

Dokumentation | DE

# BX5100

Busklemmen-Controller für CANopen





# 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	BX5100 - Einführung .....	13
2.3	Technische Daten .....	14
2.3.1	Technische Daten - BX .....	14
2.3.2	Technische Daten - CANopen .....	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	CANopen Verkabelung .....	27
<b>4</b>	<b>Parametrierung und Inbetriebnahme</b> .....	<b>35</b>
4.1	Anlaufverhalten des Busklemmen-Controllers .....	35
4.2	Konfiguration .....	35
4.2.1	Überblick .....	35
4.2.2	Anlegen einer TwinCAT-Konfiguration .....	37
4.2.3	Download einer TwinCAT-Konfiguration .....	38
4.2.4	Upload einer TwinCAT-Konfiguration .....	40
4.2.5	Ressourcen im Busklemmen-Controller .....	41
4.2.6	ADS-Verbindung über die serielle Schnittstelle .....	44
4.2.7	CANopen Slave-Schnittstelle .....	45
4.2.8	K-Bus .....	49
4.2.9	PLC .....	51
4.2.10	SSB .....	54
4.2.11	Echtzeit-Uhr (RTC) .....	75
4.2.12	COM-Port .....	76
4.2.13	Menü .....	77
4.2.14	Konfigurations-Software KS2000 .....	82

<b>5</b>	<b>Programmierung</b>	<b>83</b>
5.1	PLC-Eigenschaften der BX-Controller	83
5.2	TwinCAT PLC	83
5.3	TwinCAT PLC - Fehler-Codes	84
5.4	Persistente Daten	86
5.5	Remanente Daten	87
5.6	Lokierte Merker	88
5.7	Lokales Prozessabbild im Auslieferungszustand	89
5.8	Mapping der Busklemmen	90
5.9	Erzeugen eines Boot-Projekts	91
5.10	Lokales Prozessabbild in der TwinCAT-Konfiguration	92
5.11	Kommunikation zwischen TwinCAT und BX/BCxx50	93
5.12	Up- und Download von Programmen	94
5.13	Bibliotheken	98
5.13.1	Bibliotheken - Übersicht	98
5.13.2	TcBaseBX	101
5.13.3	TcSystemBX	112
5.13.4	TcComPortBX	115
5.13.5	TcTwinSAFE	126
5.14	Programmübertragung	133
5.14.1	Programmübertragung über die serielle Schnittstelle	133
5.14.2	Programmierung über CANopen	135
<b>6</b>	<b>CANopen Kommunikation</b>	<b>137</b>
6.1	CANopen Einführung	137
6.2	Protokollbeschreibung	138
6.2.1	Netzwerkmanagement	138
6.2.2	Prozessdatenobjekte (PDO)	143
6.2.3	PDO-Parametrierung	150
6.2.4	Servicedatenobjekte (SDO)	152
6.2.5	Identifizier-Verteilung	156
6.3	Objektverzeichnis	158
6.3.1	Objektverzeichnis - Struktur	158
6.3.2	Objektliste	159
6.3.3	Objekte und Daten	161
6.3.4	Objekte und Daten der BX5100/BC5150	183
6.4	ADS-Kommunikation	184
6.4.1	ADS-Dienste	184
<b>7</b>	<b>Fehlerbehandlung und Diagnose</b>	<b>185</b>
7.1	Diagnose	185
7.2	Diagnose-LEDs	186
7.3	Diagnose-Display	190
<b>8</b>	<b>Anhang</b>	<b>191</b>
8.1	Schnelleinstieg für erfahrene Anwender	191
8.2	CAN Identifizier-Liste	194
8.3	Firmware-Update	207

8.4	CFC-Client*.....	209
8.5	Beispielprogramme - Übersicht .....	212
8.6	Allgemeine Betriebsbedingungen .....	212
8.7	Zulassungen .....	214
8.8	Prüfnormen für Geräteprüfung .....	214
8.9	Literaturverzeichnis .....	215
8.10	Abkürzungsverzeichnis.....	216
8.11	Support und Service .....	217



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

### Firmware des BX5100

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.

Firmware	Beschreibung
1.27	<ul style="list-style-type: none"> <li>Ab dieser Version kann der BX auch ohne K-Bus-Klemmen verwendet werden, wenn diese im System Manager nicht angelegt sind, es wird dann kein Fehler erzeugt</li> </ul>
1.20	<ul style="list-style-type: none"> <li>Unterstützung der TwinSAFE-Klemmen, maximal eine Logik-Klemme am K-Bus mit maximal 7 Verbindungen erlaubt</li> <li>Umschaltung der COM 2 Schnittstelle zwischen RS232 und RS485 geändert</li> </ul> <p><b>ACHTUNG</b> Die Firmware 1.20 ist nicht auf älteren Hardware-Versionen (keiner 3.5) lauffähig. Den HW Stand entnehmen Sie dem Aufkleber des BX Controllers</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 vorher ‚9.‘</li> <li>TC-Config 4 und 8-fach analog Klemmen gefixt</li> </ul>
1.14	<ul style="list-style-type: none"> <li>K-Bus aktualisiert</li> <li>CAN-Slave-Schnittstelle 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> <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: CFC-Client Unterstützung implementiert (Controller Flash Copy - Client)</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 am SSB</li> <li>Disable der roten PLC Zyklusüberwachungs-LED</li> </ul>
1.04 / 0.99d	<ul style="list-style-type: none"> <li>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>ab PDO 5 können in der Default Konfiguration bis zu PDO 32 aktiviert werden</li> <li>Wenn nach einem Online Change sofort 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>Feldbus Diagnose in der TwinCAT Konfiguration implementiert</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

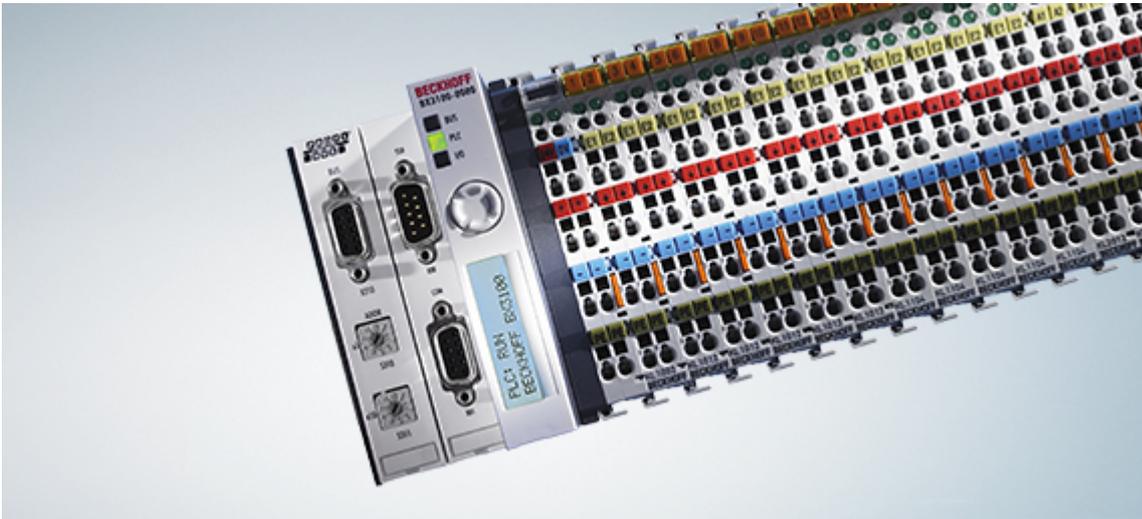


Abb. 1: Busklemmen-Controller der BX-Serie

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 BX5100 - Einführung

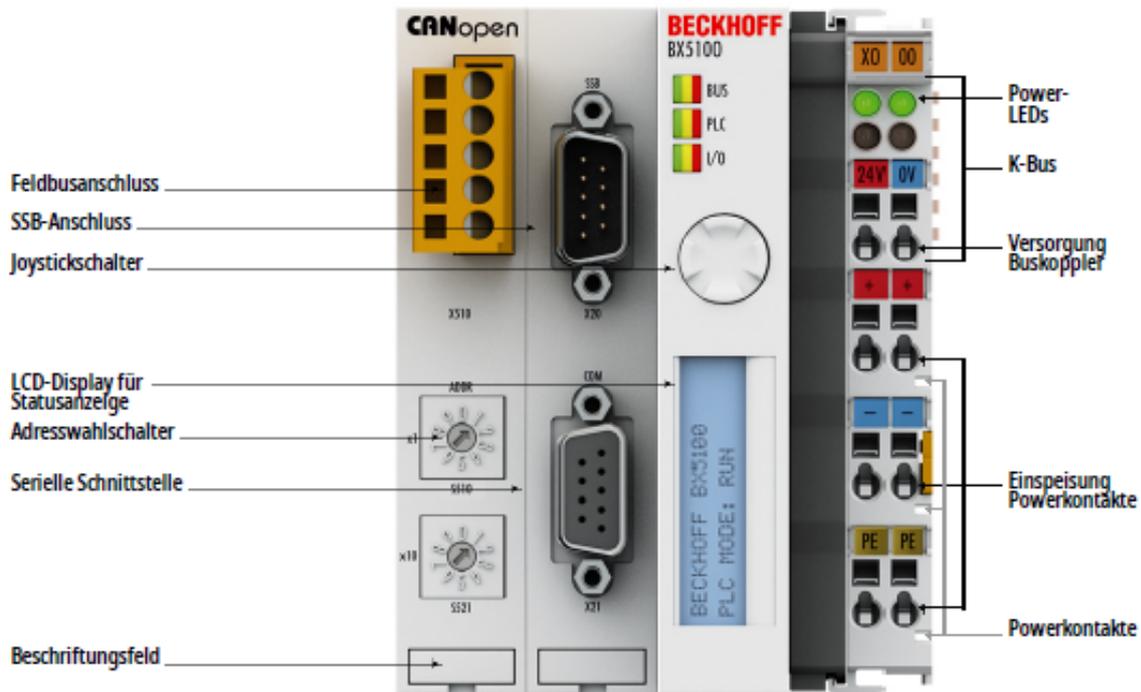


Abb. 2: BX5100

Der Busklemmen Controller BX5100 besitzt eine CANopen-Slaveschnittstelle. Er verfügt über eine automatische Baudratenerkennung bis 1 MBaud und einen Adresswahlschalter für die Adressvergabe. Es können bis zu 16 TxPDOs und 16 RxPDOs mit der Steuerung ausgetauscht werden.

Eine Einheit besteht aus dem Busklemmen Controller BX5100 mit bis zu 64 Busklemmen und einer Busendklemme. Mit dem System der Klemmenbusverlängerung ist der Anschluss von bis zu 255 Busklemmen möglich. Programmiert wird der Controller über die COM1- oder über die CANopen-Schnittstelle der PC-Feldbuskarte FC510x.

## 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 - CANopen

Systemdaten	CANopen (BX5100)							
Anzahl der Knoten	63, mit Repeater 99							
Anzahl der E/A-Punkte	steuerungsabhängig							
Übertragungsmedium	abgeschirmtes, verdrehtes Kupferkabel, 2 x Signal, 1 x CAN Ground (empfohlen)							
Leitungslänge	5000 m	2500 m	1000 m	500 m	250 m	100 m	50 m	25 m
Übertragungsrate [kBaud ]	10	20	50	125	250	500	800	1000
I/O-Kommunikationsarten	ereignisgesteuert, zyklisch, synchron, Polling							
Anzahl PDOs	32 Sende- und 32 Empfangsprozessdatenobjekte							
Baudrate	automatische Baudratenerkennung (mögliche Baudraten siehe Übertragungsrate)							

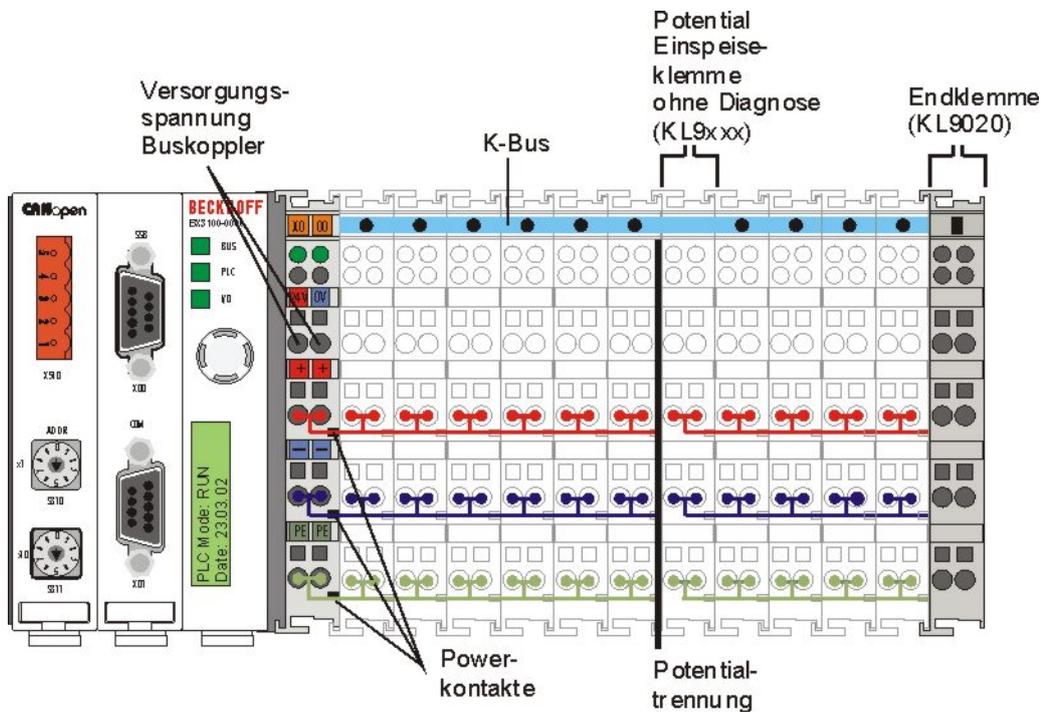
### 2.3.3 Technische Daten - SSB-Interface

Systemdaten	SSB-Interface
Max. Anzahl an Slaves	8
Max. Anzahl an PDOs	32 RxPODs / 32 TxPDOs
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



## 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.

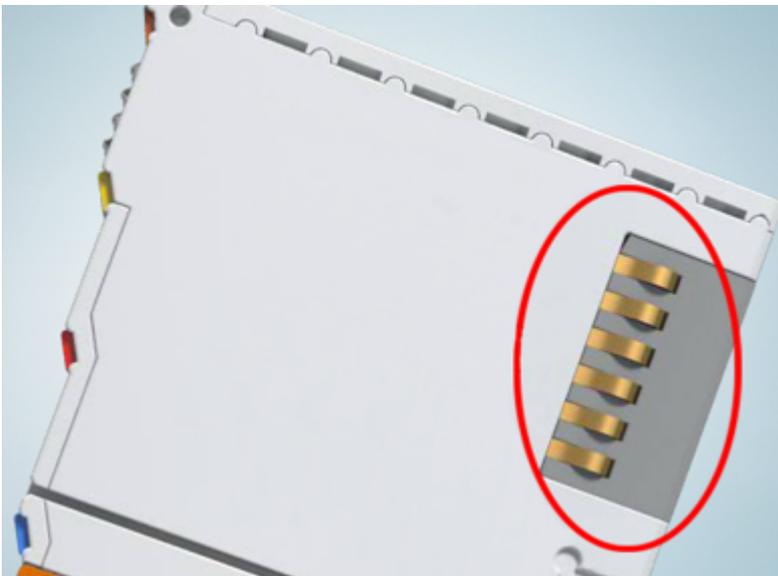


Abb. 3: Federkontakte der Beckhoff I/O-Komponenten

### 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.

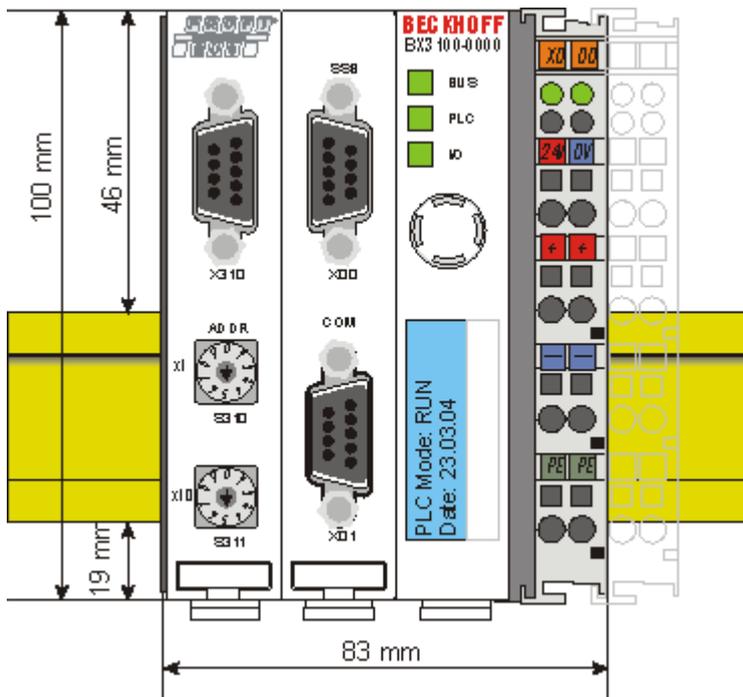


Abb. 4: BX3100, BX5100, BX5200, BX9000

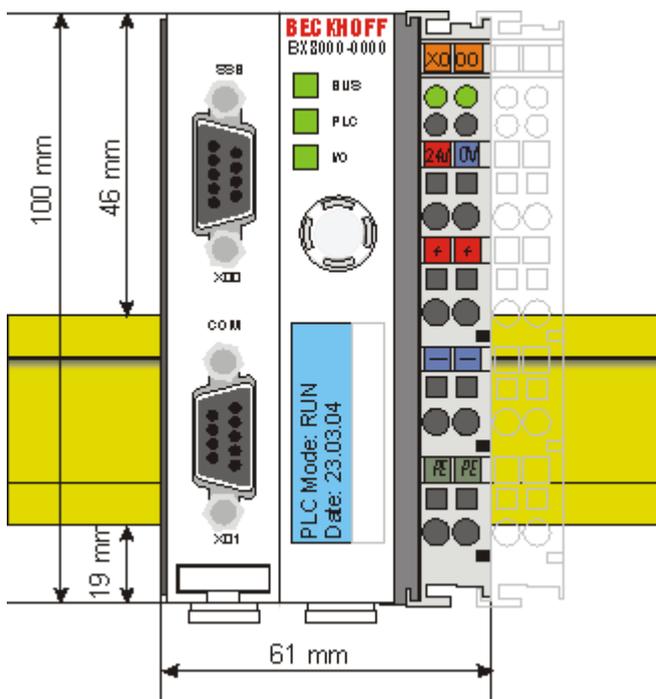


Abb. 5: BX8000

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.

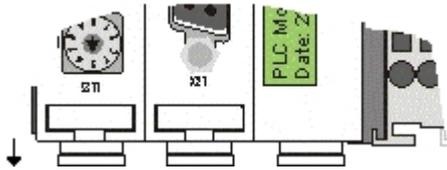


Abb. 6: Entriegelter BX-Controller

**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.

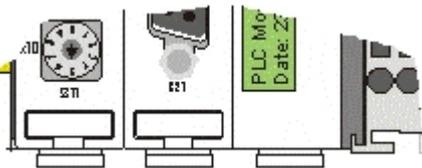


Abb. 7: Verriegelter BX-Controller

#### 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.

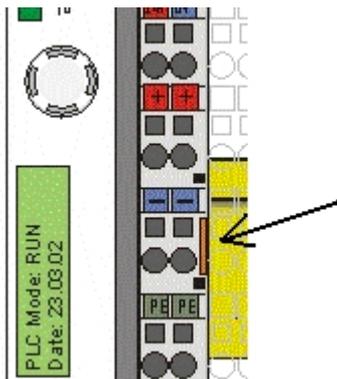


Abb. 8: Demontage

### 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.

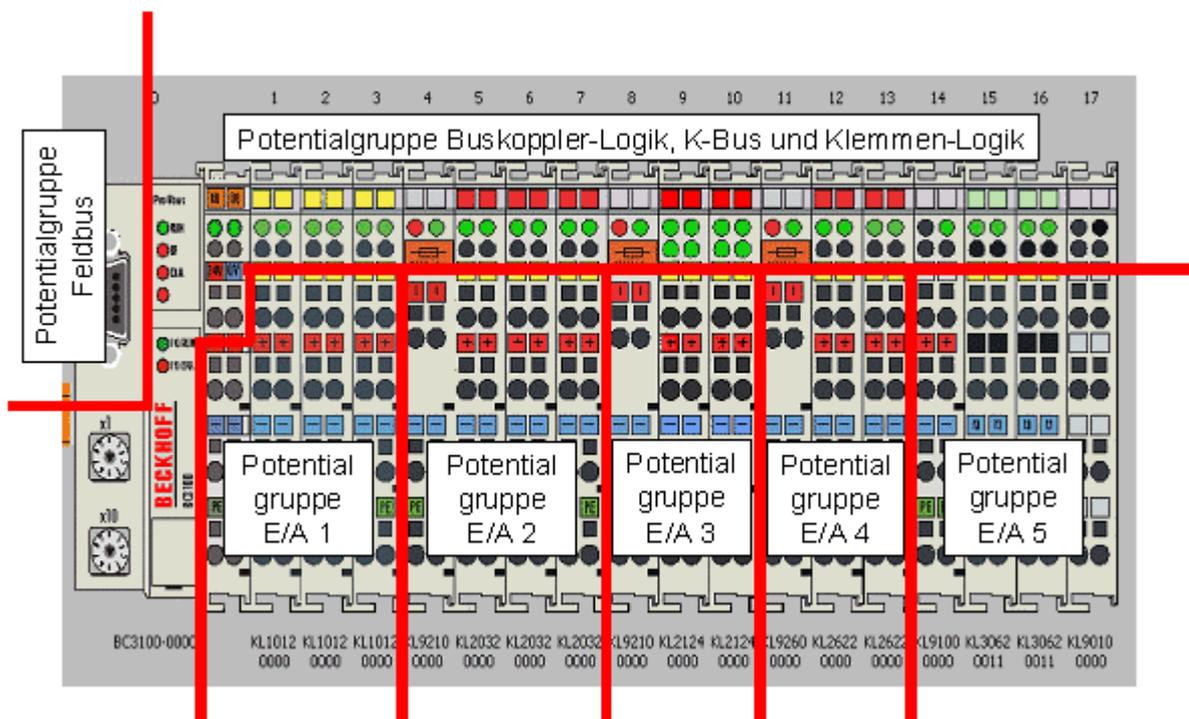


Abb. 9: Potentialgruppen eines Busklemmenblocks

#### 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 Schutzerde eingesetzt werden. Der Kontakt ist aus Sicherheitsgründen beim Zusammenstecken voreilend und kann Kurzschlussströme bis 125 A ableiten.

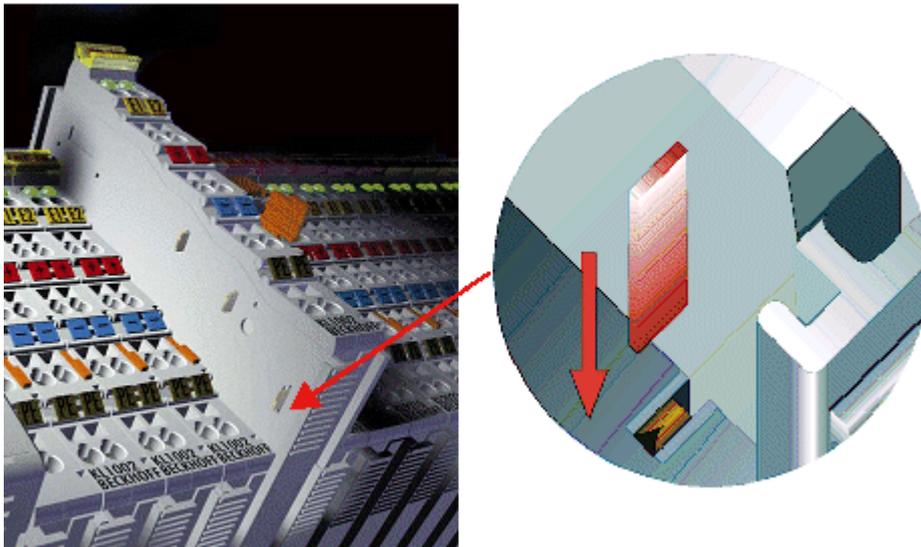


Abb. 10: Linksseitiger Powerkontakt

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>

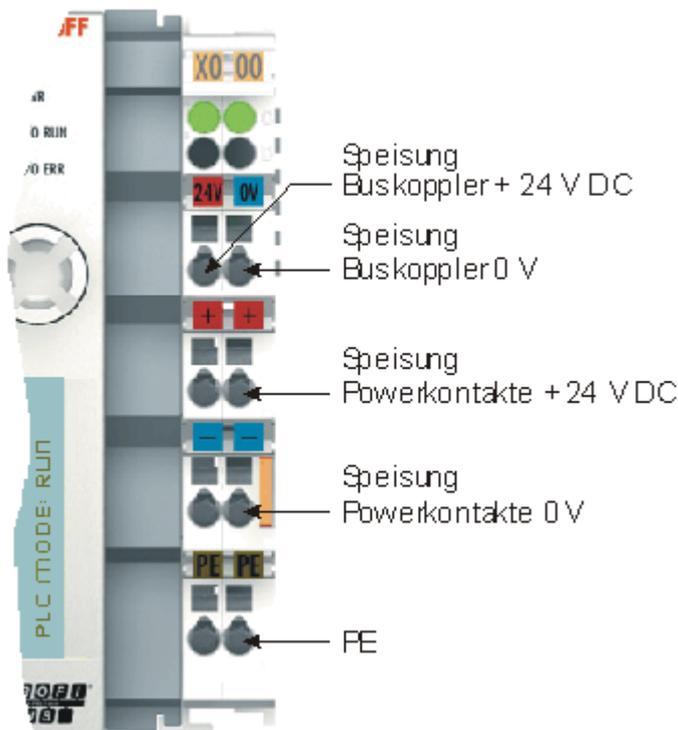


Abb. 11: Klemmstellen zur Versorgung des Busklemmen-Controllers

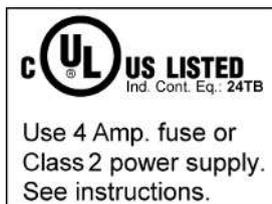


Abb. 12: UL-Kennzeichnung der BX-Controller

### 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 Schlitzen 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.

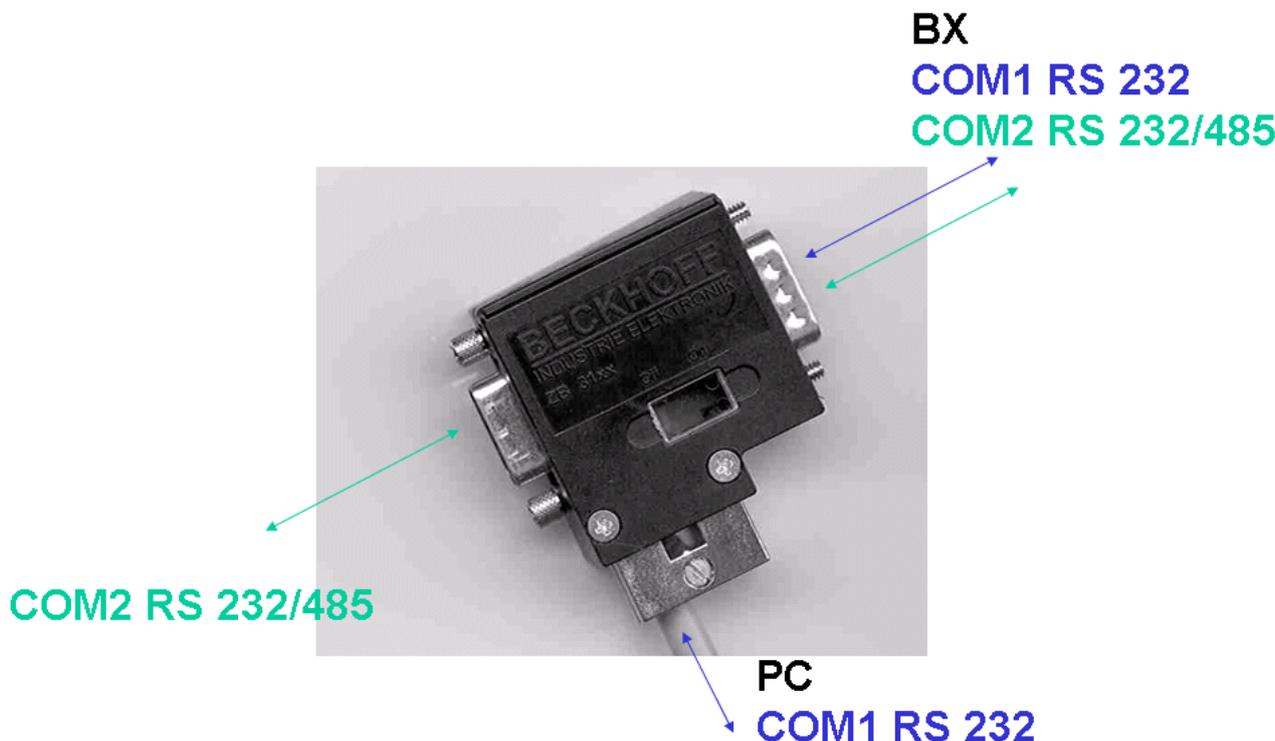


Abb. 13: Programmierkabel ZK1000-0030 - COM 1 und COM 2

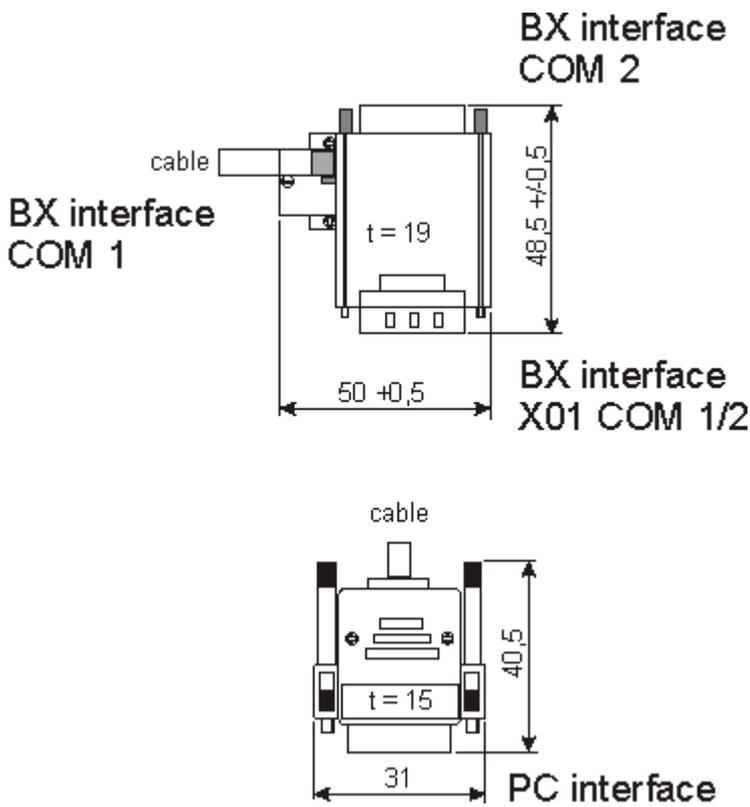


Abb. 14: Programmierkabel ZK1000-0030 - Abmessungen der Stecker

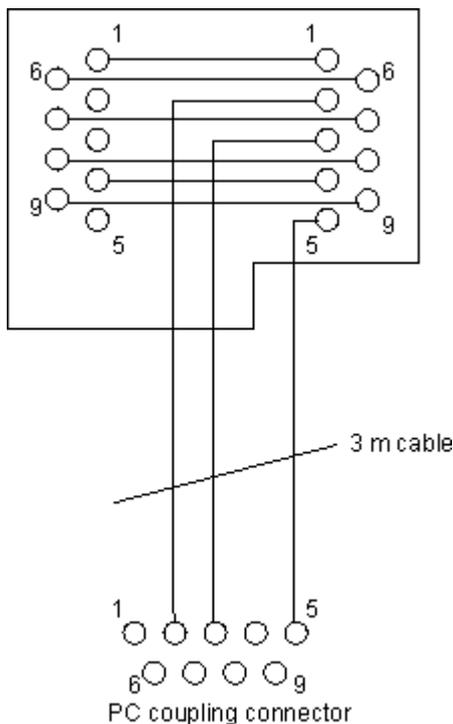


Abb. 15: Programmierkabel ZK1000-0030 - Pinning

### 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 [► 104] zur Verfügung.

**SSB-Schnittstelle**

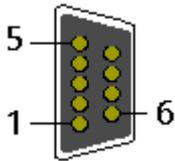


Abb. 16: SSB-Schnittstelle

**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**

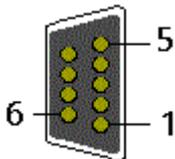


Abb. 17: COM1- (RS 232) und COM2-Schnittstelle (RS 232/485)

**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 CANopen Verkabelung**

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

### 3.4.5.1 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!

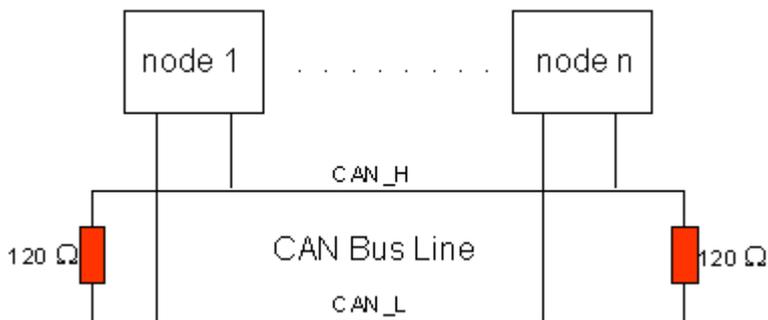


Abb. 18: Abschluss des Busses mit Abschlusswiderstand 120 Ohm

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.

