersionTcTwinSAFE

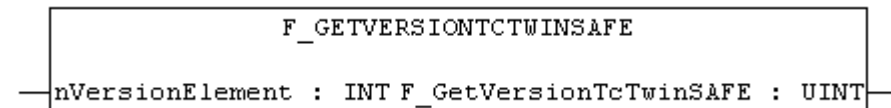


Abb. 123: Funktionsbaustein F\_GETVERSIONTCTWINSAFE

Mit dieser Funktion können Versionsinformationen der SPS-Bibliothek ausgelesen werden.

#### FUNCTION F\_GetVersionTcTwinSAFE : UINT

```
VAR_INPUT
    nVersionElement : INT;
END_VAR
```

**nVersionElement:** Versionselement, das gelesen werden soll. Mögliche Parameter:

- 1 : major number;
- 2 : minor number;
- 3 : revision number;

Entwicklungsumgebung	Zielplattform	IO-Hardware	Einzubindende SPS-Bibliotheken
TwinCAT v2.10.0 Build > 914	PC (i386)	-	TcTwinSAFE.Lib (Standard.Lib, TcBase.Lib und TcSystem.Lib werden automatisch eingebunden)
TwinCAT v2.10.0	BX-Serie	-	TcTwinSAFE.LBX (Standard.LBX; TcBaseBX.LBX; TcSystemBX.LBX werden automatisch eingebunden)

### 5.13.5.3 FUNCTION\_BLOCK FB\_TwinSAFE\_KLx904\_input

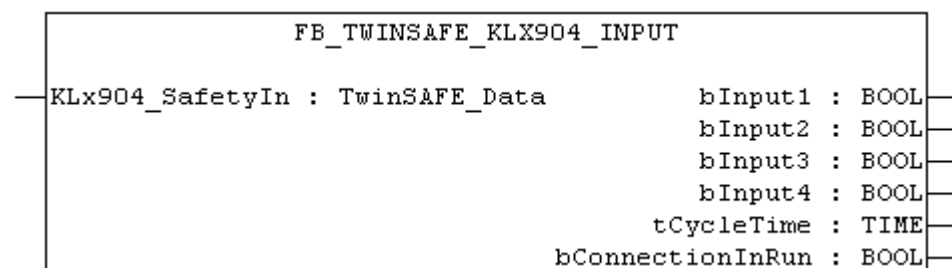


Abb. 124: Funktionsbaustein FB\_TWINSAFE\_KLX904\_INPUT

Mit dem Funktionsbaustein *FB\_TwinSAFE\_KLx904\_input* kann eine Auswertung der TwinSAFE Daten, welche von einer KL1904 oder KL2904 zu einer KL6904 gesendet werden durchgeführt werden. Der Eingangsparameter wird mit den SafetyIn Daten einer KL1904 oder KL2904 im System Manager doppelt verknüpft.

**VAR\_INPUT**

```
VAR_INPUT
    KLx904_SafetyIn AT%I* : TwinSAFE_Data; (* Additional link to "SafetyIn" *)
END_VAR
```

**KLx904\_SafetyIn:** TwinSAFE Telegramm, welches von einer KL1904 oder KL2904 zu einer KL6904 gesendet wird. Dieser Parameter wird im System Manager auf SafetyIn (Eingangsdaten der KLx904) doppelt verknüpft.

**VAR\_OUTPUT**

```
VAR_OUTPUT
    bInput1          : BOOL;
    bInput2          : BOOL;
    bInput3          : BOOL;
    bInput4          : BOOL;
    tCycleTime       : TIME;
    bConnectionInRun : BOOL;
END_VAR
```

**bInput1:** Liefert den Eingang 1 einer KL1904. Ist dieser Baustein für eine Verbindung zu einer KL2904 verwendet, ist der Wert immer 0.

**bInput2:** Liefert den Eingang 2 einer KL1904. Ist dieser Baustein für eine Verbindung zu einer KL2904 verwendet, ist der Wert immer 0.

**bInput3:** Liefert den Eingang 3 einer KL1904. Ist dieser Baustein für eine Verbindung zu einer KL2904 verwendet, ist der Wert immer 0.

**bInput4:** Liefert den Eingang 4 einer KL1904. Ist dieser Baustein für eine Verbindung zu einer KL2904 verwendet, ist der Wert immer 0.

**tCycleTime:** Liefert die Zykluszeit in ms, die benötigt wird um das TwinSAFE Telegramm zwischen den Teilnehmern auszutauschen.

**bConnectionInRun:** Liefert ein TRUE wenn kein Fehler in der Verbindung von der KLx904 zur KL6904 ansteht.

**Beispiel für einen Aufruf im FUB:**

```
PROGRAM MAIN
VAR
    fbTwinSAFE_KLx904_input      : FB_TwinSAFE_KLx904_input;
    bInput1_KL1904_S_Address_113 : BOOL;
    bInput2_KL1904_S_Address_113 : BOOL;
    bInput3_KL1904_S_Address_113 : BOOL;
    bInput4_KL1904_S_Address_113 : BOOL;
    tCycleTime_KL1904_KL6904     : TIME;
    bConnection3_In_Run          : BOOL;
END_VAR
```

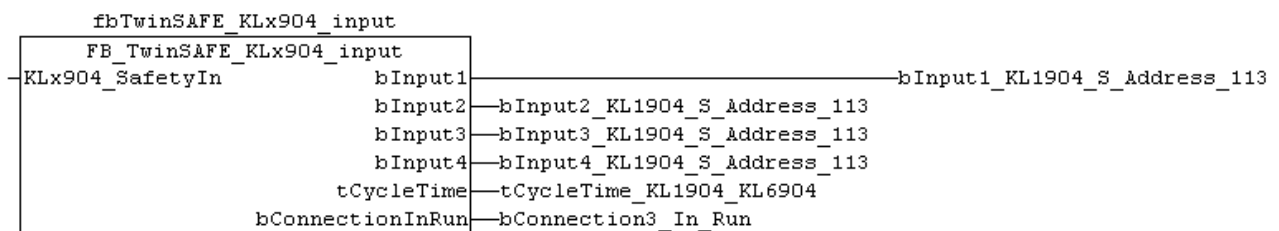


Abb. 125: Funktionsbaustein FB\_TWINSAFE\_KLX904\_input

Im Beispiel werden die Werte der KL1904 Eingangsdaten auf die angeschlossenen Variablen geschrieben. Ist der Ausgang bConnectionInRun FALSE werden die Ausgänge generell auf FALSE gesetzt.

Zum Verknüpfen der Eingangsdaten markieren Sie den Parameter KLx904\_SafetyIn und wählen aus dem Kontext Menü "Verknüpfung ändern..."

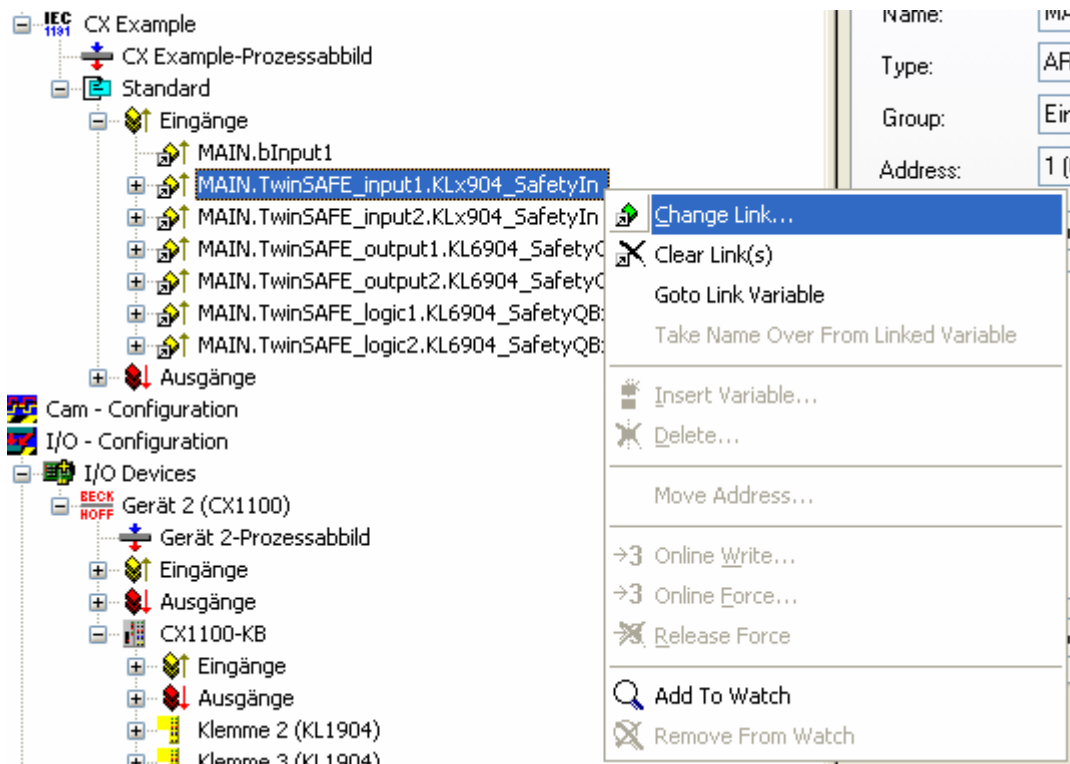


Abb. 126: Verknüpfen der Eingangsdaten

und wählen im folgenden Dialog die entsprechende SafetyIn-Variable

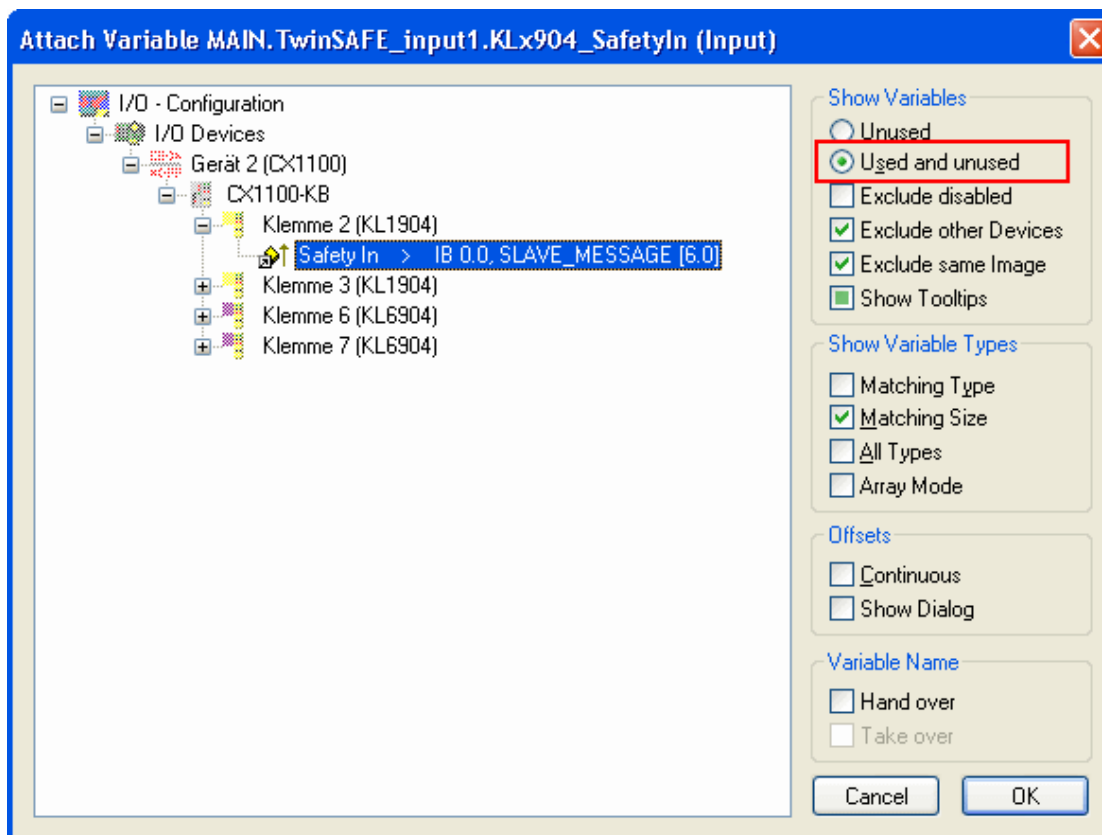


Abb. 127: Auswahl der SafetyIn-Variable

Entwicklungsumgebung	Zielplattform	IO-Hardware	Einzubindende SPS-Bibliotheken
TwinCAT v2.10.0 Build > 914	PC (i386)	KLx904	TcTwinSAFE.Lib (Standard.Lib, TcBase.Lib und TcSystem.Lib werden automatisch eingebunden)
TwinCAT v2.10.0 Build > 914	BX-Serie	KLx904	TcTwinSAFE.LBX (Standard.LBX, TcBaseBX.LBX und TcSystemBX.LBX werden automatisch eingebunden)

### 5.13.5.4 FUNCTION\_BLOCK FB\_TwinSAFE\_KLx904\_output

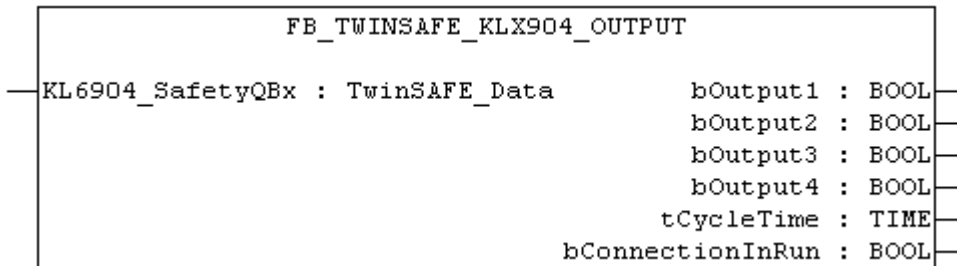


Abb. 128: Funktionsbaustein FB\_TWINSAFE\_KLX904\_OUTPUT

Mit dem Funktionsbaustein *FB\_TwinSAFE\_KLx904\_output* kann eine Auswertung der TwinSAFE Daten, welche von einer KL6904 zu einer KL1904 oder KL2904 gesendet werden durchgeführt werden. Der Eingangsparameter wird mit den SafetyQBx Daten einer KL6904 im System Manager doppelt verknüpft.

#### VAR\_INPUT

```
VAR_INPUT
    KL6904_SafetyQBx AT%I* : TwinSAFE_Data; (* Additional link to "SafetyQBx" *)
END_VAR
```

**KL6904\_SafetyQBx:** TwinSAFE Telegramm, welches von einer KL6904 zu einer KL1904 oder KL2904 gesendet wird. Dieser Parameter wird im System Manager auf SafetyQBx (Eingangsdaten der KL6904) doppelt verknüpft, wobei das x für Ziffern zwischen 1 und 15 steht entsprechend der verwendeten TwinSAFE Connection.

#### VAR\_OUTPUT

```
VAR_OUTPUT
    bOutput1      : BOOL;
    bOutput2      : BOOL;
    bOutput3      : BOOL;
    bOutput4      : BOOL;
    tCycleTime    : TIME;
    bConnectionInRun : BOOL;
END_VAR
```

**bOutput1:** Liefert den Ausgang 1 einer KL2904. Wird der Baustein für eine Verbindung zur KL1904 verwendet, ist dieser Wert grundsätzlich 0.

**bOutput2:** Liefert den Ausgang 2 einer KL2904. Wird der Baustein für eine Verbindung zur KL1904 verwendet, ist dieser Wert grundsätzlich 0.

**bOutput3:** Liefert den Ausgang 3 einer KL2904. Wird der Baustein für eine Verbindung zur KL1904 verwendet, ist dieser Wert grundsätzlich 0.

**bOutput4:** Liefert den Ausgang 4 einer KL2904. Wird der Baustein für eine Verbindung zur KL1904 verwendet, ist dieser Wert grundsätzlich 0.

**tCycleTime:** Liefert die Zykluszeit in ms, die benötigt wird um das TwinSAFE Telegramm zwischen den Teilnehmern auszutauschen.

**bConnectionInRun:** Liefert ein TRUE wenn kein Fehler in der Verbindung von der KL6904 zur KLx904 ansteht.

**Beispiel für einen Aufruf im FUB**

```
PROGRAM MAIN
VAR
  fbTwinSAFE_KLx904_output          : FB_TwinSAFE_KLx904_output;
  bOutput1_KL6904_Connection_to_113 : BOOL;
  bOutput2_KL6904_Connection_to_113 : BOOL;
  bOutput3_KL6904_Connection_to_113 : BOOL;
  bOutput4_KL6904_Connection_to_113 : BOOL;
  tCycleTime_KL6904_KL1904         : TIME;
  bConnection3_In_Run_2             : BOOL;
END_VAR
```

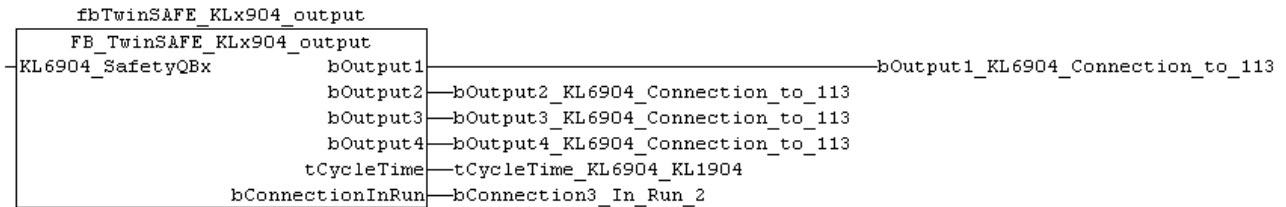


Abb. 129: Aufruf des Funktionsbausteins FB\_TWINSAFE\_KLX904\_OUTPUT

Im Beispiel werden die Werte der TwinSAFE Klemme KL6904 zur KL1904 ausgewertet. Da in dieser Verbindung keine Ausgangssignale verwendet werden, sind die Ausgänge grundsätzlich FALSE. Es kann nur tCycleTime und bConnectionInRun ausgewertet werden.

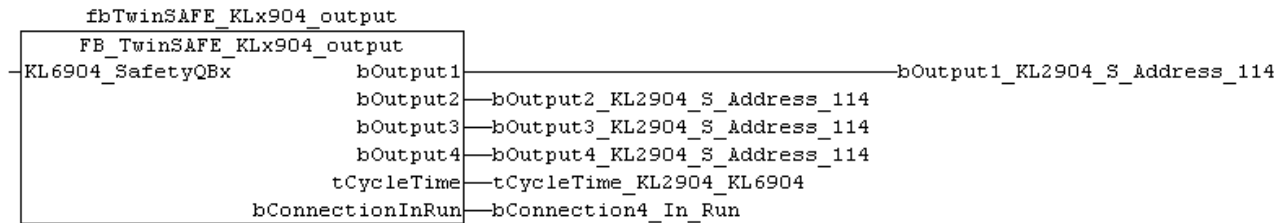


Abb. 130: Aufruf des Funktionsbausteins FB\_TWINSAFE\_KLX904\_OUTPUT

Im Beispiel werden die Werte der TwinSAFE Klemme KL6904 zur KL2904 ausgewertet. In dieser Verbindung werden die Ausgangssignale zur KL2904 geschrieben und vom Baustein auf die angeschlossenen Variablen kopiert. Ist der Ausgang bConnectionInRun auf FALSE werden die Ausgänge auf FALSE gesetzt.