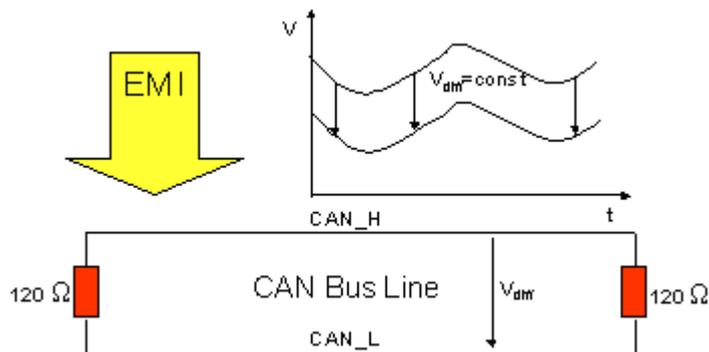


Abb. 19: Unempfindlichkeit gegen eingeprägte Störungen

### 3.4.5.2 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.

### 3.4.5.3 Sticheleitungen

Sticheleitungen ("drop lines") sind nach Möglichkeit zu vermeiden, da sie grundsätzlich zu Signalreflexionen führen. Die durch Sticheleitungen 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 Sticheleitungslängen nicht überschritten werden:

Bit-Rate	Länge Sticheleitung	gesamte Länge aller Sticheleitungen
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

Sticheleitungen dürfen nicht mit Abschlusswiderständen versehen werden.

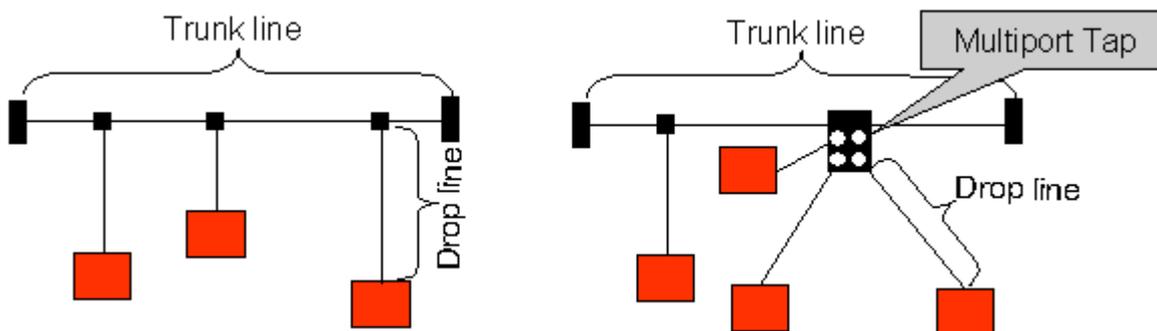


Abb. 20: Beispieltopologie Sticheleitungen

### 3.4.5.4 Sternverteiler (Multiport Tap)

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

Bit-Rate	Länge Sticheleitung bei Multiport Topologie	Länge Trunk Line (ohne Sticheleitungen)
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

### 3.4.5.5 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)

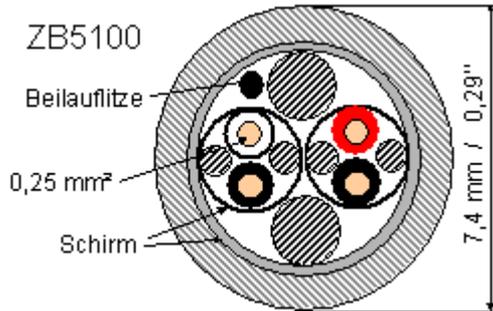


Abb. 21: Aufbau CAN-Kabel ZB5100

### 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 Beilauflitze
- 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

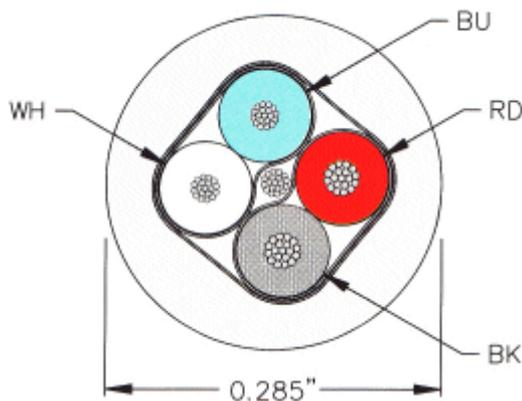


Abb. 22: Aufbau CAN-/DeviceNet-Kabel ZB5200

### 3.4.5.6 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.

### 3.4.5.7 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>

### 3.4.5.8 BK5151, FC51xx, CX mit CAN Interface und EL6751: D-Sub 9polig

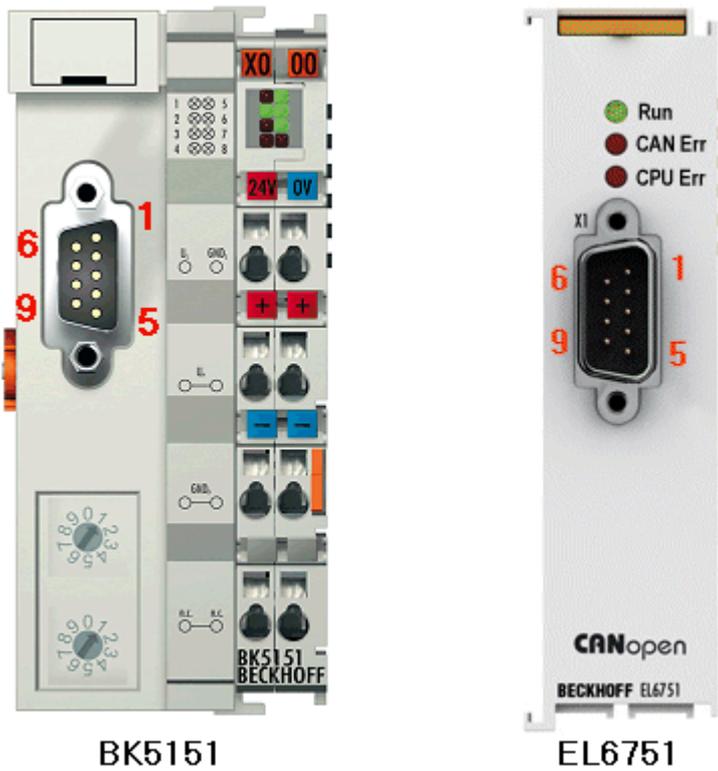
Die CAN Busleitung wird an die FC51x1, FC51x2 CANopen Karten und bei der EL6751 CANopen Master-/Slaveklemme über 9polige Sub-D-Buchsen mit folgender Steckerbelegung angeschlossen.

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. 23: Pinbelegung BK5151, EL6751

#### FC51x2:

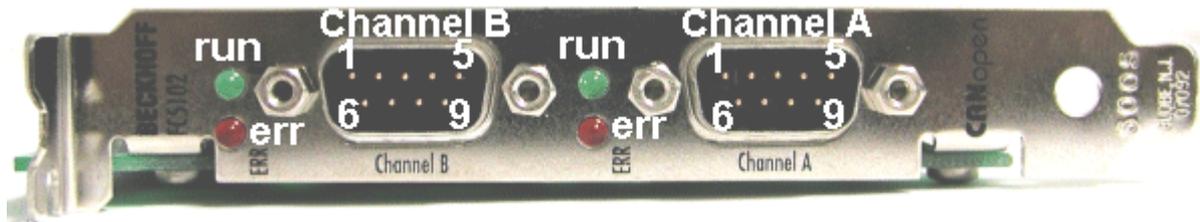


Abb. 24: FC51x2

### 3.4.5.9 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. 25: 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.

### 3.4.5.10 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.

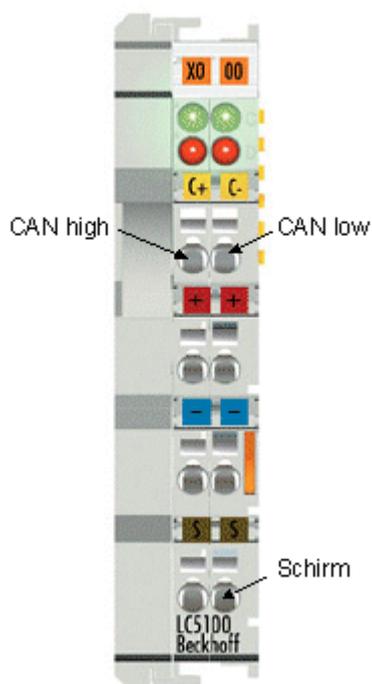


Abb. 26: LC5100

**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.

**3.4.5.11 Feldbus Box: M12 CAN Buchse**

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

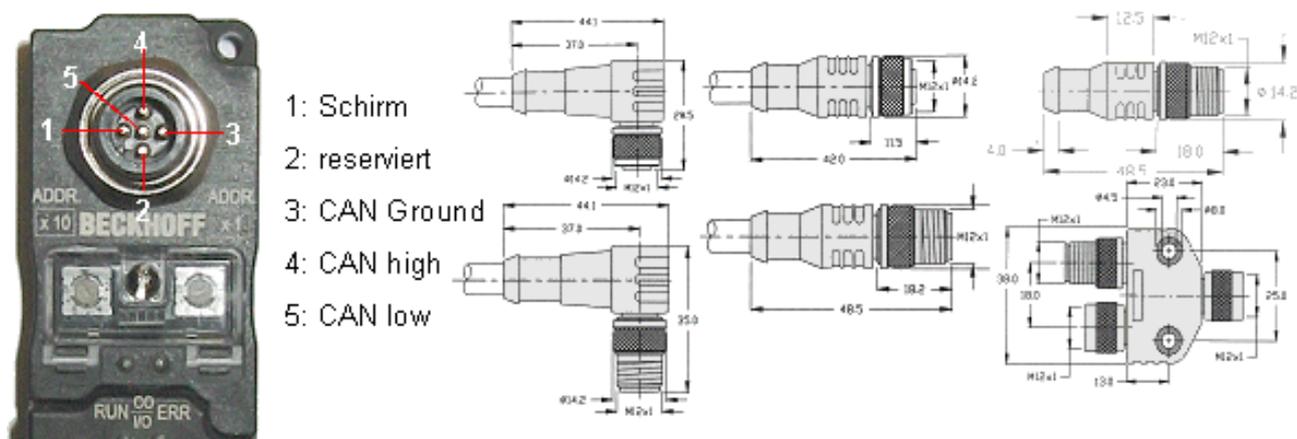


Abb. 27: Pinbelegung M12 Stecker Feldbus Box

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 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*).

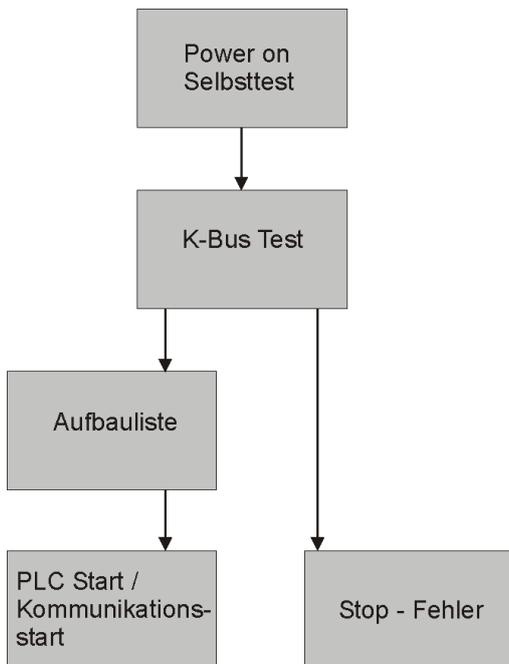


Abb. 28: Anlaufverhalten des Busklemmen-Controllers

## 4.2 Konfiguration

### 4.2.1 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 parametrierbar 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 seriell 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 parametrisiert 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

## 4.2.2 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.

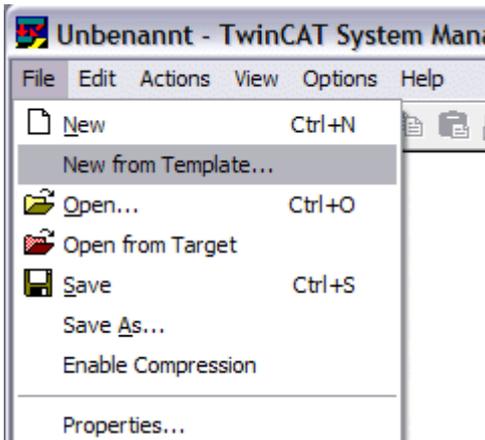


Abb. 29: Anlegen einer TwinCAT-Konfiguration

Wählen Sie den entsprechenden Busklemmen-Controller aus.

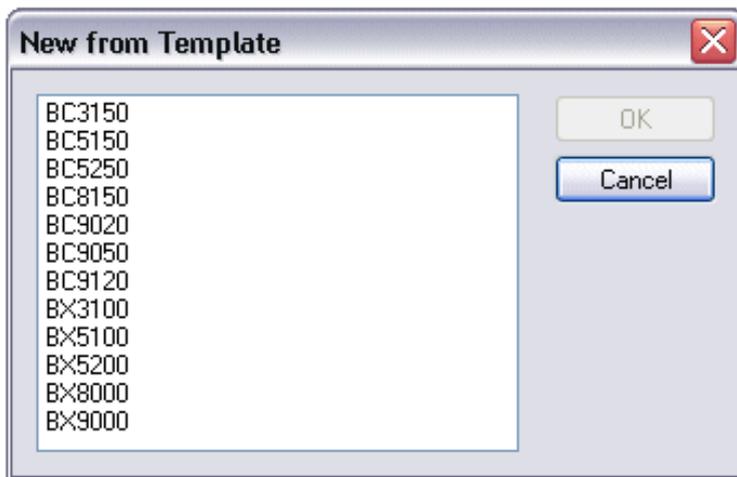


Abb. 30: Auswahl des Busklemmen-Controllers

Nun sind alle Komponenten des Busklemmen-Controllers vorhanden:

- Feldbusschnittstelle
- [K-Bus Interface \[► 49\]](#)
- [PLC Programm \[► 51\]](#)
- [SSB \[► 54\]](#) (nur Busklemmen-Controller der BX-Serie)

Die Konfiguration der Geräte entnehmen Sie den entsprechenden Kapiteln.

### 4.2.3 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 \[► 44\]](#) 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-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.

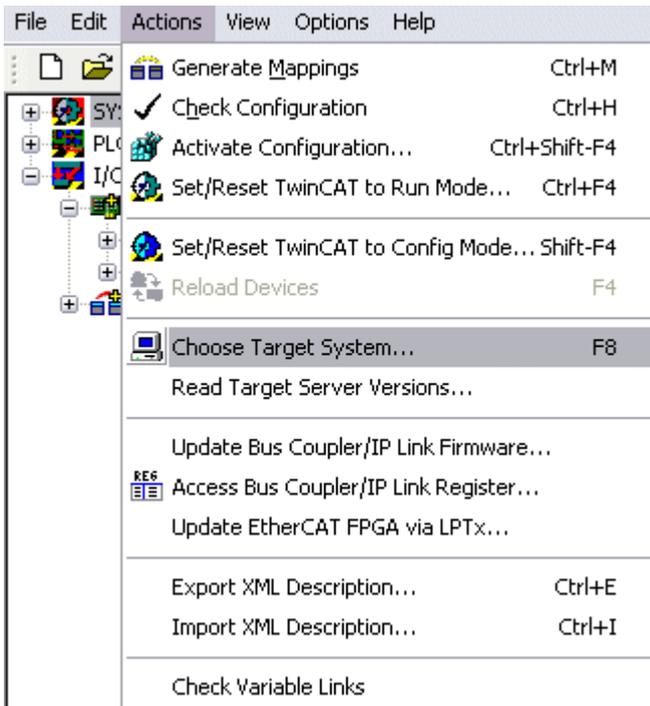


Abb. 31: Download einer TwinCAT-Konfiguration

Wählen Sie den entsprechenden Busklemmen-Controller aus.

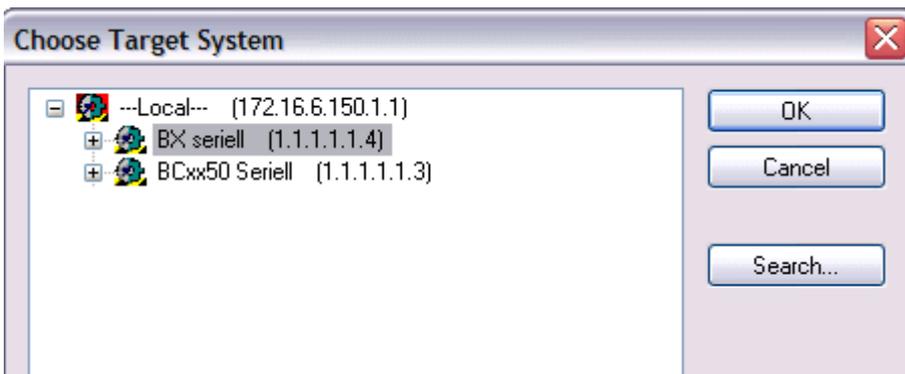


Abb. 32: Auswahl des Busklemmen-Controllers

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

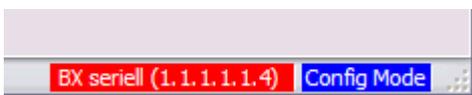


Abb. 33: Zustand des Busklemmen-Controllers

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*.

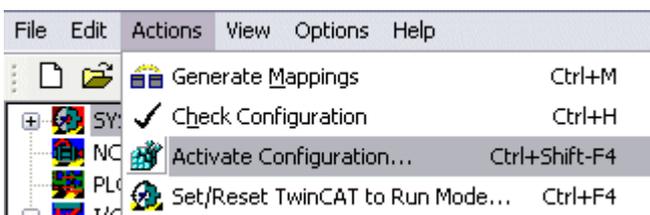


Abb. 34: Aktivieren der TwinCAT-Konfiguration

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).

## 4.2.4 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 \[▶ 44\]](#) 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 indem Sie Ihr File auf das entsprechende Gerät herunterladen können.

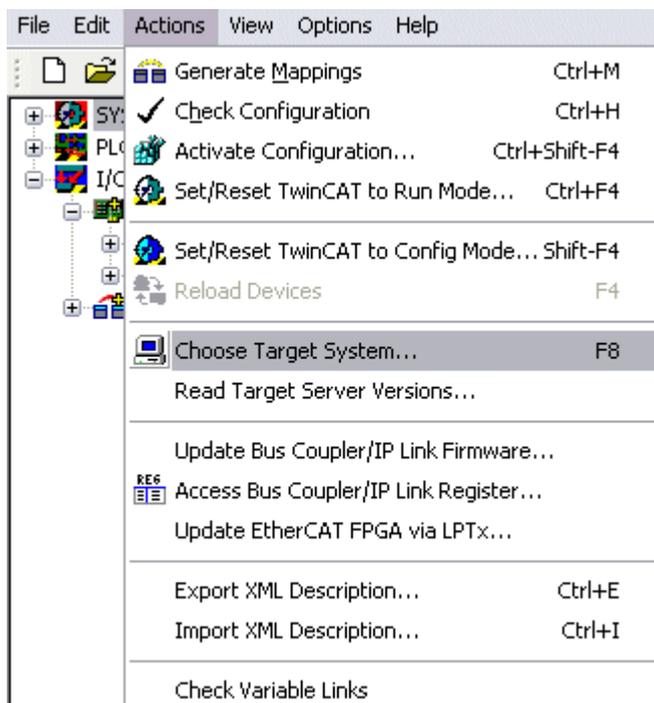


Abb. 35: Auswahl des Zielsystems

Wählen Sie den entsprechenden Busklemmen-Controller aus.

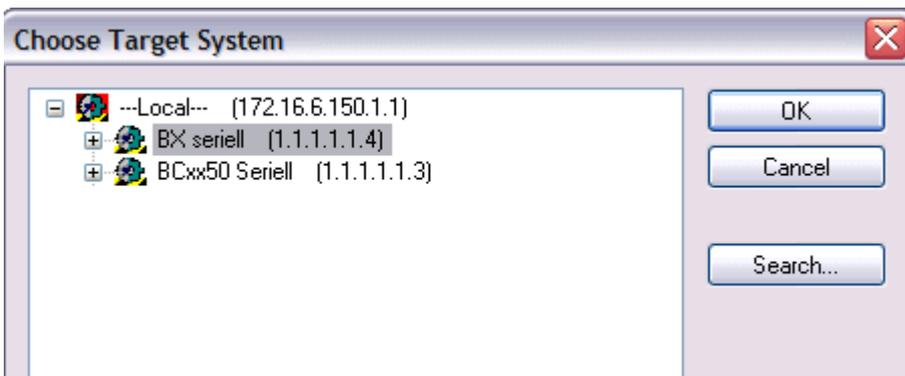


Abb. 36: Auswahl des Busklemmen-Controllers

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

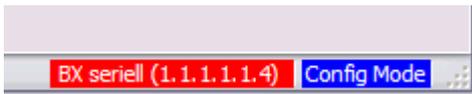


Abb. 37: Zustand des Busklemmen-Controllers

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

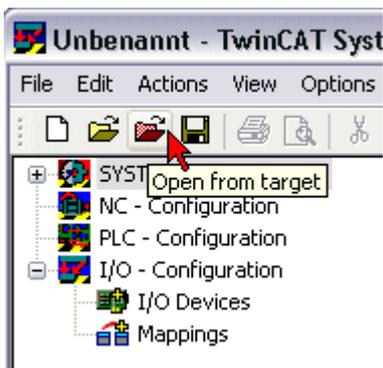


Abb. 38: Hochladen der TwinCAT-Konfiguration

## 4.2.5 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.

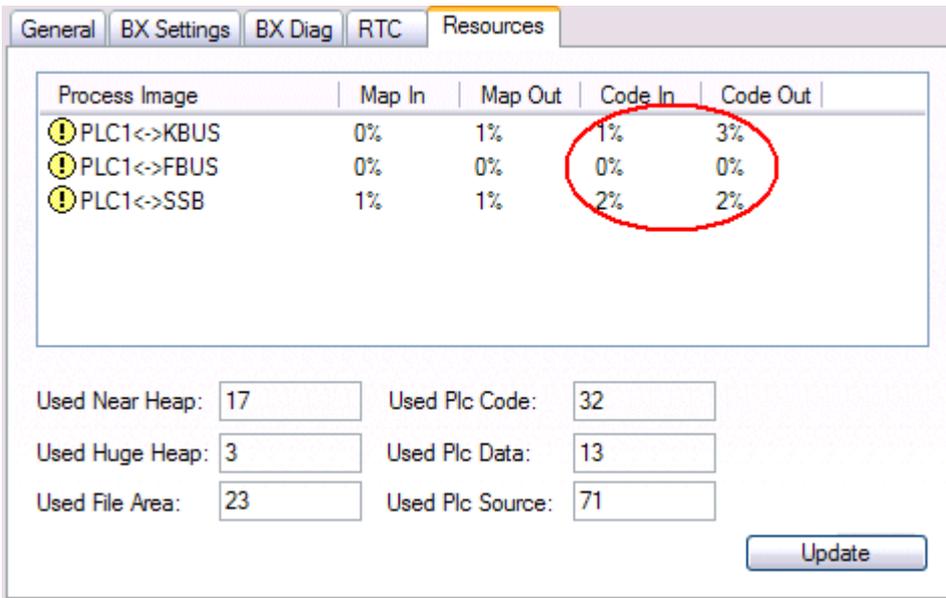


Abb. 39: Speicher für das Code Mapping

**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.

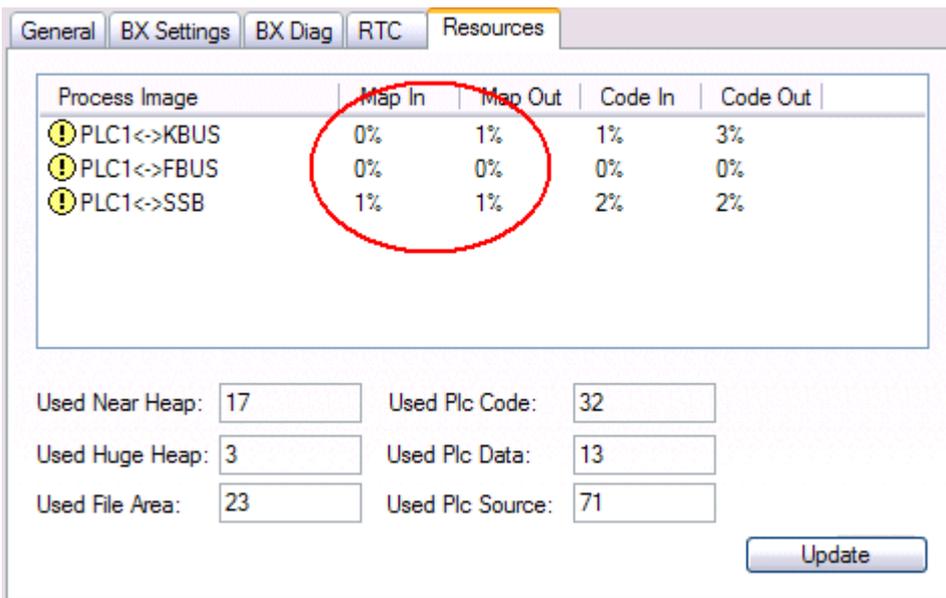


Abb. 40: Daten Speicher Mapping

**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 %.

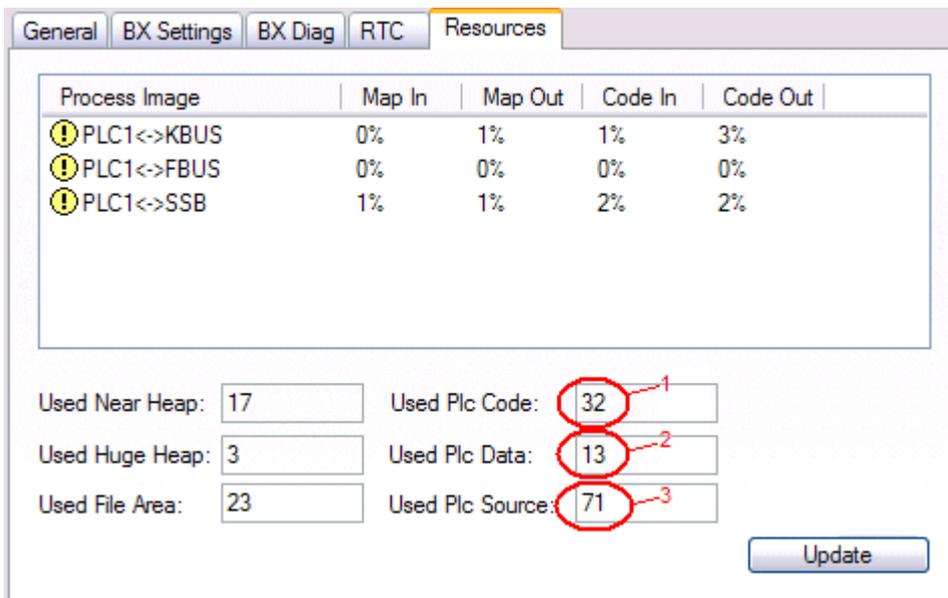


Abb. 41: Code und Daten Speicher

**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 %.

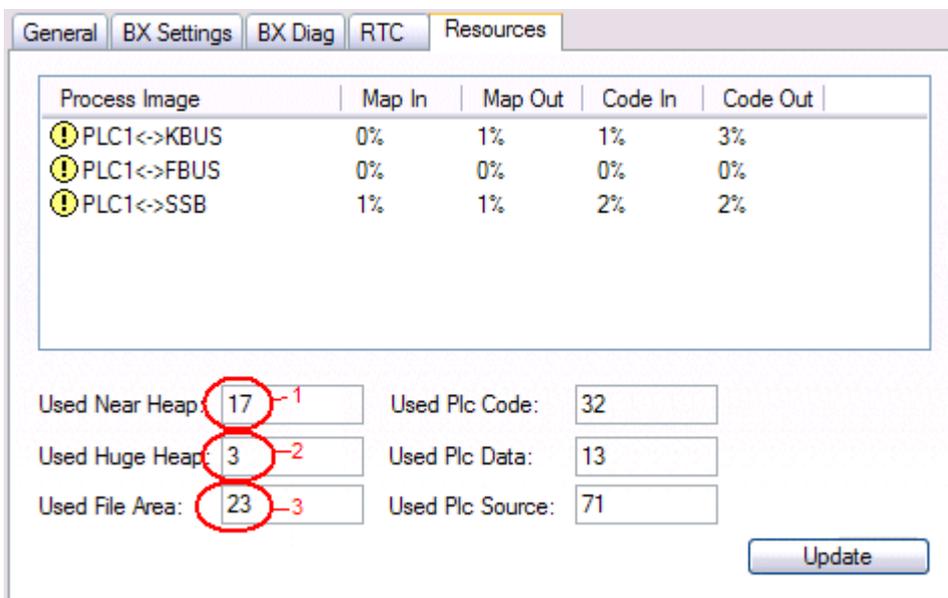


Abb. 42: Sonstiger Speicher

## 4.2.6 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 (TwnCAT Level PLC, NC oder NCI)

### ● **Verwenden Sie nur eine serielle Verbindung**

**i** 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.

### ● **AMS-Net-ID im Auslieferungszustand (Default)**

#### **i** 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.

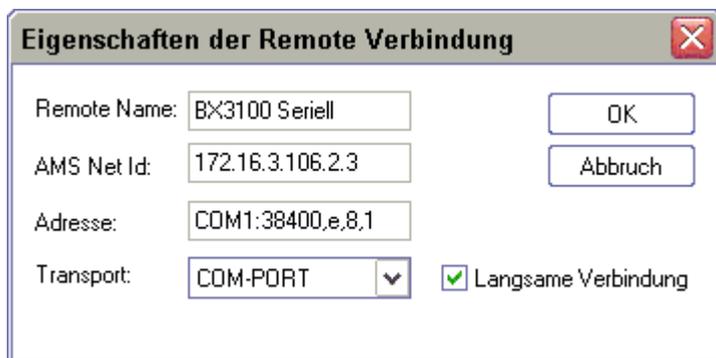


Abb. 43: Eigenschaften der Remote-Verbindung

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.

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

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.

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

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.

**AMS**                      AMS-Net-ID  
 1.1.1.1.1.1

**4.2.7                      CANopen Slave-Schnittstelle**

**4.2.7.1                      CANopen Slave-Schnittstelle**

Es gibt zwei Konfigurationsarten. In der Default Konfiguration (Auslieferungszustand) mappen sich die CANopen Daten, der CANopen Slave Schnittstelle, ab der Adresse 1000 des BX5100/BC5150 und es sind die ersten 4 PDOs aktiviert. In der TwinCAT Konfiguration kann man über den System Manager eine beliebige Konfiguration erstellen und Variablen beliebig mit der CANopen Slave Schnittstelle verknüpfen.

**Default Konfiguration**

In dieser Konfiguration mappen sich die CANopen Daten wie folgt:

**CANopen Daten**

PDO Nummer	Lesen/Schreiben	BX Prozessabbild
PDO 1	Rx/Tx	%IB1000...%IB1007/QB1000...%QB1007
PDO 2	Rx/Tx	%IB1008...%IB1015/QB1008...%QB1015
PDO 3	Rx/Tx	%IB1016...%IB1023/QB1016...%QB1023
PDO 4	Rx/Tx	%IB1024...%IB1031/QB1024...%QB1031

Die Tx-PDOs werden in der Default Konfiguration nur bei Änderung übertragen.

**Mehr als 4 PDOs**

Möchte man mehr als 4 PDOs nutzen kann das in der TwinCAT Konfiguration als auch in der Default Konfiguration erfolgen. Bei der TwinCAT Konfiguration erstellt man die benötigten PDOs in seinem System Manager File und lädt diese Konfiguration zum BX5100/BC5150. In der Default Konfiguration muss vor dem Starten des Knoten dies durch die Objekte 0x14xx (für TxPDOs) und 0x18xx (für die RxPDOs) erfolgen. Dabei ist der Sub-Index 1 die COB ID einzutragen und im Sub-Index 2 der Typ der Datenübertragung.

Beispiel für PDO 5 des Konten mit der Adresse 11:

0x1804 Sub-Index 1 Länge 4 Wert 0x68B

0x1804 Sub-Index 2 Länge 1 Wert 0xFF (nicht unbedingt notwendig zur Aktivierung des PDOs)

0x1404 Sub-Index 1 Länge 4 Wert 0x78B

0x1404 Sub-Index 2 Länge 1 Wert 0xFF (nicht unbedingt notwendig zur Aktivierung des PDOs)

Es können beim BX5100 maximal 32 PDOs in jede Richtung verwendet werden (32 TxPDOs und 32 RxPDOs). Es können beim BC5150 maximal 16 PDOs in jede Richtung verwendet werden (16 TxPDOs und 16 RxPDOs).

PDO Nummer	Lesen/Schreiben	BX Prozessabbild
PDO 5	Rx/Tx	%IB1032...%IB1039/QB1032...%QB1039
PDO 6	Rx/Tx	%IB1040...%IB1047/QB1040...%QB1047
PDO 7	Rx/Tx	%IB1048...%IB1055/QB1048...%QB1055
...	Rx/Tx	...
PDO 32	Rx/Tx	%IB1248...%IB1255/QB1248...%QB1255

### Azyklisch Synchron

PDOs der Übertragungsart 0 arbeiten synchron, aber nicht zyklisch. Ein RxPDO wird erst nach Empfang des nächsten SYNC-Telegramms ausgewertet. Damit lassen sich beispielsweise Achsgruppen nacheinander mit neuen Zielpositionen versehen, die alle beim nächsten SYNC gültig werden - ohne dass ständig Stützstellen ausgegeben werden müssen. Ein Gerät, dessen TxPDO auf Übertragungsart 0 konfiguriert ist, ermittelt seine Eingangsdaten beim Empfang des SYNC (synchrones Prozessabbild) und sendet sie anschließend, falls die Daten einem Ereignis entsprechen (beispielsweise eine Eingangsänderung) eingetreten ist. Die Übertragungsart 0 kombiniert also den Sendegrund "ereignisgesteuert" mit dem Sende- (und möglichst Sample-) bzw. Verarbeitungs-Zeitpunkt *SYNC-Empfang*.

### Zyklisch Synchron

Bei Übertragungsart 1-240 wird das PDO zyklisch gesendet: nach jedem "n-ten" SYNC (n=1...240). Da die Übertragungsart nicht nur im Netz, sondern auch auf einem Gerät kombiniert werden dürfen, kann so z. B. ein schneller Zyklus für digitale Eingänge vereinbart werden (n=1), während die Daten der Analogeingänge in einem langsameren Zyklus übertragen werden (z. B. n=10). RxPDOs unterscheiden in der Regel nicht zwischen den Übertragungsarten 0...240: ein empfangenes PDO wird beim nächsten SYNC-Empfang gültig gesetzt. Die Zykluszeit (SYNC-Rate) kann überwacht werden (Objekt 0x1006), das Gerät reagiert bei SYNC-Ausfall dann entsprechend der Definition des Geräteprofils und schaltet z. B. seine Ausgänge in den Fehlerzustand.

Die CANopen PC-Karten CIFn0 senden stets ereignisgesteuert - auch wenn die Übertragungsart im Bereich von 1-240 eingestellt ist. Dieses Verhalten entspricht etwa der Übertragungsart 0. Die PC Karten FC510x unterstützen die Zyklisch Synchron Übertragungsart vollständig.

### Asynchron

Die Übertragungsarten 254 und 255 sind asynchron oder auch ereignisgesteuert. Bei Übertragungsart 254 ist das Ereignis herstellerspezifisch, bei 255 im Geräteprofil definiert. Im einfachsten Fall ist das Ereignis die Veränderung eines Eingangswertes - es wird also jede Werteänderung übertragen.

### Sendeverzögerungszeit (Inhibit Time)

(Diese Funktion wird von BC5150 und BX5100 nicht unterstützt.)

Über den Parameter *Inhibit Time* kann ein Sende-Filter aktiviert werden, der die Reaktionszeit bei der relativ ersten Eingangsänderung nicht verlängert, aber bei unmittelbar darauf folgenden Änderungen aktiv ist. Die Sendeverzögerungszeit beschreibt die Zeitspanne, die zwischen dem Versenden zweier gleicher Telegramme mindestens abgewartet werden muss. Wenn die Inhibit Time genutzt wird, so kann die maximale Busbelastung und damit die Latenzzeit im "worst case"-Fall ermittelt werden.

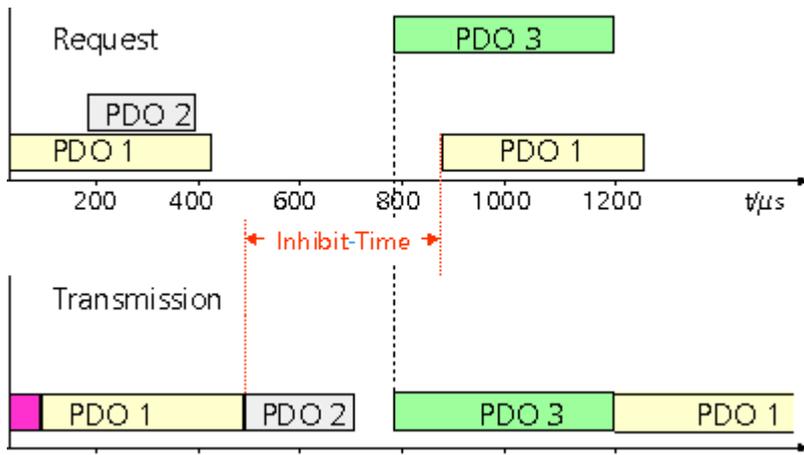


Abb. 44: Sendeverzögerungszeit (Inhibit Time)

Eine Spreizung der gesendeten PDOs (Sendeverzögerung) ergibt sich automatisch aus der gewählten Zyklus-Zeit der SPS - und es macht wenig Sinn, die SPS schneller laufen zu lassen als es die Busbandbreite zulässt. Zudem kann die Busbelastung wirkungsvoll über die synchrone Kommunikation beeinflusst werden.

**Event Timer**

Über Subindex 5 der Kommunikationsparameter lässt sich ein Event Timer (Ereignis-Timer) für Sende-PDOs festlegen. Der Ablauf dieses Timers wird als zusätzlich eingetretenes Ereignis für das entsprechende PDO gewertet, das PDO wird also dann gesendet. Wenn das Applikationsereignis während einer Timer-Periode auftritt, so wird ebenfalls gesendet und der Timer wird zurückgesetzt.

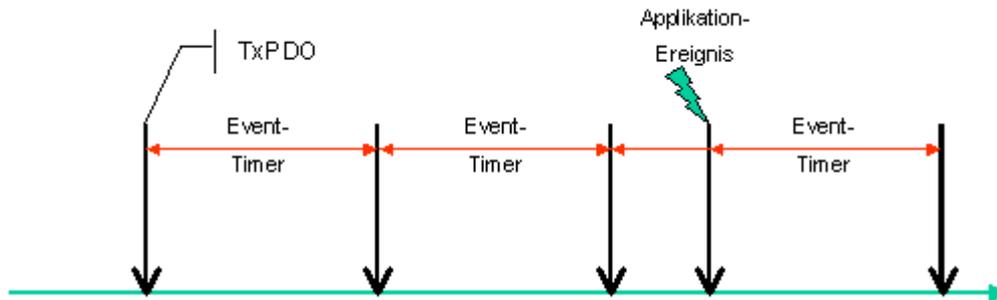


Abb. 45: Event Timer (Ereignis-Timer)

Bei Empfangs-PDOs wird der Timer-Parameter dazu verwendet, die Überwachungszeit für dieses PDO anzugeben: Die Applikation wird benachrichtigt, wenn kein entsprechendes PDO innerhalb der eingestellten Zeit empfangen wurde.

**4.2.7.2 TwinCAT-Konfiguration**

Die TwinCAT-Konfiguration ermöglicht das freie, adressunabhängige Mapping der PLC-Daten zur CAN Slave-Schnittstelle.

**TwinCAT Konfiguration**

Für die Konfiguration benötigen Sie den TwinCAT System Manager und eine ADS-Verbindung zum Busklemmen-Controller.

Die ADS-Verbindung kann über die serielle Schnittstelle (siehe serielles ADS) oder den Feldbus (PCI Karte FC510x und die CAN Slave-Schnittstelle des BX5100) erfolgen.

### 4.2.7.3 Slave-Adresse

Die Slave-Adresse ist über die zwei Drehwahlschalter einzustellen. Die Default-Einstellung beträgt 11. Es sind alle Adressen von 0 bis 63 erlaubt. Jede Adresse darf im Netzwerk nur einmal vorkommen. Die Adresse wird im ausgeschalteten Zustand des BX-Controllers 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 BX-Controller wieder eingeschaltet wird.

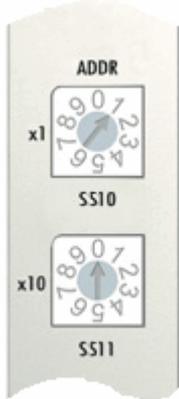


Abb. 46: Einstellen der Node ID

#### Beispiel

- Sie wollen Adresse 34 einstellen.
- unterer Drehwahlschalter S311: 3
  - oberer Drehwahlschalter S310: 4

### 4.2.7.4 Baud-Rate

#### Auto Baud-Rate

Für die automatische Erkennung der Baud-Rate ist es erforderlich, dass mehrere gültige Telegramme der gewünschten Baud-Rate auf dem Bus vorhanden sind. Während die Baud-Rate gesucht wird blinken die LEDs RUN und CAN ERR in schnellem Wechsel. Sobald eine Baud-Rate erkannt und übernommen wurde fährt der Busklemmen Controller mit der Initialisierung fort.

Ein Software-Reset führt nicht zur erneuten Suche der Baud-Rate. Die vorher aktive Baud-Rate bleibt erhalten.

#### Bit Timing

Folgende Baud-Raten und Bit-Timing RegisterEinstellungen werden von den BECKHOFF CANopen-Geräten unterstützt:

Baud-Rate [kBaud]	BTR0	BTR1	Sampling Point
1000	0x00	0x14	75%
800	0x00	0x16	80%
500	0x00	0x1C	87%
250	0x01	0x1C	87%
125	0x03	0x1C	87%
100	0x04	0x1C	87%
50	0x09	0x1C	87%
20	0x18	0x1C	87%
10	0x31	0x1C	87%

Die angegebenen Bit-Timing Registerinstellungen (BTR0, BTR1) gelten z. B. für die CAN-Controller Philips 82C200, SJA1000, Intel 80C527, Siemens 80C167, und andere. Sie sind für maximale Buslänge optimiert.

## 4.2.8 K-Bus

### ● Busklemme und Endklemme erforderlich

**I** 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

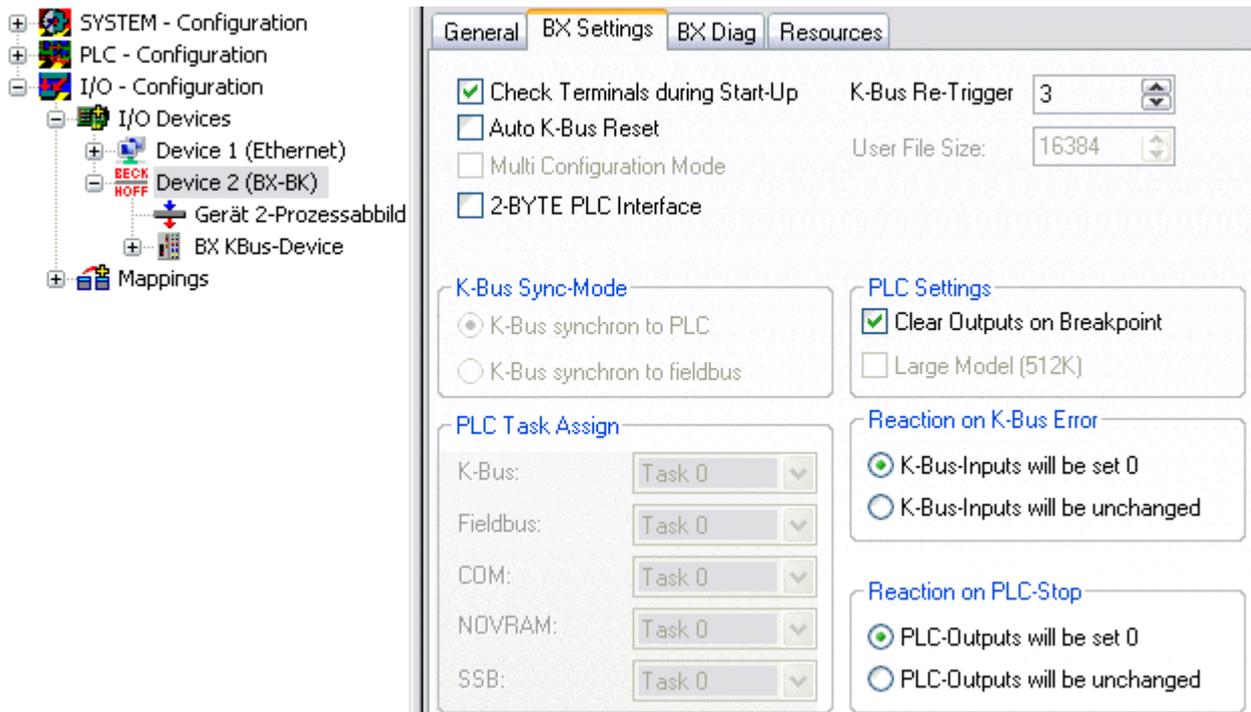


Abb. 47: Karteireiter BX Settings

#### 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, dass 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

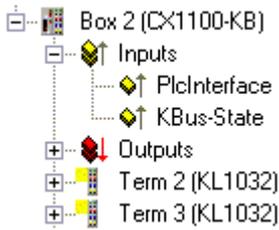
Anzeige der Zykluszeit für Task 1, K-Bus, Bearbeitung Feldbus und die Auslastung des SSB.

	Actual Value	Maximum Value	
PLC-Task 1 (µs):	72	144	
PLC-Task 2 (µs):			
PLC-Task 3 (µs):			
PLC-Task 4 (µs):			
K-Bus (µs):	246	303	
Fieldbus (µs):	21	31	
SSB (µs):			
SSB-Overhead (%):			Read CurrentConfig.xml...
Display 1:	TWINCAT-CONFIG		Reset Maximum Values
Display 2:	BC9020PROJEKT		Factory Settings

Abb. 48: 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

**4.2.9 PLC**

**4.2.9.1 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](#) [▶ 93])

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.

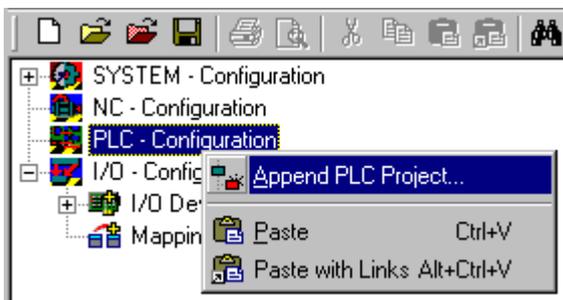


Abb. 49: Auswahl des PLC-Projekts

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

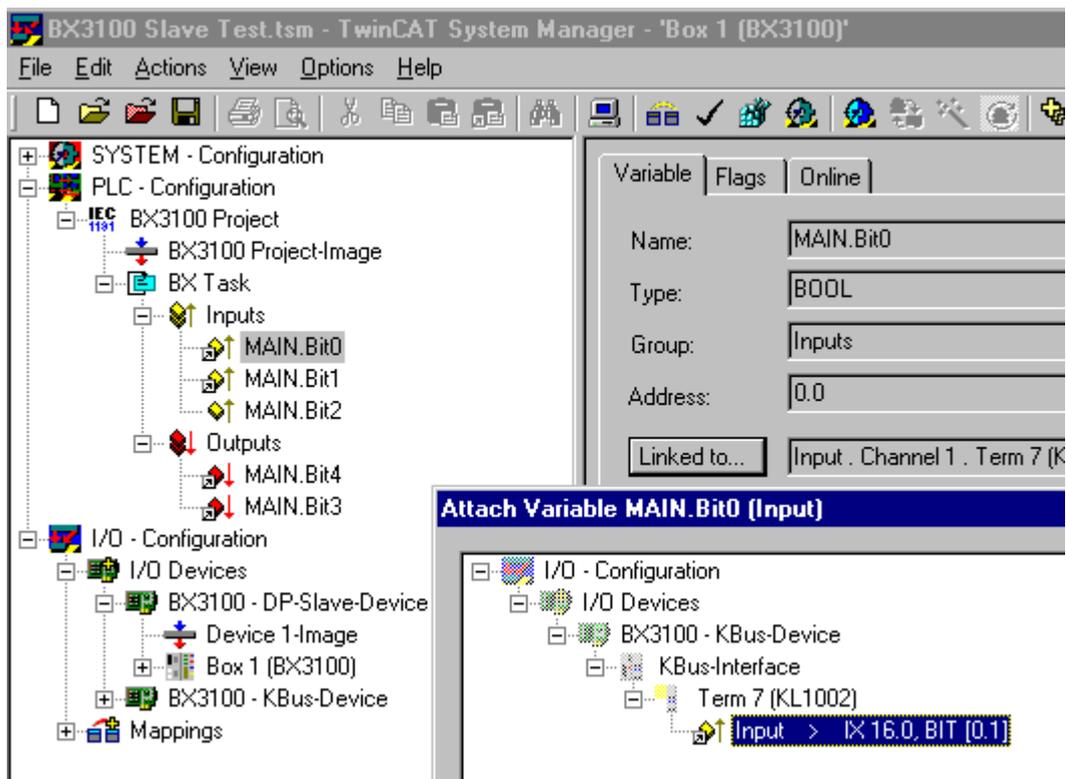


Abb. 50: Verbinden vom PLC-Variable und Hardware

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).

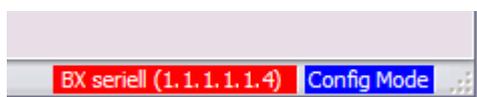


Abb. 51: Anzeige des Ziel-Systems

### 4.2.9.2 Messen der PLC-Zykluszeit

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

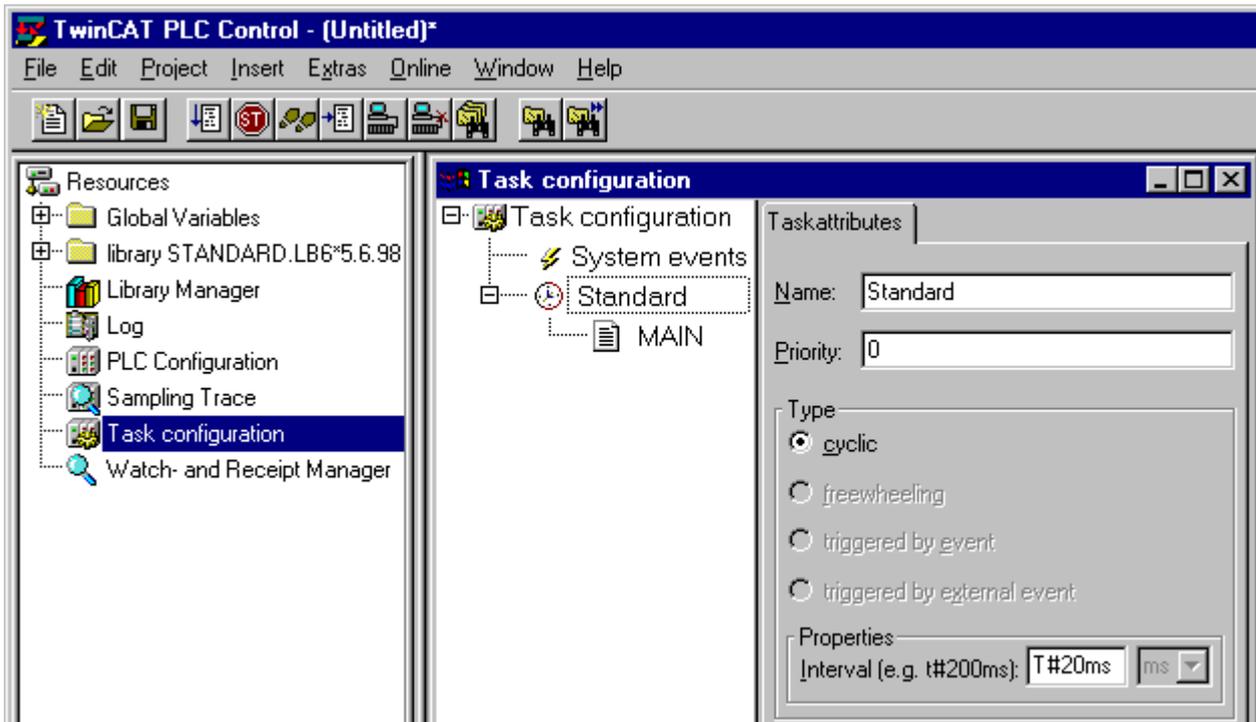


Abb. 52: Einstellen der Task-Zeit

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](#) [► 49]. Die gesamte Zykluszeit wird mit der TcBase Bibliothek angezeigt (siehe TcBase.lbx oder TcBaseBCxx50.lbx).

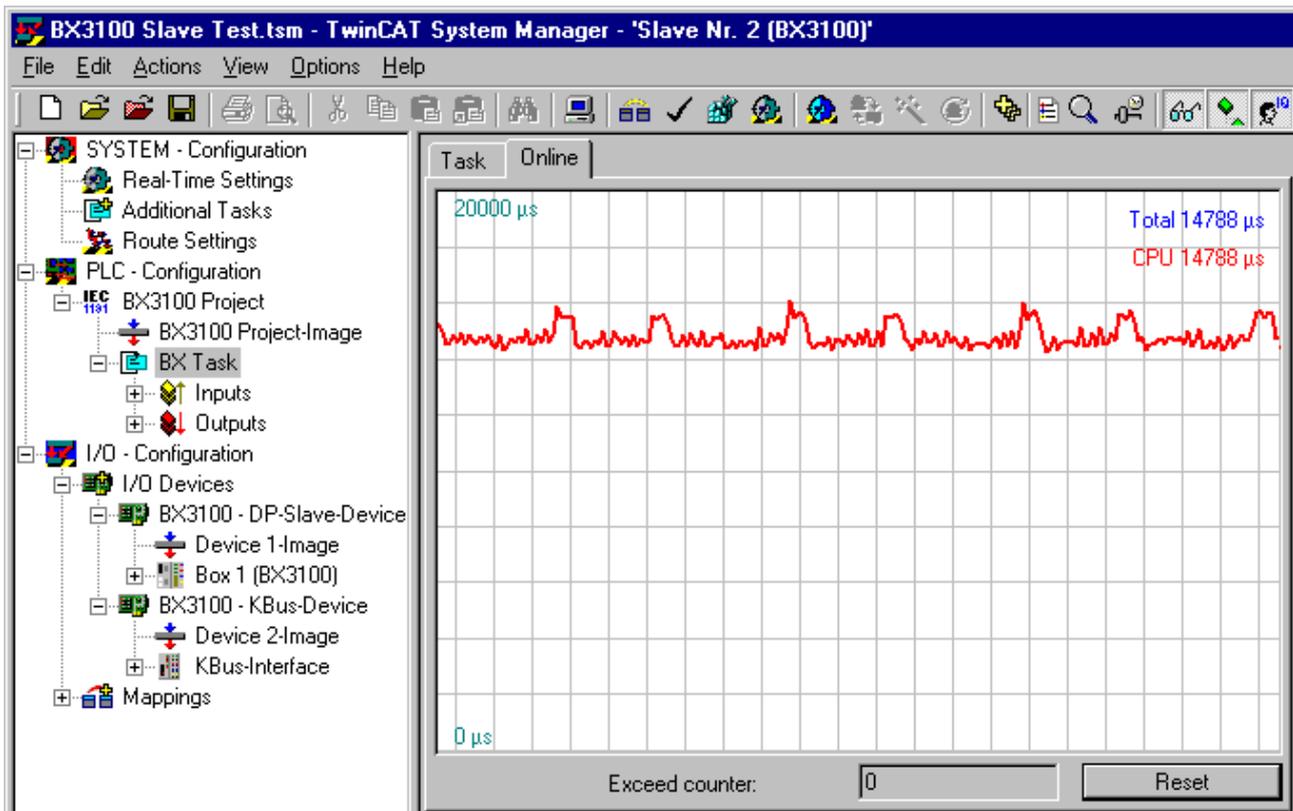


Abb. 53: Anzeige der PLC-Zykluszeit

## 4.2.10 SSB

### 4.2.10.1 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 TxPODs
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.