Zum Verknüpfen der Eingangsdaten markieren Sie den Parameter KL6904\_SafetyQBx und wählen aus dem Kontext Menü "Verknüpfung ändern..."

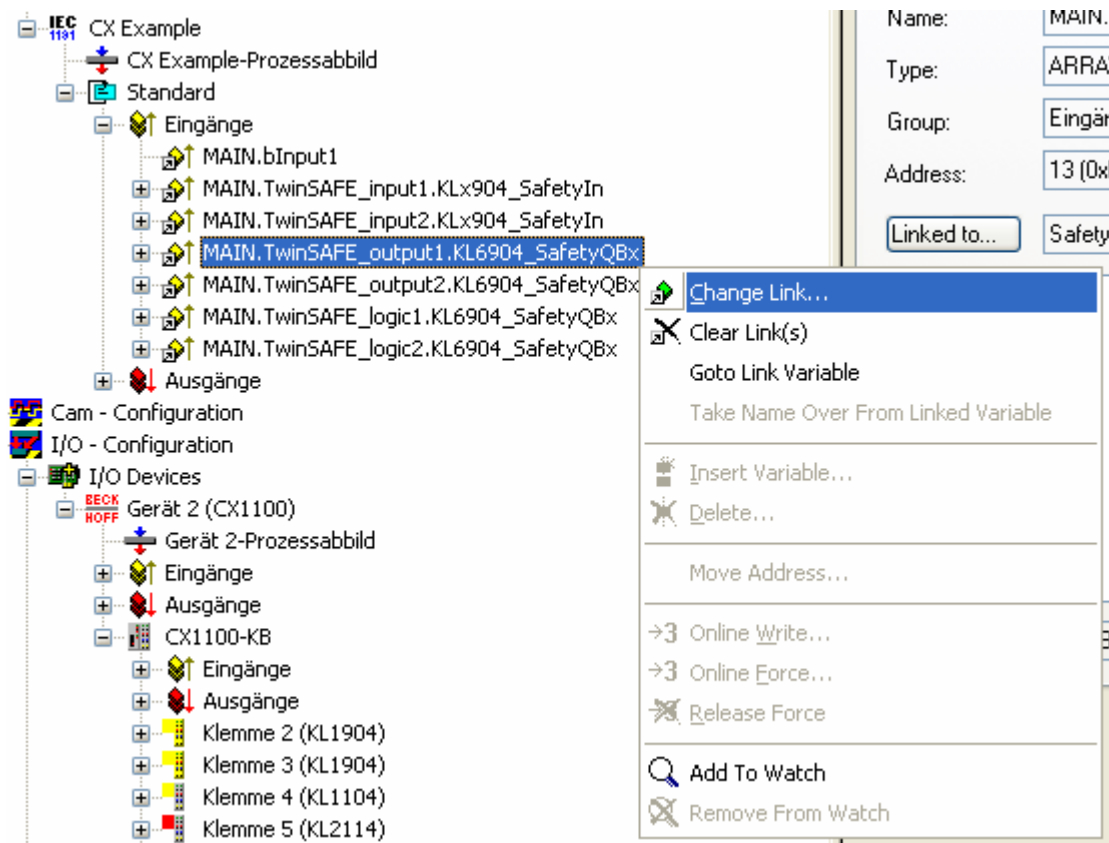


Abb. 131: Verknüpfen der Eingangsdaten

und wählen im folgenden Dialog die entsprechende SafetyQBx-Variable

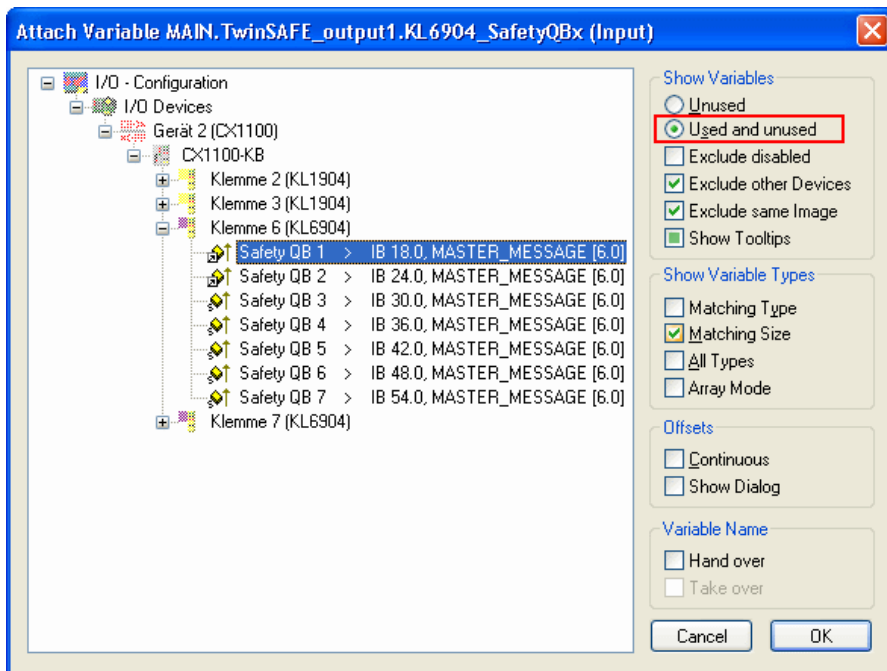


Abb. 132: Auswahl der entsprechenden SafetyQBx-Variable



Entwicklungsumgebung	Zielplattform	IO-Hardware	Einzubindende SPS-Bibliotheken
TwinCAT v2.10.0 Build > 914	PC (i386)	KLx904	TcTwinSAFE.Lib (Standard.Lib, TcBase.Lib und TcSystem.Lib werden automatisch eingebunden)
TwinCAT v2.10.0 Build > 914	BX-Serie	KLx904	TcTwinSAFE.LBX (Standard.LBX, TcBaseBX.LBX und TcSystemBX.LBX werden automatisch eingebunden)

## 5.14 Programmübertragung

### 5.14.1 Programmübertragung über die serielle Schnittstelle

Jeder Busklemmen-Controller kann über die RS232-Schnittstelle des PCs programmiert werden.

Wählen Sie im TwinCAT PLC Control die serielle Schnittstelle an.

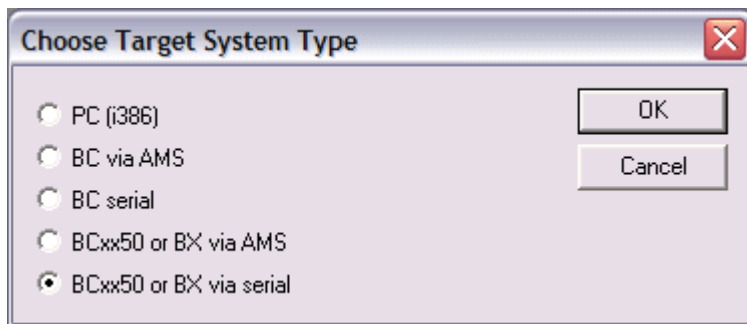


Abb. 133: Auswahl des Datenübertragungswegs - Serielle Schnittstelle

Unter Online/Kommunikationsparameter finden Sie im PLC Control die Einstellungen zur seriellen Schnittstelle, Portnummer, Baud-Rate usw.

Der Busklemmen Controller benötigt folgende Einstellung:

- Baud-Rate: 9600/19200/38400/57600 Baud (automatische Baudratenerkennung)
- Stop Bits: 1
- Parity: Gerade

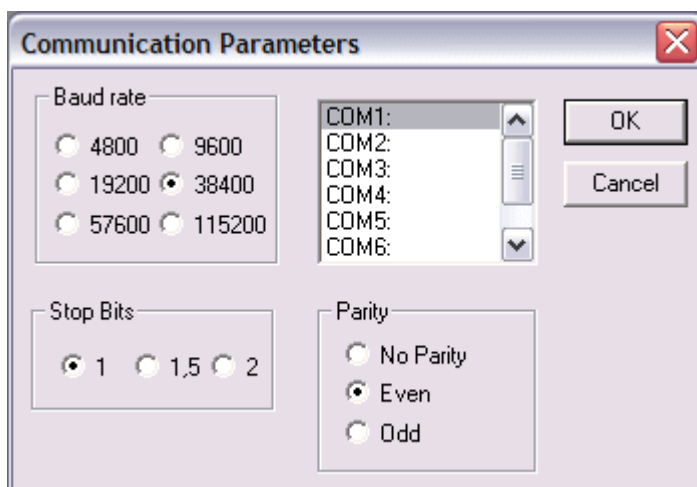


Abb. 134: Parametrierung der seriellen Schnittstelle

**Programmübertragung über die serielle Schnittstelle per ADS**

Der Busklemmen-Controller kann über die RS232-Schnittstelle des PC programmiert werden. Bevor Sie damit arbeiten können muss der Busklemmen-Controller dem TwinCAT bekannt gemacht werden (siehe serielles ADS [▶ 41]).

Wählen Sie im TwinCAT PLC-Control die ADS-Verbindung an.

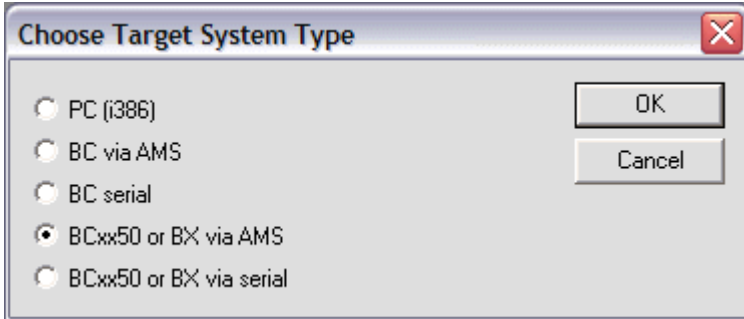


Abb. 135: Auswahl des Datenübertragungswegs - AMS

Im PLC Control kann man sich unter *Online/Communication Parameters...* einwählen.

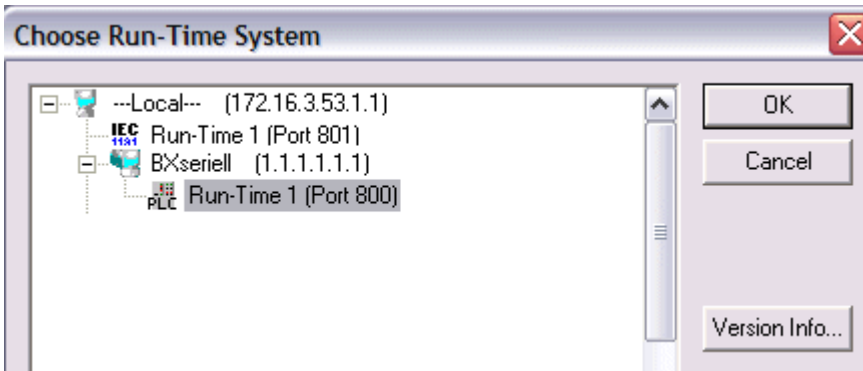


Abb. 136: Auswahl des Gerätes

## 5.14.2 Programmübertragung über PROFIBUS

TwinCAT bietet die Möglichkeit, das Anwenderprogramm über den Feldbus auf den Busklemmen-Controller zu übertragen. Im PLC Control kann nach dem Sichern in der Registry und einem Restart des TwinCAT-Systems, als Zielsystem der BC/BX ausgewählt werden. Erforderlich ist der TwinCAT-Level TwinCAT PLC.

Mindestanforderungen:

- FC310x mit Firmware ab 2.55
- TwinCAT 2.9 Build 1025

### Initialisieren des Busklemmen-Controller

Um den Koppler im PLC Control auswählen zu können, muss er dem System erst einmal bekannt gemacht werden.

Tragen Sie den Busklemmen Controller in dem System Manager ein, legen Sie Art, Anzahl und Größe der Feldbusvariablen an und verknüpfen sie diese mit einer Task. Für den Späteren Programm Download über den PROFIBUS muss beim Busklemmen-Controller im Karteireiter ADS das ADS-Interface aktiviert werden. Sichern Sie die Einstellungen und aktivieren Sie die Konfiguration. Danach starten Sie das TwinCAT System und die zyklische Task.

### TwinCAT System Manager

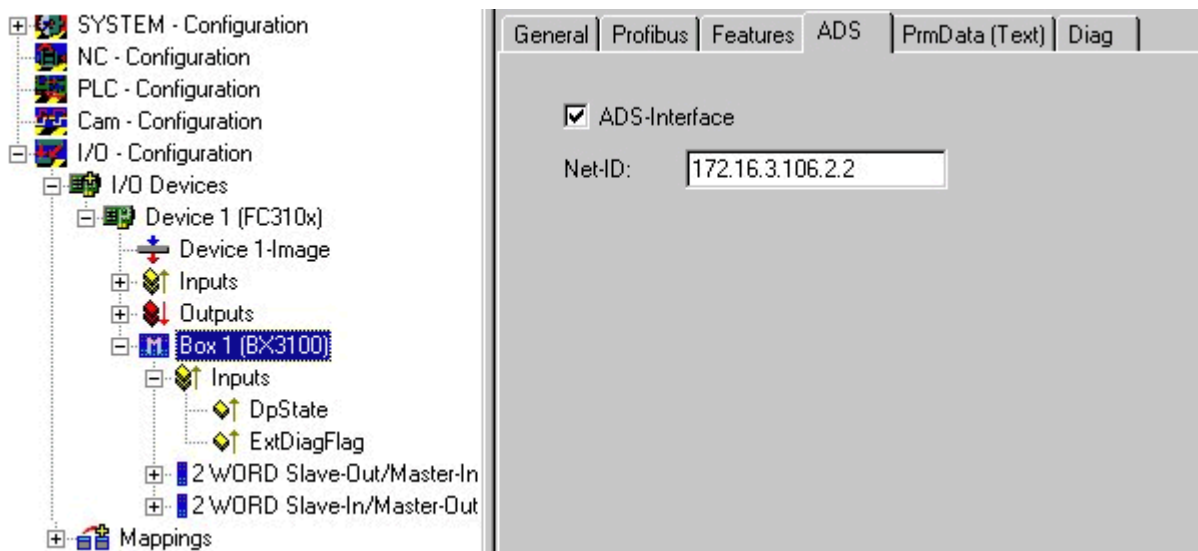


Abb. 137: ADS-Interface

### PLC Control

Beim Neustart des TwinCAT PLC Control fragt TwinCAT nach der Zielplattform, d.h. auf welchem Gerät später das Anwender-Programm laufen soll. TwinCAT bietet als Steuerung zwei Zielplattformen, den PC oder den Busklemmen-Controller.

Für die Übertragung zum Busklemmen-Controller stehen Ihnen zwei Wege zur Verfügung:

- AMS für BCxx00 (Busklemmen Controller ohne Online Change, eine Task)
- AMS für BCxx50 und BX (Busklemmen Controller mit Online Change, zwei Tasks)
- BC seriell, das serielle Kabel für die Kommunikation über die [RS232-Schnittstelle](#) [► 145] des PCs und die Programmierschnittstelle des Busklemmen-Controllers

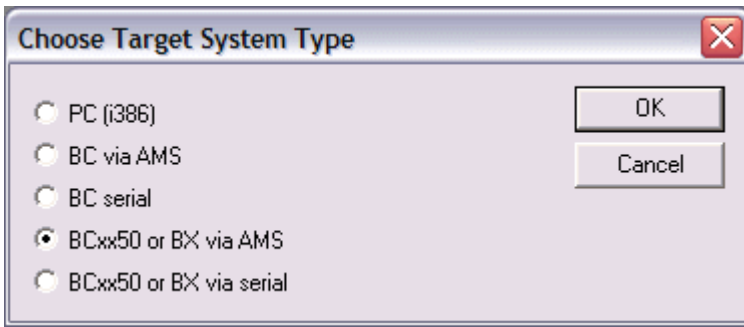


Abb. 138: Auswahl der Zielplattform

Wählen Sie nach Erstellung Ihres Programms unter der Symbolleiste *Online* das Zielsystem aus. Hierzu muss TwinCAT gestartet sein. Im Beispiel ist dies die Ethernet-Karte mit der Box 1 und die Run-Time 1 des Busklemmen-Controllers.

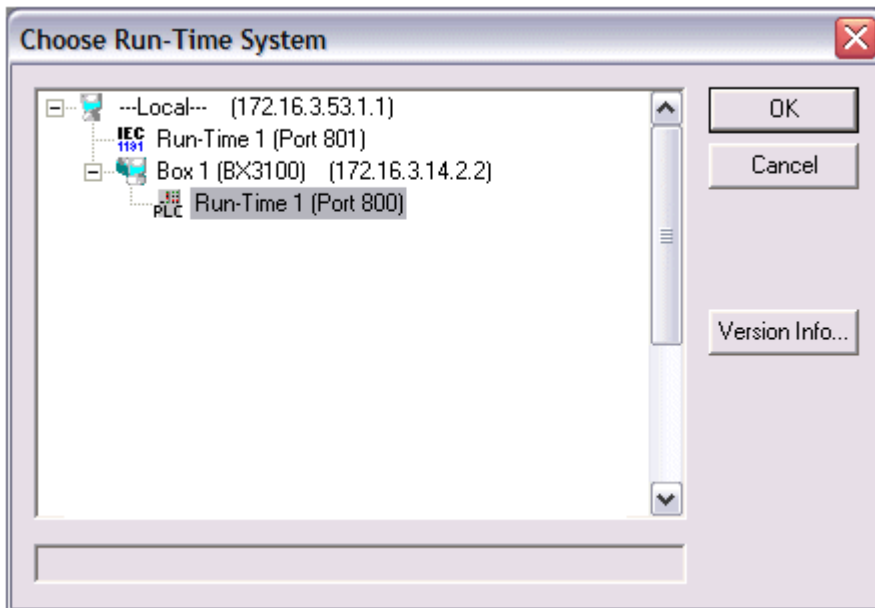


Abb. 139: Auswahl des Zielsystems

## 5.15 Prozessabbild

### 5.15.1 PROFIBUS-Prozessabbild

Der BX3100 ist ein PROFIBUS Slave Gerät. Für die Konfiguration auf der Master Seite steht eine Geräte-Stamm-Datei (GSD) zur Verfügung, die in die jeweilige Master Konfigurations- Software eingebunden sein muss. Auf der Slave Seite, d.h. im BX3100 wird festgelegt welche und wie viele Daten zum PROFIBUS Master übertragen werden sollen. Es können maximal 244 Byte Eingangsdaten und 244 Byte Ausgangsdaten vom BX3100 bearbeitet werden. Zwei Konfigurationsarten stehen zur Verfügung.

Für die DEFAULT CONFIG ist keine besondere Konfiguration notwendig. Die PROFIBUS Daten fangen ab einem bestimmten Adress- Offset an und im Master konfiguriert man die ART und Anzahl der Daten.

Für die TWINCAT CONFIG ist der TwinCAT System Manager notwendig. Hier kann man Byte- weise die PROFIBUS Daten mit den SPS Daten verknüpfen oder direkt Daten vom K-Bus zum PROFIBUS.

Die Eingangsdaten des BX3100 sind Ausgangsdaten des Masters und Ausgangsdaten des BX3100 sind Eingangsdaten des Slaves.

#### DEFAULT CONFIG

In der DEFAULT CONFIG werden alle lokierten Variablen ab der Adresse 1000 und größer, zum Master übertragen. Die Art und die Anzahl der Variablen konfiguriert man auf der Master-Seite.

#### Beispiel

Var\_1 AT %IB1000 :INT; auf der Masterseite muss eine 2 Byte Output Variable konfiguriert sein.  
Var\_2 AT %IB1002 :DINT; auf der Masterseite muss eine 4 Byte Output Variable konfiguriert sein.

Die Reihenfolge der Variablen ist auf der Master-Seite einzuhalten. Lücken in der Adressierung sind nur zulässig, wenn diese auch auf der Masterseite konfiguriert sind. Sonst muss ab Adresse 1000 die Adressierung fortlaufend sein.

#### TWINCAT CONFIG

Für die TWINCAT CONFIG müssen Sie mit dem System Manager eine Konfiguration erstellen. Die Art und Anzahl der Variablen werden im System Manager festgelegt und die Variablen werden dort auch Verknüpft. Deshalb können die PROFIBUS-Daten überall im zulässigen, lokierten Prozessabbild liegen.

Zulässiges Prozessabbild:  
INPUT %IB0 - %IB2047  
OUTPUT %QB0 - %QB2047

## 6 PROFIBUS

### 6.1 Systemvorstellung

#### 6.1.1 Feldbusübersicht

PROFIBUS ist ein herstellerunabhängiger, offener Feldbusstandard mit breitem Anwendungsbereich in der Fertigungs- und Prozessautomatisierung. Herstellerunabhängigkeit und Offenheit sind durch die internationalen Normen EN 50170 und EN 50254 garantiert. PROFIBUS ermöglicht die Kommunikation von Geräten verschiedener Hersteller ohne besondere Schnittstellenanpassungen. PROFIBUS ist sowohl für schnelle, zeitkritische Anwendungen, als auch für komplexe Kommunikationsaufgaben geeignet.

PROFIBUS bietet funktional abgestufte Kommunikationsprotokolle: DP und FMS. Als Übertragungstechniken stehen hierfür, je nach Anwendungsbereich, RS-485, IEC 1158-2 oder Lichtwellenleiter zur Verfügung.

PROFIBUS legt die technischen Merkmale eines seriellen Feldbussystems fest, mit dem verteilte digitale Automatisierungsgeräte von der Feldebene bis zur Zellenebene miteinander vernetzt werden können. PROFIBUS ist ein Multi-Master System und ermöglicht dadurch den gemeinsamen Betrieb von mehreren Automatisierungs-, Engineering- oder Visualisierungssystemen mit den dezentralen Peripheriegeräten an einem Bus.

#### PROFIBUS Gerätetypen

PROFIBUS unterscheidet folgende Gerätetypen:

**Master-Geräte** bestimmen den Datenverkehr auf dem Bus. Ein Master darf Nachrichten ohne externe Aufforderung aussenden, wenn er im Besitz der Buszugriffsberechtigung (Token) ist. Master werden auch als aktive Teilnehmer bezeichnet.

**Slave-Geräte** sind Peripheriegeräte wie beispielsweise Ein-/Ausgangsgeräte, Ventile, Antriebe, Messumformer und die Beckhoff PROFIBUS-Slaves der Reihe BK3xx0, BC3xx0, IPxxx-B310, IL230x-B310 und IL230x-C310. Sie erhalten keine Buszugriffsberechtigung, d.h. sie dürfen nur empfangene Nachrichten quittieren oder auf Anfrage eines Masters Nachrichten an diesen übermitteln. Slaves werden als passive Teilnehmer bezeichnet. Sie benötigen nur einen geringen Anteil des Busprotokolls, dadurch wird eine aufwandsarme Implementierung ermöglicht.

#### PROFIBUS-DP

PROFIBUS-DP ist für den effizienten Datenaustausch in der Feldebene konzipiert. Hier kommunizieren die zentralen Automatisierungsgeräte, wie z. B. SPS/PC oder Prozessleitsysteme, über eine schnelle serielle Verbindung mit dezentralen Feldgeräten wie E/A, Antriebe, Ventile etc. Der Datenaustausch mit den dezentralen Geräten erfolgt vorwiegend zyklisch. Die dafür benötigten Kommunikationsfunktionen sind durch die DP-Grundfunktionen gemäß EN 50170 festgelegt.

Über diese Grundfunktionen hinaus bietet PROFIBUS-DP auch erweiterte azyklische Kommunikationsdienste für zum Beispiel Parametrierung und Bedienung welche auch durch die Beckhoff PROFIBUS-Slaves der Reihe IPxxx-B310, IL230x-B310 und IL230x-C310 unterstützt werden. Eine zentrale Steuerung (Master) liest zyklisch die Eingangsinformationen von den Slaves und schreibt die Ausgangsinformationen zyklisch an die Slaves. Hierbei sollte die Buszykluszeit kürzer sein als die Programmzykluszeit des zentralen Automatisierungssystems, die in vielen Anwendungen etwa 10 ms beträgt.

Ein hoher Datendurchsatz alleine genügt nicht für den erfolgreichen Einsatz eines Bussystems. Vielmehr müssen die einfache Handhabung, gute Diagnosemöglichkeiten und eine störsichere Übertragungstechnik gegeben sein, um die Anforderungen der Anwender zu erfüllen. Bei PROFIBUS-DP wurden diese Eigenschaften optimal kombiniert.

**Systemkonfiguration und Gerätetypen**

Mit PROFIBUS-DP können Mono- oder Multi-Master-Systeme realisiert werden. Dadurch wird ein hohes Maß an Flexibilität bei der Systemkonfiguration ermöglicht. Es können maximal 126 Geräte (Master oder Slaves) an einem Bus angeschlossen werden. Mit den Beckhoff PROFIBUS-Slaves der Reihe IPxxxx-B310, IL230x-B310 und IL230x-C310 kann eine Stationsadresse zwischen 0 und 99 gewählt werden. Die Festlegungen zur Systemkonfiguration beinhalten die Anzahl der Stationen, die Zuordnung der Stationsadresse zu den E/A-Adressen, Datenkonsistenz der E/A-Daten, Format der Diagnosemeldungen und die verwendeten Busparameter. Jedes PROFIBUS-DP-System besteht aus unterschiedlichen Gerätetypen. Es werden drei Gerätetypen unterschieden:

Klasse	Beschreibung
<b>DP-Master Klasse 1 (DPM1)</b> z. B. Beckhoff FC310x: PROFIBUS-Masterkarte für PCs	Hierbei handelt es sich um eine zentrale Steuerung, die in einem festgelegten Nachrichtenzyklus zyklisch Informationen mit den dezentralen Stationen (Slaves) austauscht. Typische Geräte sind z. B. Speicherprogrammierbare Steuerungen (SPS) oder PC.
<b>DP-Master Klasse 2 (DPM2)</b>	Geräte dieses Typs sind Engineering-, Projektierungs- oder Bediengeräte. Sie werden bei der Inbetriebnahme und zur Wartung und Diagnose eingesetzt, um die angeschlossenen Geräte zu konfigurieren, Messwerte und Parameter auszuwerten sowie den Gerätezustand abzufragen.
<b>DP-Slave</b> z. B. Beckhoff IPxxxx-B310: Feldbus Box für PROFIBUS	Ein PROFIBUS-DP-Slave ist ein Peripheriegerät (E/A, Antriebe, Messumformer etc.), das Eingangsinformationen einliest und Ausgangsinformationen an die Peripherie abgibt. Es sind auch Geräte möglich, die nur Eingangs- oder nur Ausgangsinformationen bereitstellen. Die Menge der Eingangs- und Ausgangsinformationen ist geräteabhängig und darf max. 246 Byte Eingangs- und 246 Byte Ausgangsdaten betragen.

**Mono-Master-Systeme**

Bei Mono-Master-Systemen ist in der Betriebsphase des Bussystems nur ein Master am Bus aktiv. Die SPS-Steuerung ist die zentrale Steuerungskomponente. Die Slaves sind über das Übertragungsmedium dezentral an die SPS-Steuerung gekoppelt. Mit dieser Systemkonfiguration wird die kürzeste Buszykluszeit erreicht.

**Multi-Master-Betrieb**

Im Multi-Master-Betrieb befinden sich an einem Bus mehrere Master. Sie bilden entweder voneinander unabhängige Subsysteme, bestehend aus je einem DPM1 und den zugehörigen Slaves, oder zusätzliche Projektierungs- und Diagnosegeräte. Die Eingangs- und Ausgangsabbilder der Slaves können von allen DP-Mastern gelesen werden. Das Schreiben der Ausgänge ist nur für einen DP-Master (den bei der Projektierung zugeordneten DPM1) möglich. Multi-Master-Systeme erreichen eine mittlere Buszykluszeit. In zeitkritischen Anwendungen sollten Sie die Vergrößerung der Buszykluszeit durch Zuschalten eines Diagnosewerkzeugs beachten.

**Gerätstammdatei (GSD)**

Bei PROFIBUS-DP werden die Leistungsmerkmale der Geräte in Form eines Gerätedatenblattes und einer Gerätstammdatendatei von den Herstellern dokumentiert und den Anwendern zur Verfügung gestellt. Aufbau, Inhalt und Kodierung dieser Gerätstammdaten (GSD) sind standardisiert. Sie ermöglichen die komfortable Projektierung beliebiger PROFIBUS-DP-Slaves mit Projektierungsgeräten verschiedener Hersteller. Die PROFIBUS-Nutzer-Organisation (PNO) archiviert diese Informationen herstellerübergreifend und gibt auf Anfrage Auskünfte über die GSD aller Hersteller. Die GSD - Daten werden von einer PROFIBUS-Master - Konfigurationssoftware gelesen und entsprechende Einstellungen in den PROFIBUS-Master übertragen. Die Beschreibung entnehmen Sie bitte dem entsprechenden Softwarehandbuch des Masterherstellers.

Die Beckhoff GSD-Dateien sind über das Internet unter <http://www.beckhoff.de> erhältlich.

**Diagnosefunktionen**

Die umfangreichen Diagnosefunktionen von PROFIBUS-DP ermöglichen die schnelle Fehlerlokalisierung. Die Diagnose der Beckhoff Buskoppler ist in der Defaulteinstellung der Typdatei und der GSD-Datei nicht eingeschaltet. Die Diagnosemeldungen werden über den Bus übertragen und beim Master zusammengefasst.

Sie werden in drei Ebenen eingeteilt:

Diagnoseart	Beschreibung
<b>Stationsbezogen</b>	Meldungen zur allgemeinen Betriebsbereitschaft eines Teilnehmers wie z. B. Übertemperatur oder Unterspannung
<b>Modulbezogen</b>	Diese Meldungen zeigen an, dass innerhalb eines bestimmten E/A Teilbereichs (z. B. 8 Bit Ausgangs - Modul) eines Teilnehmers eine Diagnose ansteht
<b>Kanalbezogen</b>	Hier wird die Fehlerursache bezogen auf ein einzelnes Ein- / Ausgangs - Bit (Kanal) angegeben, wie z. B. Kurzschluss auf Ausgang 2

Die Beckhoff PROFIBUS-Slaves der Reihe IPxxxx-B310, IL230x-B310 und IL230x-C310 unterstützen die Diagnosefunktionen des PROFIBUS-DP. Die Auswertung der Diagnosedaten über die Steuerung hängt von der Unterstützung des PROFIBUS-Masters ab. Entnehmen Sie bitte den Gerätehandbüchern der Masteranschlüssen die Handhabung der Diagnose.

**Sync- und Freeze-Mode**

Zusätzlich zu dem Teilnehmer bezogenen Nutzdatenverkehr, der automatisch vom DPM1 abgewickelt wird, besteht für die DP-Master die Möglichkeit, Steuerkommandos an einen, eine Gruppe oder an alle DP-Slaves gleichzeitig zu senden. Diese Steuerkommandos werden als Multicast übertragen. Mit diesen Steuerkommandos können die Sync- und Freeze-Betriebsarten zur Synchronisation der DP-Slaves vorgegeben werden. Sie ermöglichen eine ereignisgesteuerte Synchronisation der DP-Slaves.

Die DP-Slaves beginnen den **Sync-Mode**, wenn sie vom zugeordneten DP-Master ein Sync-Steuerkommando empfangen. In diesem Betriebszustand werden bei allen adressierten DP-Slaves die Ausgänge auf den momentanen Zustand eingefroren. Bei den folgenden Nutzdatenübertragungen werden die Ausgangsdaten bei den DP-Slaves gespeichert, die Ausgangszustände bleiben jedoch unverändert. Erst beim Empfang des nächsten Sync-Steuerkommandos vom Master werden die gespeicherten Ausgangsdaten an die Ausgänge durchgeschaltet. Mit einem Unsync-Steuerkommando wird der Sync-Betrieb beendet.

Analog dazu bewirkt ein Freeze-Steuerkommando den **Freeze-Mode** der angesprochenen DP-Slaves. In dieser Betriebsart werden die Zustände der Eingänge auf den momentanen Wert eingefroren. Die Eingangsdaten werden erst dann wieder aktualisiert, wenn der DP-Master das nächste Freeze-Steuerkommando an die betroffenen Geräte gesendet hat. Mit Unfreeze wird der Freeze-Betrieb beendet.

**Systemverhalten**

Um eine weitgehende Geräte austauschbarkeit zu erreichen, wurde bei PROFIBUS-DP auch das Systemverhalten standardisiert. Es wird im Wesentlichen durch den Betriebszustand des DPM1 bestimmt. Dieser kann entweder lokal oder über den Bus vom Projektierungs-Gerät gesteuert werden.

Es werden folgende drei Hauptzustände unterschieden:

Betriebsart	Beschreibung
<b>Stop</b>	Es findet kein Datenverkehr zwischen dem DPM1 und den DP-Slaves statt. Der Buskoppler spricht die Busklemmen nur einmal nach dem Einschalten der Versorgungsspannung an (keine der E/A - LEDs leuchtet).
<b>Clear</b>	Der DPM1 liest die Eingangsinformationen der DP-Slaves, und hält die Ausgänge der DP-Slaves im sicheren Zustand (Abhängig von der Reaktion auf Feldbusfehler leuchtet die grüne E/A-LED und werden die Ausgänge gesetzt).
<b>Operate</b>	Der DPM1 befindet sich in der Datentransferphase. In einem zyklischen Datenverkehr werden die Eingänge von den DP - Slaves gelesen und die Ausgangsinformationen an die DP-Slaves übertragen (Die grüne E/A-LED leuchtet).



Der DPM1 sendet seinen lokalen Status in einem konfigurierbaren Zeitintervall mit einem Multicast-Kommando zyklisch an alle ihm zugeordneten DP-Slaves. Die Systemreaktion nach dem Auftreten eines Fehlers in der Datentransferphase des DPM1, wie z. B. Ausfall eines DP-Slaves, wird durch den Betriebsparameter *Auto-Clear* bestimmt. Wurde dieser Parameter auf *True* gesetzt, dann schaltet der DPM1 die Ausgänge aller zugehörigen DP-Slaves in den sicheren Zustand, sobald ein DP-Slave nicht mehr bereit für die Nutzdatenübertragung ist. Danach wechselt der DPM1 in den Clear-Zustand. Ist dieser Parameter *False*, dann verbleibt der DPM1 auch im Fehlerfall im Operate-Zustand und der Anwender kann die Systemreaktion selbst bestimmen.

**Datenverkehr zwischen DPM1 und den DP-Slaves**

Der Datenverkehr zwischen dem DPM1 und den ihm zugeordneten DP-Slaves wird in einer festgelegten immer wiederkehrenden Reihenfolge automatisch durch den DPM1 abgewickelt. Bei der Projektierung des Bussystems legt der Anwender die Zugehörigkeit eines DP-Slaves zum DPM1 fest. Weiterhin wird definiert, welche DP-Slaves in den zyklischen Nutzdatenverkehr aufgenommen oder ausgenommen werden sollen.

Der Datenverkehr zwischen dem DPM1 und den DP-Slaves gliedert sich in die Phasen Parametrierung, Konfigurierung und Datentransfer.

Bevor ein DP-Slave in die Datentransferphase aufgenommen wird, prüft der DPM1 in der Parametrierungs- und Konfigurations-Phase, ob die projektierte Sollkonfiguration mit der tatsächlichen Gerätekonfiguration übereinstimmt. Bei dieser Überprüfung müssen der Gerätetyp, die Format- und Längeninformationen sowie die Anzahl der Ein- und Ausgänge übereinstimmen. Der Benutzer erhält dadurch einen zuverlässigen Schutz gegen Parametrierungsfehler. Zusätzlich zum Nutzdatentransfer, der vom DPM1 automatisch durchgeführt wird, besteht die Möglichkeit neue Parametrierungsdaten auf Anforderung des Benutzers an die DP-Slaves zu senden.

**Schutzmechanismen**

Im Bereich der dezentralen Peripherie ist es aus Sicherheitsgründen erforderlich, die Systeme mit hochwirksamen Schutzfunktionen gegen Fehlparametrierung oder Ausfall der Übertragungseinrichtungen zu versehen. PROFIBUS-DP verwendet Überwachungsmechanismen beim DP-Master und bei den DP-Slaves. Sie werden als Zeitüberwachungen realisiert. Das Überwachungsintervall wird bei der Projektierung des DP-Systems festgelegt.

<b>Schutzmechanismen</b>	<b>Beschreibung</b>
<b>Auf dem DP-Master</b>	Der DPM1 überwacht den Nutzdatentransfer der Slaves mit dem <i>Data_Control_Timer</i> . Für jeden zugeordneten Slave wird ein eigener Überwachungszeitgeber benutzt. Die Zeitüberwachung spricht an, wenn innerhalb eines Überwachungsintervalls kein ordnungsgemäßer Nutzdatentransfer erfolgt. In diesem Fall wird der Benutzer informiert. Falls die automatische Fehlerreaktion ( <i>Auto_Clear = True</i> ) freigegeben wurde, verlässt der DPM1 den Zustand <i>Operate</i> , schaltet die Ausgänge der zugehörigen Slaves in den sicheren Zustand und geht in den Betriebszustand <i>Clear</i> über.
<b>Auf dem DP-Slave</b>	Der Slave führt zur Erkennung von Fehlern des Masters oder der Übertragungstrecke die Ansprechüberwachung durch. Findet innerhalb des Ansprechüberwachungsintervalls kein Datenverkehr mit dem zugeordneten Master statt, dann schaltet der Slave die Ausgänge selbständig in den sicheren Zustand. Zusätzlich ist für die Ein- und Ausgänge der Slaves beim Betrieb in Multi - Master - Systemen ein Zugriffsschutz erforderlich damit sichergestellt ist, dass der direkte Zugriff nur vom berechtigten Master erfolgt. Für alle anderen Master stellen die Slaves ein Abbild der Eingänge und Ausgänge zur Verfügung, das von jedem beliebigen Master auch ohne Zugriffsberechtigung gelesen werden kann.