**4.2.10.2 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.

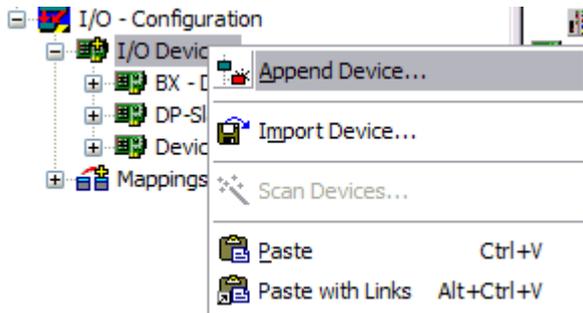


Abb. 54: Anfügen eines weiteren Gerätes

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

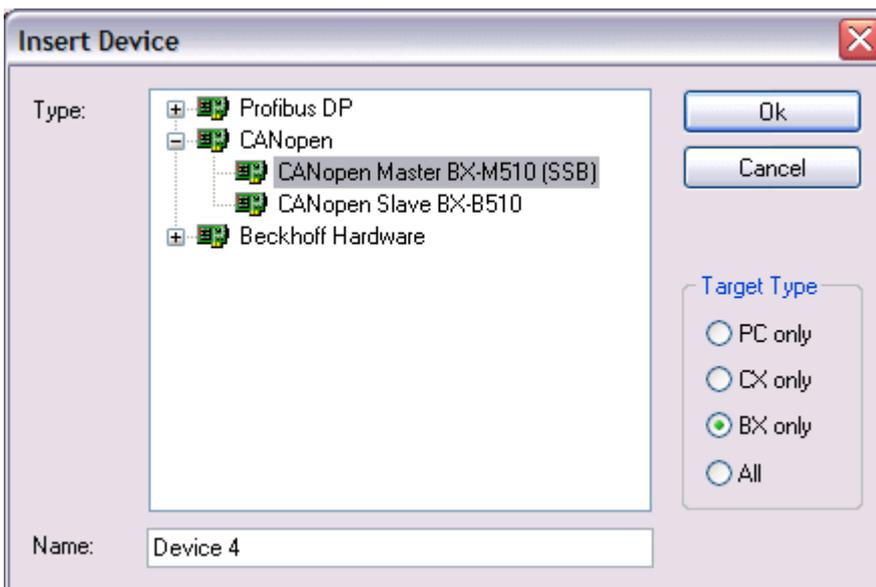


Abb. 55: Auswahl des CANopen Masters SSB

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

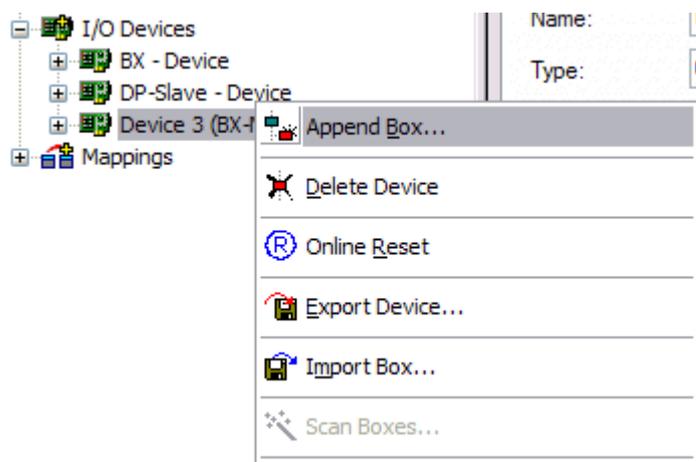


Abb. 56: Anfügen eines CANopen Geräts

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

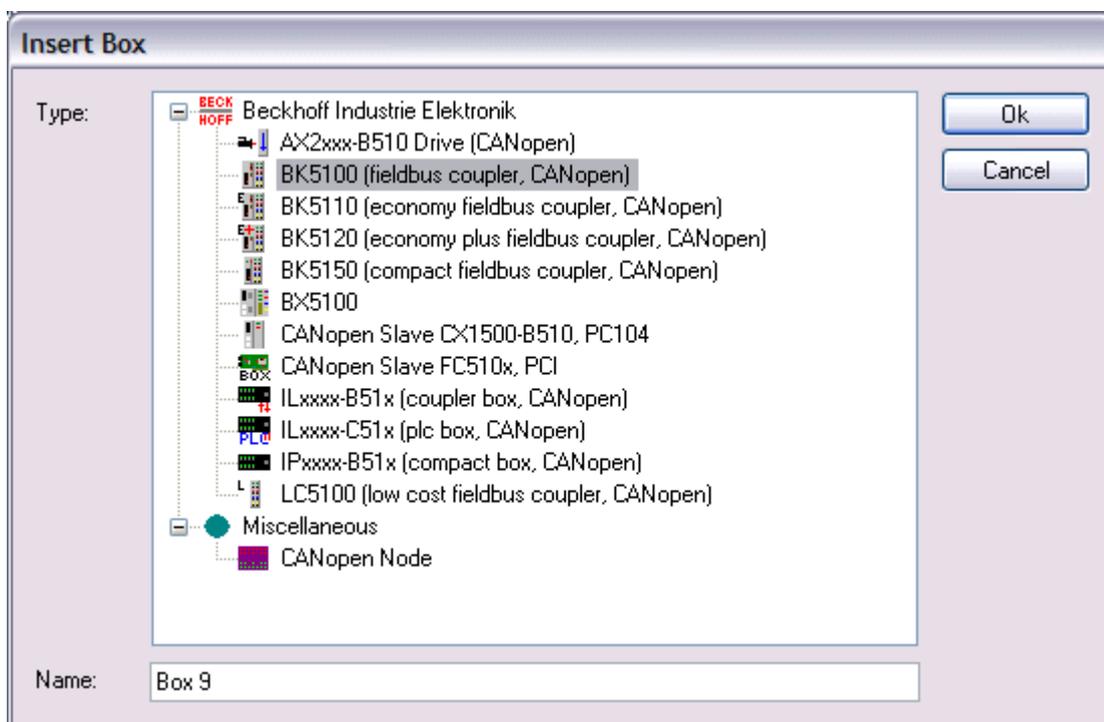


Abb. 57: Auswahl eines CANopen-Knotens

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

### 4.2.10.3 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.

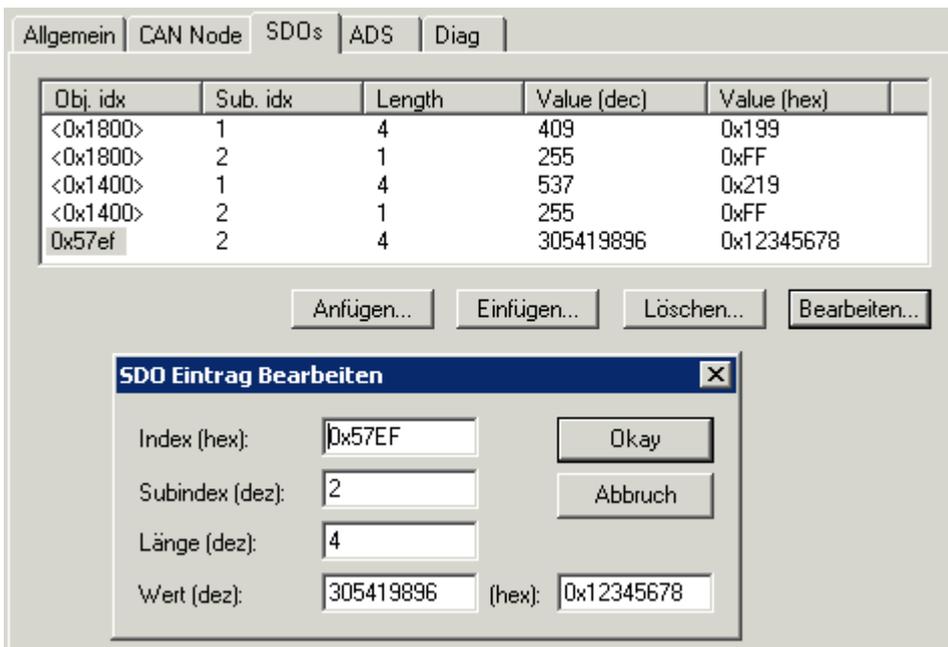


Abb. 58: Anfügen/Bearbeiten von Objektverzeichniseinträgen

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.

#### 4.2.10.4 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 \(https://infosys.beckhoff.com/content/1031/bx5100/Resources/zip/3207257611.zip\)](https://infosys.beckhoff.com/content/1031/bx5100/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/bx5100/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)

**4.2.10.5 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.

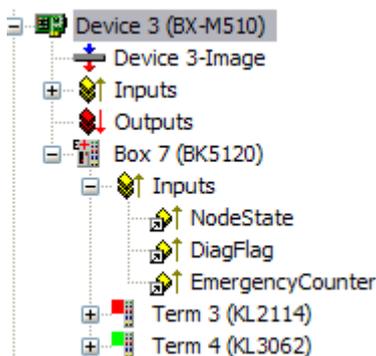


Abb. 59: NodeState, DiagFlag und EmergencyCounter

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/bx5100/Resources/zip/3207261963.zip>)

 Download Beispiel System-Manager File BX (<https://infosys.beckhoff.com/content/1031/bx5100/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/bx5100/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

**4.2.10.6 Beispiele**

**4.2.10.6.1 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/bx5100/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/bx5100/Resources/zip/3207270667.zip>)



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

#### 4.2.10.6.2 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. 60: 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/bx5100/Resources/zip/3207272843.zip>)



- Programm File BX\_ONE (<https://infosys.beckhoff.com/content/1031/bx5100/Resources/zip/3207275019.zip>)



- TwinCAT System Manager File BX\_TWO (<https://infosys.beckhoff.com/content/1031/bx5100/Resources/zip/3207277195.zip>)



- Programm File BX\_ONE (<https://infosys.beckhoff.com/content/1031/bx5100/Resources/zip/3207279371.zip>)



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

#### 4.2.10.6.3 AX2000 am SSB



Abb. 61: 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/bx5100/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/bx5100/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.

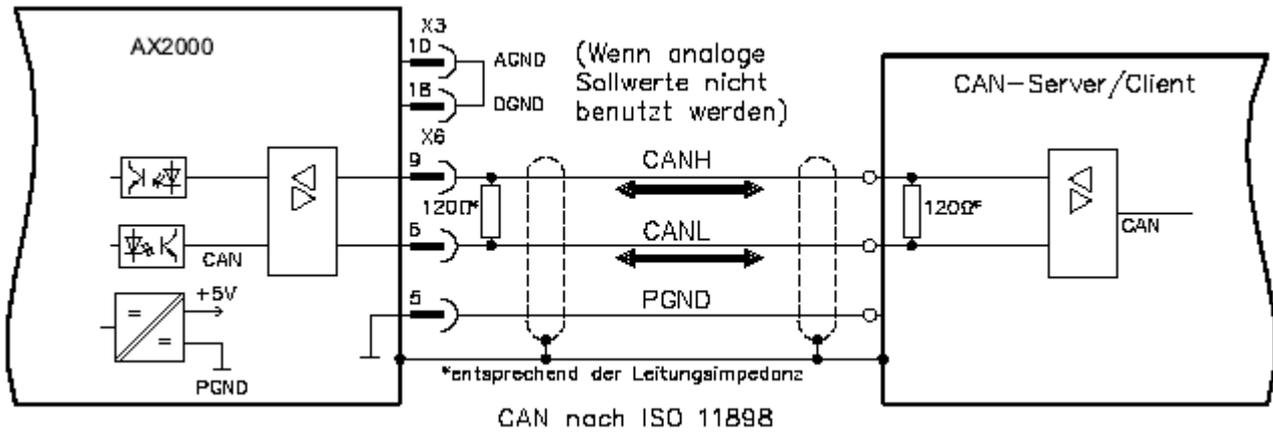


Abb. 62: CANopen Interface (X6)

#### 4.2.10.6.4 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. 63: 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/bx5100/Resources/zip/3207285899.zip>)



- Beispielkonfiguration für den BX: (<https://infosys.beckhoff.com/content/1031/bx5100/Resources/zip/3207288075.zip>)



- Beispiel für Cimrex 41: (<https://infosys.beckhoff.com/content/1031/bx5100/Resources/zip/3207290251.zip>)



- Baudrate 500 kBaud
- CAN Slave Adresse 10

#### 4.2.10.6.5 IclA-Drive am SSB



# *IclA*<sup>®</sup>

Abb. 64: 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/bx5100/Resources/zip/3207292427.zip>)



- TwinCAT-PLC-File (<https://infosys.beckhoff.com/content/1031/bx5100/Resources/zip/3207294603.zip>)



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

- TwinCAT-System-Manager-File (<https://infosys.beckhoff.com/content/1031/bx5100/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/bx5100/Resources/zip/3207298955.zip>)



**IclA D065 - Beschreibung**

Folgende Abschnitte sind Auszüge aus der Dokumentation des IclA-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**

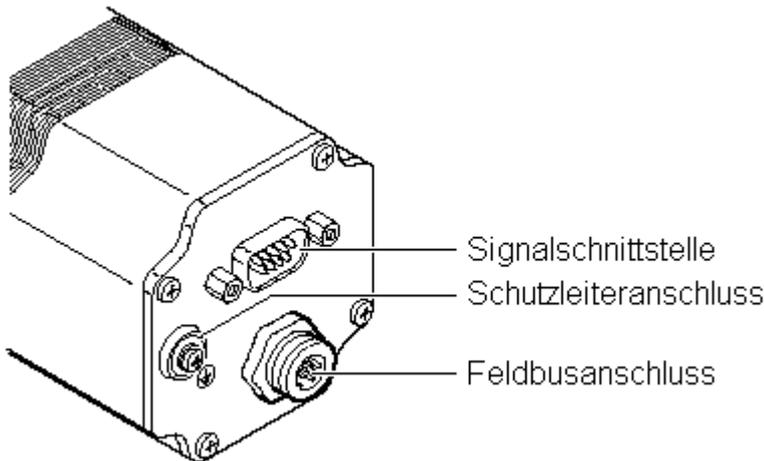


Abb. 65: Anschlüsse des IclA-Drives

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

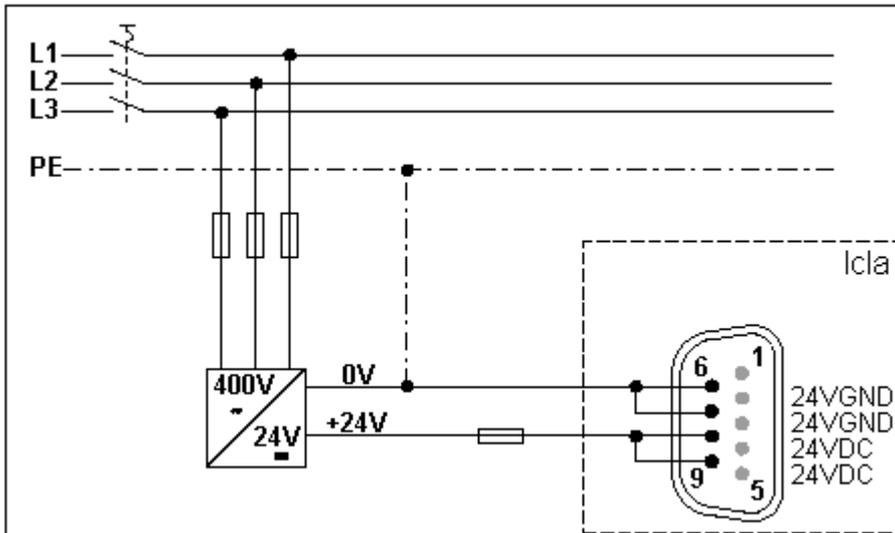


Abb. 66: Signalschnittstelle

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

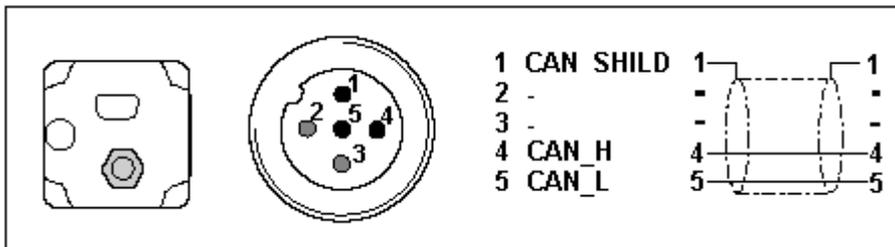


Abb. 67: Feldbusanschluss

**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.

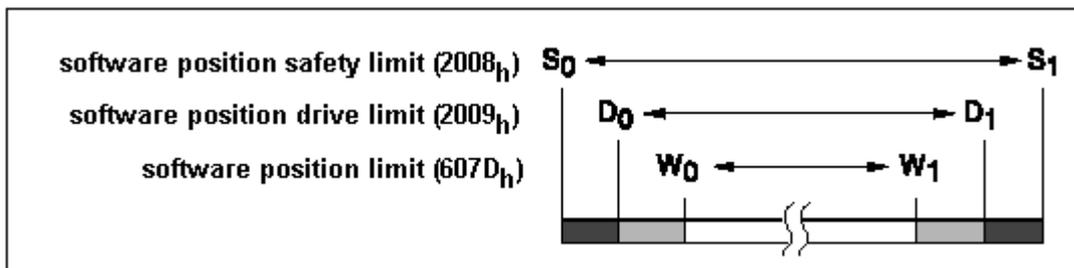


Abb. 68: Referenzierungsbereiche

- 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. 69: Listing der Referenzierungswerte

#### 4.2.10.6.6 Lenze Frequenzumrichter am SSB

# Lenze



Abb. 70: 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.

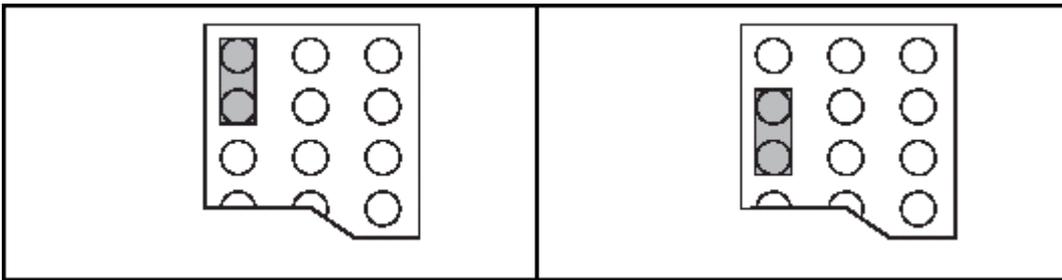


Abb. 71: Externe Spannungsversorgung - Interne Spannungsversorgung (Auslieferungszustand)

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

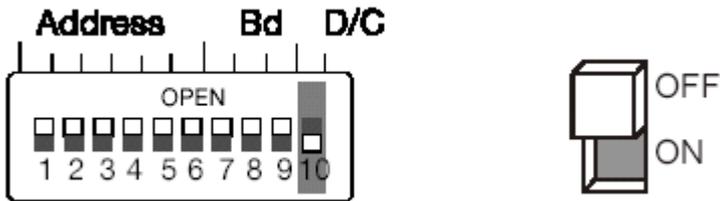


Abb. 72: DIP-Schalter

**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:

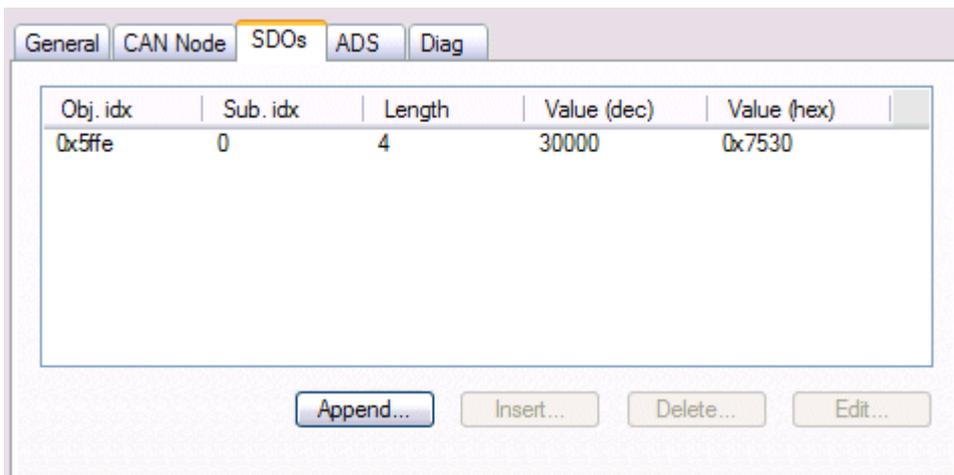
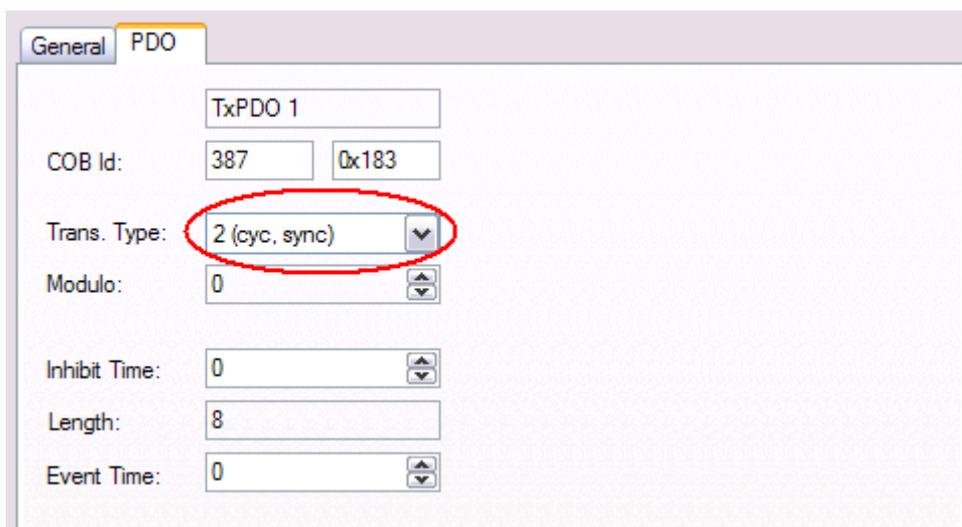


Abb. 73: Freigabe des Kommunikationsmoduls

### 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/bx5100/Resources/zip/3207301131.zip>)



- TwinCAT-PLC-File: (<https://infosys.beckhoff.com/content/1031/bx5100/Resources/zip/3207303307.zip>)



## 4.2.11 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*.

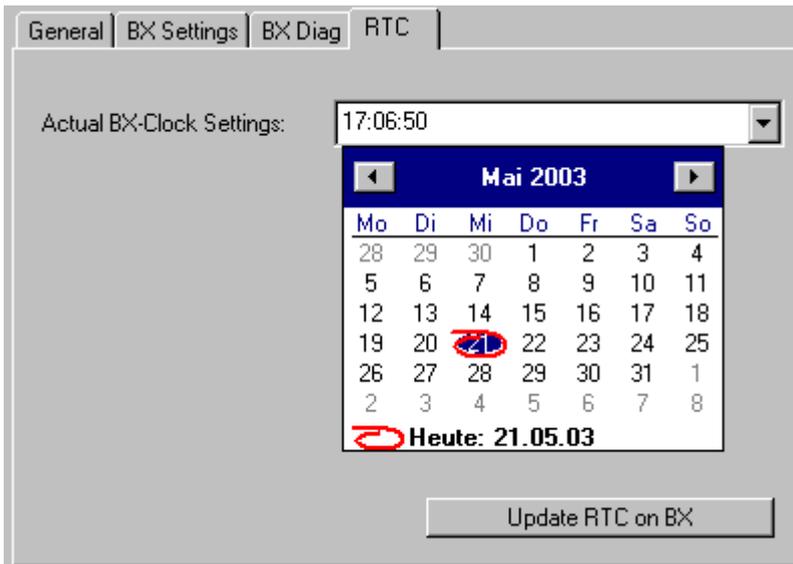


Abb. 74: Einstellen der Echtzeit-Uhr (RTC)



### Lebensdauer des Akkus

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

### Lesen der RTC auf dem BX-Controller (siehe Beispiel [▶ 112] 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. [[▶ 77](#)]

**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)

**4.2.12 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="#">▶ 104</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](#) [▶ 104]
- [Beispiel](#) [▶ 109]
- [Bibliothek](#) [▶ 104]

## 4.2.13 Menü

### 4.2.13.1 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. 75: Navigationsschalter der BX-Controller

**Schalterbelegung**

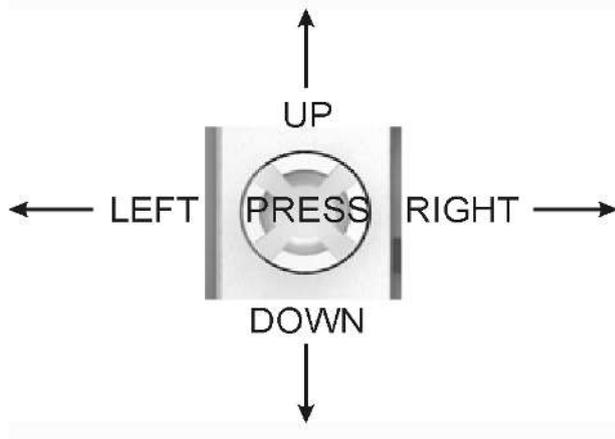


Abb. 76: Schalterbelegung

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 rebootet
	Reboot	Aktivieren?	wenn die Taste gedrückt wird, rebootet 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

**Busspezifische 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

**4.2.13.2 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/bx5100/Resources/zip/3207305483.zip>)

## 4.2.14 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**

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

---

### ● **Erforderliche KS2000-Version**

**i** 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 [[▶ 133](#)]
- 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 POUs

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. 77: Maximale Anzahl der POUs überschritten

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

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

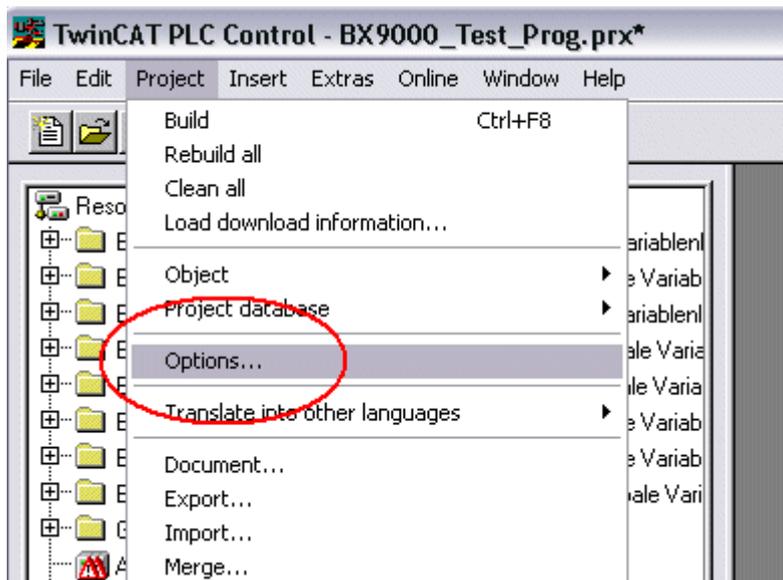


Abb. 78: Menüpfad Projekte / Optionen / Controller Settings

...die Controller Settings.

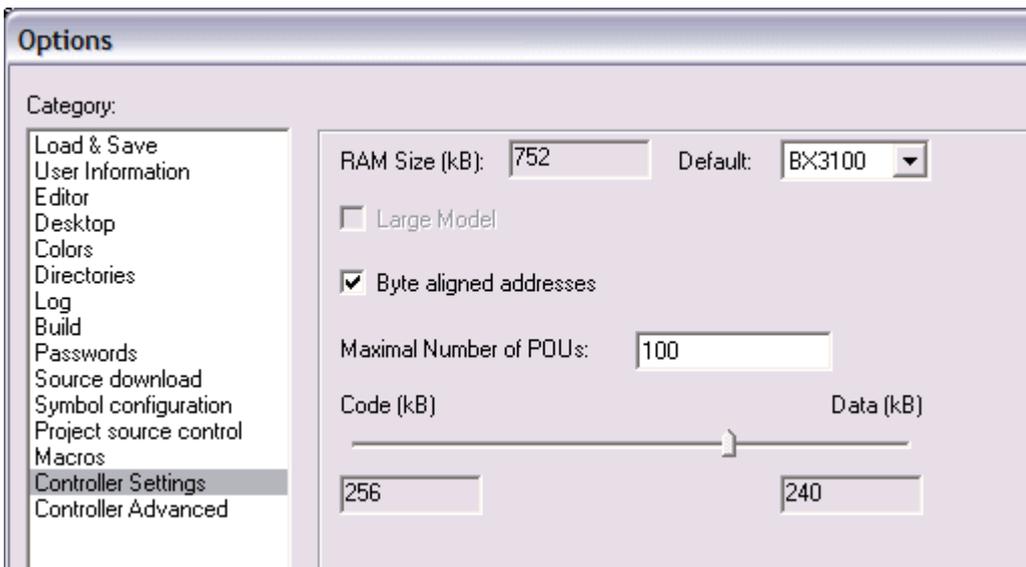


Abb. 79: Controller Settings

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. 80: 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.

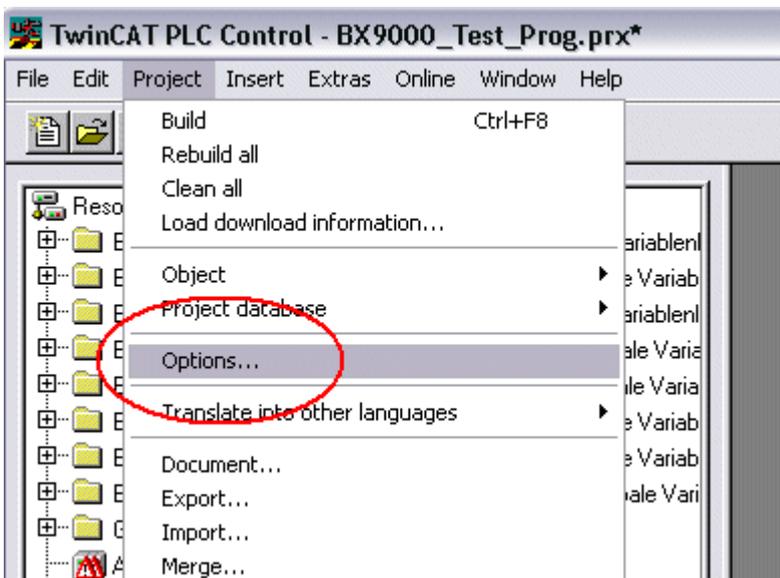


Abb. 81: Menüpfad Projekte / Optionen / Build

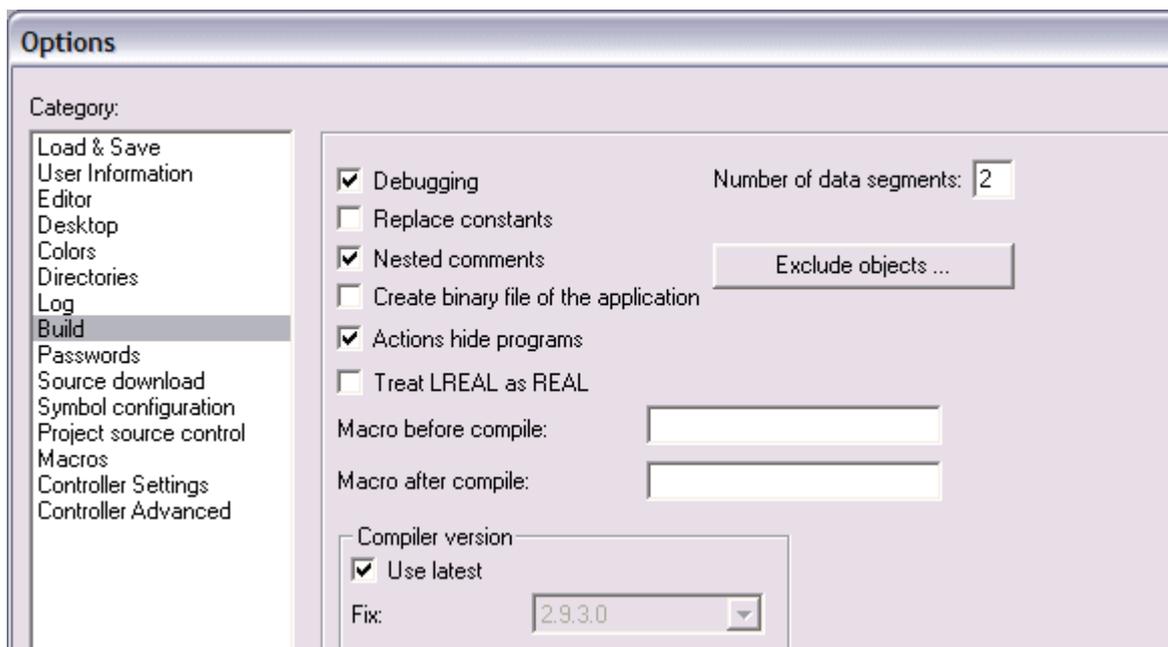


Abb. 82: Build

## 5.4 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](#) [► 98]).

### 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/bx5100/Resources/zip/3207307659.zip>) um ein Beispielprogramm aus dieser Dokumentation herunterzuladen.