**Ident-Nummer**

Jeder DP-Slave und jeder DPM1 muss eine individuelle Ident-Nummer haben. Sie wird benötigt, damit ein DP-Master ohne signifikanten Protokoll-Overhead die Typen der angeschlossenen Geräte identifizieren kann. Der Master vergleicht die Ident-Nummer der angeschlossenen DP-Geräte mit den Ident-Nummern in den vom DPM2 vorgegebenen Projektierungsdaten. Der Nutzdatentransfer wird nur dann begonnen, wenn die richtigen Geräte-Typen mit den richtigen Stationsadressen am Bus angeschlossen wurden. Dadurch wird

Sicherheit gegenüber Projektierungsfehlern garantiert. Die Vergabe der Herstellerspezifischen Ident-Nummern erfolgt durch die PROFIBUS Nutzerorganisation (PNO). Die PNO verwaltet die Ident-Nummern zusammen mit den Gerätestammdaten (GSD).

## 6.1.2 PROFIBUS DP

In PROFIBUS DP-Systemen kommuniziert in der Regel ein Master (SPS, PC, etc.) mit vielen Slaves (I/Os, Antriebe, etc.), wobei nur der Master aktiv auf den Bus zugreifen (unaufgefordert Telegramme senden), während ein DP-Slave nur Telegramme sendet, wenn er von einem Master dazu aufgefordert wurde.

### DP-StartUp

Bevor Master und Slave zyklischen Datenaustausch miteinander durchführen, werden während des DP-StartUps die Parameter- und Konfigurationsdaten vom Master an die Slaves übertragen. Nach dem Senden der Parameter- und Konfigurationsdaten fragt der Master solange die Diagnosedaten des Slaves ab, bis der Slave seine Bereitschaft zum Datenaustausch signalisiert. Je nach Umfang der Berechnungen, die der Slave durch den Empfang von Parameter- und Konfigurationsdaten durchzuführen hat, bis er bereit zum Datenaustausch ist, kann das bis zu einigen Sekunden dauern. Der Slave besitzt daher die folgenden Zustände:

### Parameterdaten

Die Parameterdaten werden mit dem SetPrmLock-Request-Telegramm von dem Master an die Slaves gesendet, das SetPrmLock-Response-Telegramm enthält keine Daten und besteht daher nur aus einem Byte, der Kurzquittung. Die Parameterdaten bestehen aus DP-Parametern (z. B. Einstellung des DP-Watchdogs, Überprüfung der IdentNumber (eindeutig für jedes DP-Gerät)), aus DPV1-/DPV2-Parametern und aus anwendungsspezifischen Parametern, die nur einmal während des StartUps übertragen werden müssen. Wenn ein Fehler in den Parameterdaten auftritt, wird das in den Diagnosedaten gekennzeichnet und der Slave bleibt bzw. geht in den Zustand WAIT-PRM.

### Konfigurationsdaten

Die Konfigurationsdaten werden mit dem ChkCfg-Request-Telegramm von dem Master an die Slaves gesendet, das ChkCfg-Response-Telegramm enthält keine Daten und besteht daher nur aus einem Byte, der Kurzquittung. Die Konfigurationsdaten beschreiben die Zuordnung der DP-Module zu den zyklischen I/O-Daten, die mit dem Data\_Exchange-Telegramm während des zyklischen Datenaustauschs zwischen Master und Slave und ausgetauscht werden. Die Reihenfolge der im DP-Konfigurationstool an einen Slave angefügten DP-Module bestimmt die Reihenfolge der zugehörigen I/O-Daten im Data\_Exchange-Telegramm.

### Diagnosedaten

Die Diagnosedaten werden mit einem SlaveDiag-Request-Telegramm ohne Daten vom Master angefordert, der Slave sendet die Diagnosedaten mit einem SlaveDiag-Response-Telegramm. Die Diagnosedaten bestehen aus der Standard DP-Diagnose (z. B. Zustand des Slaves, IdentNumber) und anwendungsspezifischen Diagnosedaten.

### Zyklischer Datenaustausch

Kernstück des PROFIBUS DP-Protokolls ist der zyklische Datenaustausch, bei dem innerhalb eines PROFIBUS DP-Zyklus der Master mit jedem Slave einen I/O-Datenaustausch durchführt. Dabei sendet der Master an jeden Slave die Outputs mit einem DataExchange-Request-Telegramm, der Slave antwortet mit den Inputs in einem DataExchange-Response-Telegramm. Sämtliche Output- bzw. Inputdaten werden also mit je einem Telegramm übertragen, wobei die DP-Konfiguration (Reihenfolge der DP-Module) die Zuordnung der Output- bzw. Inputdaten zu den realen Prozessdaten des Slaves festlegt.

### Diagnose während des zyklischen Datenaustausches

Während des zyklischen Datenaustausches kann ein Slave eine Diagnose an den Master melden. In dem Fall setzt der Slave ein Flag im DataExchange-Response-Telegramm, woran der Master erkennt, dass bei dem Slave neue Diagnosedaten vorliegen, die er dann mit dem SlaveDiag-Telegramm abholt. Die Diagnosedaten sind also nicht in Echtzeit mit den zyklischen I/O-Daten in der Steuerung, sondern mindestens immer einen DP-Zyklus später.

### Synchronisierung mit Sync und Freeze

Mit den Sync- und Freeze-Kommandos im GlobalControl-Request-Telegramm (Broadcast-Telegramm) kann der Master die Ausgabe der Outputs (Sync) bzw. das Einlesen der Inputs (freeze) bei mehreren Slaves synchronisieren. Bei Verwendung des Sync-Kommandos werden die Slaves zunächst in den Sync-Mode geschaltet (wird in den Diagnosedaten quittiert), dann werden sequentiell die I/O-Daten per DataExchange-Telegramm mit den Slaves ausgetauscht, ein Senden des Sync-Kommandos im GlobalControl-Telegramm hat dann zur Folge, dass die Slaves die zuletzt empfangenen Outputs ausgeben. Im Freeze-Betrieb wird zunächst ein Freeze-Kommando im GlobalControl-Telegramm gesendet, woraufhin alle Slaves ihre Inputs latchen, die dann wiederum mit dem DataExchange-Telegramm vom Master sequentiell abgeholt werden.

### Zustände im Master

Der Master unterscheidet die Zustände CLEAR (alle Outputs auf dem Fail\_Safe-Wert) und OPERATE (alle Outputs haben den Prozesswert). In der Regel wird der Master in den CLEAR-Mode geschaltet, wenn z. B. die SPS auf STOP geht.

### DP-Master Class 1 und Class 2

Mit dem Class 1-Master wird die Steuerung bezeichnet, die zyklischen I/O-Datenaustausch mit den Slaves durchführt, während ein Class 2-Master ein B&B-Gerät ist, das in der Regel nur lesend auf die I/O-Daten des Slaves zugreift.

## 6.1.3 PROFIBUS DPV1

PROFIBUS DPV1 bezeichnet im Wesentlichen die azyklischen Read- und Write-Telegramme, mit denen azyklisch auf Datensätze im Slave zugegriffen wird. Auch bei DPV1 wird zwischen Class 1 und einem Class 2 Master unterschieden. Die azyklische Class 1 (C1) bzw. Class 2 (C2) Verbindungen unterscheiden sich dadurch, dass die azyklische C1-Verbindung mit dem DP-StartUp des zyklischen DP-Betriebs aufgebaut wird (ab dem Zustand WAIT-CFG des Slaves können azyklische DPV1-C1-Read- und Write-Telegramme vom Master zum Slave gesendet werden, während die C2-Verbindung einen separaten Verbindungsaufbau unabhängig von der zyklischen DP-Verbindung hat, der in der Regel von einem zweiten C2-Master durchgeführt wird, so dass z. B. ein herstellerspezifisches Projektierungs- und Diagnosetool auf die Daten des Slaves zugreifen kann.

Bei Verwendung von zwei Mastern ist allerdings immer zu beachten, dass diese sich den Buszugriff teilen (es wird ein Token ausgetauscht), so dass die zeitlichen Verhältnisse ungünstiger als bei einem Mono-Master-System sind.

## 6.1.4 Stecker, Kabel und Schalter

### Die Physik der Übertragung

Die physikalische Datenübertragung ist in der PROFIBUS-Norm definiert. Siehe PROFIBUS Schicht 1 (Physical Layer).

Der Einsatzbereich eines Feldbus - Systems wird wesentlich durch die Wahl des Übertragungs- Mediums und der physikalischen Busschnittstelle bestimmt. Neben den Anforderungen an die Übertragungssicherheit sind die Aufwendungen für Beschaffung und Installation des Buskabels von entscheidender Bedeutung. Die PROFIBUS-Norm sieht daher verschiedene Ausprägungen der Übertragungstechnik unter Beibehaltung eines einheitlichen Busprotokolls vor.

Kabelgebundene Übertragung: Diese Version, gemäß dem US-Standard EIA RS-485, wurde als Grundversion für Anwendungen im Bereich der Fertigungstechnik, Gebäudeleittechnik und Antriebstechnik festgelegt. Es wird ein verdrehtes Kupferkabel mit einem Leiterpaar verwendet. Die Abschirmung kann in Abhängigkeit des beabsichtigten Einsatzgebietes (EMV-Gesichtspunkte beachten) entfallen.

### Kabelbedingte Störungen

Es stehen zwei Leitungstypen mit unterschiedlichen Höchstleitungslängen zur Verfügung, siehe Tabelle „RS485“. Die Belegung des Steckverbinders und die Verdrahtung sind unten im Bild dargestellt. Beachten Sie die besonderen Anforderungen an das Datenkabel bei Baud-Raten von mehr als 1,5 Mbaud. Das richtige Kabel ist Grundvoraussetzung für den störungsfreien Betrieb des Bussystems. Bei der Verwendung eines einfachen 1,5 Mbaud-Kabels kann es durch Reflexionen und zu großer Dämpfung zu erstaunlichen Phänomenen kommen. Das kann zum Beispiel sein: eine beliebige Station ist ohne Verbindung und durch Abziehen der benachbarten Station nimmt diese die Verbindung wieder auf. Oder es kommt zu Übertragungsfehlern, wenn ein bestimmtes Bitmuster übertragen wird. Das kann bedeuten, dass der PROFIBUS ohne Funktion der Anlage störungsfrei arbeitet und nach dem Hochlauf zufällig Busfehler meldet. Die Reduzierung der Baud-Rate (< 93,75 kbaud) beseitigt das geschilderte Fehlverhalten.

Führt die Verringerung der Baud-Rate nicht zur Beseitigung des Fehlers, liegt in häufigen Fällen ein Verdrahtungsfehler vor. Die beiden Datenleitungen sind an einem oder mehreren Steckern gedreht oder die Abschlusswiderstände sind nicht eingeschaltet oder an falschen Stellen aktiviert.

#### **i** Vorkonfektionierte Kabel von Beckhoff

Mit den vorkonfektionierten Kabeln von Beckhoff vereinfacht sich die Installation erheblich. Verdrahtungsfehler werden vermieden und die Inbetriebnahme führt schneller zum Erfolg. Das Programm umfasst Feldbuskabel, Stromversorgungskabel und Sensorkabel sowie Zubehör wie Abschlusswiderstände und T-Stücke. Ebenso sind jedoch auch Feldkonfektionierbare Stecker und Kabel erhältlich.

### PROFIBUS-Anschluss der Feldbus Box Module

Die M12 Buchse ist invers Codiert und besitzt 5 Kontaktstifte. Stift 1 überträgt 5 V<sub>DC</sub> und Stift 3 GND für den aktiven Abschlusswiderstand. Diese dürfen auf keinen Fall für andere Funktionen missbraucht werden, da dies zu Zerstörung des Gerätes führen kann. Stift 2 und Stift 4 übertragen die Signale des PROFIBUS. Diese dürfen auf keinen Fall getauscht werden, da sonst die Kommunikation gestört ist. Stift 5 überträgt den Schirm der kapazitiv mit der Grundfläche der Feldbus Box verbunden ist.

### Pinbelegung PROFIBUS-Buchse

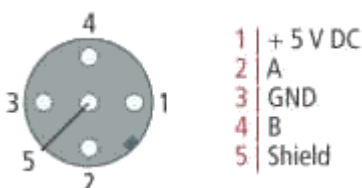


Abb. 140: Pinbelegung M12 Buchse

**Leitungsfarben PROFIBUS**

PROFIBUS-Leitung	M12	D-Sub
B rot	Stift 4	Stift 3
A grün	Stift 2	Stift 8

**RS485 - Grundlegende Eigenschaften**

RS-485 Übertragungstechnik nach PROFIBUS-Norm	
Netzwerk Topologie	Linearer Bus, aktiver Busabschluss an beiden Enden, Stichleitungen sind möglich
Medium	Abgeschirmtes verdrehtes Kabel, Schirmung darf abhängig von den Umgebungsbedingungen (EMV) entfallen
Anzahl der Stationen	32 Stationen in jedem Segment ohne Repeater. Mit Repeater erweiterbar bis 127 Stationen
Max. Bus Länge ohne Repeater	100 m bei 12 MBit/s 200 m bei 1500 KBit/s, bis zu 1,2 km bei 93,75 KBit/s
Max. Bus Länge mit Repeater	Durch Leitungsverstärker (Repeater) kann die max. Buslänge bis in den 10 km Bereich vergrößert werden. Die Anzahl der möglichen Repeater ist mindestens 3 und kann je nach Hersteller bis zu 10 betragen
Übertragungsgeschwindigkeit	9,6, 19,2, 93,75, 187,5, 500, 1500 KBit/s, bis 12 MBit/s in Stufen einstellbar
Steckverbinder	9-Pin D-Sub-Steckverbinder für IP20 M12 Rundsteckverbinder für IP65/67

**Verkabelung für PROFIBUS-DP und PROFIBUS\_FMS**

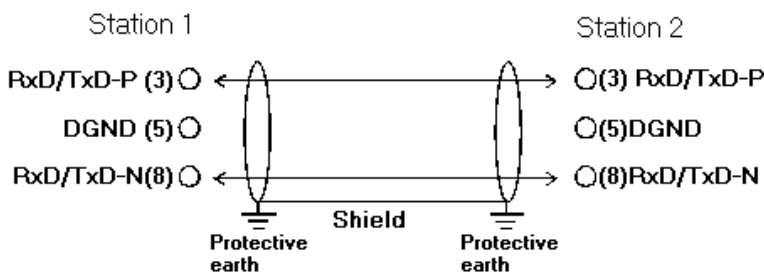


Abb. 141: Belegung des PROFIBUS-Kabels

**● Abschlusswiderstände**

**i** In Systemen mit mehr als zwei Stationen werden alle Teilnehmer parallel verdrahtet. An den Leitungsenden muss das PROFIBUS-Kabel in jedem Fall mit Widerständen abgeschlossen werden, um Reflexionen und damit Übertragungsprobleme zu vermeiden.

**Adressierung**

**Einstellung der Stationsadressen**

Die Adresse bei PROFIBUS ist über die zwei Drehwahlschalter hinter der Klarsichtabdeckung einzustellen. Die Default Einstellung beträgt 11. Es sind alle Adressen erlaubt, dabei darf jede Adresse im Netzwerk nur einmal vorkommen. Die Adresse wird im ausgeschalteten Zustand der Feldbus Box (Buskoppler) geändert. Dafür lösen Sie die Abdeckung (nur Feldbus Box), und verändern Sie mit Hilfe eines Schraubendrehers die Schalter auf die gewünscht Position. Dabei ist zu beachten, dass die Schalter richtig einrasten. Die Adressänderung wird aktiv, sobald das Gerät eingeschaltet wird.

**Adresse Feldbus Box**

Der linke Schalter ist der Zehner-Multiplikator und der rechte Schalter der Einer-Multiplikator.



Abb. 142: Adresswahlschalter

### Adresse des Buskopplers

Der untere Schalter S311 ist der Zehner-Multiplikator und der obere Schalter S310 der Einer-Multiplikator.

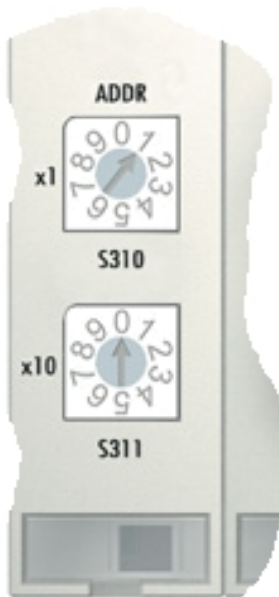


Abb. 143: Adresswahlschalter

## 6.1.5 Topologie

- Ein Bussegment darf aus maximal 32 Teilnehmern bestehen (einschließlich den Repeatern).
- Die maximale Leitungslänge eines Segmentes ist abhängig von der verwendeten Übertragungsgeschwindigkeit und der Qualität der verwendeten Busleitung.
- Zwischen zwei Teilnehmer dürfen maximal 9 Repeater installiert werden.
- Stichleitungen sind zu vermeiden und ab 1,5 Mbaud nicht zulässig.
- Maximale Anzahl an Teilnehmern 127
- Unterbrechungen der Versorgungsspannungen von Leitungsanschlüssen durch abschalten des Repeaters/Slave oder abziehen des Steckers ist nicht zulässig.

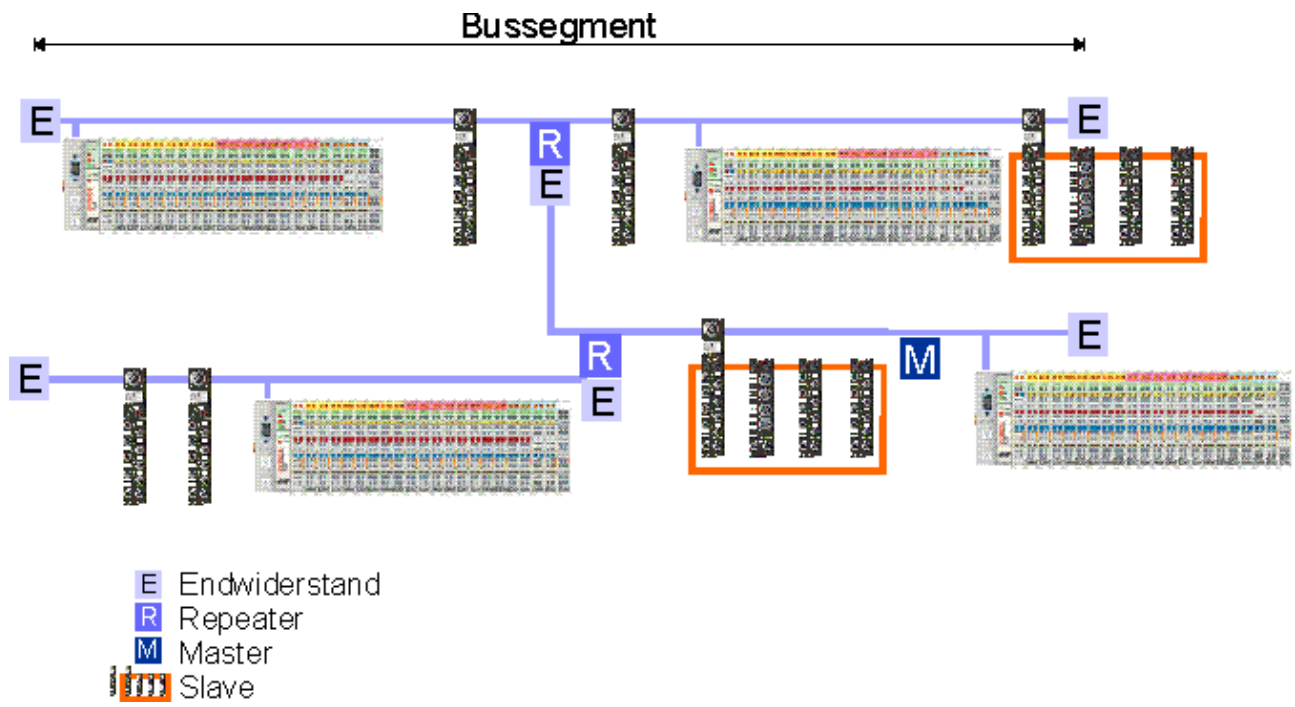


Abb. 144: RS485-Topologie mit 3 Segmenten und 2 Repeatern

## 7 Fehlerbehandlung und Diagose

### 7.1 Diagnose

#### Zustand des PROFIBUS

In vielen Fällen ist es wichtig zu wissen ob die Kommunikation mit dem übergeordneten Master noch funktioniert. Verknüpfen Sie hierfür die "DpState" Variable mit Ihrem SPS-Programm.

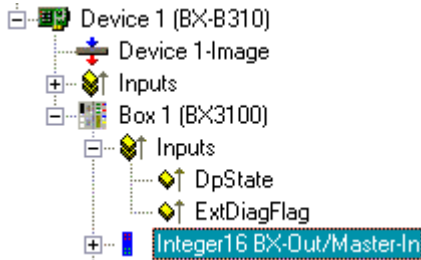


Abb. 145: PROFIBUS Diagnose Byte im System Manager

Fehler Nummer	Beschreibung	Abhilfe
0	kein Fehler	-
129	Wartet auf Konfigurationsdaten	PROFIBUS starten
130	Wartet auf Parameterdaten	PROFIBUS starten
131	kein Master vorhanden	PROFIBUS Kabel und Stecker überprüfen

#### Beispiel

Wenn der PROFIBUS unterbrochen wird, zum Beispiel weil das Kabel abgezogen oder die SPS ausgeschaltet wurde, dann meldet das der Busklemmen-Controller mit 130 im DP-State. Das heißt, der Busklemmen-Controller wartet auf Parameterdaten vom Master.

*Reihenfolge wie ein Slave nach einem TimeOut oder starten des Masters hochbootet:*

Parameterdaten - Konfigurationsdaten - Datenaustausch

#### Auslesen des Feldbusstatus per ADS

In der Default Konfiguration oder in TwinCAT Konfiguration kann der Feldbusstatus über ADSREAD ausgelesen werden.

Parameter ADSREAD Baustein	Beschreibung
NetID	lokal - Leerstring
Port	1
IndexGroup	16#0006
IndexOffset	BX3100: 16#000C_A0F4 BC3150: 16#000C_A080
LEN	1

#### Zustand des K-Bus

Sollte der interne Bus oder einer seiner Busklemmen ein Problem haben wird dies im K-Bus-State angezeigt. Eine genauere Fehlerursache kann mit einem Funktionsbaustein ausgelesen werden (in Vorbereitung). Verknüpfen Sie hierfür die "K-Bus-State" Variable mit Ihrem SPS-Programm.



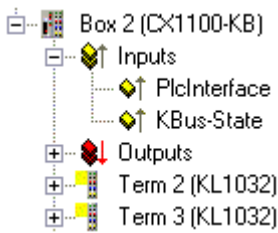


Abb. 146: K-Bus-Status

Fehlerbit	Beschreibung	Fehlerart
0	kein Fehler	Kein FEHLER
Bit 0	K-Bus Fehler	FEHLER
Bit 2	K-Bus wird Nachgetriggert	HINWEIS

### Auslesen des K-Bus Status per ADS

In der Default Konfiguration oder in TwinCAT Konfiguration kann der Feldbusstatus über ADSREAD ausgelesen werden.

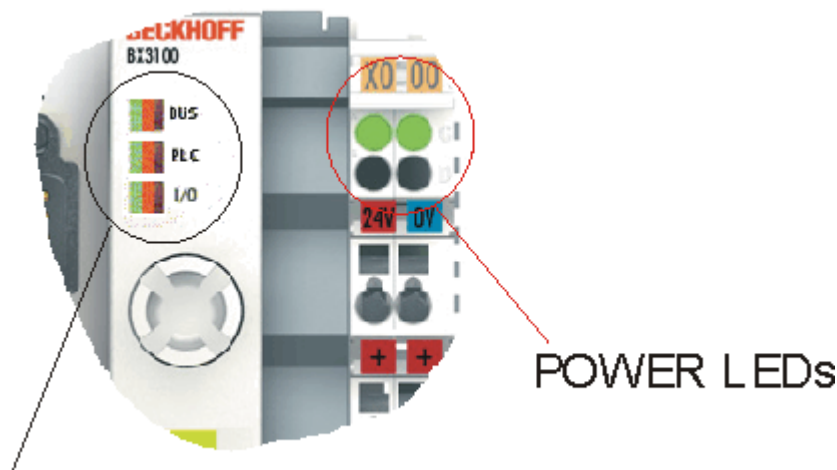
Parameter ADSREAD Baustein	Beschreibung
NetID	lokal - Leerstring
Port	1
IndexGroup	16#0006
IndexOffset	16#000C_9000
LEN	1

## 7.2 Diagnose-LEDs

Zur Statusanzeige besitzt der Buskoppler zwei Gruppen von LEDs. Die linken Gruppe "DIAG LEDs" geben einen Zustand des Feldbusses, der PLC und des K-Busses an. Auf der rechten oberen Seite des Buskopplers befinden sich zwei grüne LEDs (Power LEDs). Diese dienen der Anzeige der Versorgungsspannung des BX-Contollers und der 24 V<sub>DC</sub> der Powerkontakte.

### LEDs zur Diagnose der Spannungsversorgung

LED (Power LEDs)	Bedeutung
Linke LED aus	Buskoppler hat keine Spannung 24 V <sub>DC</sub>
Rechte LED aus	Keine Spannungsversorgung 24 V <sub>DC</sub> an den Powerkontakten angeschlossen



## DIAG LEDs

Abb. 147: Diagnose LEDs für den Feldbus, die SPS, den K-Bus und die Spannungsversorgungen

Die DIAG LEDs sind unterteilt in:

- Bus: Feldbus-Diagnose
- PLC: SPS-Diagnose
- I/O: K-Bus-Diagnose

Die LEDs können die Zustände aus, grün, orange und rot annehmen.



Abb. 148: Diagnose LEDs für den Feldbus, die SPS und den K-Bus

Nach dem Einschalten überprüft der Buskoppler sofort die angeschlossene Konfiguration. Der fehlerfreie Hochlauf wird durch das Verlöschen der LED I/O signalisiert. Das rote Leuchten der LED I/O zeigt einen Fehler im Bereich der Busklemmen an. Die Fehlerart wird im Display angezeigt. Das ermöglicht eine schnelle Fehlerbeseitigung.

### LED BUS - Feldbus-Diagnose

LED	Bedeutung
LED aus	kein Feldbusangeschlossen, Buskoppler sucht die Baudrate
LED rot	Fehler Blinken - Fehlerart - Anzeige Display
LED orange	Buskoppler hat Baudrate gefunden, wartet auf Config- und Parameterdaten
LED grün	Buskommunikation ist in Ordnung, BX-Controller ist im Datenaustausch

**LED PLC - SPS-Diagnose**

<b>LED</b>	<b>Bedeutung</b>
LED aus	PLC Stop oder kein Programm vorhanden
LED rot	blinken - die eingestellte Taskzeit wird ab und zu überschritten an - die eingestellte Taskzeit wird immer überschritten Disable der roten LED
LED orange	PLC ohne Bootprojekt läuft (nur während des Zyklus an), beim Erzeugen des Bootprojektes blinkt die LED orange
LED grün	Bootprojekt - PLC läuft (nur während des Zyklus an)

**LED I/O - K-Bus-Diagnose**

<b>LED</b>	<b>Bedeutung</b>
LED aus	über den K-Bus werden keine Daten ausgetauscht
LED rot	Fehler Blinken - Fehlerart - Anzeige Display
LED orange	Register- bzw. KS2000-Online-Zugriff
LED grün	K-Bus ist in Ordnung und läuft

## Fehler-Codes zur K-Bus-Diagnose

Fehler-Code	Fehler-Argument	Beschreibung	Abhilfe
0	-	EMV-Probleme	<ul style="list-style-type: none"> <li>Spannungsversorgung auf Unter- oder Überspannungsspitzen kontrollieren</li> <li>EMV-Maßnahmen ergreifen</li> <li>Liegt ein K-Bus-Fehler vor, kann durch erneutes Starten (Aus- und Wiedereinschalten des BX-Controllers) der Fehler lokalisiert werden</li> </ul>
1	0	EEPROM-Prüfsummenfehler	Herstellereinstellung mit der Konfigurationssoftware KS2000 setzen
	1	Überlauf im Code Buffer	Weniger Busklemmen stecken. Bei prog. Konfiguration sind zu viele Einträge in der Tabelle
	2	Unbekannter Datentyp	Software-Update des BX-Controllers notwendig
2	-	Reserve	-
3	0	K-Bus-Kommandofehler	<ul style="list-style-type: none"> <li>Keine Busklemme gesteckt</li> <li>Eine der Busklemmen ist defekt, angehängte Busklemmen halbieren und prüfen ob der Fehler bei den übrigen Busklemmen noch vorhanden ist. Dies weiter durchführen, bis die defekte Busklemme lokalisiert ist.</li> </ul>
	n	Bruchstelle hinter Busklemme n	Kontrollieren ob die Busendklemme KL9010 gesteckt ist
4	0	K-Bus-Datenfehler, Bruchstelle hinter dem BX-Controllers	Prüfen ob die n+1 Busklemme richtig gesteckt ist, gegebenenfalls tauschen
	n	Bruchstelle hinter Busklemme n	Kontrollieren ob die Busendklemme KL9010 gesteckt ist
5	n	K-Bus-Fehler bei Register-Kommunikation mit Busklemme n	n-te Busklemme tauschen
6	0	Fehler bei der Initialisierung	BX-Controllers austauschen
	1	Interner Datenfehler	Hardware-Reset des BX-Controllers (Aus - und Wiedereinschalten)
	2	DIP-Schalter nach einem Software-Reset verändert	Hardware-Reset des BX-Controllers (Aus - und Wiedereinschalten)
	3	IP Adressen Kollision	Prüfen Sie ob die IP-Adresse schon im Netzwerk vorhanden ist.
7	0	Hinweis: Zykluszeit wurde überschritten	Warning: Die eingestellte Zykluszeit wurde überschritten. Dieser Hinweis (blinken der LEDs) kann nur durch erneutes booten des BX-Controllers gelöscht werden. Abhilfe: Zykluszeit erhöhen
9	0	Checksummenfehler im Programm-Flash	Programm erneut zum BX-Controllers übertragen
	1	Falsche oder fehlerhaft Bibliothek implementiert	Entfernen Sie die fehlerhaft Bibliothek
10	n	Die Busklemme n stimmt nicht mit der Konfiguration, die beim Erstellen des Boot-Projektes existiert überein	Die n-te Busklemme überprüfen. Sollte eine n-te Busklemme gewollt eingefügt worden sein, muss das Bootprojekt gelöscht werden.
14	n	n-te Busklemme hat das falsche Format	BX-Controller erneut starten, falls der Fehler erneut auftritt, die Busklemme tauschen.
15	n	Anzahl der Busklemmen stimmt nicht mehr	BX-Controller erneut starten. Falls der Fehler erneut auftritt, BX-Controller mit der Konfigurationssoftware auf Auslieferungszustand zurücksetzen.
16	n	Länge der K-Bus-Daten stimmt nicht mehr	BX-Controller erneut starten. Falls der Fehler erneut auftritt, BX-Controller mit der Konfigurationssoftware auf Auslieferungszustand zurücksetzen.

## 7.3 Diagnose-Display

Während des Hochstarten zeigt das Display für ca. drei Sekunden die aktuelle Firmware-Version an.

Sollte beim Starten ein Fehler vorliegen wird ihnen dieser per Blink-Sequenz der entsprechenden LED angezeigt.

Konfigurationsfehler werden im Display mit TC-Config und einer Fehlernummer angegeben. Bitte überprüfen Sie in diesem Fall Ihre Hardwarekonfiguration im System Manager oder setzen Sie sich mit dem Support in Verbindung.

Anzeige Display	Bedeutung
TC-Config 0xE02E	Eine komplexe Busklemme liegt auf einer Bitadresse. Überprüfen Sie die TwinCAT Konfiguration.
TC-Config 0xF0nn	Die nn' te Busklemme stimmt nicht mit der Konfiguration überein. Vergleichen Sie für die nn'te Busklemme den Busaufbau mit der Konfiguration.
TC-Config 0xC0nn	Die nn' te Busklemme stimmt nicht mit der Konfiguration überein. Vergleichen Sie für die nn'te Busklemme den Busaufbau mit der Konfiguration.

Firmware Fehler werden im Display mit FW-Error und einer Fehlernummer angegeben. Bitte setzen Sie sich mit dem Support in Verbindung.

Anzeige Display	Bedeutung
FW-Error 0xnxxx	Wenden Sie sich an den Support

## 7.4 PROFIBUS Diagnose-Daten

Der BX-Controller ermöglicht Ihnen aus der PLC heraus Diagnose-Daten zu verschicken. So können Sie sich Ihre eigene Diagnosemitteilung für den Master schreiben und diese individuell mit Daten füllen (siehe [Gerätespezifische Diagnosedaten](#) [► 166]).

### DP-Diagnosedaten (DiagData)

Die DP-Diagnosedaten bestehen aus 6 Bytes DP-Standard-Diagnose und bis zu 238 Bytes gerätespezifischen Diagnosedaten.

Wenn sich die DP-Diagnosedaten ändern, meldet das der Slave dem Master, der die geänderten Diagnosedaten in der Regel daraufhin automatisch abholt. Die DP-Diagnosedaten sind also nicht in Echtzeit zu den DP-Prozessdaten verfügbar, sondern erreichen die Steuerung immer erst einige Zyklen später.

In TwinCAT werden die DP-Diagnosedaten per ADS von der DP-Master-Anschaltung (FC310x, CX1500-M310) ausgelesen (s. Kapitel Slave-Diagnose in der FC310x-Dokumentation).

**DP-Standard-Diagnosedaten**

Offset	Bedeutung
0x00.0	StationNonExistent: Slave beim letzten Telegramm nicht geantwortet
0x00.1	StationNotReady: Slave verarbeitet noch das Set_Prm bzw. Chk_Cfg-Telegramm
0x00.2	CfgFault: Slave meldet einen Konfigurationsfehler
0x00.3	ExtDiag: Extended DiagData sind vorhanden und gültig
0x00.4	NotSupported: Slave unterstützt ein Feature nicht, das mit Set_Prm oder Global_Control gefordert wurde
0x00.5	InvalidSlaveResponse: Slave antwortet nicht DP-konform
0x00.6	PrmFault: Slave meldet einen Parametrierfehler
0x00.7	MasterLock: Slave ist im Datenaustausch mit einem anderen Master
0x01.0	PrmReq: Slave muss neu parametrierung und konfiguriert werden
0x01.1	StatDiag: Slave meldet statische Diagnose bzw. Applikation des DPV1-Slave noch nicht bereit für den Datenaustausch
0x01.2	PROFIBUS-DP-Slave
0x01.3	WdOn: DP-Watchdog ist eingeschaltet
0x01.4	FreezeMode: DP-Slave ist im Freeze-Mode
0x01.5	SyncMode: DP-Slave ist im Sync-Mode
0x01.6	reserviert
0x01.7	Deactivated: DP-Slave wurde deaktiviert
0x02.0	reserviert
0x02.1	reserviert
0x02.2	reserviert
0x02.3	reserviert
0x02.4	reserviert
0x02.5	reserviert
0x02.6	reserviert
0x02.7	ExtDiagOverflow: zu viele Extended DiagData vorhanden
0x03	MasterAdd: Stationsadresse des Masters, der mit dem Slave Datenaustausch macht
0x04,0x05	IdentNumber
ab 0x06	gerätespezifische Diagnosedaten (Extended DiagData)

**Gerätespezifische Diagnosedaten**

Für das Absetzen der Diagnosedaten wird der Funktionsbaustein ADSWRITE verwendet. Die aktuelle DP-Diagnose, wie sie auf den Bus gesendet wird kann per ADSREAD gelesen werden. Achten Sie beim Lesen darauf, dass Sie 6 Byte (die PROFIBUS Standard DP-Diagnose) mitberücksichtigen müssen, d.h. 6 Byte mehr auslesen, als Sie geschrieben haben. Die ADS-Parameter sind mit dem Lesen identisch.

Eingangsparameter	Beschreibung
NETID	lokale NetId des BX
PORT Nummer	100
IDXGRP	16#0003_0000
IDXOFFS	0
LEN	1-238 Bytes (schreiben), 7-244 (lesen)
SRCADDR	Pointer auf die Diagnosedaten

**●** **Diagnosetelegramm ist ereignisabhängig**

**I** Es wird erst dann ein Diagnosetelegramm zur Steuerung übertragen, wenn sich die Diagnosedaten geändert haben.

## 8 Anhang

### 8.1 Erste Schritte mit dem BX3100

Für das folgende Beispiel werden folgende Hardware und Software Komponenten benötigt:


#### Hardware

- FC310x ab Firmware 2.0
- BX3100
- KL10x4
- KL20x4
- KL9010
- PROFIBUS Kabel + Verkabelungsmaterial (wie 24 V<sub>DC</sub> Netzteil etc.)

#### Software

- TwinCAT 2.9 Build 947 (mindestens TwinCAT PLC Level)

#### Beispiel 1: Default Konfiguration

a.) Öffnen sie folgende Datei  (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3238787467.zip>). Dieser File beinhaltet den System Manager File für die Master-Konfiguration mit der FC310x Karte und den BX3100 Slave. Wählen Sie die richtige PCI-Adresse für die FC3101.