## 5.5 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);
```

## 5.6 Lokierte Merker

Es stehen 4 kByte an lokierte Merkern 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 lokierte 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
```

## 5.7 Lokales Prozessabbild im Auslieferungszustand

Das Prozessabbild der Busklemmen-Controller besteht aus einem 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 hat man auf dem Busklemmen-Controller 256 kByte Speicher zur Verfügung. Für die lokierten Daten stehen 2048 Byte Ein- und 2048 Byte Ausgangsdaten bereit. In der Default Config, d.h. im Auslieferungszustand des Busklemmen-Controllers werden alle angeschlossenen Busklemmen einer festen Adresse zugewiesen. Die Daten für die CANopen Kommunikation liegen ab dem Adress-Offset 1000dez. Die Anzahl der CANopen Daten beträgt in der Default Configuration 4 x kByte (4 PDOs).

INPUTS	OUTPUTS
Busklemme %IB0 ...	Busklemmen %QB0 ...
CANopen DATEN (SPS-Variablen) %IB1000 ...	CANopen DATEN (SPS-Variablen) %QB1000 ...
... %IB2047 Maximal	... %QB2047 Maximal

### CANopen Daten

PDO Nummer	Lesen/Schreiben	BX Prozessabbild
PDO 1	Rx/Tx	%IB1000...%IB1007/QB1000...%QB1007
PDO 2	Rx/Tx	%IB1008...%IB1015/QB1008...%QB1015
PDO 3	Rx/Tx	%IB1016...%IB1023/QB1016...%QB1023
PDO 4	Rx/Tx	%IB1024...%IB1031/QB1024...%QB1031

Weitere PDO Daten 5-32 müssen vom Master freigeschaltet werden (siehe CAN-Konfiguration [▶ 45]).

### 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 dann die digitalen Busklemmen, die zu einem Byte aufgefüllt werden. Das Default-Mapping der komplexen Busklemmen ist:

- komplette Auswertung
- Intel-Format
- Wordalignment

### Beispiel

Busklemmen Controller: 1 x BX5100  
 Position 1: 1 x KL1012  
 Position 2: 1 x KL1104  
 Position 3: 1 x KL2012  
 Position 4: 1 x KL2034  
 Position 5: 1 x KL1502  
 Position 6: 1 x KL3002  
 Position 7: 1 x KL4002  
 Position 8: 1 x KL6001  
 Position 9: 1 x KL9010

**Prozessabbild**

Busklemme	Position	Eingangsabbild	Ausgangsabbild
KL1501	5	%IB0...%IB5	%QB0...%QB5
KL3002	6	%IB6...%IB13	%QB6...%QB13
KL4002	7	%IB14...%IB21	%QB14...%QB21
KL6001	8	%IB22...%IB29	%QB22...%QB29
KL1012	1	%IX30.0..30.1	-
KL1104	2	%IX30.1..30.5	-
KL2012	3	-	%QX30.0..30.1
KL2034	4	-	%QX30.2..30.5
KL9010	9	-	-

**i Adresse der Busklemmen in der lokalen SPS**

Falls Sie nicht wissen, auf welcher Adresse sich die Busklemmen befinden, die Sie der lokalen SPS (BCxx00) zugewiesen haben:

Konfigurieren Sie im System Manager Ihre Hardware-Konfiguration. Nachdem Sie alle Busklemmen und SPS-Variablen eingetragen haben klicken Sie im Hardware-Baum mit der rechten Maustaste auf den BCxx00 und wählen Sie den Menü Punkt Export Variablen Info.... Es wird eine Datei gespeichert, die Sie im System Manager unter Projekt Importieren einfügen können. Nun haben Sie unter den globalen Variablen den Eintrag TwinCAT Import, indem Sie alle Variablen finden, die Sie der lokalen SPS (BCxx00) zugeordnet haben.

## 5.8 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

## 5.9 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.

---

### ● Übernahme des aktuellen Projektes als Boot-Projekt

**I** 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

## 5.10 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.

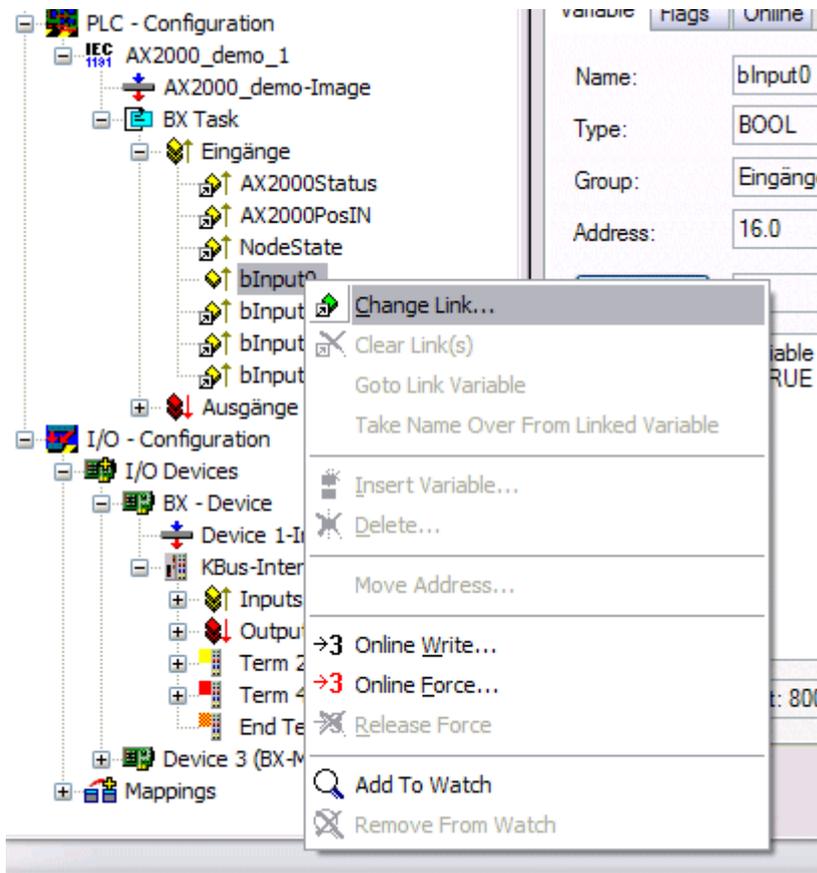


Abb. 83: Ändern der Verknüpfung von Variablen

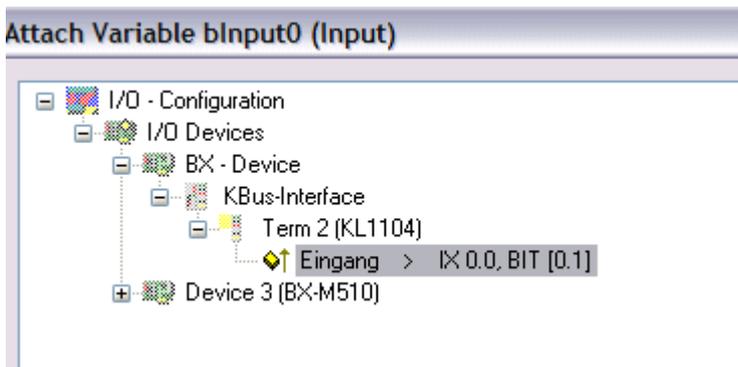


Abb. 84: Verknüpfen einer Variable mit einem Eingang

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](#) [▶ 35]). Ein Hochladen einer bestehenden TwinCAT-Konfiguration ist auch möglich.

## 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
- Boolesche Variablen sollten nie einzeln in eine Struktur gelegt werden, da sie so immer 1 Byte belegen würden. Boolesche 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

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.

**i Größere Feldbusdatenblöcke**

Sie können auch größere Feldbusdatenblöcke übertragen, um eine Reserve für Ihre Struktur zu haben. Nachteil: Diese Reserven werden dann mit jedem Feldbustelegamm 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.

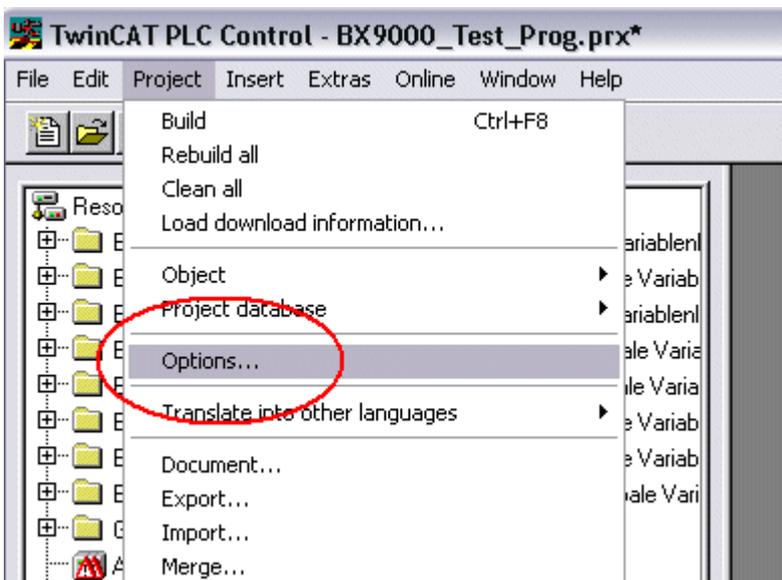


Abb. 85: Öffnen des Optionsmenüs

Wählen Sie nun den Source Download an.

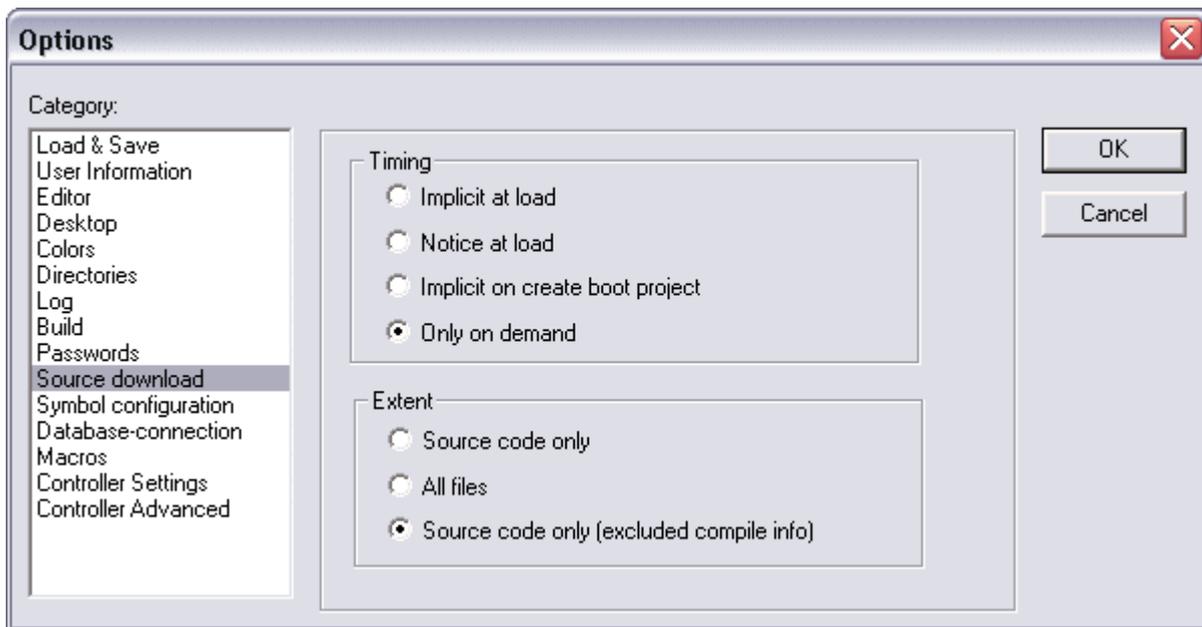


Abb. 86: Auswahl des Source Downloads

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. 87: Download des Programm Codes

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

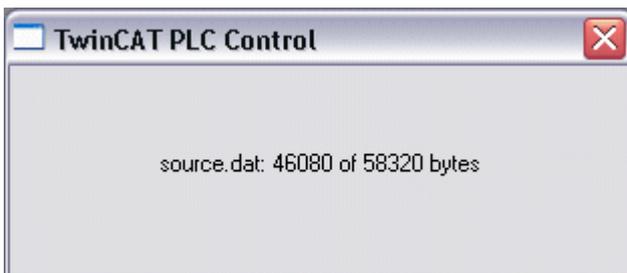


Abb. 88: Vorschrift des Downloads

### 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.

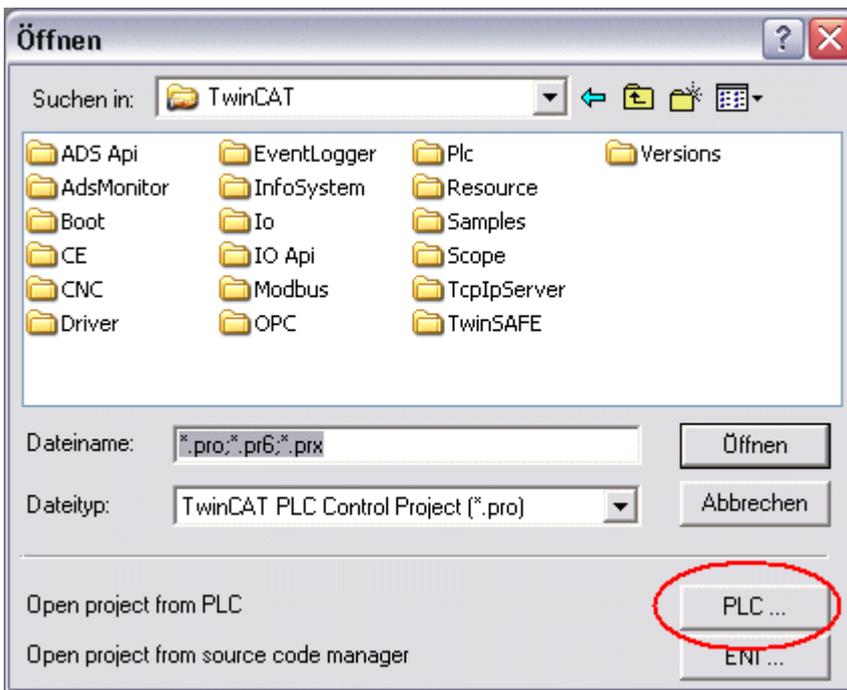


Abb. 89: Upload eines Programms

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.

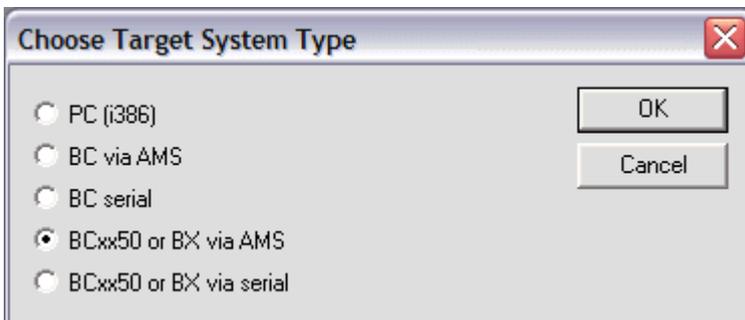


Abb. 90: Auswahl des Datenübertragungswegs

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

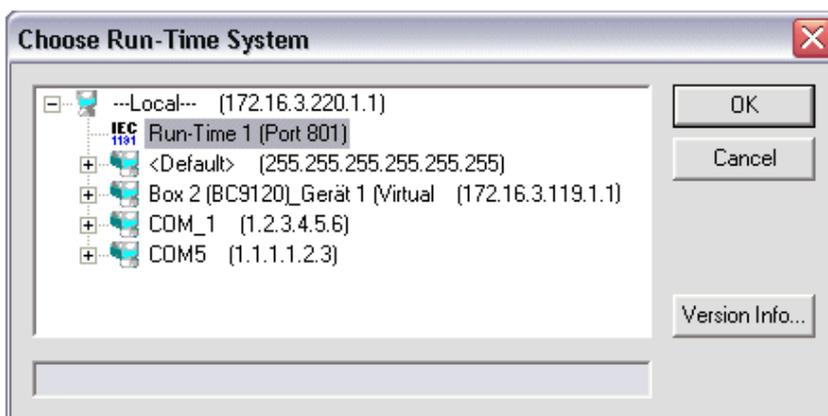


Abb. 91: Auswahl des Gerätes

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/bx5100/Resources/zip/3207309835.zip>)  

- TcSystemBX (<https://infosys.beckhoff.com/content/1031/bx5100/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/bx5100/Resources/zip/3207314187.zip>)  

- ChrAscBX.lbx (<https://infosys.beckhoff.com/content/1031/bx5100/Resources/zip/3207316363.zip>)  


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

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_DisWrite ▶ 111]	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> [▶ 110]	08.08.03	0.59	0.14	1.00	0.02	1.00
<a href="#">F_StartDebugTimer</a> [▶ 110]	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> [▶ 110]	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> [▶ 116]	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` [► 101]. 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` [► 101] 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	Zielformat	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="#">▶ 102</a> ] <sub>dez</sub>	Lesen und schreiben von Registern und Tabellen aus den Koppler und den komplexen Busklemmen
150 [ <a href="#">▶ 75</a> ] <sub>dez</sub>	Lesen und schreiben der RTC (Echt-Zeit-Uhr)
153 [ <a href="#">▶ 58</a> ] <sub>dez</sub>	SSB - auslesen der Emergency-Nachricht
800 [ <a href="#">▶ 102</a> ] <sub>dez</sub>	Lokales Prozessabbild der PLC, siehe auch Port 801
801 [ <a href="#">▶ 102</a> ] <sub>dez</sub>	Lokales Prozessabbild der PLC, siehe auch Port 800
0x1000 + Node ID [ <a href="#">▶ 57</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
TMOU	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` [► 101] 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> [► 106]	Eine serielle Verbindung zu einem Partner öffnen.
<code>FB_ComPortClose</code> [► 106]	Eine serielle Verbindung zu einem Partner schließen.

#### Funktionen

Name	Beschreibung
<code>F_ComPortRead</code> [► 105]	Daten aus dem COM-Puffer lesen
<code>F_ComPortWrite</code> [► 105]	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](#) [► 115].

- 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**

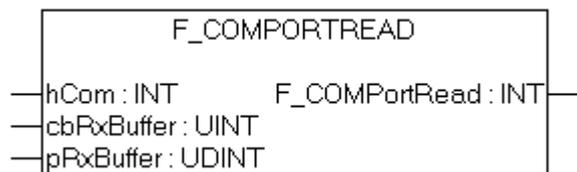


Abb. 92: Funktionsbaustein F\_COMPORTRREAD

**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**

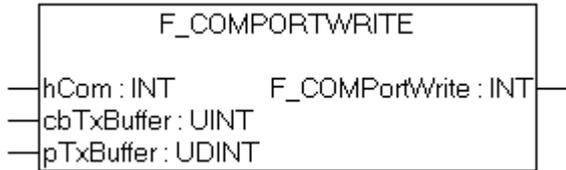


Abb. 93: Funktionsbaustein F\_COMPORWRITE

**FUNCTION F\_COMPORWRITE**

**VAR\_INPUT**

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

**Legende**

hCom: wir mit dem iHandle des FB\_COMPOROPEN 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**

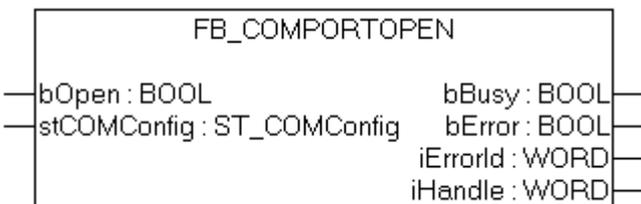


Abb. 94: Funktionsbaustein FB\_COMPOROPEN

**FUNCTION\_BLOCK FB\_COMPOROPEN**

**VAR\_INPUT**

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

**Legende**

bOpen: Positive Flanke startet den Baustein

stComConfig [▶ 108]: 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**

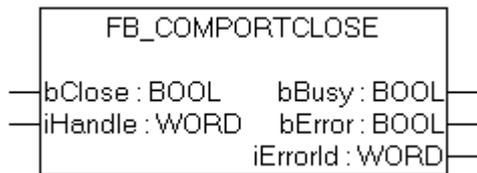


Abb. 95: Funktionsbaustein FB\_COMPORTCLOSE

**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/bx5100/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

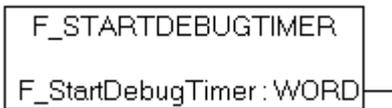


Abb. 96: Funktionsbaustein F\_STARTDEBUGTIMER

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

#### Funktion Read Debug Timer

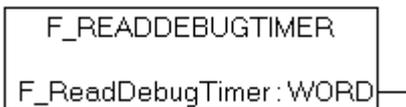


Abb. 97: Funktionsbaustein F\_READDEBUGTIMER

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.

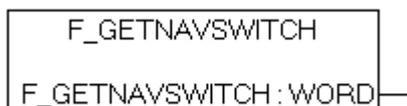


Abb. 98: Funktionsbaustein F\_GETNAVSWITCH

#### 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/bx5100/Resources/zip/3207322891.zip>

**5.13.2.9 Display**

**5.13.2.9.1 FB DISPWRITE**

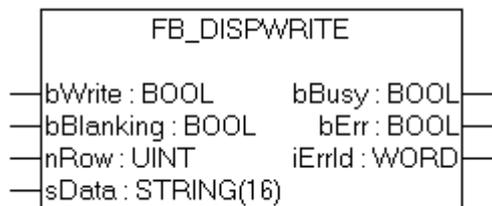


Abb. 99: Funktionsbaustein FB\_DISPWRITE

**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/bx5100/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
```

### 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/bx5100/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.

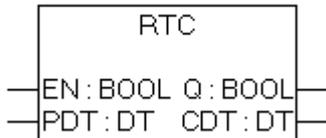


Abb. 100: Funktionsbaustein RTC

**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/bx5100/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

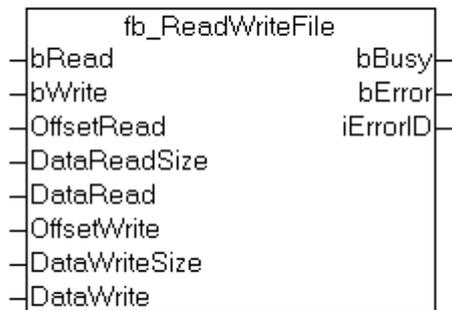


Abb. 101: Funktionsbaustein fb\_ReadWriteFile

#### 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/bx5100/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> <a href="#">[► 116]</a>	BK8x00 COM Port Funktionsbaustein, Kommunikation mit Buskoppler BK8x00 oder BC8x00
<a href="#">fb_BX_BK8x00_Slave</a> <a href="#">[► 116]</a>	BK8x00 COM Port Funktionsbaustein, Kommunikation mit PC. Es wird ein BK8x00 simuliert.

Name	Beschreibung
FB_BX_COM_5 [▶ 120]	Baustein zum Emulieren einer KL60x1 (wenn COMlib.lib6 oder ModbusRTU.lib6 verwendet wird).
FB_BX_COM_64 [▶ 120]	Baustein zum Emulieren einer PC-Schnittstelle (wenn COMlib.lib oder ModbusRTU.lib verwendet wird).
FB_BX_COM_64ex [▶ 121]	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.

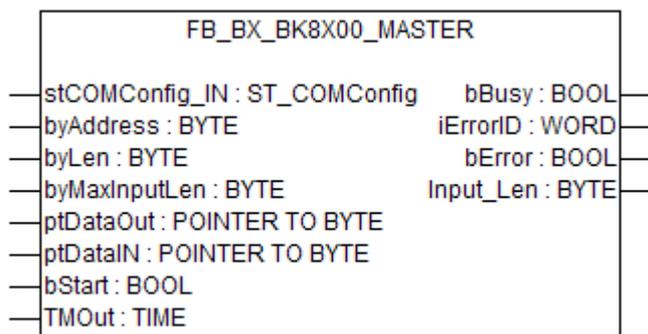


Abb. 102: Funktionsbaustein FB\_BX\_BK8X00\_MASTER

#### 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/bx5100/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.

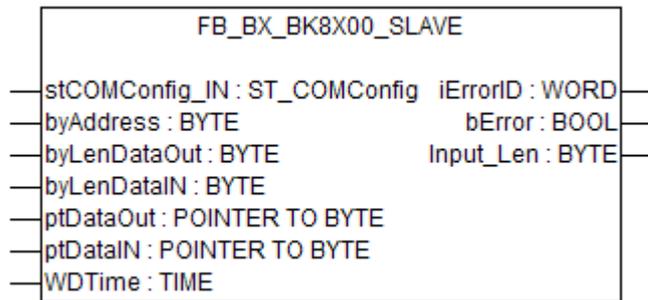


Abb. 103: Funktionsbaustein FB\_BX\_BK8X00\_SLAVE

**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/bx5100/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

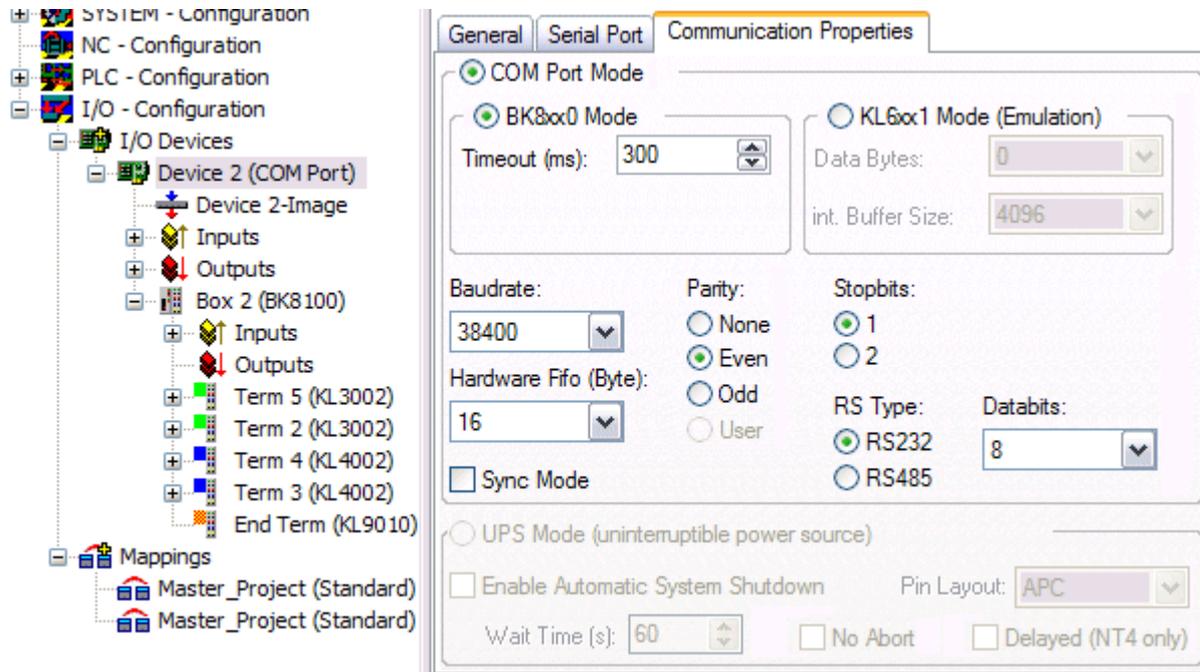


Abb. 104: Kommunikationseigenschaften

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. 105: 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 [▶ 108]: Parametrierung der COM Schnittstelle

 Download Beispielprogramm in ST für die Verknüpfung COMLib zum BX: (<https://infosys.beckhoff.com/content/1031/bx5100/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. 106: 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 [▶ 108]: Parametrierung der COM Schnittstelle

 Download Beispielprogramm in ST für die Verknüpfung ModbusRTU zum BX: (<https://infosys.beckhoff.com/content/1031/bx5100/Resources/zip/3207340299.zip>)

 Download Beispielprogramm in ST für die Verknüpfung ModbusRTU Version 2 zum BX: (<https://infosys.beckhoff.com/content/1031/bx5100/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. 107: 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 [[▶ 108](#)]:

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](#)) [[▶ 106](#)]

 Download Beispielprogramm in ST für die Verknüpfung ModbusRTU zum BX: (<https://infosys.beckhoff.com/content/1031/bx5100/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/bx5100/Resources/zip/3207346827.zip>)

 **Download Beispiel Programm in ST für die Verknüpfung ModbusRTU Master Version 2 zum BX:** (<https://infosys.beckhoff.com/content/1031/bx5100/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/bx5100/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. 108: 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 ( $\leq 5$  m) und kleinen Baudraten ( $\leq 19200$  Baud) nicht erforderlich

 Download Beispielprogramm in ST für den BX: (<https://infosys.beckhoff.com/content/1031/bx5100/Resources/zip/3207340299.zip>)

 Download Beispiel mit Cimrex-Panel: (<https://infosys.beckhoff.com/content/1031/bx5100/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/bx5100/Resources/zip/3207355531.zip>)

 Download Beispielprogramm PC: (<https://infosys.beckhoff.com/content/1031/bx5100/Resources/zip/3207357707.zip>)

 Download Beispiel System Manager File PC: (<https://infosys.beckhoff.com/content/1031/bx5100/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. 109: Mobiltelefon am COM-Port des BX-Controllers

 Download: (<https://infosys.beckhoff.com/content/1031/bx5100/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 [▶ 127]	Versionsnummer der Bibliothek
FB_TwinSAFE_KLx904_input [▶ 127]	Auswertung der TwinSAFE Daten, welche von einer KL1904 oder KL2904 zu einer KL6904 gesendet werden
FB_TwinSAFE_KLx904_output [▶ 130]	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/bx5100/Resources/zip/3207364235.zip>)

### 5.13.5.2 FUNCTION F\_GetVersionTcTwinSAFE

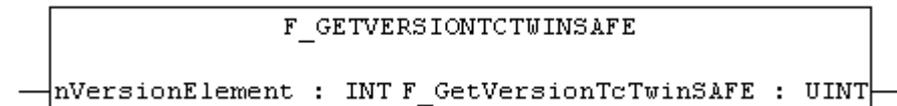


Abb. 110: Funktionsbaustein F\_GETVERSIONTCTWINSAFE

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

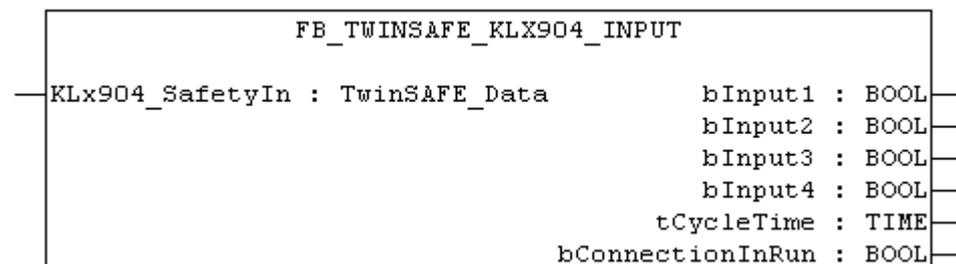


Abb. 111: Funktionsbaustein FB\_TWINSAFE\_KLX904\_INPUT

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
```

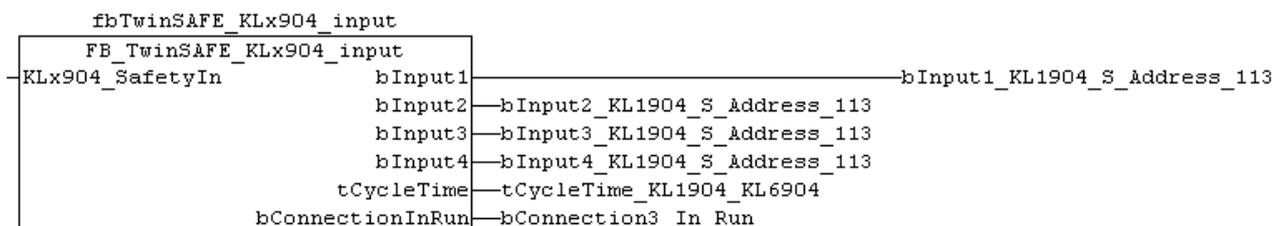


Abb. 112: Funktionsbaustein FB\_TWINSAFE\_KLX904\_input

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..."

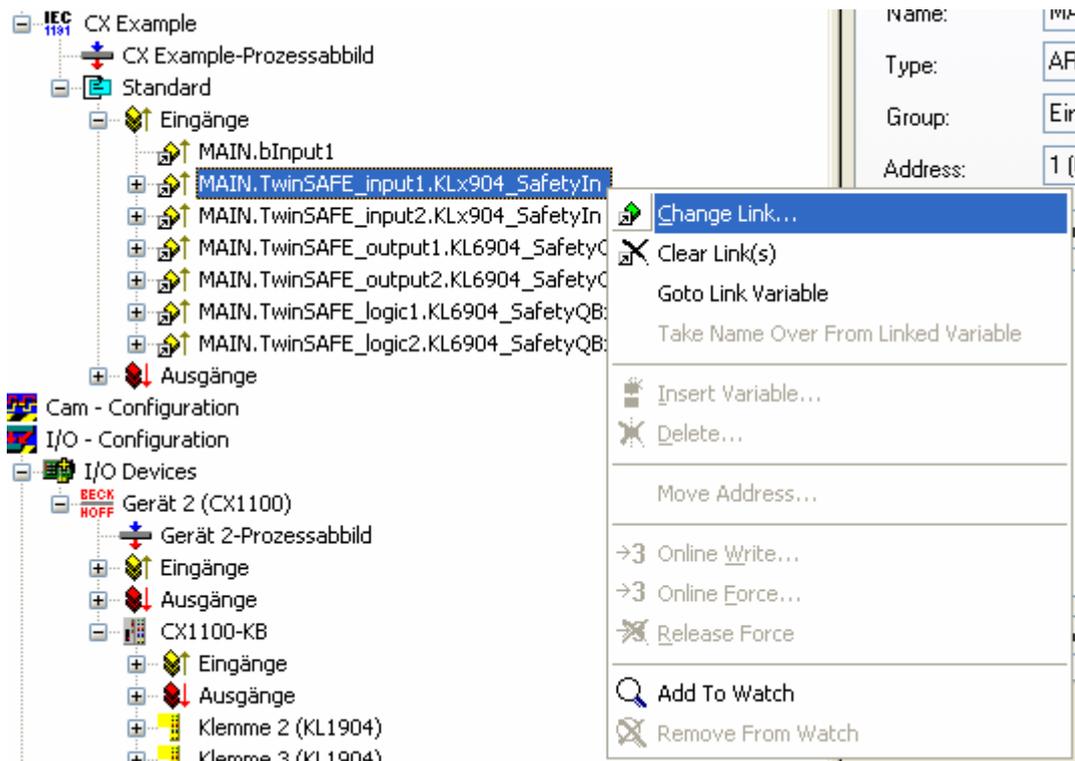


Abb. 113: Verknüpfen der Eingangsdaten

und wählen im folgenden Dialog die entsprechende SafetyIn-Variable

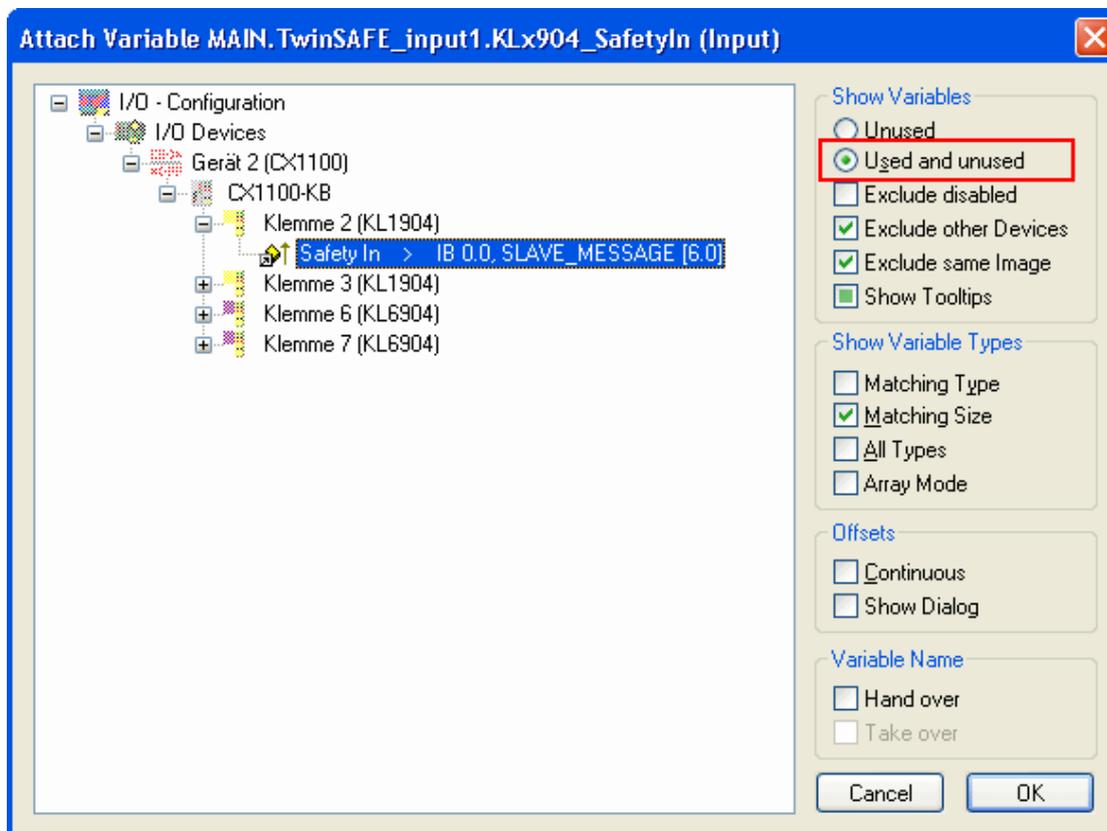


Abb. 114: Auswahl der SafetyIn-Variable

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

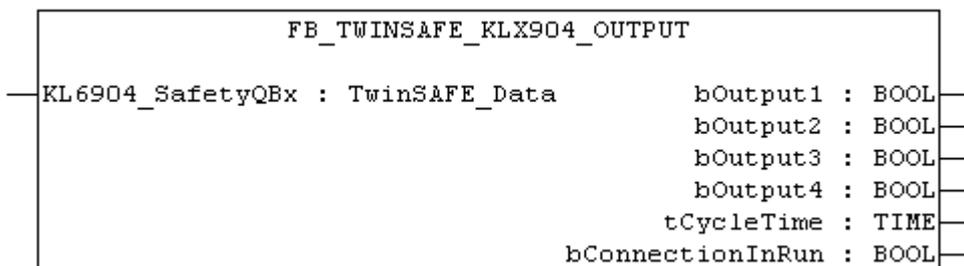


Abb. 115: Funktionsbaustein FB\_TWINSAFE\_KLX904\_OUTPUT

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
```

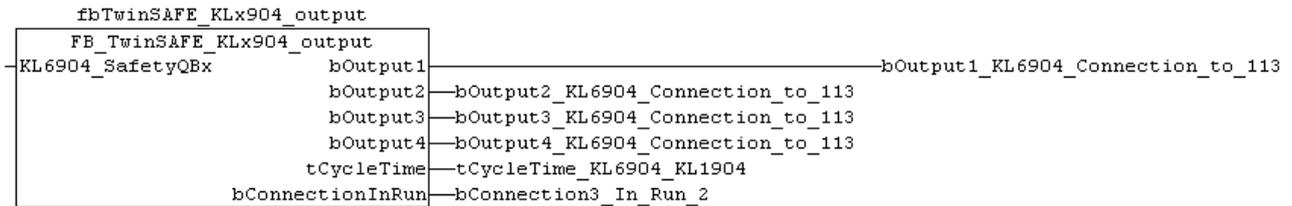


Abb. 116: Aufruf des Funktionsbausteins FB\_TWINSAFE\_KLX904\_OUTPUT

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.

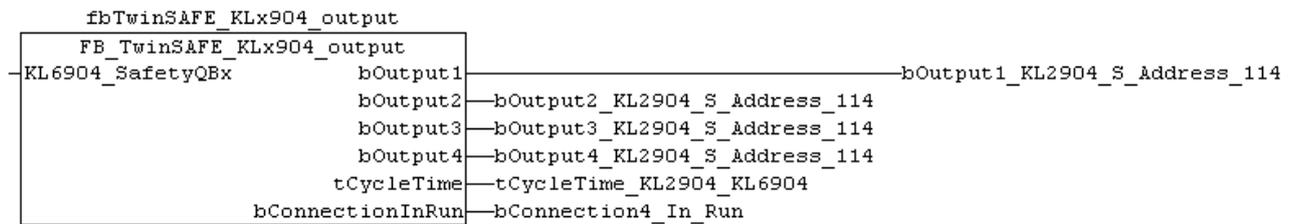


Abb. 117: Aufruf des Funktionsbausteins FB\_TWINSAFE\_KLX904\_OUTPUT

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..."

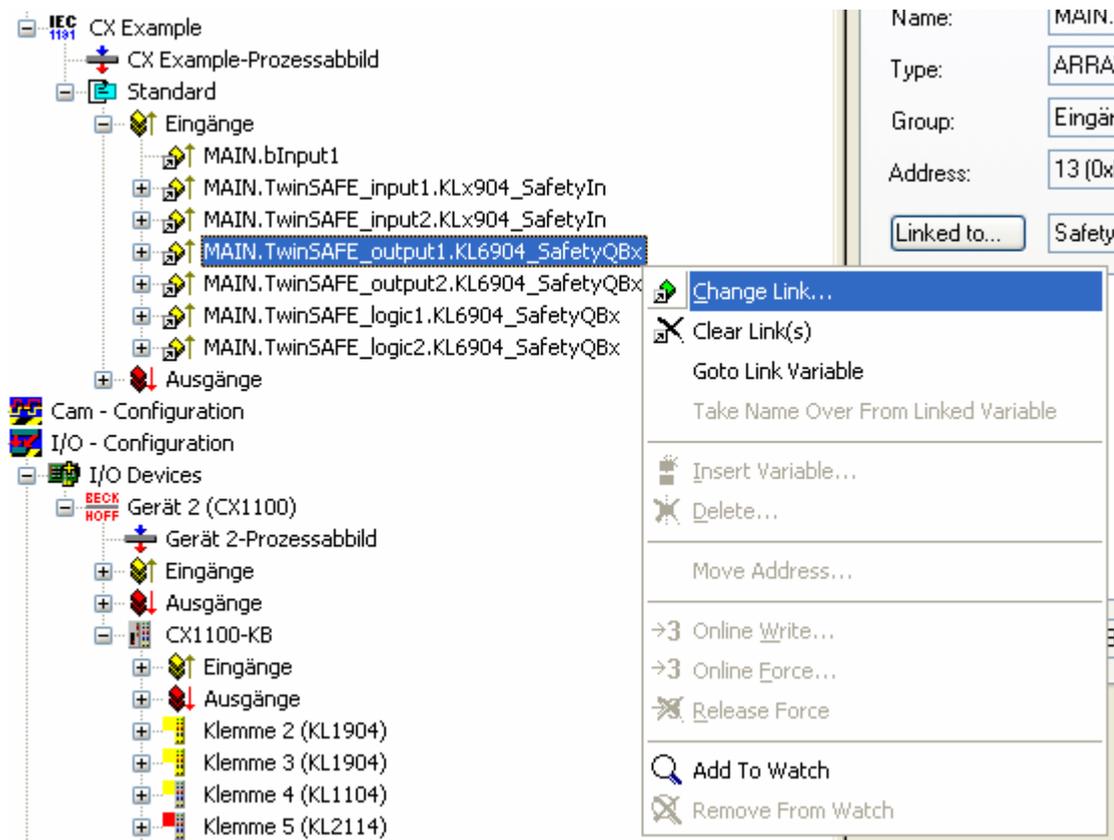


Abb. 118: Verknüpfen der Eingangsdaten

und wählen im folgenden Dialog die entsprechende SafetyQBx-Variable

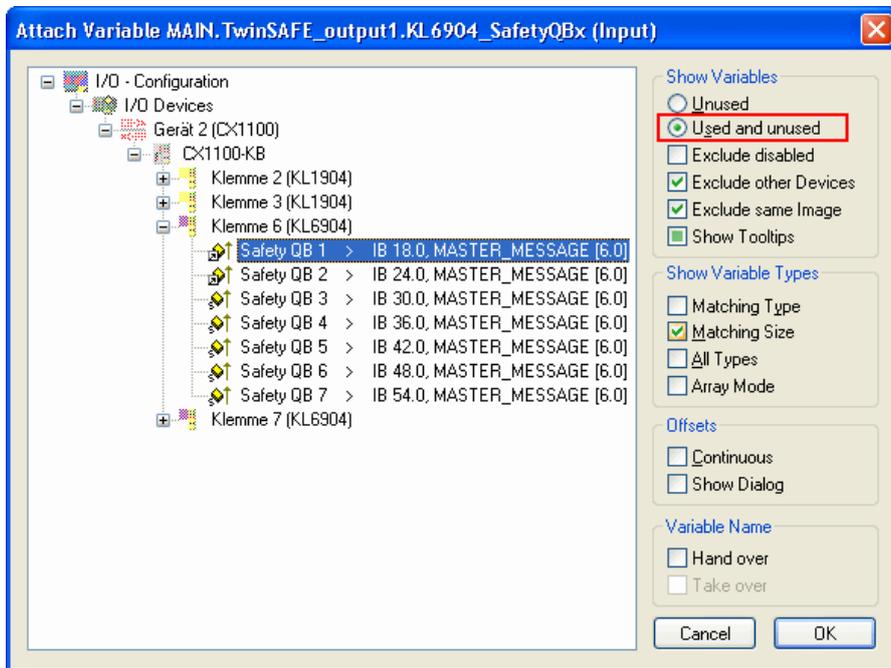


Abb. 119: Auswahl der entsprechenden SafetyQBx-Variable

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.

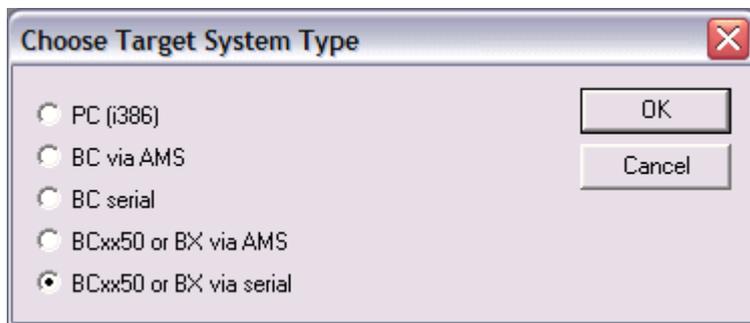


Abb. 120: Auswahl des Datenübertragungswegs - Serielle Schnittstelle

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

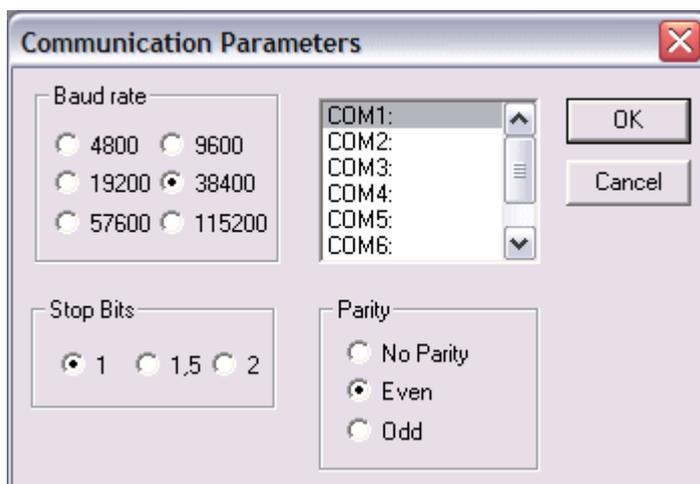


Abb. 121: Parametrierung der seriellen Schnittstelle

**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 [▶ 44]).

Wählen Sie im TwinCAT PLC-Control die ADS-Verbindung an.

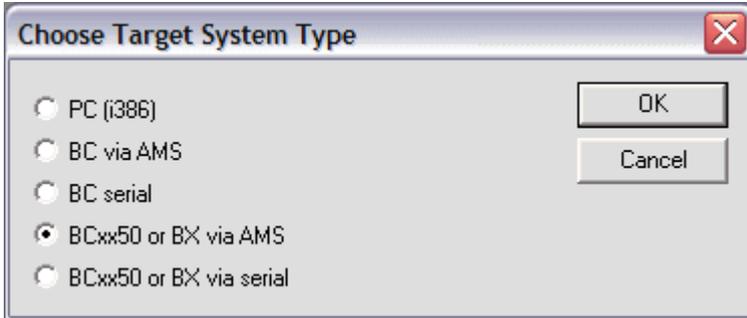


Abb. 122: Auswahl des Datenübertragungswegs - AMS

Im PLC Control kann man sich unter *Online/Communication Parameters...* einwählen.

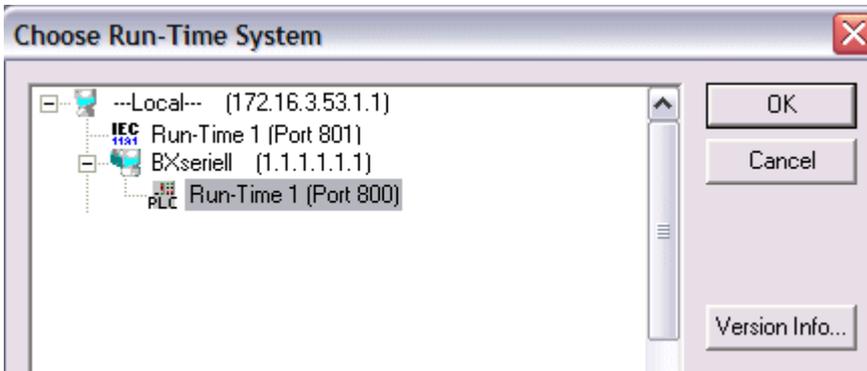


Abb. 123: Auswahl des Gerätes

## 5.14.2 Programmierung über CANopen

TwinCAT bietet die Eigenschaft, 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 angewählt werden. Erforderlich ist der TwinCAT-Level TwinCAT PLC.

### Mindestanforderungen

FC510x mit Firmware ab 1.55  
TwinCAT 2.9 Build 948

### 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 Felbusvariablen an und verknüpfen sie diese mit einer Task. Für den Späteren Programm Download über den CANopen 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

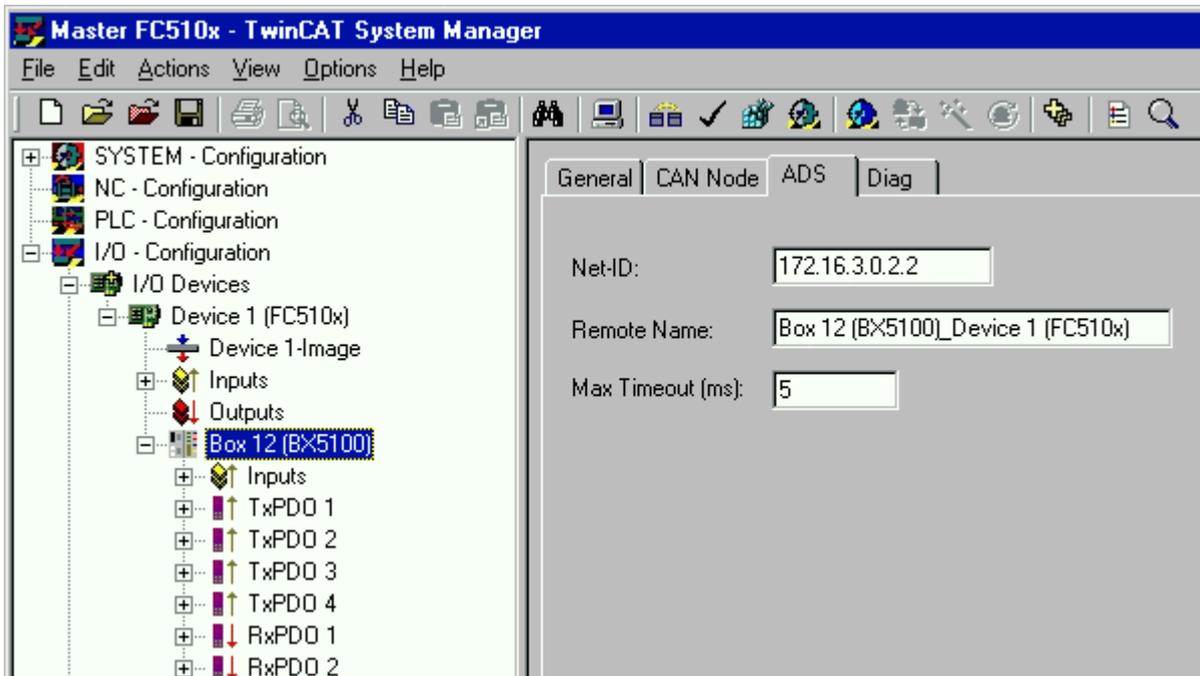


Abb. 124: Darstellung des BX5100 im TwinCAT System Manager

### 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 Task)
- BC seriell, das serielle Kabel für die Kommunikation über die RS232-Schnittstelle des PCs und die Programmierschnittstelle des Busklemmen-Controllers

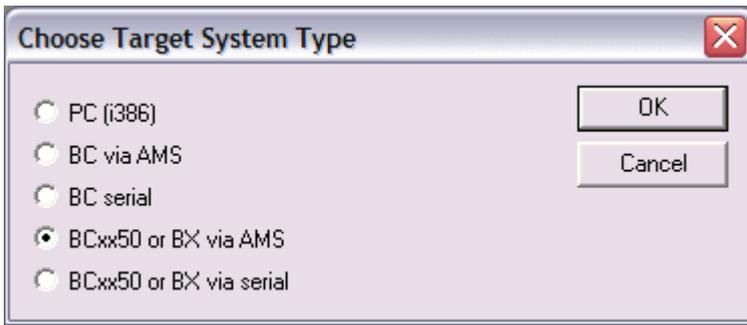


Abb. 125: Auswahl des Zielsystems

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 2 und die Run-Time 1 des Busklemmen-Controllers.

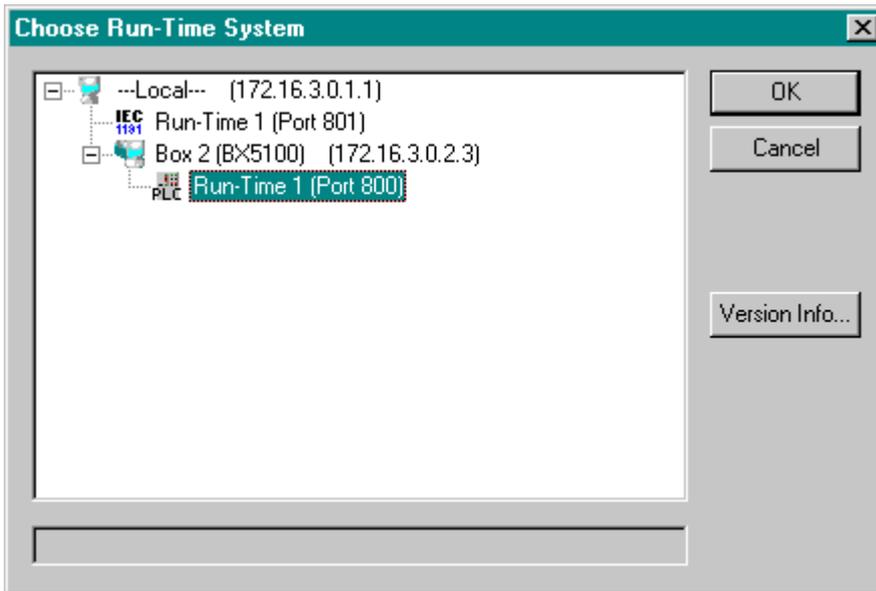


Abb. 126: Auswahl des Zielsystems - Box 2, Run-Time 1 des Busklemmen-Controllers

# 6 CANopen Kommunikation

## 6.1 CANopen Einführung



Abb. 127: CANopenLogo

CANopen ist eine weit verbreitete CAN-Anwendungsschicht, die im Verband CAN-in-Automation (CiA, <http://www.can-cia.org>) entwickelt und inzwischen zur internationalen Normung angenommen wurde.

### Gerätemodell

CANopen besteht aus der Protokolldefinition (Kommunikationsprofil) sowie den Geräteprofilen, die den Dateninhalt für die jeweilige Geräteklasse normieren. Zur schnellen Kommunikation der Ein- und Ausgangsdaten dienen die Prozessdatenobjekte (PDO) [▶ 143]. Die CANopen-Geräteparameter und Prozessdaten sind in einem Objektverzeichnis strukturiert. Der Zugriff auf beliebige Daten dieses Objektverzeichnisses erfolgt über die Servicedatenobjekte (SDO). Weiter gibt es einige Spezialobjekte (bzw. Telegrammarten) für Netzwerkmanagement (NMT), Synchronisation, Fehlermeldungen etc.

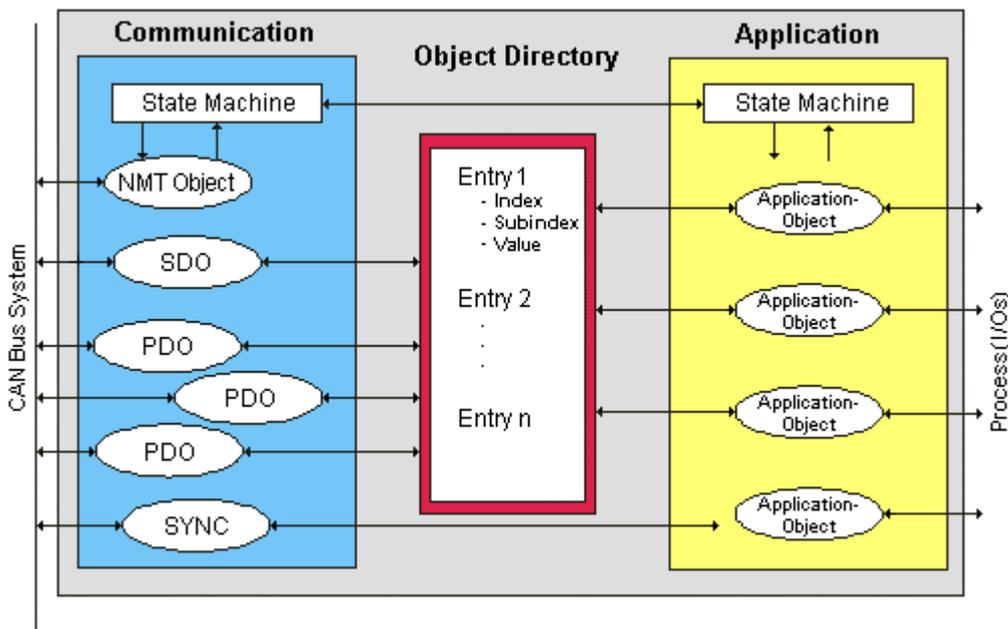


Abb. 128: CANopen Gerätemodell

### Kommunikationsarten

CANopen definiert mehrere Kommunikationsarten für die Ein- und Ausgangsdaten (Prozessdatenobjekte):

- Ereignisgesteuert [▶ 145]: Telegramme werden versendet, sobald sich der Inhalt geändert hat. Hier wird nicht ständig das Prozessabbild, sondern nur die Änderung desselben übertragen.
- Zyklisch synchron [▶ 146]: Über ein SYNC Telegramm werden die Baugruppen veranlasst, die vorher empfangenen Ausgangsdaten zu übernehmen und neue Eingangsdaten zu senden.
- Angefordert (gepollt) [▶ 143]: Über ein CAN Datenanforderungstelegramm werden die Baugruppen veranlasst ihre Eingangsdaten zu senden.

Die gewünschte Kommunikationsart wird über den Parameter Transmission Type [► 143] eingestellt.

### Geräteprofil

Die BECKHOFF CANopen-Geräte unterstützen alle E/A- Kommunikationsarten und entsprechen dem Geräteprofil für digitale und analoge Ein-/Ausgabebaugruppen (DS401 Version 1). Aus Gründen der Abwärtskompatibilität wurde das Default Mapping nicht der Profilversion DS401 V2 angepasst.

### Übertragungsraten

Neun Übertragungsraten von 10 kBit/s bis 1 MBit/s stehen für unterschiedliche Buslängen zur Verfügung. Durch die effektive Nutzung der Busbandbreite erreicht CANopen kurze Systemreaktionszeiten bei vergleichsweise niedrigen Datenraten.

### Topologie

CAN basiert auf einer linienförmigen Topologie [► 28]. Die Anzahl der Teilnehmer pro Netz ist dabei von CANopen logisch auf 128 begrenzt, physikalisch erlaubt die aktuelle Treiber-Generation bis zu 64 Knoten in einem Netzsegment. Die bei einer bestimmten Datenrate maximal mögliche Netzausdehnung ist durch die auf dem Busmedium erforderliche Signallaufzeit begrenzt. Bei 1 MBit/s ist z. B. eine Netzausdehnung von 25 m, bei 50 kBit/s eine Netzausdehnung von 1000 m möglich. Bei niedrigen Datenraten kann die Netzausdehnung durch den Einsatz von Repeatern erhöht werden, diese ermöglichen auch den Aufbau von Baumstrukturen.

### Buszugriffsverfahren

CAN arbeitet nach dem Verfahren Carrier Sense Multiple Access (CSMA), d.h. jeder Teilnehmer ist bezüglich des Buszugriffs gleichberechtigt und kann auf den Bus zugreifen, sobald dieser frei ist (Multi-Master-Buszugriff). Der Nachrichtenaustausch ist dabei nicht Teilnehmerbezogen sondern Nachrichtenbezogen. Das bedeutet, dass jede Nachricht mit einem priorisierten Identifier eindeutig gekennzeichnet ist. Damit beim Verschicken der Nachrichten verschiedener Teilnehmer keine Kollisionen auf dem Bus entstehen, wird beim Start der Datenübertragung eine bitweise Busarbitrierung durchgeführt. Die Busarbitrierung vergibt die Busbandbreite an die Nachrichten in der Reihenfolge ihrer Priorität, am Ende der Arbitrierungsphase belegt jeweils nur ein Busteilnehmer den Bus, Kollisionen werden vermieden und die Bandbreite wird optimal genutzt.

### Konfiguration und Parametrierung

Mit dem TwinCAT System Manager können alle CANopen Parameter komfortabel eingestellt werden. Für die Parametrierung der Beckhoff CANopen-Geräte mit Konfigurationstools dritter Hersteller steht Ihnen auf der Beckhoff Website (<http://www.beckhoff.de>) ein eds-File (electronic data sheet) zur Verfügung.

### Zertifizierung

Die Beckhoff CANopen-Geräte verfügen über eine leistungsfähige Protokollimplementierung und sind vom Verband CAN-in-Automation (<http://www.can-cia.org>) zertifiziert.

## 6.2 Protokollbeschreibung

### 6.2.1 Netzwerkmanagement

#### Einfacher Boot-Up

CANopen erlaubt einen sehr einfachen Boot-Up des verteilten Netzwerkes. Die Module befinden sich nach der Initialisierung automatisch im Zustand *Pre-Operational*. In diesem Zustand kann bereits über Service-Datenobjekte (SDOs) mit Default-Identifiern auf das Objektverzeichnis zugegriffen werden, die Module können also konfiguriert werden. Da für alle Einträge im Objektverzeichnis Default-Einstellungen vorhanden sind, kann in den meisten Fällen auf eine Konfiguration verzichtet werden.

Zum Starten der Module ist dann nur eine einzige CAN-Nachricht erforderlich: `Start_Remote_Node`: Identifier 0, zwei Datenbytes: 0x01, 0x00. Sie überführt die Knoten in den Zustand *Operational*.

**Netzwerkstatus**

Die Zustände im CANopen Boot-Up und die Zustandsübergänge sind aus dem Zustandsdiagramm ersichtlich:

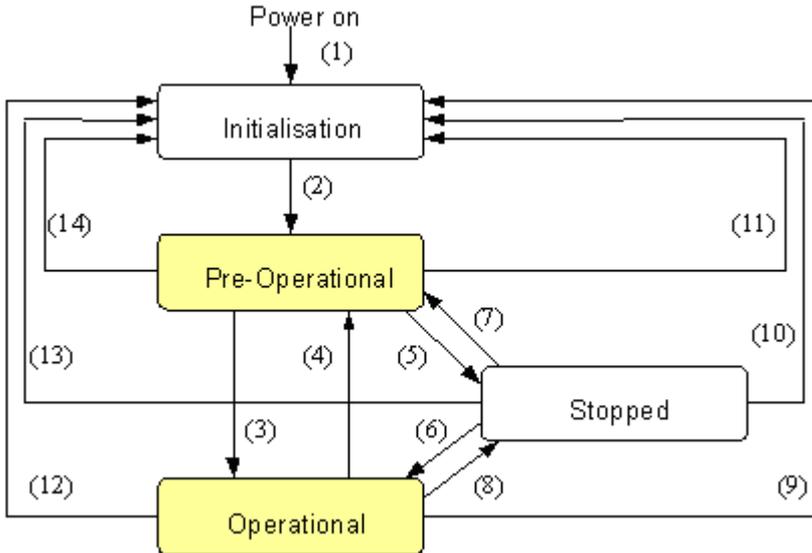


Abb. 129: Zustandsdiagramm CANopen Boot-up

**Pre-Operational**

Nach der Initialisierung geht der Buskoppler automatisch, d.h. ohne Befehl von außen, in den Zustand *Pre-Operational* über. In diesem Zustand kann er konfiguriert werden, denn die Servicedatenobjekte (SDOs) sind bereits aktiv. Die Prozessdatenobjekte sind hingegen noch gesperrt.

**Operational**

Im Zustand *Operational* sind auch die Prozessdatenobjekte aktiv.

Wenn der Buskoppler aufgrund äußerer Einflüsse (z. B. CAN-Störung, keine Ausgangs-Spannung) oder innerer Einflüsse (z. B. K-Bus-Fehler) nicht mehr in der Lage ist, Ausgänge zu setzen oder Eingänge zu lesen bzw. zu kommunizieren, so versucht er eine entsprechende Emergency-Nachricht zu senden, geht in den Fehlerzustand und fällt dabei in den Zustand *Pre-Operational* zurück. Damit kann auch die NMT-Statusmaschine des Netzwerkmasters fatale Fehler sofort erkennen.

**Stopped**

Im Zustand *Stopped* (früher *Prepared*) ist keine Datenkommunikation mit dem Koppler möglich - lediglich NMT-Nachrichten werden empfangen. Die Ausgänge gehen in den Fehlerzustand.

**Statusübergänge**

Die Netzwerkmanagement-Nachrichten haben einen sehr einfachen Aufbau: CAN-Identifier 0 mit zwei Byte Dateninhalt. Das erste Datenbyte enthält den sogenannten Command-Specifier (cs), das zweite Datenbyte die Knotenadresse, wobei die Knotenadresse 0 alle Knoten anspricht (Broadcast).

11-bit Identifier	2 Byte Nutzdaten						
0x00	cs	Node-ID					

Die folgende Tabelle gibt einen Überblick über alle CANopen Statusübergänge und die dazugehörigen Kommandos (Command Specifier im NMT Master-Telegramm):

Statusübergang	Command Specifier cs	Erläuterung
(1)	-	Der Initialisierungs-Status wird beim Einschalten selbsttätig erreicht
(2)	-	Nach der Initialisierung wird der Status Pre-Operational automatisch erreicht - dabei wird die Boot-Up-Nachricht abgeschickt.
(3), (6)	cs = 1 = 0x01	Start_Remote_Node. Startet Modul, gibt Ausgänge frei, Startet Übertragung von PDOs.
(4), (7)	cs = 128 = 0x80	Enter_Pre-Operational. Stoppt PDO-Übertragung, SDO weiter aktiv.
(5), (8)	cs = 2 = 0x02	Stop_Remote_Node. Ausgänge gehen in den Fehlerzustand, SDO und PDO abgeschaltet.
(9), (10), (11)	cs = 129 = 0x81	Reset_Node. Führt Reset durch. Alle Objekte werden auf Power-On Defaults zurückgesetzt.
(12), (13), (14)	cs = 130 = 0x82	Reset_Communication. Führt Reset der Kommunikationsfunktionen durch. Objekte 0x1000 - 0x1FFF werden auf Power-On Defaults zurückgesetzt

### Beispiel 1

Mit folgendem Telegramm werden netzwerkweit alle Baugruppen in den Fehlerzustand (Ausgänge sicherer Zustand) überführt:

11-bit Identifier	2 Byte Nutzdaten						
0x00	0x02	0x00					

### Beispiel 2

Mit folgendem Telegramm wird Knoten 17 zurückgesetzt (resetted):

11-bit Identifier	2 Byte Nutzdaten						
0x00	0x81	0x11					

### Boot-Up-Nachricht

Nach der Initialisierungsphase und dem Selbsttest sendet der Buskoppler die Boot-Up-Nachricht, eine CAN-Nachricht mit einem Datenbyte (0) auf dem Identifier der Guarding- bzw. Heartbeat-Nachricht: CAN-ID = 0x700 + Node-ID. Damit kann ein temporärer Ausfall einer Baugruppe während des Betriebs (z. B. durch einen Spannungseinbruch) oder eine nachträglich eingeschaltete Baugruppe zuverlässig auch ohne Node Guarding festgestellt werden. Der Sender kann über den Identifier der Nachricht (siehe Default-Identifier-Verteilung) bestimmt werden.

Außerdem ist es mit Hilfe der Boot-Up-Nachricht möglich, die beim Aufstarten am Netz befindlichen Knoten mit einem einfachen CAN-Monitor zu erkennen, ohne dass ein Schreibzugriff (z. B. Scannen des Netzwerks durch Auslesen von Parameter 0x1000) auf den Bus erforderlich ist.

Schließlich wird durch die Boot-Up-Nachricht das Ende der Initialisierungsphase kommuniziert; der Buskoppler signalisiert, dass er nun konfiguriert bzw. gestartet werden kann.



#### Firmwarestand BA

Bis Firmwarestand BA wurde für die Boot-Up-Nachricht der Emergency Identifier genutzt.

**Format Boot-Up Nachricht**

11-bit Identifier	1 Byte Nutzdaten						
0x700 (=1792) + Node-ID	0x00						

**Knotenüberwachung**

Für die Ausfallüberwachung des CANopen Netzwerkes stehen Heartbeat und Guarding-Mechanismen zur Verfügung. Diese sind bei CANopen besonders wichtig, da sich die Baugruppen in der ereignisgesteuerten Betriebsart nicht regelmäßig melden. Beim Guarding werden die Teilnehmer per Datenanforderungstelegramm (Remote Frame) zyklisch nach ihrem Status gefragt, beim Heartbeat senden die Knoten ihren Status von selbst.

**Guarding: Node Guarding und Life Guarding**

Über Node Guarding werden die dezentralen Peripherie-Baugruppen überwacht, die ihrerseits über Life Guarding den Ausfall des Guarding-Masters erkennen können. Beim Guarding setzt der Master Remote Frames (remote transmit request, Nachrichten-Anforderungstelegramme) auf die Guarding Identifier der zu überwachenden Slaves ab. Diese antworten mit der Guarding-Nachricht. Diese enthält den Status-Code des Slaves sowie ein Toggle-Bit, das nach jeder Nachricht wechseln muss. Falls Status- oder Toggle-Bit nicht mit den vom NMT-Master erwarteten übereinstimmen oder falls keine Antwort erfolgt geht der Master von einem Slave-Fehler aus.

**Guarding-Verfahren**

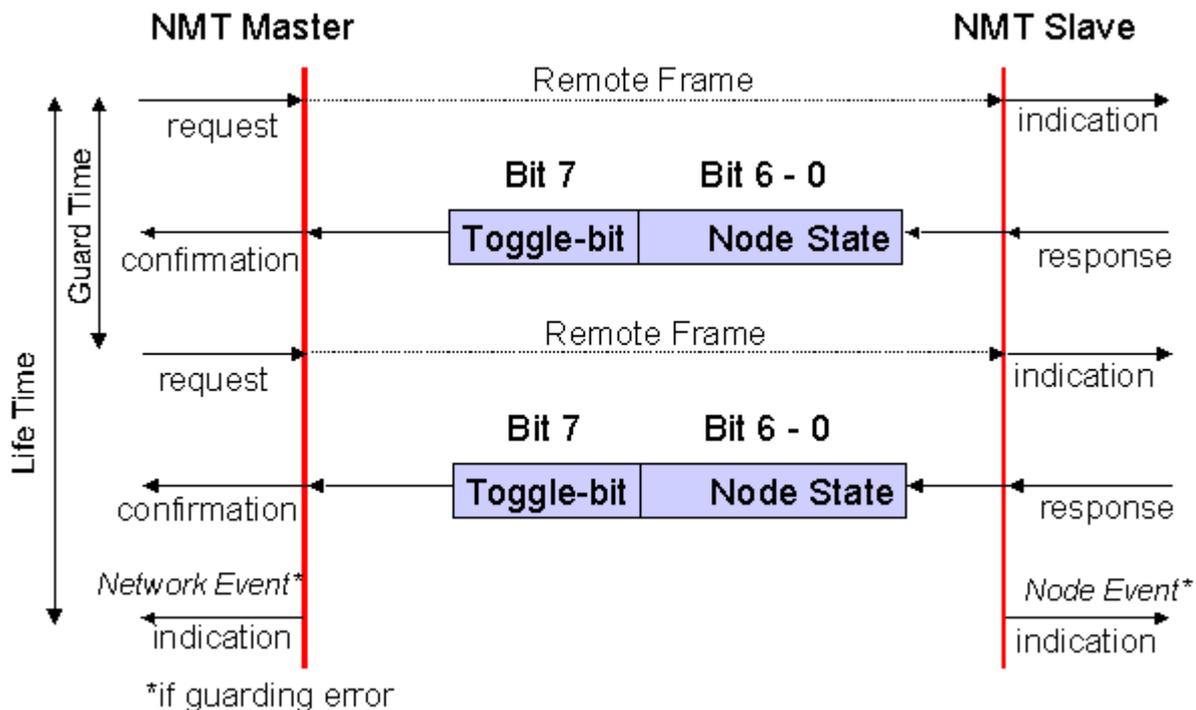


Abb. 130: Schematische Darstellung „Guarding-Verfahren“

**Protokoll**

Das im ersten Guarding-Telegramm übertragene Toggle-Bit (t) hat den Wert 0. Anschließend wechselt (toggelt) das Bit in jedem Guarding-Telegramm und signalisiert so, ob ein Telegramm verloren ging. In den restlichen sieben Bit gibt der Knoten seinen Netzwerk Status (s) an:

s	Status
4 = 0x04	Stopped (früher: Prepared)
5 = 0x05	Operational
127 = 0x7F	Pre-Operational

### Beispiel

Die Garding Nachricht des Knotens 27 (0x1B) muss mit einem Remote Frame mit Identifier 0x71B (1819<sub>dez</sub>) angefragt werden. Wenn der Knoten *Operational* ist, wechselt das erste Datenbyte der Antwort-Nachricht zwischen 0x05 und 0x85, im Zustand *Pre-Operational* wechselt es zwischen 0x7F und 0xFF.

### Guard Time und Life Time Factor

Wenn der Master die Guard-Nachrichten streng zyklisch anfordert, kann der Slave den Ausfall des Masters erkennen. Falls der Slave in diesem Fall innerhalb der eingestellten *Node Life Time* keine Nachrichtenanforderung vom Master erhält (Guarding-Fehler), geht er von einem Masterausfall aus (Watchdog-Funktion). Dann setzt er seine Ausgänge in den Fehlerzustand, sendet ein Emergency-Telegramm und fällt in den Zustand *Pre-Operational* zurück. Nach einem Guarding Time-Out kann das Verfahren durch Übertragen eines erneuten Guarding-Telegramms wieder angeregt werden.

Die Node Life-Time berechnet sich aus den Parametern Guard-Time (Objekt 0x100C) und Life-Time-Factor (Objekt 0x100D):