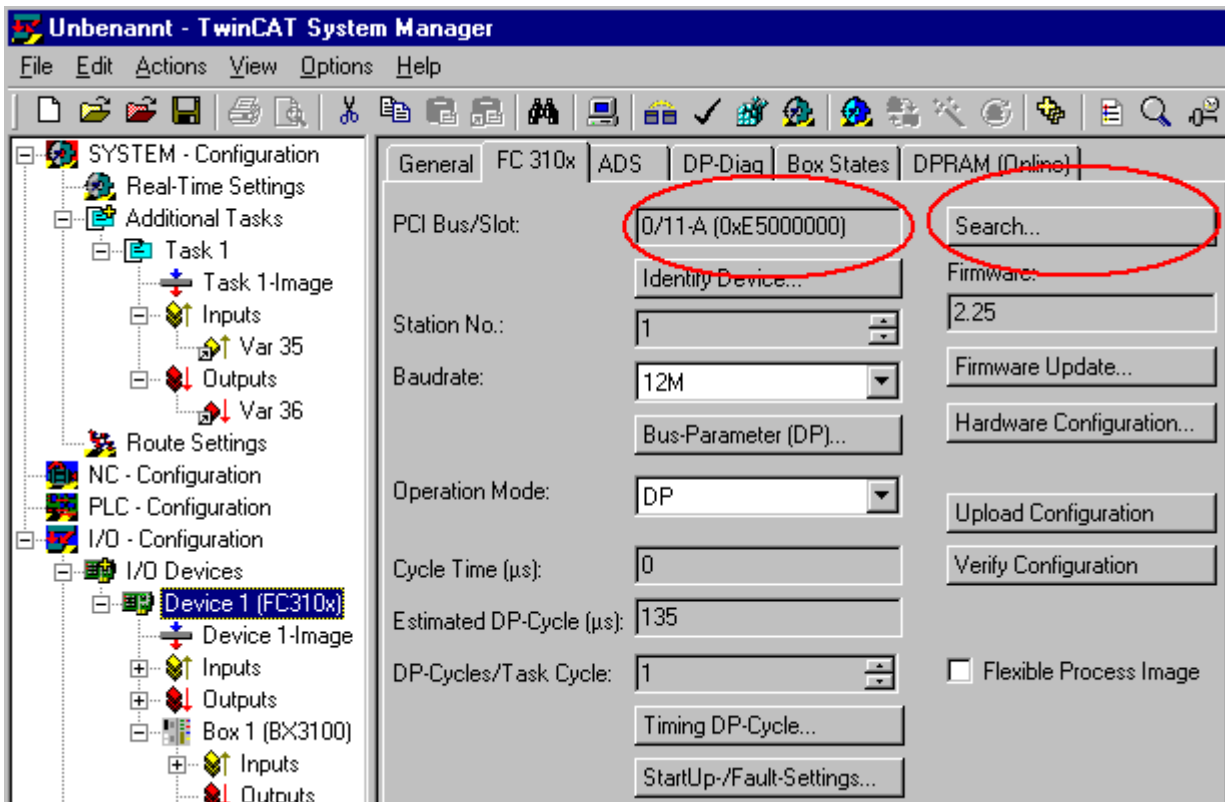


Abb. 149: Auswahl der PCI-Adresse für die FC3101

und stellen Sie auf dem BX3100 die PROFIBUS Adresse [► 32] 2 ein.

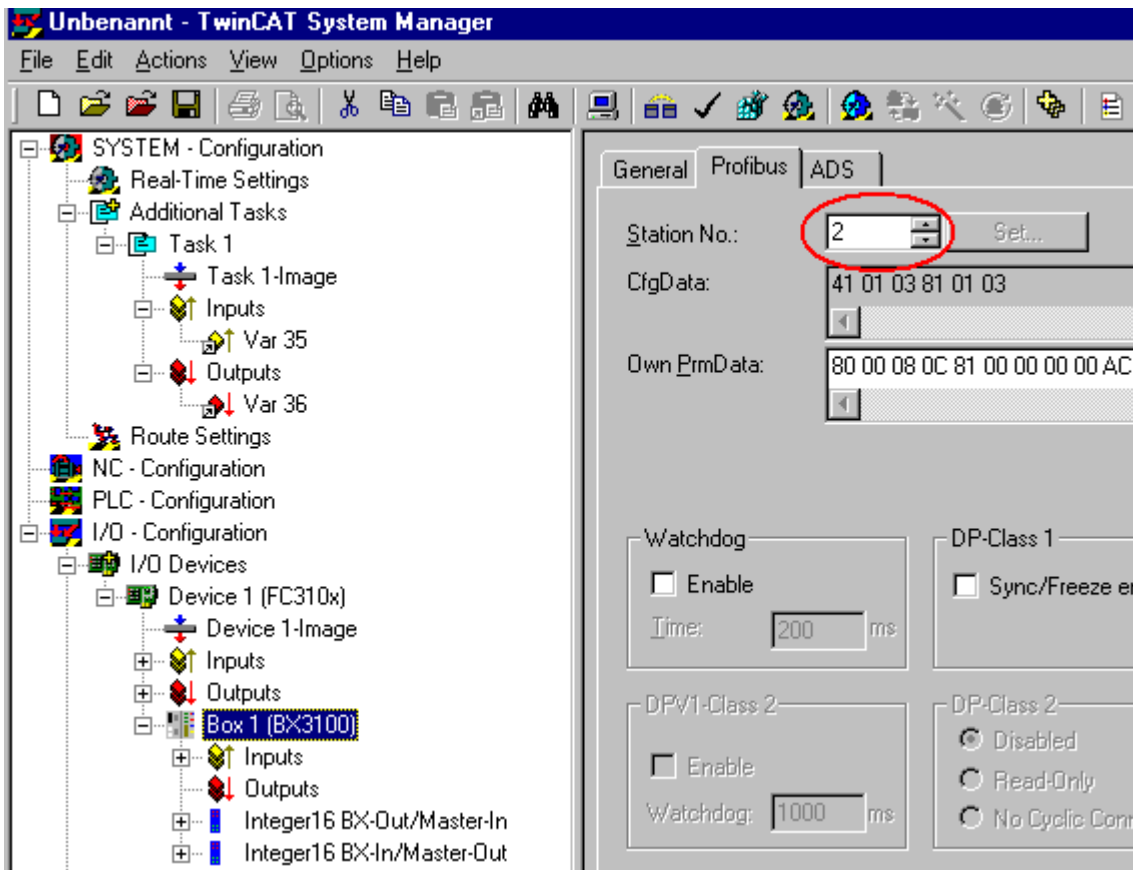


Abb. 150: Einstellen der PROFIBUS-Adresse

Aktivieren Sie die Konfiguration und starten Sie das System. Das TwinCAT Icon (rechts unten auf dem Desktop) muss grün sein. Die Bus LED auf dem BX3100 muss ebenfalls grün leuchten. Sollte dies nicht der Fall sein, überprüfen Sie die Adresse des BX3100 und die physikalische Verbindung zum PROFIBUS Master (Endwiderstände etc.).

b.) Öffnen Sie den folgenden File  (File). Dieser File ist das PLC Projekt für den BX3100. Wählen Sie unter >Online/Auswahl des Zielsystems< den BX3100 aus.

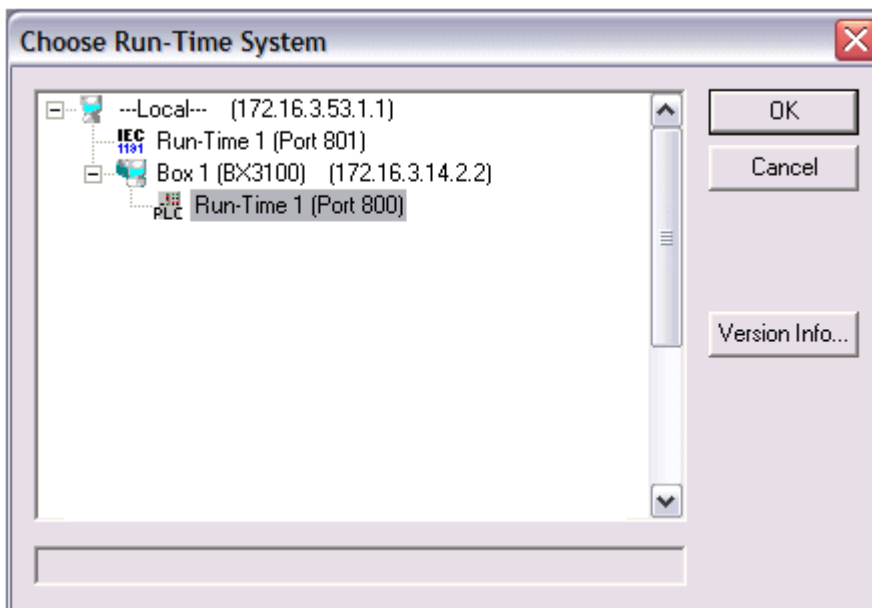


Abb. 151: Auswahl des Zielsystems



Laden Sie das Projekt auf den Controller (>Online|einloggen<) und starten Sie das Programm (>Online|Start<). Die PLC LED auf dem BX3100 muss jetzt orange leuchten.

### Programmbeschreibung

Das Programm zählt einen Wert >Profibus\_Out\_1< solange hoch, wie die Variable Profibus\_Input\_1 den Wert = 0 beinhaltet. Gleichzeitig taktet der erste digitale Ausgang mit ca. 2 Hz. Sie können im System Manager die Variabel >Var 36< auf einen Wert ungleich Null forcen. Damit bleibt der Zähler stehen und der erste digitale Ausgang wird konstant auf "eins" gesetzt.

## 8.2 Umstieg zwischen den Controllern

### Umstieg vom BCxx00 zum BCxx50/BCxx20

#### Dateinamen

Bei Busklemmen-Controllern der Serie BCxx50 und BCxx20 heißen Bibliotheken \*.lbx und Programme werden als \*.prx gespeichert.

#### Merker Variablen

Die lokierten Merker Variablen

- der BCxx00 belegen %MB0...%MB511 (außer BC9000/BC9100: %MB0...%B4095).
- der BCxx20 belegen %MB0...%MB4095
- der BCxx50 belegen %MB0...%MB4095

Zustandsinformationen wie K-Bus-/Feldbus-Status, Zyklustick werden nicht in den BCxx50/BCxx20 kopiert. Diese Informationen stehen in der TcSystemBCxx50.lbx für den BCxx50/BCxx20 als Funktion zur Verfügung.

Die lokierten Merker arbeiten **nicht** als Retain-Variablen.

#### Retain-Daten

Die Retain-Daten müssen als `VAR_RETAIN [► 102]` deklariert werden. Es stehen bis zu 2 kByte zur Verfügung.

#### SPS-Variablen

In der Default-Config fangen die SPS-Variablen ab %IB1000 und %QB1000 an.

#### Large Model

Gibt es nicht bei BCxx50 und BCxx20.

Max. Speicher:

- BCxx50: 48 kByte
- BCxx20: 128 kByte

#### Task Time

Die Task-Zeit wird im PLC-Control festgelegt, diese sollte auf eine realistisch Größe eingestellt sein (messen der PLC Zykluszeit und des K-Busses). Die Background-Zeit entfällt.

#### Task-Konfiguration

Es steht maximal eine Task zur Verfügung. Diese Task muss konfiguriert werden.

## PLC und Feldbusklemmen

Bei den Standard Busklemmen Controllern (BCxx00) gab es die Möglichkeit zu wählen, ob eine Busklemme dem Feldbus oder der lokalen PLC zugewiesen wird.

In der Default-Konfiguration des BCxx50/BCxx20 sind alle Busklemmen der lokalen PLC zugewiesen. Eine Zuordnung zum Feldbus ist hier nicht möglich.

## Umstieg vom BCxx00 zum BXxx00

### Dateinamen

Bei Busklemmen-Controllern der Serie BXxx00 heißen Bibliotheken \*.lxb und Programme werden als \*.prx gespeichert.

### Merker-Variablen

Die lokierten Merker-Variablen

- der BCxx00 belegen %MB0...%MB511 (außer BC9000/BC9100: %MB0...%B4095).
- der BXxx00 belegen %MB0...%MB4095

Zustandsinformationen wie K-Bus-/Feldbus-Status, Zyklustick werden nicht in den BXxx00 kopiert. Diese Informationen stehen in der TcSystemBX.lxb für den BXxx00 als Funktion zur Verfügung.

Die lokierten Merker arbeiten **nicht** als Retain-Variablen.

### Retain-Daten

Die Retain-Daten müssen als `VAR RETAIN [► 102]` deklariert werden. Es stehen bis zu 2 kByte zur Verfügung.

### SPS-Variablen

In der Default-Config fangen die SPS-Variablen ab %IB1000 und %QB1000 an.

### Large Model

Gibt es nicht bei BXxx00. Max. Speicher: 256 kByte.

### Task Time

Die Task-Zeit wird im PLC-Control festgelegt, diese sollte auf eine realistisch Größe eingestellt sein (messen der PLC Zykluszeit und des K-Buses). Die Background-Zeit entfällt.

### Task-Konfiguration

Es steht maximal eine Task zur Verfügung. Diese Task muss konfiguriert werden.

## PLC und Feldbusklemmen

Bei den Standard Busklemmen Controller (BCxx00) gab es die Möglichkeit zu wählen, ob eine Busklemme dem Feldbus oder der lokalen PLC zugewiesen wird.

In der Default-Konfiguration des BXxx00 sind alle Busklemmen der lokalen PLC zugewiesen. Eine Zuordnung zum Feldbus ist hier nicht möglich.

## Umstieg vom PC zum BCxx50/BCxx20/BXxx00

### Dateinamen

Bei den Busklemmen-Controllern der Serien BCxx50/BCxx20 und BXxx00 heißen Bibliotheken \*.lxb und Programme werden als \*.prx gespeichert.

### Lokierte Variablen

Es stehen bei den Busklemmen-Controllern der Serien BCxx50/BCxx20 und BXxx00 eine begrenzte Anzahl von lokierten Daten zur Verfügung:

- Inputs 2 kByte, %IB0...2048
- Outputs 2 kByte, %QB0...2048
- Merker 4 kByte, %MB0...4095

### Task-Konfiguration

Maximal steht eine Task zur Verfügung. Es ist eine sinnvolle Task-Zeit zu wählen. Passen Sie die Task-Zeit Ihrer Anwendung an, messen Sie dazu die benötigte Systemzeit (PLC + K-Bus + Feldbus + sonstiges).

### Retain-Daten

Bei den Busklemmen-Controllern der Serien BCxx50, BCxx20 und BXxx00 stehen bis zu 2 kByte Retain-Daten zur Verfügung. Achten Sie daher darauf das Sie keine (oder nur sehr beschränkt) Retain-Daten in Funktionsblöcken verwenden (siehe [RETAIN Daten](#) [▶ 102]).

## 8.3 Firmware-Update


### Firmware-Update-Programm

Um eine neue Firmware auf den Buskoppler zu laden, ist das Firmware Update Programm erforderlich. Das Programm wird über die serielle Schnittstelle übertragen.

Hinweis für BX3100:

Die Firmware 0.64 (oder kleiner) des BX3100 erlaubt kein Update. Sollte ein Update dieser Geräte erforderlich sein, schicken Sie den BX3100 mit dem Hinweis auf ein Update zum Hersteller.

Beckhoff Automation GmbH & Co. KG  
Abt. Service  
Stahlstr. 31  
D-33415 Verl

 Firmware Update Programm 241 (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3238791819.zip>) 71 kByte (für Windows NT4.0 SP6, 2000, XP).

Das Programm *FirmwareUpdate.exe* und die *TcRouterHelper.dll* müssen im gleichen Verzeichnis stehen. Mit einem Doppelklick auf *FirmwareUpdate.exe* öffnen Sie das Programm.

### Update für Busklemmen-Controller

#### BX-Serie

Wählen sie das entsprechende Gerät aus - hier "Serial Interface (BX)".

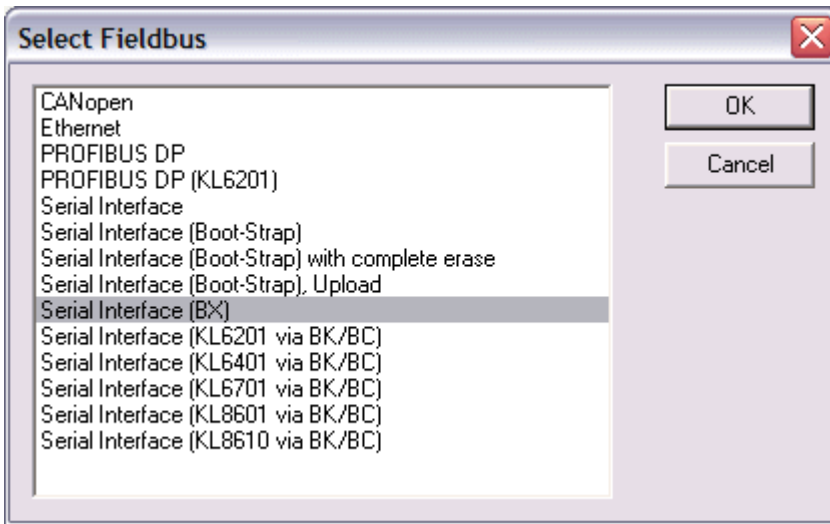


Abb. 152: Auswahl des Busklemmen-Controllers der BX-Serie

### BCxx50 Serie

Wählen sie das entsprechende Gerät aus - hier "Serial Interface".

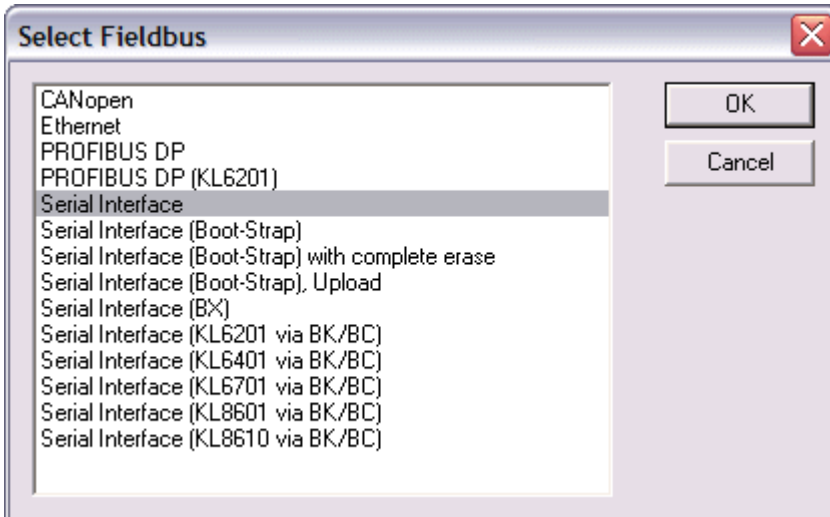


Abb. 153: Auswahl des Busklemmen-Controllers der BC-Serie

### BX- und BCxx50-Serie

Anschließen wählen Sie den COM-Port aus.

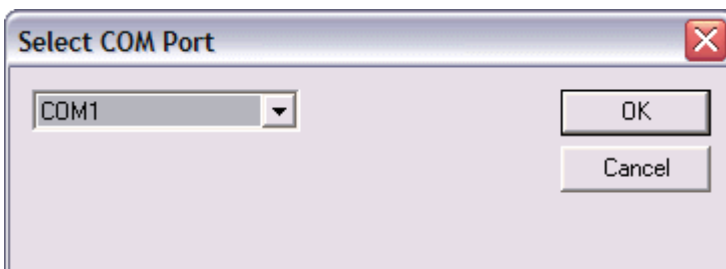


Abb. 154: Auswahl des COM-Ports

Öffnen Sie das entsprechende File, das Sie herunterladen möchten.

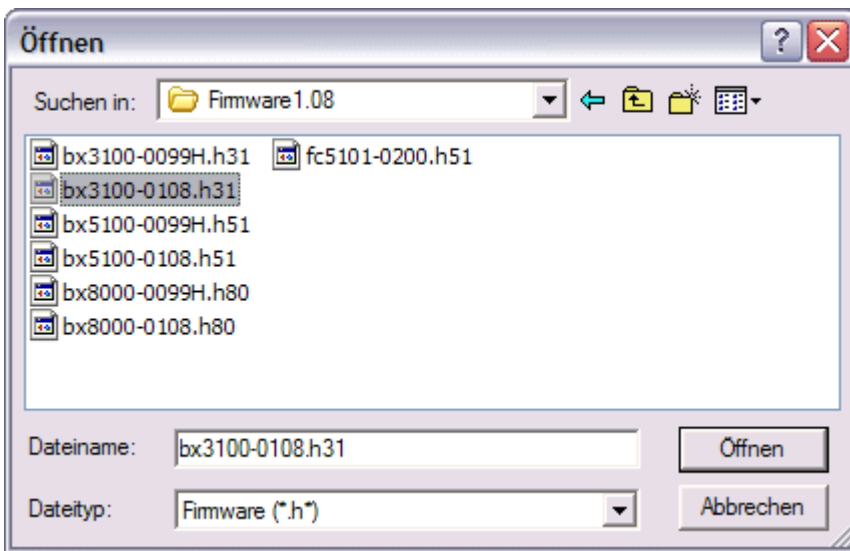


Abb. 155: Öffnen des Firmware-Files

Starten Sie den Download mit der grünen Ampel. Nach ca. einer Minute beginnt der Download, der dann auch im Display des BX angezeigt wird. Nach erfolgreichem Download (ca. 2 bis 3 Minuten) bootet der Busklemmen-Controller automatisch neu.

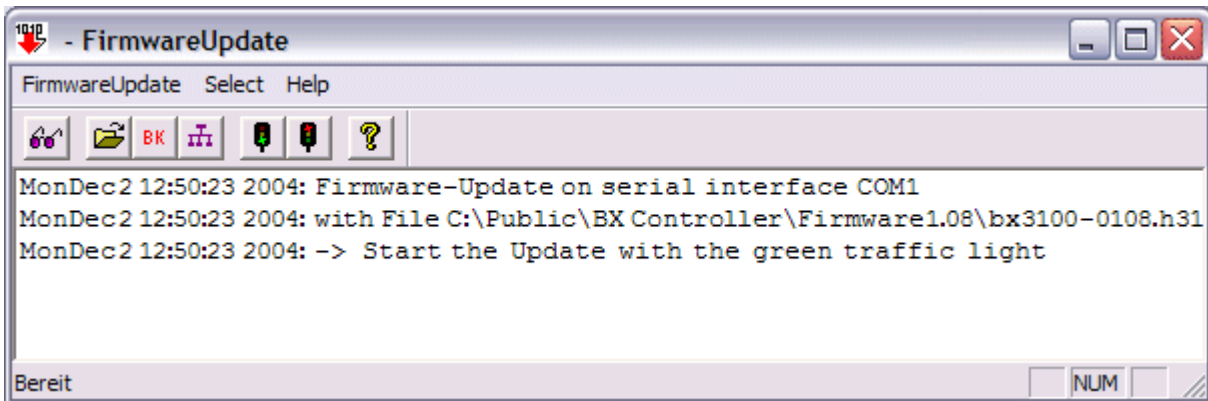



Abb. 156: Statusmeldungen des Firmware-Updates

## 8.4 CFC-Client\*

Der BX bietet mit den CFC-Client die Möglichkeit den gesamten Speicherinhalt eines BX-Controllers zu kopieren. Die Daten aus dem Speicher werden in einer BIN Datei gespeichert und können auf einen identischen Controller geladen werden.

Der CFC-Client läuft auf Windows 9x, NT, 2000 und XP und ist völlig unabhängig von TwinCAT.

 CFC-Client (<https://infosys.beckhoff.com/content/1031/bx3100/Resources/zip/3238793995.zip>)  
(Hinweis: Die DLLs müssen Sie manuell per regsvr32 Registrieren)

Wenn Sie den CFC-Client starten stellen Sie als erstes die COM-Parameter ein.

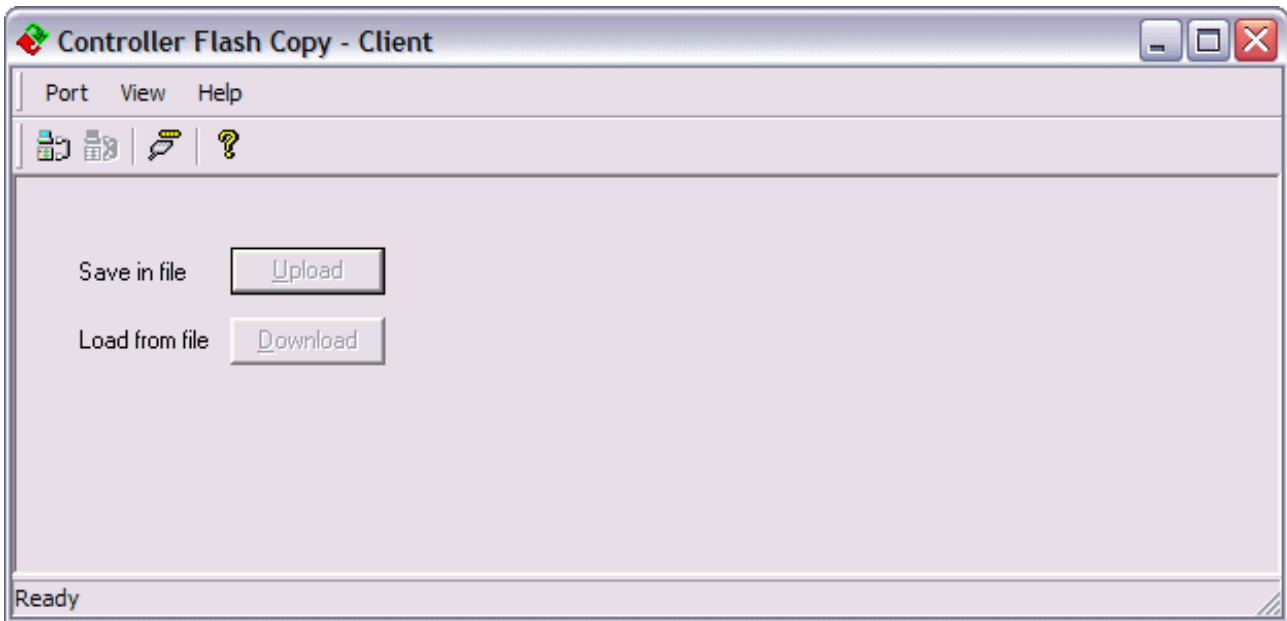


Abb. 157: CFC-Client

Port/Port Setup...

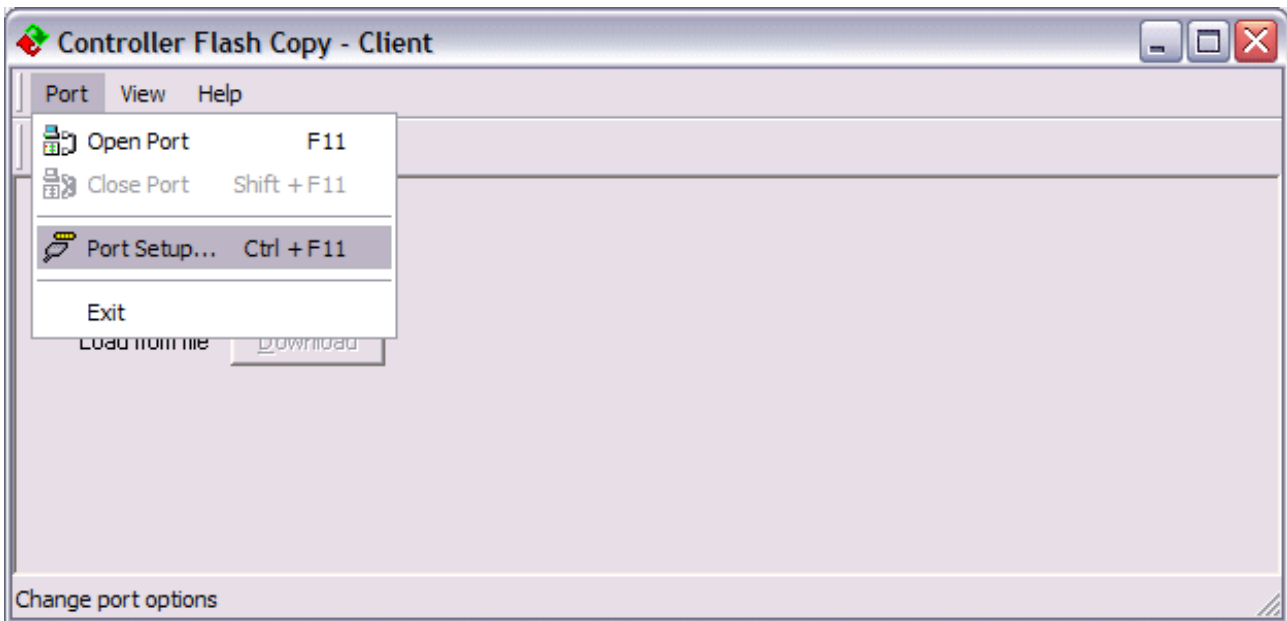


Abb. 158: Aufruf des Port-Setups

Wählen Sie die Einstellung aus.

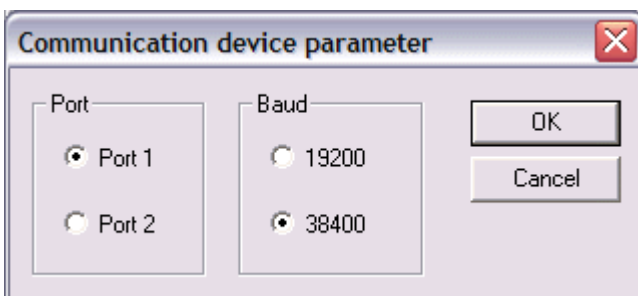


Abb. 159: Einstellen der COM-Parameter

Bevor Sie den COM-Port öffnen müssen Sie eine physikalische Verbindung zum BX-Controller herstellen. Öffnen Sie dann den COM Port.

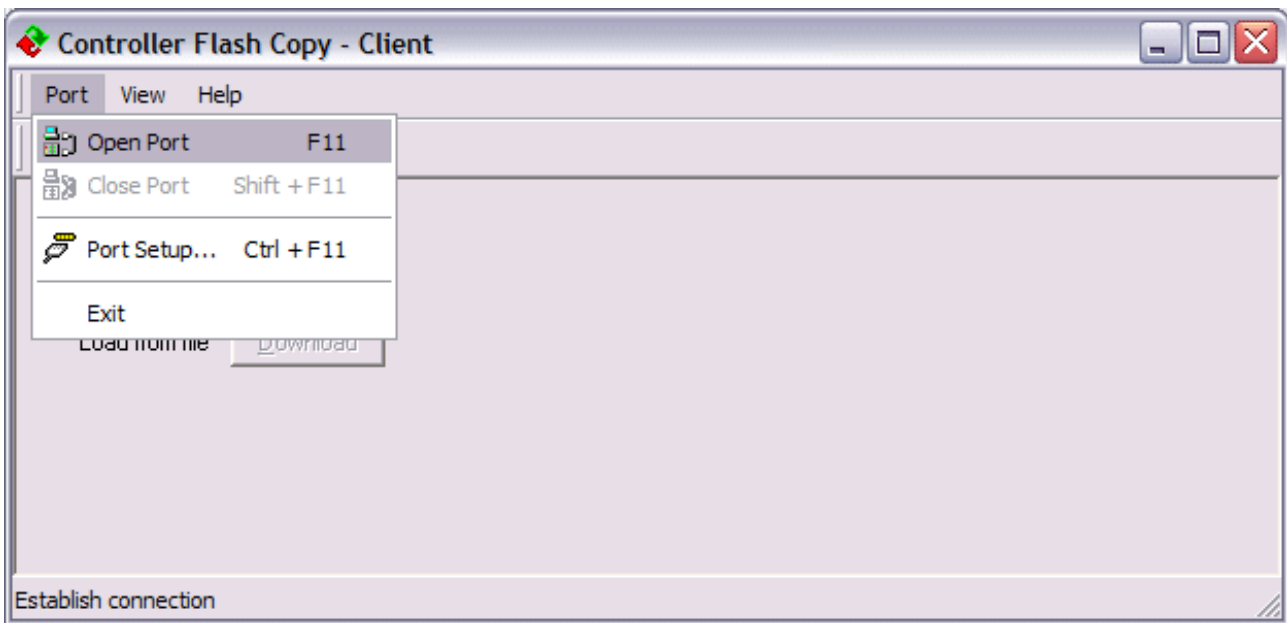


Abb. 160: Öffnen des COM-Ports

Mit Upload erstellen Sie eine Kopie des BX-Speichers speichern diese als BIN-Datei.

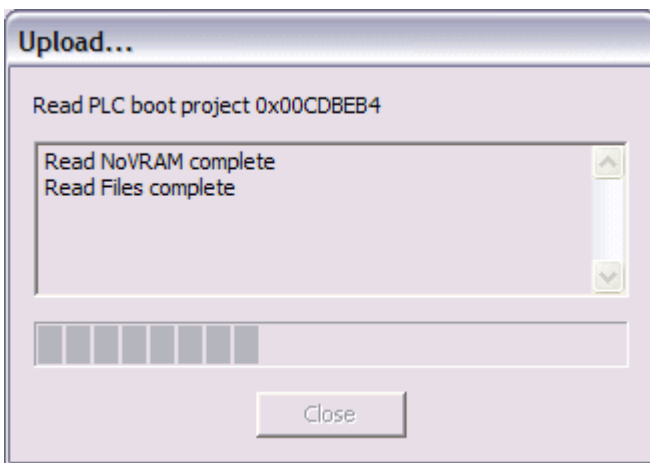


Abb. 161: Upload des BX-Speicherinhalts in ein BIN-File

Mit Download wird das BIN-File zum BX-Controller geladen. Der Controller startet automatisch nach erfolgreichem Download.

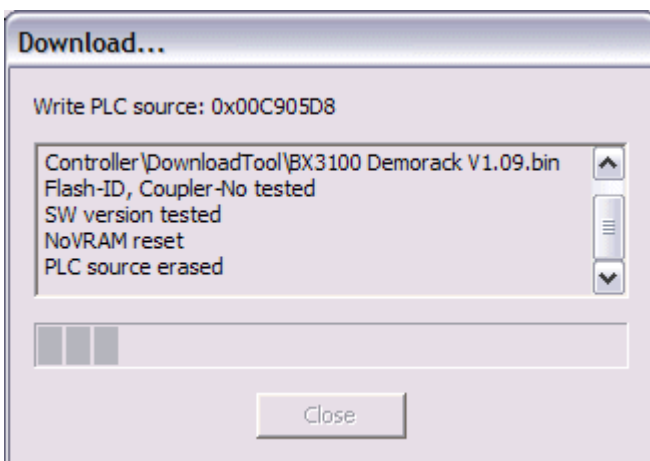


Abb. 162: Download eines BIN-Files auf den BX-Controller

\* ab Firmware 1.09 ist der CFC-Server auf der BX-Controller Reihe verfügbar

## 8.5 Beispielprogramme - Übersicht

Benennung	Beschreibung
<a href="#">Display [► 123]</a>	Beispiel zum Ansteuern des Displays
<a href="#">Navigationsschalter [► 122]</a>	Auslesen des Navigationsschalter aus der PLC heraus
<a href="#">Menü [► 94]</a>	Beispiel für ein eignes Menü mit Navigations-Schalter und Display
<a href="#">RTC [► 87]</a>	Beispiel zum Auslesen der Real Time Clock (RTC) über Funktionsbausteine
<a href="#">COM Port - BK/BC8x00 Master Interface [► 128]</a>	COM1- oder COM2-Schittstelle als Master mit dem BK8x00 Protokoll
<a href="#">COM Port - BK8x00 Slave Interface [► 129]</a>	COM1- oder COM2-Schittstelle als Slave mit dem BK8x00 Protokoll
<a href="#">COM Port - Cimrex 12 [► 135]</a>	Beispiel zum Ansteuern eines Cimrex 12 Displays über ModbusRTU
<a href="#">COM Port - ModbusRTU Slave [► 132]</a>	Verknüpfung der ModbusRTU Lib mit der COM 1 oder COM 2 Schnittstelle des BXes
<a href="#">COM Port - ModbusRTU Master [► 134]</a>	Verknüpfung der ModbusRTU Lib mit der COM 1 oder COM 2 Schnittstelle des BXes
<a href="#">COM Port - RK512 Protokoll [► 137]</a>	RK512-Protokoll über die COM 1 oder COM 2
<a href="#">COM Port - SMS über COM Port [► 138]</a>	Anschluss eines Siemens S35 Mobiltelefons an COM-Schnittstelle zum Versenden von SMS-Nachrichten
<a href="#">COM Port - COMlibV2 [► 135]</a>	String verschicken und empfangen mit der COMlibV2
<a href="#">SSB - Display [► 78]</a>	Cimrex Panel am SSB
<a href="#">SSB - AX2000 [► 76]</a>	AX2000 am SSB
<a href="#">SSB - BK51x0 [► 74]</a>	BK5120 am SSB
<a href="#">SSB - BX / BX Kommunikation [► 74]</a>	Kommunikation von BX zu BX (über SSB)
<a href="#">SSB - IclA Drive [► 79]</a>	IclA Drive am SSB
<a href="#">SSB - Lenze Drive [► 84]</a>	Lenze Frequenzumrichter am SSB

## 8.6 Allgemeine Betriebsbedingungen

Um einen fehlerfreien Betrieb der Feldbuskomponenten zu erreichen, müssen die nachfolgenden Bedingungen eingehalten werden.