Life-Time = Guard-Time x Life-Time-Factor

Falls einer der beiden Parameter "0" ist (Default-Einstellung), erfolgt keine Überwachung des Masters (kein Life Guarding).

### Heartbeat: Knotenüberwachung ohne Remote Frame

Beim Heartbeat-Verfahren senden die Knoten ihre jeweilige Statusmeldung zyklisch selbsttätig. Es kann daher auf Remote Frames verzichtet werden und es wird weniger Buslast erzeugt als beim Guarding-Verfahren.

Der Master sendet sein Heartbeat-Telegramm ebenfalls zyklisch, die Slaves können somit den Ausfall des Masters ebenfalls erkennen.

Heartbeat-Verfahren

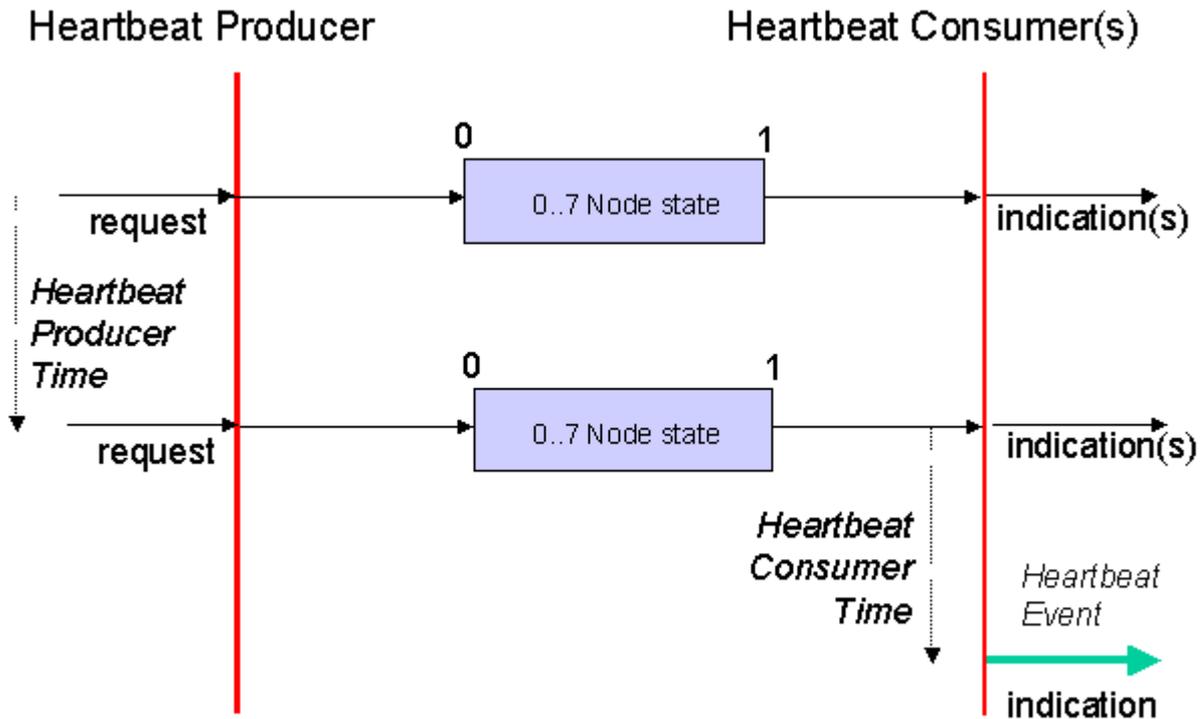


Abb. 131: Schematische Darstellung „Heartbeat-Verfahren“

Protokoll

Beim Heartbeat-Verfahren wird auf das Toggle-Bit verzichtet, die Knoten senden zyklisch Ihren Status (s). Siehe [Guarding \[141\]](#).

## 6.2.2 Prozessdatenobjekte (PDO)

Einführung

Bei vielen Feldbus-Systemen wird ständig das gesamte Prozessabbild übertragen - meist mehr oder weniger zyklisch. CANopen ist nicht auf dieses Kommunikationsprinzip beschränkt, da CAN durch die Multi-Master Buszugriffsregelung auch andere Möglichkeiten bietet: die Prozessdaten werden bei CANopen nicht im Master/Slave-Verfahren übertragen, sondern folgen dem Produzenten/Konsumenten-Modell (Producer/Consumer). Hierbei sendet ein Busknoten seine Daten von sich aus (Producer), beispielsweise durch den Eintritt eines Ereignisses getriggert; alle anderen Knoten hören mit und entscheiden anhand des Identifiers, ob sie sich für dieses Telegramm interessieren und verarbeiten es entsprechend (Consumer).

Bei CANopen werden die Prozessdaten in Segmente zu maximal 8 Byte aufgeteilt. Diese Segmente heißen Prozessdatenobjekte (PDOs). Die PDOs entsprechen jeweils einem CAN-Telegramm und werden über dessen spezifischen CAN-Identifier zugeordnet und in ihrer Priorität bestimmt. Man unterscheidet Empfangs-PDOs (Receive-PDOs, RxPDOs) und Sende-PDOs (Transmit-PDOs, TxPDOs), wobei die Bezeichnung jeweils aus Gerätesicht erfolgt: eine Ein-/Ausgabebaugruppe sendet ihre Eingangsdaten mit TxPDOs, und empfängt die Ausgangsdaten in den RxPDOs. **Diese Bezeichnung wird im TwinCAT-System-Manager beibehalten.**

Kommunikationsparameter

Die PDOs können je nach Applikationsanforderung mit unterschiedlichen Kommunikationsparametern versehen werden. Wie alle CANopen-Parameter stehen auch diese im Objektverzeichnis des Gerätes, auf sie kann über die Servicedatenobjekte zugegriffen werden. Die Parameter für die Empfangs-PDOs stehen

bei Index 0x1400 (RxPDO1) und folgende, bis zu 512 RxPDOs können vorhanden sein (Bereich bis Index 0x15FF). Entsprechend finden sich die Einträge für die Sende-PDOs bei Index 0x1800 (TxPDO1) bis 0x19FF (TxPDO512).

Für den Prozessdatenaustausch stehen auf den Beckhoff Buskopplern bzw. Feldbus Koppler Box Baugruppen jeweils 16 RxPDO und TxPDOs zur Verfügung (bei den Economy- und LowCost-Kopplern BK5110 und LC5100 sowie den Feldbus Boxen sind es jeweils 5 PDOs, da diese Geräte über weniger Prozessdaten verfügen). Die FC510x CANopen Master Karte unterstützt - beschränkt durch die DPRAM-Größe - je Kanal bis zu 192 Sende- und 192 Empfangs-PDOs. Die CANopen Klemme EL6751 organisiert das Prozessabbild dynamisch, d.h. die Prozessdaten werden hintereinander geschrieben, was eine höhere Datenübertragungsrate ermöglicht. Im Slave Mode können bis zu 32 TxPDOs und 32 RxPDOs verarbeitet werden.

Für jedes vorhandene Prozessdatenobjekt ist ein zugehöriges Kommunikationsparameter-Objekt vorhanden. Der TwinCAT-System-Manager ordnet die eingestellten Parameter automatisch den jeweiligen Objektverzeichniseinträgen zu. Im Folgenden werden diese Einträge samt ihrer Bedeutung für das Kommunikationsverhalten der Prozessdaten erläutert.

### PDO-Identifizier

Der wichtigste Kommunikationsparameter eines PDOs ist der CAN-Identifizier (auch Communication Object Identifier, COB-ID genannt). Er dient zur Identifizierung der Daten und bestimmt deren Priorität beim Buszugriff. Für jedes CAN-Datentelegramm darf es nur einen Sendeknoten (Producer) geben; da CAN jedoch alle Nachrichten im Broadcast-Verfahren sendet kann ein Telegramm wie beschrieben von beliebig vielen Knoten empfangen werden (Consumer). Ein Knoten kann also seine Eingangsinformation mehreren Busteilnehmern gleichzeitig zur Verfügung stellen - auch ohne Weiterleitung durch einen logischen Busmaster. Der Identifizier steht in Subindex 1 des Kommunikationsparametersatzes. Er ist als 32-Bit Wert kodiert, wobei die niederwertigsten 11 Bits (Bit 0...10) den eigentlichen Identifizier enthalten. Die Datenbreite des Objektes von 32 Bit erlaubt auch den Eintrag von 29 Bit-Identifiern nach CAN 2.0B, allerdings beziehen sich die Default-Identifizier stets auf die üblichere 11 Bit-Variante. Allgemein geht CANopen sparsam mit den zur Verfügung stehenden Identifiern um, sodass der Einsatz der 29 Bit-Variante auf Sonderanwendungen beschränkt bleibt - und daher auch von den Beckhoff CANopen Geräten nicht unterstützt wird. Über das höchstwertige Bit (Bit 31) lässt sich das Prozessdatenobjekt aktivieren bzw. abschalten.

Im Anhang finden Sie eine komplette [Identifizier-Liste \[► 194\]](#).

### PDO Linking

Im System der Default-Identifizier kommunizieren alle Knoten (hier: Slaves) mit einer Zentrale (Master), da kein Slave-Knoten per Default auf die Sende-Identifizier eines anderen Slave-Knotens hört).

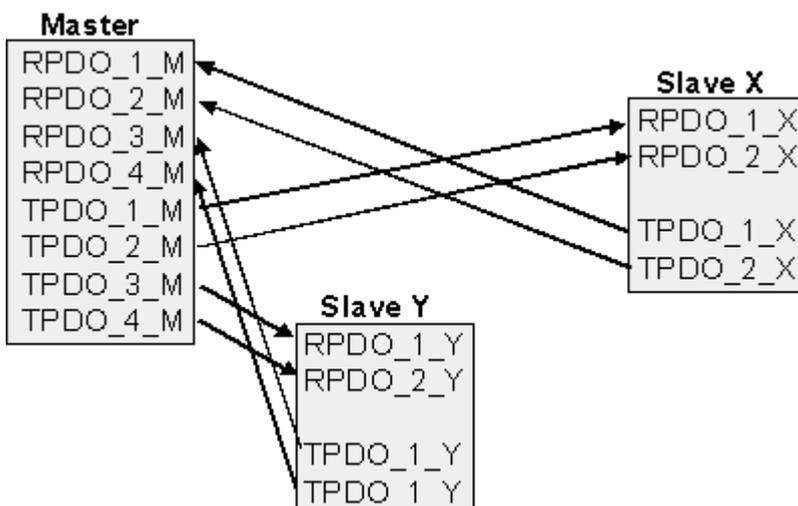


Abb. 132: Default Identifizier-Verteilung: Master/Slave

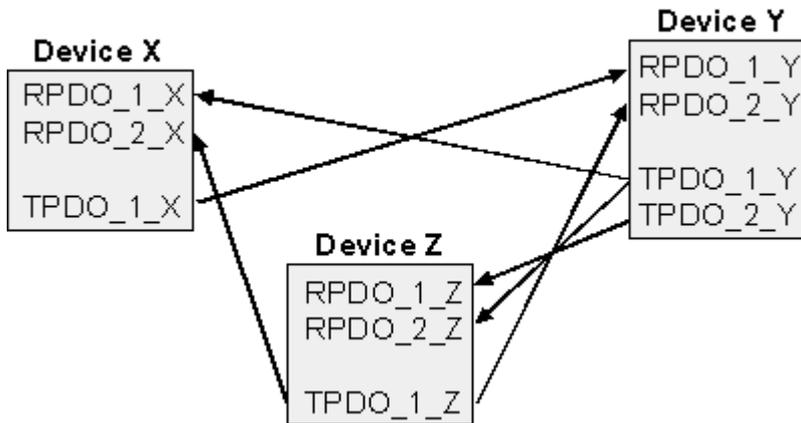


Abb. 133: PDO Linking: Peer to Peer

Wenn das Consumer-Producer-Modell der CANopen PDOs zum direkten Datenaustausch zwischen Knoten (ohne Master) genutzt werden soll, so muss die Identifier-Verteilung entsprechend angepasst werden, damit der TxPDO-Identifier des Producers mit dem RxPDO-Identifier des Consumers übereinstimmt. Dieses Verfahren nennt man PDO Linking. Es ermöglicht beispielsweise den einfachen Aufbau von elektronischen Getrieben, bei denen mehrere Slave-Achsen gleichzeitig auf den Ist-Wert im TxPDO der Master-Achse hören.

**PDO-Kommunikationsarten: Überblick**

CANopen bietet vielfältige Möglichkeiten, die Prozessdaten zu übertragen (siehe auch: [Hinweise zur PDO Parametrierung](#) [▶ 150])

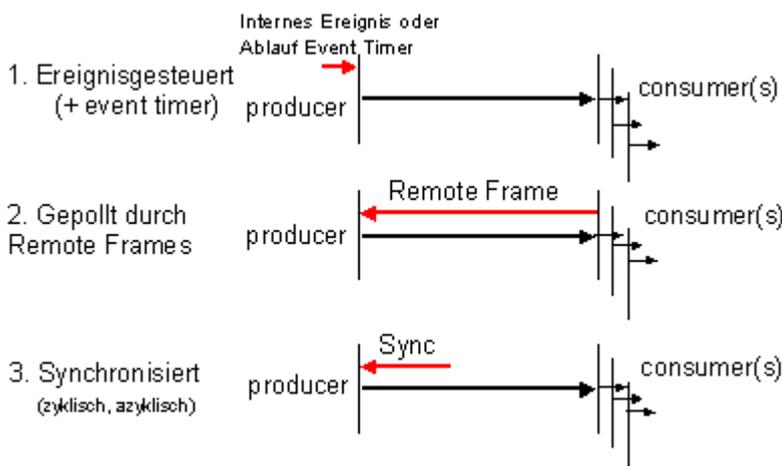


Abb. 134: Darstellung Übertragung CAN-Prozessdaten

**Ereignisgesteuert**

Das "Ereignis" ist die Änderung eines Eingangswertes, die Daten werden sofort nach dieser Änderung verschickt. Durch die Ereignissteuerung wird die Busbandbreite optimal ausgenutzt, da nicht ständig das Prozessabbild, sondern nur die Änderung desselben übertragen wird. Gleichzeitig wird eine kurze Reaktionszeit erreicht, da bei Änderung eines Eingangswertes nicht erst auf die nächste Abfrage durch einen Master gewartet werden muss.

Ab CANopen Version 4 kann die ereignisgesteuerte Kommunikationsart mit einem zyklischen Update kombiniert werden. Auch wenn gerade kein Ereignis aufgetreten ist, werden ereignisgesteuerte TxPDO nach Ablauf des Event Timers verschickt. Beim Auftreten eines Ereignisses wird der Event Timer zurückgesetzt. Bei RxPDOs wird der Event Timer als Watchdog benutzt um das Eintreffen von ereignisgesteuerten PDOs zu überwachen. Sollte innerhalb der eingestellten Zeit kein PDO eingetroffen sein, so geht der Busknoten in den Fehlerzustand.

## Gepollt

Die PDOs können auch durch Datenanforderungstelegramme (Remote Frames) gepollt werden. Auf diese Art kann etwa das Eingangsprozessabbild bei ereignisgesteuerten Eingängen auch ohne deren Änderung auf den Bus gebracht werden, beispielsweise bei einem zur Laufzeit ins Netz aufgenommenen Monitor- oder Diagnosegerät. Das zeitliche Verhalten von Remote Frame und Antworttelegramm hängt von den verwendeten CAN-Controllern ab. Bausteine mit integrierter kompletter Nachrichtenfilterung ("FullCAN") beantworten ein Datenanforderungstelegramm in der Regel direkt und versenden sofort die im entsprechenden Sendebuffer stehenden Daten - dort muss die Applikation dafür Sorge tragen, dass die Daten ständig aktualisiert werden. CAN-Controller mit einfacher Nachrichtenfilterung (BasicCAN) reichen die Anforderung dagegen an die Applikation weiter, die nun das Telegramm mit den aktuellen Daten zusammenstellen kann. Das dauert länger, dafür sind die Daten aktuell. Beckhoff verwendet CAN Controller nach dem Basic CAN Prinzip.

Da dieses Geräteverhalten für den Anwender meist nicht transparent ist und zudem noch CAN-Controller in Verwendung sind, die Remote Frames überhaupt nicht unterstützen, kann die gepollte Kommunikationsart nur bedingt für den laufenden Betrieb empfohlen werden.

## Synchronisiert

Nicht nur bei Antriebsanwendungen ist es sinnvoll, das Ermitteln der Eingangsinformation sowie das Setzen der Ausgänge zu synchronisieren. CANopen stellt hierzu das SYNC-Objekt zur Verfügung, ein CAN-Telegramm hoher Priorität ohne Nutzdaten, dessen Empfang von den synchronisierten Knoten als Trigger für das Lesen der Eingänge bzw. für das Setzen der Ausgänge verwendet wird.

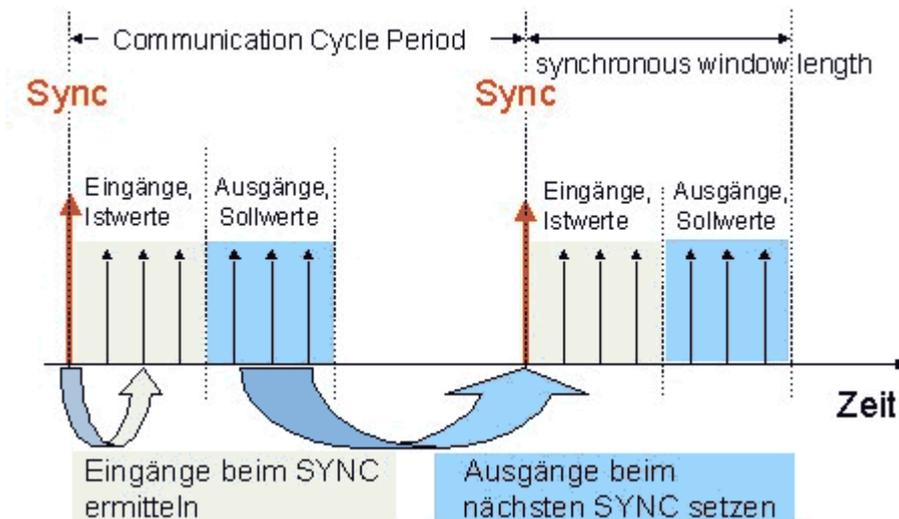


Abb. 135: Darstellung CAN Telegramm „SYNC“

## PDO-Übertragungsart: Parametrierung

Der Parameter PDO-Übertragungsart (Transmission Type) legt fest, wie das Versenden des PDOs ausgelöst wird bzw. wie empfangene PDOs behandelt werden:

Übertragungsart	Zyklisch	Azyklisch	Synchron	Asynchron	Nur RTR
0		X	X		
1-240	X		X		
241-251	- reserviert -				
252			X		X
253				X	X
254, 255				X	

Die Übertragungsart wird für RxPDOs in den Objekten 0x1400ff, Subindex 2, und für TxPDOs in den Objekten 0x1800ff, Subindex 2 parametrierung.

## Azyklisch Synchron

PDOs der Übertragungsart 0 arbeiten synchron, aber nicht zyklisch. Ein RxPDO wird erst nach Empfang des nächsten SYNC-Telegramms ausgewertet. Damit lassen sich beispielsweise Achsgruppen nacheinander mit neuen Zielpositionen versehen, die alle beim nächsten SYNC gültig werden - ohne dass ständig Stützstellen ausgegeben werden müssen. Ein Gerät, dessen TxPDO auf Übertragungsart 0 konfiguriert ist, ermittelt seine Eingangsdaten beim Empfang des SYNC (synchrones Prozessabbild) und sendet sie anschließend, falls die Daten einem Ereignis entsprechen (beispielsweise eine Eingangsänderung) eingetreten ist. Die Übertragungsart 0 kombiniert also den Sendegrund "ereignisgesteuert" mit dem Sende- (und möglichst Sample-) bzw. Verarbeitungs-Zeitpunkt "SYNC-Empfang".

## Zyklisch Synchron

Bei Übertragungsart 1-240 wird das PDO zyklisch gesendet: nach jedem "n-ten" SYNC ( $n=1\dots240$ ). Da die Übertragungsart nicht nur im Netz, sondern auch auf einem Gerät kombiniert werden dürfen, kann so z. B. ein schneller Zyklus für digitale Eingänge vereinbart werden ( $n=1$ ), während die Daten der Analogeingänge in einem langsameren Zyklus übertragen werden (z. B.  $n=10$ ). RxPDOs unterscheiden in der Regel nicht zwischen den Übertragungsarten 0...240: ein empfangenes PDO wird beim nächsten SYNC-Empfang gültig gesetzt. Die Zykluszeit (SYNC-Rate) kann überwacht werden (Objekt 0x1006), das Gerät reagiert bei SYNC-Ausfall dann entsprechend der Definition des Geräteprofils und schaltet z. B. seine Ausgänge in den Fehlerzustand.

Die FC510x Karte / EL6751Klemme unterstützen die synchrone Kommunikationsart vollständig: das Versenden des SYNC Telegramms ist mit der verknüpften Task gekoppelt, sodass zu jedem Taskbeginn neue Eingangsdaten zur Verfügung stehen. Das Ausbleiben eines synchronen PDOs wird erkannt und an die Applikation gemeldet.

## Nur RTR

Die Übertragungsarten 252 und 253 gelten für Prozessdatenobjekte, die ausschließlich auf Anforderung durch ein Remote Frame übertragen werden. 252 ist synchron: beim Empfang des SYNCs werden die Prozessdaten ermittelt, gesendet werden sie nur auf Anforderung. 253 ist asynchron, hier werden die Daten ständig ermittelt und auf Anforderung verschickt. Diese Übertragungsart ist generell nicht zu empfehlen, da das Abholen der Eingangsdaten von einigen CAN Controllern nur unvollständig unterstützt wird. Da die CAN Controller zudem teilweise selbsttätig auf Remote Frames antworten (ohne vorher aktuelle Eingangs-Daten anzufordern), ist die Aktualität der gepollten Daten unter Umständen fragwürdig. Die Übertragungsart 252 und 253 wird aus diesen Gründen von den Beckhoff PC-Karten / Klemmen nicht unterstützt.

## Asynchron

Die Übertragungsarten 254 + 255 sind asynchron oder auch ereignisgesteuert. Bei Übertragungsart 254 ist das Ereignis herstellerspezifisch, bei 255 im Geräteprofil definiert. Im einfachsten Fall ist das Ereignis die Veränderung eines Eingangswertes - es wird also jede Werteänderung übertragen. Die Asynchrone Übertragungsart kann mit dem Event Timer gekoppelt werden und liefert so auch dann Eingangsdaten, wenn aktuell kein Ereignis aufgetreten ist.

## Inhibit Zeit

Über den Parameter "Inhibit-Zeit" kann ein "Sende-Filter" aktiviert werden, der die Reaktionszeit bei der relativ ersten Eingangsänderung nicht verlängert, aber bei unmittelbar darauffolgenden Änderungen aktiv ist. Die Inhibit-Zeit (Sendeverzögerungszeit) beschreibt die Zeitspanne, die zwischen dem Versenden zweier gleicher Telegramme mindestens abgewartet werden muss. Wenn die Inhibit-Zeit genutzt wird, so kann die maximale Busbelastung und damit die Latenzzeit im "worst case"-Fall ermittelt werden.

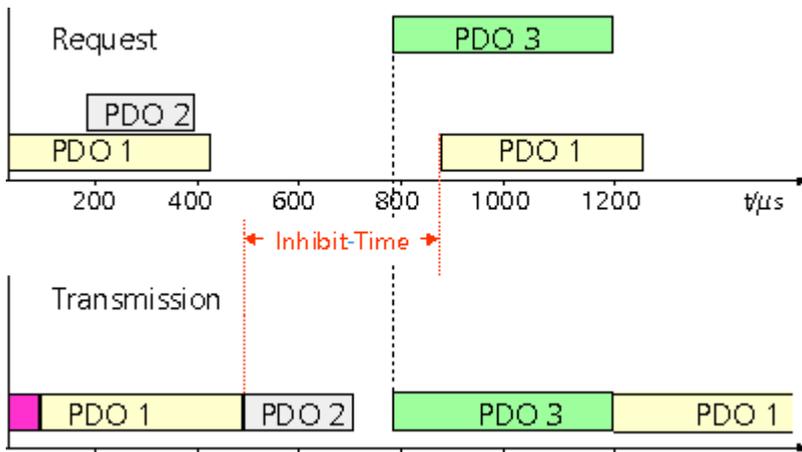


Abb. 136: Zeitl. Diagramm „Inhibit-Time“

Die Beckhoff PC-Karten FC510x / EL6751 Klemme können zwar die Inhibit-Zeit auf Slave-Geräten parametrieren, unterstützen sie jedoch selbst nicht. Eine Spreizung der gesendeten PDOs (Sendeverzögerung) ergibt sich automatisch aus der gewählten Zyklus-Zeit der SPS - und es macht wenig Sinn, die SPS schneller laufen zu lassen als es die Busbandbreite zulässt. Zudem kann die Busbelastung wirkungsvoll über die synchrone Kommunikation beeinflusst werden.

### Event Timer

Über Subindex 5 der Kommunikationsparameter lässt sich ein Ereignis-Timer (Event Timer) für Send-PDOs festlegen. Der Ablauf dieses Timers wird als zusätzlich eingetretenes Ereignis für das entsprechende PDO gewertet, das PDO wird also dann gesendet. Wenn das Applikationsereignis während einer Timer-Periode auftritt, so wird ebenfalls gesendet und der Timer wird zurückgesetzt.

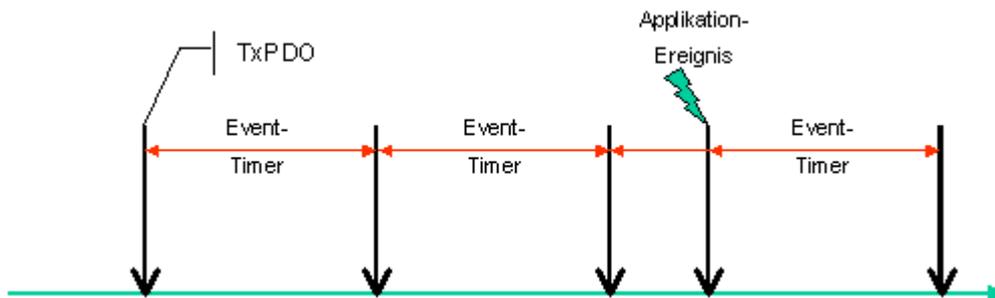


Abb. 137: Zeitliche Darstellung des Event-Timers

Bei Empfangs-PDOs wird der Timer-Parameter dazu verwendet, die Überwachungszeit für dieses PDO anzugeben: Die Applikation wird benachrichtigt, wenn kein entsprechendes PDO innerhalb der eingestellten Zeit empfangen wurde. Auf diese Art kann die FC510x / EL6751 jedes einzelne PDO individuell überwachen.

[Hinweise zur PDO Parametrierung \[► 150\]](#)

### PDO Mapping

Unter PDO-Mapping versteht man die Abbildung der Applikationsobjekte (Echtzeitdaten) aus dem Objektverzeichnis in die Prozessdatenobjekte. Die CANopen-Geräteprofile sehen für jeden Gerätetyp ein Default Mapping vor, das für die meisten Anwendungen passend ist. So bildet das Default Mapping für digitale E/A einfach die Ein- bzw. Ausgänge ihrer physikalischen Reihenfolge gemäß in die Send- bzw. Empfangs-Prozessdatenobjekte ab.

Die Default-PDOs für Antriebe enthalten jeweils 2 Byte Steuer- bzw. Statuswort und Soll- bzw. Istwert für die betreffende Achse.

Das aktuelle Mapping kann über entsprechende Einträge im Objektverzeichnis, die sogenannten Mapping-Tabellen, gelesen werden. An erster Stelle der Mapping Tabelle (Subindex 0) steht die Anzahl der gemappten Objekte, die im Anschluss aufgelistet sind. Die Tabellen befinden sich im Objektverzeichnis bei Index 0x1600 ff. für die RxPDOs bzw. 0x1A00ff für die TxPDOs.

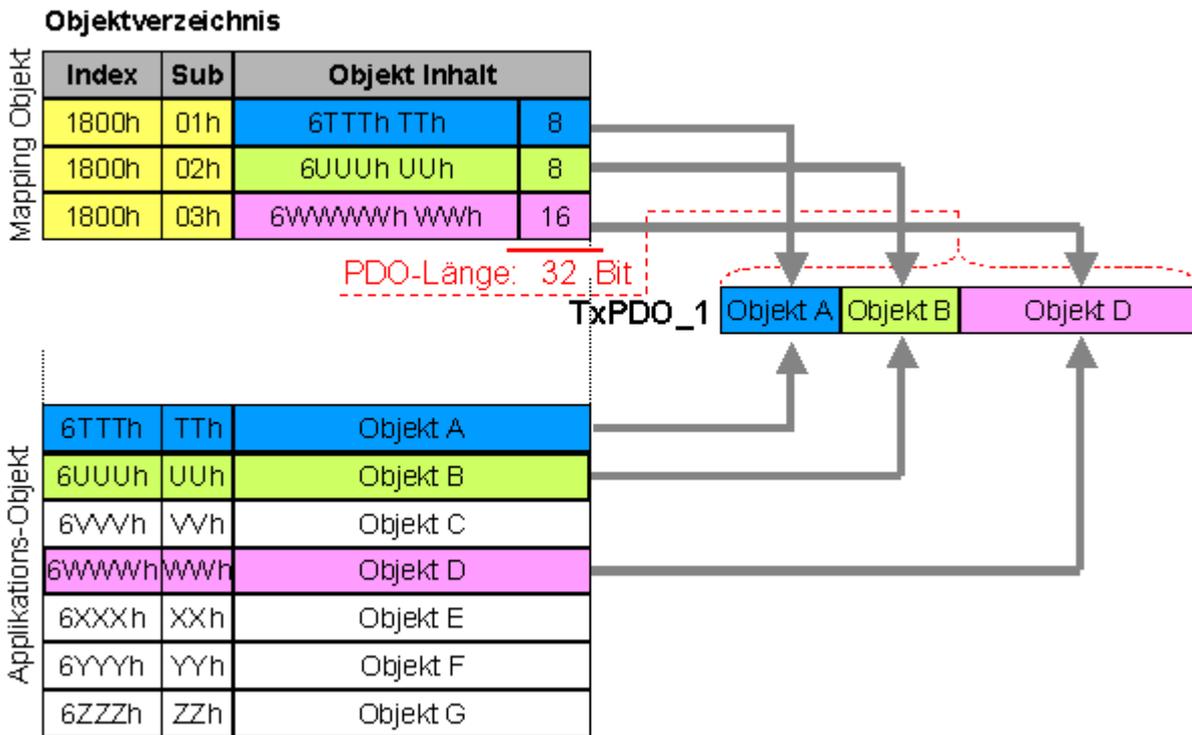


Abb. 138: Darstellung Mapping

**Digitale und analoge Ein-/Ausgabebaugruppen: E/A-Anzahl auslesen**

Die aktuelle Anzahl der digitalen und analogen Ein-/Ausgänge lässt sich durch Auslesen der entsprechenden Applikationsobjekte im Objektverzeichnis ermitteln bzw. verifizieren:

Parameter	Adresse Objektverzeichnis
Anzahl digitale Eingangsbytes	Index 0x6000, Subindex 0
Anzahl digitale Ausgangsbytes	Index 0x6200, Subindex 0
Anzahl analoge Eingänge	Index 0x6401, Subindex 0
Anzahl analoge Ausgänge	Index 0x6411, Subindex 0

**Variables Mapping**

In der Regel genügt die Default-Belegung der Prozessdatenobjekte (Default Mapping) bereits den Anforderungen. Für spezielle Anwendungsfälle kann die Belegung jedoch verändert werden: So unterstützen beispielsweise die Beckhoff CANopen Buskoppler das variable Mapping, bei dem die Applikationsobjekte (Ein- und Ausgangsdaten) frei den PDOs zugeordnet werden können. Hierzu müssen die Mapping-Tabellen konfiguriert werden: Ab CANopen Version 4 ist nur noch die folgende Vorgehensweise zulässig, die genau eingehalten werden muss:

1. Zunächst PDO löschen (0x1400ff, bzw. 0x1800ff, Subindex 1, Bit 31 auf "1" setzen)
2. Subindex 0 im Mapping Parameter (0x1600ff bzw. 0x1A00ff) auf "0" setzen
3. Mapping Einträge (0x1600ff bzw. 0x1A00ff, SI 1..8) verändern
4. Subindex 0 im Mapping Parameter auf gültigen Wert setzen. Das Gerät überprüft dann die Einträge auf Konsistenz.
5. PDO anlegen durch Eintragen d. Identifiers (0x1400ff bzw. 0x1800ff Subindex 1).

**Dummy-Mapping**

Ein weiteres Feature von CANopen ist das Mappen von Platzhaltern (Dummy-Einträgen). Als Platzhalter dienen die im Objektverzeichnis hinterlegten Datentyp-Einträge, die ja selbst nicht mit Daten versehen sind. Sind solche Einträge in der Mapping-Tabelle enthalten, so werden die entsprechenden Daten vom Gerät

nicht ausgewertet. Auf diese Art können beispielsweise mehrere Antriebe über ein einziges CAN-Telegramm mit neuen Sollwerten versorgt werden oder Ausgänge auf mehreren Knoten auch im ereignisgesteuerten Modus gleichzeitig gesetzt werden.

## 6.2.3 PDO-Parametrierung

Auch wenn die meisten CANopen-Netze in der Default-Einstellung und damit mit minimalem Konfigurationsaufwand zufrieden stellend arbeiten, so sollte zumindest überprüft werden, ob die vorhandene Buslast vertretbar ist. 80% Busauslastung mag für ein rein zyklisch synchron arbeitendes Netzwerk akzeptabel sein, für ein rein ereignisgesteuertes Netz ist dieser Wert in der Regel zu hoch, da kaum Bandbreite für zusätzliche Ereignisse zur Verfügung steht.

### Applikationsanforderungen berücksichtigen

Die Prozessdatenkommunikation sollte hinsichtlich einiger sich teilweise widersprechender Applikationsanforderungen optimiert werden. Hierzu gehören

- Geringer Parametrierungsaufwand - optimal sind brauchbare Default-Werte
- Garantierte Reaktionszeit auf bestimmte Ereignisse
- Zykluszeit bei Regelvorgängen über den Bus
- Sicherheitsreserven für Busstörungen (genügend Bandbreite für Nachrichtenwiederholung)
- Maximale Baud-Rate - hängt von der maximalen Buslänge ab
- Gewünschte Kommunikationspfade - wer spricht mit wem

Der bestimmende Faktor ist meist die zur Verfügung stehende Busbandbreite (Buslast).

### Baud-Rate

Allgemein wird man beginnen, die Baud-Rate so groß zu wählen, wie es die Buslänge erlaubt. Hierbei sollte man berücksichtigen, dass serielle Bussysteme grundsätzlich um so empfindlicher auf Störeinflüsse reagieren, je höher die Baud-Rate ist. Es gilt also die Regel: so schnell wie nötig. 1000 kBit/s sind meist nicht erforderlich und uneingeschränkt nur bei Netzwerken innerhalb eines Schaltschranks ohne galvanische Trennung der Busknoten empfehlenswert. Die Erfahrung zeigt auch, dass das Abschätzen der verlegten Buskabelänge häufig zu optimistisch erfolgt - die tatsächliche Kabelänge also größer ist.

### Kommunikationsart bestimmen

Ist die Baud-Rate gewählt, so gilt es nun die PDO-Kommunikationsart(en) zu bestimmen. Diese haben unterschiedliche Vor- und Nachteile:

- Die zyklisch synchrone Kommunikation ergibt eine genau vorhersagbare Busbelastung und damit ein definiertes Zeitverhalten - man könnte auch sagen, der worst case ist Standard. Sie ist einfach zu konfigurieren: mit dem Parameter SYNC-Rate kann die Buslast global eingestellt werden. Die Prozessabbilder werden synchronisiert: Eingänge werden gleichzeitig gelesen, Ausgangsdaten gleichzeitig gültig gesetzt - die Qualität dieser Synchronisierung ist allerdings implementierungsabhängig. Die BECKHOFF PC-Karten FC510x / CANopen-Klemme EL6751 sind in der Lage, das CANopen Bussystems mit den Zyklen der Anwendungsprogramme (SPS bzw. NC) zu synchronisieren.

Die garantierte Reaktionszeit ist bei der zyklisch synchronen Kommunikation immer mindestens so groß wie die Zykluszeit, und die Busbandbreite wird nicht optimal genutzt, da auch alte, sich nicht ändernde Daten ständig übertragen werden. Es ist aber möglich, das Netz durch die Wahl unterschiedlicher SYNC-Vielfacher (Transmission Types 1...240) zu optimieren und sich langsam ändernde Daten seltener zu übertragen als z. B. zeitkritische Eingänge. Berücksichtigt werden sollte jedoch, dass Eingangszustände, die kürzer anstehen als die Zykluszeit, nicht unbedingt kommuniziert werden. Ist dies gefordert, so sollten die entsprechenden PDOs für asynchrone Kommunikation vorgesehen werden.

- Die ereignisgesteuerte, asynchrone Kommunikation ist optimal hinsichtlich Reaktionszeit und Verwendung der Busbandbreite - man könnte sie als "CAN pur" bezeichnen. Bei ihrer Wahl muss allerdings berücksichtigt werden, dass unter Umständen viele Ereignisse gleichzeitig auftreten und sich

dann entsprechende Verzögerungszeiten einstellen können, bis ein relativ niederprioreres PDO verschickt werden kann - eine seriöse Netzwerkplanung erfordert demnach eine worst-case Betrachtung. Auch muss, z. B. durch Verwendung der Inhibit Zeit [▶ 143], verhindert werden, dass ein sich ständig ändernder Eingang mit hoher PDO-Priorität den Bus blockiert (Fachbegriff: "babbling idiot"). Aus diesem Grund ist beispielsweise die Ereignissteuerung bei Analogeingängen im Geräteprofil per Default abgeschaltet und muss gezielt aktiviert werden. Über den Ablauf-Timer lassen sich Zeitfenster für die Sende-PDOs einstellen: Das Telegramm wird frühestens nach Ablauf der Inhibit-Zeit [▶ 143] und spätestens nach Verstreichen des Ablauf-Timers erneut gesendet.

- Parametriert wird die Kommunikationsart über den Transmission Type [▶ 143].

Es ist auch möglich, beide PDO Kommunikationsprinzipien zu kombinieren. So kann es beispielsweise sinnvoll sein, die Soll- und Istwerte einer Achsregelung zyklisch synchron auszutauschen, während Endschalter oder die mit Grenzwerten versehene Motortemperatur mit ereignisgesteuerten PDOs überwacht werden. So kombiniert man die Vorteile beider Prinzipien: Synchronität der Achskommunikation und kurze Reaktionszeit für Endschalter. Durch die dezentrale Grenzwertüberwachung wird trotz Ereignissteuerung vermieden, dass der Temperatur-Analogwert ständig zur Buslast beiträgt.

Im genannten Beispiel kann es auch sinnvoll sein, die Identifier-Verteilung gezielt zu beeinflussen, um den Buszugriff durch die Prioritätsverteilung zu optimieren: die höchste Priorität bekommt das PDO mit den Endschalterdaten, die niedrigste das mit den Temperaturwerten.

In aller Regel ist es aber nicht erforderlich, die Identifier-Verteilung anzupassen, um die Latenzzeit beim Buszugriff zu optimieren. Dagegen müssen die Identifier verändert werden, um eine masterlose Kommunikation zu ermöglichen (PDO Linking [▶ 143]). Im genannten Beispiel könnte je ein RxPDO der Achsen denselben Identifier wie das TxPDO des Endschalters zugewiesen bekommen und dadurch eine Veränderung des Eingangswertes verzögerungsfrei empfangen.

## Buslast bestimmen

In jedem Fall ist es sinnvoll, die Buslast zu bestimmen. Doch welche Buslastwerte sind zulässig bzw. sinnvoll? Unterscheiden sollte man zunächst den kurzfristigen Burst von Telegrammen, bei dem eine Anzahl CAN-Nachrichten direkt aufeinander folgt - kurzzeitig 100% Buslast. Das ist nur dann problematisch, wenn die dadurch ausgelöste Folge von Empfangsinterrupts auf den CAN-Knoten nicht mehr abgearbeitet werden kann, es also zu einem Datenüberlauf (CAN-Queue-Overrun) kommt. Das kann bei sehr hohen Baud-Raten (> 500 kBit/s) bei Knoten mit Software-Telegrammfilterung und relativ langsamen oder stark ausgelasteten Mikro-Controllern vorkommen, wenn z. B. eine direkte Folge von Remote Frames (diese enthalten keine Datenbytes und haben daher minimale Länge) auf dem Bus ist (bei 1 Mbit/s kann so alle 40 µs ein Interrupt erzeugt werden; Beispiel: ein NMT-Master sendet alle Guarding-Anforderungen direkt hintereinander). Durch geschickte Implementierung lässt sich das vermeiden, der Anwender sollte davon ausgehen können, dass von den Geräteanbietern hierfür Sorge getragen wurde. Ein Burst-Zustand ist z. B. direkt nach dem SYNC Telegramm völlig normal: vom SYNC getriggert versuchen alle synchron arbeitenden Knoten quasi gleichzeitig Ihre Daten zu senden, es finden viele Arbitrierungsvorgänge statt, die Telegramme sortieren sich nacheinander in der Reihenfolge ihrer Priorität auf den Bus. Das ist in der Regel unkritisch, da es sich hier um Telegramme mit einigen Datenbytes handelt und die Telegrammfolge damit zwar eine schnelle, aber überschaubare Folge von Empfangsinterrupts auf den CAN-Knoten auslöst.

Unter Buslast versteht man meist den gemittelten Wert über mehrere Primärzyklen, also z. B. das Mittel über 100-500 ms. CAN, und damit CANopen, ist zwar in der Lage, nahe 100% Buslast auf Dauer zu bewältigen, aber dann steht keine Bandbreite für eventuelle Wiederholungen bei Störeinflüssen, asynchrone Fehlermeldungen, Parametrierung etc. zur Verfügung. Selbstverständlich hat die vorherrschende Art der Kommunikation einen großen Einfluss auf die sinnvolle Buslast: ein komplett zyklisch synchron arbeitendes Netz befindet sich ja bereits nahe am worst case Zustand und kann daher mit Werten von 70-80% betrieben werden. Für ein rein ereignisgesteuertes Netz ist diese Zahl nur schwer anzugeben: es muss hier abgeschätzt werden, wie viele zusätzliche Ereignisse im Vergleich zum derzeitigen Anlagenzustand auftreten können und für wie lange das zu einem Burst führt - also wie lange die relativ niederpriorste Nachricht dann verzögert würde. Ist dieser Wert von der Applikation her zulässig, so ist die aktuelle Buslast akzeptabel. Als Näherungswert kann meist angenommen werden, dass ein ereignisgesteuertes Netz mit 30-40% Grundlast genügend Reserven für worst-case-Szenarien hat - diese Annahme macht aber eine sorgfältige Analyse nicht überflüssig, wenn Verzögerungen zu kritischen Anlagenzuständen führen können.

Die BECKHOFF CANopen-Master-Karten FC510x / CANopen-Masterklemme EL6751 zeigen die Buslast über den System Manager ein. Diese Variable kann auch in der SPS verarbeitet oder in der Visualisierung zur Anzeige gebracht werden.

Neben den Kommunikationsparametern ist natürlich die Datenbelegung der Prozessdatenobjekte entscheidend: das [PDO Mapping](#). [► 148]

## 6.2.4 Servicedatenobjekte (SDO)

Die im Objektverzeichnis aufgeführten Parameter werden über Servicedatenobjekte gelesen und beschrieben. Diese SDOs sind *Multiplexed Domains*, also Datenstrukturen beliebiger Größe, die mit einem Multiplexor (Adresse) versehen sind. Der Multiplexor besteht aus 16-Bit-Index und 8-Bit-Subindex, die die entsprechenden Einträge im Objektverzeichnis adressieren.

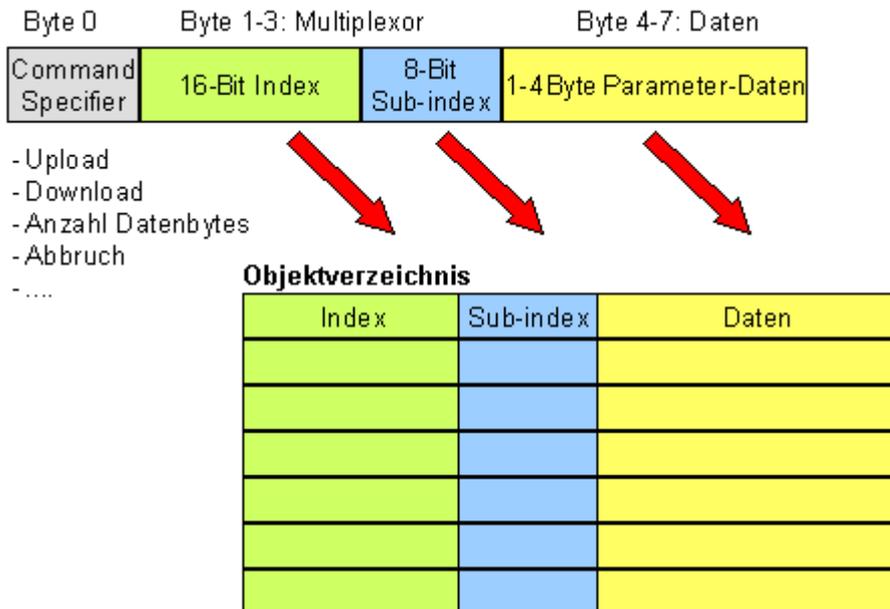


Abb. 139: SDO-Protokoll: Zugriff auf Objektverzeichnis

Die CANopen Buskoppler sind Server für das SDO, d.h. sie stellen auf Anforderung des Clients (z. B. des IPCs oder der SPS) Daten zur Verfügung (Upload) oder sie empfangen Daten vom Client (Download). Dabei findet ein Handshake zwischen Client und Server statt.

Wenn der zu übertragende Parameter bis zu 4 Bytes umfasst, genügt ein einziger Handshake (ein Telegrammpaar): Beim Download sendet der Client die Daten zusammen mit Index, Subindex und der Server bestätigt den Erhalt. Beim Upload fordert der Client die Daten an indem er Index und Subindex des gewünschten Parameters überträgt, und der Server sendet den Parameter (incl. Index und Subindex) in seinem Antworttelegramm.

Für Upload und Download wird das gleiche Identifier-Paar verwendet. In den stets 8 Byte großen Telegrammen sind im ersten Datenbyte die unterschiedlichen Dienste codiert. Bis auf die Objekte 1008h, 1009h und 100Ah (Gerätename, Hardware- bzw. Softwareversion) sind alle Parameter der Buskoppler nur bis zu 4 Byte groß, daher beschränkt sich diese Beschreibung auf die Übertragung dieser Daten im beschleunigten Transfer (Expedited Transfer).

### Protokoll

Im Folgenden wird der Aufbau der SDO-Telegramme beschrieben.

#### Client -> Server, Upload Request

11-bit Identifier	8 Byte Nutzdaten							
0x600 (=1536dez) + Node-ID	0x40	Index0	Index1	SubIdx	0x00	0x00	0x00	0x00

Parameter	Erläuterung
Index0	Index Low-Byte (Unsigned16, LSB)
Index1	Index High-Byte (Unsigned16, MSB)
SubIdx	Subindex (Unsigned8)

**Client -> Server, Upload Response**

11-bit Identifier	8 Byte Nutzdaten							
0x580 (=1408dez) + Node-ID	0x4x	Index0	Index1	SubIdx	Data0	Data1	Data2	Data3

Parameter	Erläuterung
Index0	Index Low-Byte (Unsigned16, LSB)
Index1	Index High-Byte (Unsigned16, MSB)
SubIdx	Subindex (Unsigned8)
Data0	Daten Low-Low-Byte (LLSB)
Data3	Daten High-High-Byte (MMSB)

Parameter des Datentyps Unsigned8 werden im Byte D0 übertragen, Parameter des Typs Unsigned16 in D0 und D1.

Die Anzahl der gültigen Datenbytes ist im ersten CAN-Datenbyte (0x4x) wie folgt codiert:

Anzahl Parameter-Bytes	1	2	3	4
Erstes CAN-Datenbyte	0x4F	0x4B	0x47	0x43

**Client -> Server, Download Request**

11-bit Identifier	8 Byte Nutzdaten							
0x600 (=1536dez) + Node-ID	0x22	Index0	Index1	SubIdx	Data0	Data1	Data2	Data3

Parameter	Erläuterung
Index0	Index Low-Byte (Unsigned16, LSB)
Index1	Index High-Byte (Unsigned16, MSB)
SubIdx	Subindex (