### Bedingungen an die Umgebung

#### Betrieb

An folgenden Orten dürfen die Komponenten nicht ohne Zusatzmaßnahmen eingesetzt werden:

- unter erschwerten Betriebsbedingungen, wie z.B. ätzende Dämpfe oder Gase, Staubbildung
- bei hoher ionisierender Strahlung

Bedingung	zulässiger Bereich
zulässige Umgebungstemperatur im Betrieb	siehe Technische Daten
Einbaulage	beliebig
Vibrationsfestigkeit	gemäß EN 60068-2-6
Schockfestigkeit	gemäß EN 60068-2-27
EMV-Festigkeit	gemäß EN 61000-6-2
Aussendung	gemäß EN 61000-6-4



**Transport und Lagerung**

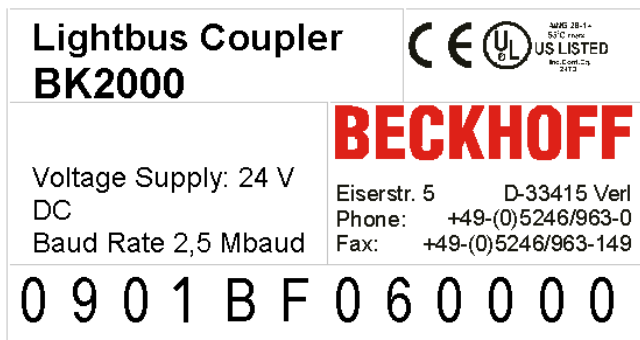
Bedingung	zulässiger Bereich
zulässige Umgebungstemperatur bei Lagerung	-25°C... +85°C
Relative Feuchte	95 %, keine Betauung
Freier Fall	originalverpackt bis 1 m

**Schutzklasse und Schutzart**


Bedingung	zulässiger Bereich
Schutzklasse nach IEC 536 (VDE 0106, Teil 1)	An der Profilschiene ist ein Schutzleiteranschluss erforderlich!
Schutzart nach IEC 529	IP20 ( Schutz gegen Berührung mit Standard Prüffinger)
Schutz gegen Fremdkörper	kleiner 12 mm im Durchmesser
Schutz gegen Wasser	kein Schutz

**Kennzeichnung der Komponenten**

Jede ausgelieferte Komponente enthält einen Aufkleber, mit Informationen über die Zulassung des Produkts. Beispiel für den Buskoppler BK2000:



Auf dem Aufkleber sind folgende Informationen abzulesen:

Aufdruck	Bedeutung für diesen Aufkleber
genaue Produktbezeichnung	Lightbus Koppler BK2000
Versorgungsspannung Us	24 V <sub>DC</sub> (Benutzen Sie eine 4 A Sicherung oder eine der Class 2 entsprechende Spannungsversorgung um die UL-Anforderungen zu erfüllen!)
Übertragungsrate	2,5 MBaud
Hersteller	Beckhoff Automation GmbH & Co. KG
CE-Zeichen	Konformitätskennzeichnung
UL-Zeichen 	Kennzeichen für UL-Zulassung. UL steht für Underwriters Laboratories Inc., die führende Zertifizierungsorganisation für Nordamerika mit Sitz in den USA. C = Kanada, US = USA, UL File Nummer: E172151
Produktionsbezeichnung	Die Zahlenfolge ergibt von links nach rechts die Produktionswoche (2 Ziffern), das Produktionsjahr (2 Ziffern), die Software- (2 Ziffern) sowie die Hardwareversion (2 Ziffern) und Sonderbezeichnungen (4 Ziffern).  In diesem Fall handelt es sich also um einen BK2000 - produziert in der 9. Kalenderwoche - de Jahres 2001 - mit der Firmware-Version BF - in der 6. Hardwareversion - ohne Sonderbezeichnung

## 8.7 Prüfnormen für Geräteprüfung

### EMV

#### EMV-Festigkeit

EN 61000-6-2

#### EMV-Aussendung

EN 61000-6-4

### Vibrations-/ Schockfestigkeit

#### Vibrationsfestigkeit

EN 60068-2-6

#### Schockfestigkeit

EN 60068-2-27

## 8.8 Literaturverzeichnis

### Deutsche Bücher

#### PROFIBUS

- PROFIBUS-DP/DPV1  
Grundlagen, Tipps und Tricks für Anwender  
von Manfred Popp  
ISBN: 3778527819

#### Feldbustechnik allgemein

- Gerhard Gruhler (Hrsg.): **Feldbusse und Geräte-Kommunikationssysteme**  
Praktisches Know-How mit Vergleichsmöglichkeiten  
Franzis Verlag 2001  
244 Seiten  
ISBN 3-7723-5745-8

### Englische Bücher

(In Vorbereitung)

#### Standards PROFIBUS-DP

- IEC 61158 und IEC 61784
- DIN 19245, Teil 3
- Euronorm EN 50 170

#### Web Seiten

- <http://www.profibus.de>

## 8.9 Abkürzungsverzeichnis

### DP

Dezentrale Peripherie. PROFIBUS-Protokoll für den schnellen zyklischen Datenaustausch.

### FMS

Übertragungsprotokoll des PROFIBUS (Fieldbus Message Specification).

### Freeze-Mode

Mit diesem Kommando friert der Salve seine Eingänge ein.

### GSD-Datei

Deutsche Geräte-Stammdatei

### GSE-Datei

Englische Geräte-Stammdatei

### IP20, IP65, IP66, IP67

Schutzart (gegen Berührung, Wasser, Staub)

**K-Bus**

Klemmenbus: Interner Bus für die Kommunikation zwischen Buskoppler und Busklemmen

**PNO**

PROFIBUS-Nutzerorganisation (siehe [www.profibus.de](http://www.profibus.de))

**Repeater**

Dienen der Signalaufbereitung und verbindet einzelne Bussegmente

**SPS**

Speicherprogrammierbare Steuerung (englisch PLC: programmable logic controller)

**Sync-Mode**

Mit diesem Kommando hält der Slave seine Ausgänge fest, bis er das Sync-Telegramm erhält.

## 8.10 Support und Service

Beckhoff und seine weltweiten Partnerfirmen bieten einen umfassenden Support und Service, der eine schnelle und kompetente Unterstützung bei allen Fragen zu Beckhoff Produkten und Systemlösungen zur Verfügung stellt.

### Beckhoff Niederlassungen und Vertretungen

Wenden Sie sich bitte an Ihre Beckhoff Niederlassung oder Ihre Vertretung für den lokalen Support und Service zu Beckhoff Produkten!

Die Adressen der weltweiten Beckhoff Niederlassungen und Vertretungen entnehmen Sie bitte unseren Internetseiten: <https://www.beckhoff.de>

Dort finden Sie auch weitere Dokumentationen zu Beckhoff Komponenten.

### Beckhoff Support

Der Support bietet Ihnen einen umfangreichen technischen Support, der Sie nicht nur bei dem Einsatz einzelner Beckhoff Produkte, sondern auch bei weiteren umfassenden Dienstleistungen unterstützt:

- Support
- Planung, Programmierung und Inbetriebnahme komplexer Automatisierungssysteme
- umfangreiches Schulungsprogramm für Beckhoff Systemkomponenten

Hotline: +49(0)5246 963 157  
Fax: +49(0)5246 963 9157  
E-Mail: [support@beckhoff.com](mailto:support@beckhoff.com)

### Beckhoff Service

Das Beckhoff Service-Center unterstützt Sie rund um den After-Sales-Service:

- Vor-Ort-Service
- Reparaturservice
- Ersatzteilservice
- Hotline-Service

Hotline: +49(0)5246 963 460  
Fax: +49(0)5246 963 479  
E-Mail: [service@beckhoff.com](mailto:service@beckhoff.com)

### Beckhoff Firmenzentrale

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20  
33415 Verl  
Deutschland

Telefon: +49(0)5246 963 0  
Fax: +49(0)5246 963 198  
E-Mail: [info@beckhoff.com](mailto:info@beckhoff.com)  
Internet: <https://www.beckhoff.de>

# Abbildungsverzeichnis

Abb. 1	Busklemmen-Controller der BX-Serie.....	11
Abb. 2	BX3100 .....	13
Abb. 3	Prinzip der Busklemme.....	16
Abb. 4	Federkontakte der Beckhoff I/O-Komponenten .....	19
Abb. 5	BX3100, BX5100, BX5200, BX9000.....	20
Abb. 6	BX8000 .....	20
Abb. 7	Enriegelter BX-Controller.....	21
Abb. 8	Verriegelter BX-Controller.....	21
Abb. 9	Demontage .....	21
Abb. 10	Potentialgruppen eines Busklemmenblocks .....	22
Abb. 11	Linksseitiger Powerkontakt.....	23
Abb. 12	Klemmstellen zur Versorgung des Busklemmen-Controllers .....	24
Abb. 13	UL-Kennzeichnung der BX-Controller .....	24
Abb. 14	Programmierkabel ZK1000-0030 - COM 1 und COM 2.....	25
Abb. 15	Programmierkabel ZK1000-0030 - Abmessungen der Stecker .....	26
Abb. 16	Programmierkabel ZK1000-0030 - Pinning .....	26
Abb. 17	SSB-Schnittstelle .....	27
Abb. 18	COM1- (RS 232) und COM2-Schnittstelle (RS 232/485) .....	27
Abb. 19	Pinbelegung M12 Buchse (-B310).....	28
Abb. 20	Pinbelegung M12 Buchse/Stecker (-B318).....	28
Abb. 21	Pinbelegung der PROFIBUS D-Sub Buchse .....	28
Abb. 22	Pin-Belegung Buchse/Stecker Feldbus Box Module .....	29
Abb. 23	Belegung des PROFIBUS-Kabels .....	30
Abb. 24	Anlaufverhalten des Busklemmen-Controllers.....	32
Abb. 25	Drehwahlschalter für die PROFIBUS-Adresse .....	33
Abb. 26	Anlegen einer TwinCAT-Konfiguration .....	35
Abb. 27	Auswahl des Busklemmen-Controllers .....	35
Abb. 28	Download einer TwinCAT-Konfiguration.....	36
Abb. 29	Auswahl des Busklemmen-Controllers .....	37
Abb. 30	Zustand des Busklemmen-Controllers.....	37
Abb. 31	Aktivieren der TwinCAT-Konfiguration.....	37
Abb. 32	Auswahl des Zielsystems .....	38
Abb. 33	Auswahl des Busklemmen-Controllers .....	38
Abb. 34	Zustand des Busklemmen-Controllers.....	38
Abb. 35	Hochladen der TwinCAT-Konfiguration .....	39
Abb. 36	Speicher für das Code Mapping .....	39
Abb. 37	Daten Speicher Mapping .....	40
Abb. 38	Code und Daten Speicher .....	40
Abb. 39	Sonstiger Speicher .....	41
Abb. 40	Eigenschaften der Remote-Verbindung.....	42
Abb. 41	Karteireiter DP-Slaves .....	43
Abb. 42	BX3100 einfügen .....	45
Abb. 43	Anlegen von PROFIBUS-Daten.....	46
Abb. 44	ADS-Interface .....	46

Abb. 45	Anlegen von PROFIBUS-Daten.....	47
Abb. 46	Einstellungen .....	49
Abb. 47	Beispiel für das Eintragen einzelner Bytes. ....	50
Abb. 48	Beispiel für das Eintragen zusammenhängender Bytes .....	51
Abb. 49	Karteireiter BX Settings .....	52
Abb. 50	Karteireiter BX Diag .....	53
Abb. 51	Auswahl des PLC-Projekts .....	54
Abb. 52	Verbinden vom PLC-Variable und Hardware.....	55
Abb. 53	Anzeige des Ziel-Systems .....	55
Abb. 54	Einstellen der Task-Zeit.....	56
Abb. 55	Anzeige der PLC-Zykluszeit .....	57
Abb. 56	Abschluss des Busses mit Abschlusswiderstand 120 Ohm .....	58
Abb. 57	Unempfindlichkeit gegen eingeprägte Störungen.....	58
Abb. 58	Beispieltopologie Stichleitungen .....	59
Abb. 59	Aufbau CAN-Kabel ZB5100.....	60
Abb. 60	Aufbau CAN-/DeviceNet-Kabel ZB5200 .....	61
Abb. 61	Pinbelegung BK5151, EL6751.....	62
Abb. 62	FC51x2 .....	62
Abb. 63	Belegung Verbindungsbuchse BK51x0/BX5100 .....	63
Abb. 64	LC5100 .....	64
Abb. 65	Pinbelegung M12 Stecker Feldbus Box.....	64
Abb. 66	Anfügen eines weiteren Gerätes .....	65
Abb. 67	Auswahl des CANopen Masters SSB.....	65
Abb. 68	Anfügen eines CANopen Geräts .....	65
Abb. 69	Auswahl eines CANopen-Knotens.....	66
Abb. 70	Anfügen/Bearbeiten von Objektverzeichniseinträgen.....	67
Abb. 71	NodeState, DiagFlag und EmergencyCounter .....	68
Abb. 72	Verdrahtungsplan für Testaufbau .....	72
Abb. 73	Kommunikation von BX-Controller zu BX-Controller (über SSB) .....	75
Abb. 74	AX2000 .....	76
Abb. 75	CANopen Interface (X6) .....	77
Abb. 76	Cimrex-Panel am SSB des BX-Controllers.....	78
Abb. 77	IclA-Drive am SSB .....	79
Abb. 78	Anschlüsse des IclA-Drives .....	80
Abb. 79	Signalschnittstelle .....	80
Abb. 80	Feldbusanschluss .....	81
Abb. 81	Referenzierungsbereiche.....	82
Abb. 82	Listing der Referenzierungswerte .....	83
Abb. 83	Frequenzumrichter der Fa. Lenze .....	84
Abb. 84	Externe Spannungsversorgung - Interne Spannungsversorgung(Auslieferungszustand).....	85
Abb. 85	DIP-Schalter .....	85
Abb. 86	Freigabe des Kommunikationsmoduls.....	86
Abb. 87	Einstellen der Echtzeit-Uhr (RTC) .....	87
Abb. 88	Navigationsschalter der BX-Controller.....	90
Abb. 89	Schalterbelegung.....	91
Abb. 90	Maximale Anzahl der POU's überschritten.....	97

Abb. 91	Menüpfad Projekte / Optionen / Controller Settings .....	97
Abb. 92	Controller Settings .....	98
Abb. 93	Globaler Speicher nicht ausreichend .....	98
Abb. 94	Menüpfad Projekte / Optionen / Build .....	98
Abb. 95	Build .....	99
Abb. 96	Ändern der Verknüpfung von Variablen .....	103
Abb. 97	Verknüpfen einer Variable mit einem Eingang .....	104
Abb. 98	Öffnen des Optionsmenüs .....	106
Abb. 99	Auswahl des Source Downloads .....	107
Abb. 100	Download des Programm Codes .....	108
Abb. 101	Vorschritt des Downloads .....	108
Abb. 102	Upload eines Programms .....	109
Abb. 103	Auswahl des Datenübertragungswegs .....	109
Abb. 104	Auswahl des Gerätes .....	109
Abb. 105	Funktionsbaustein F_COMPORTEAD .....	117
Abb. 106	Funktionsbaustein F_COMPORWRITE .....	118
Abb. 107	Funktionsbaustein FB_COMPOROPEN .....	118
Abb. 108	Funktionsbaustein FB_COMPORCLOSE .....	119
Abb. 109	Funktionsbaustein F_STARTDEBUGTIMER .....	122
Abb. 110	Funktionsbaustein F_READDEBUGTIMER .....	122
Abb. 111	Funktionsbaustein F_GETNAVSWITCH .....	122
Abb. 112	Funktionsbaustein FB_DISPWRITE .....	123
Abb. 113	Funktionsbaustein RTC .....	125
Abb. 114	Funktionsbaustein fb_ReadWriteFile .....	126
Abb. 115	Funktionsbaustein FB_BX_BK8X00_MASTER .....	128
Abb. 116	Funktionsbaustein FB_BX_BK8X00_SLAVE .....	130
Abb. 117	Kommunikationseigenschaften .....	131
Abb. 118	Funktionsbaustein FB_BX_COM_5 .....	132
Abb. 119	Funktionsbaustein FB_BX_COM_64 .....	132
Abb. 120	Funktionsbaustein FB_BX_COM_64EX .....	133
Abb. 121	Cimrex-Panel am COM-Port des BX-Controllers .....	135
Abb. 122	Mobiltelefon am COM-Port des BX-Controllers .....	138
Abb. 123	Funktionsbaustein F_GETVERSIONTCTWINSAFE .....	139
Abb. 124	Funktionsbaustein FB_TWINSAFE_KLX904_INPUT .....	139
Abb. 125	Funktionsbaustein FB_TWINSAFE_KLX904_input .....	140
Abb. 126	Verknüpfen der Eingangsdaten .....	141
Abb. 127	Auswahl der SafetyIn-Variable .....	141
Abb. 128	Funktionsbaustein FB_TWINSAFE_KLX904_OUTPUT .....	142
Abb. 129	Aufruf des Funktionsbausteins FB_TWINSAFE_KLX904_OUTPUT .....	143
Abb. 130	Aufruf des Funktionsbausteins FB_TWINSAFE_KLX904_OUTPUT .....	143
Abb. 131	Verknüpfen der Eingangsdaten .....	144
Abb. 132	Auswahl der entsprechenden SafetyQBx-Variable .....	144
Abb. 133	Auswahl des Datenübertragungswegs - Serielle Schnittstelle .....	145
Abb. 134	Parametrierung der seriellen Schnittstelle .....	145
Abb. 135	Auswahl des Datenübertragungswegs - AMS .....	146
Abb. 136	Auswahl des Gerätes .....	146



Abb. 137 ADS-Interface .....	147
Abb. 138 Auswahl der Zielplattform .....	148
Abb. 139 Auswahl des Zielsystems .....	148
Abb. 140 Pinbelegung M12 Buchse .....	156
Abb. 141 Belegung des PROFIBUS-Kabels .....	157
Abb. 142 Adresswahlschalter.....	158
Abb. 143 Adresswahlschalter.....	158
Abb. 144 RS485-Topologie mit 3 Segmenten und 2 Repeatern.....	159
Abb. 145 PROFIBUS Diagnose Byte im System Manager .....	160
Abb. 146 K-Bus-Status.....	161
Abb. 147 Diagnose LEDs für den Feldbus, die SPS, den K-Bus und die Spannungsversorgungen .....	162
Abb. 148 Diagnose LEDs für den Feldbus, die SPS und den K-Bus .....	162
Abb. 149 Auswahl der PCI-Adresse für die FC3101 .....	167
Abb. 150 Einstellen der PROFIBUS-Adresse .....	168
Abb. 151 Auswahl des Zielsystems .....	168
Abb. 152 Auswahl des Busklemmen-Controllers der BX-Serie .....	172
Abb. 153 Auswahl des Busklemmen-Controllers der BC-Serie .....	172
Abb. 154 Auswahl des COM-Ports .....	172
Abb. 155 Öffnen des Firmware-Files .....	173
Abb. 156 Statusmeldungen des Firmware-Updates .....	173
Abb. 157 CFC-Client.....	174
Abb. 158 Aufruf des Port-Setups .....	174
Abb. 159 Einstellen der COM-Parameter.....	174
Abb. 160 Öffnen des COM-Ports .....	175
Abb. 161 Upload des BX-Speicherinhalts in ein BIN-File .....	175
Abb. 162 Download eines BIN-Files auf den BX-Controller.....	175



Mehr Informationen:  
[www.beckhoff.de/BX3100](http://www.beckhoff.de/BX3100)

Beckhoff Automation GmbH & Co. KG  
Hülshorstweg 20  
33415 Verl  
Deutschland  
Telefon: +49 5246 9630  
[info@beckhoff.de](mailto:info@beckhoff.de)  
[www.beckhoff.de](http://www.beckhoff.de)

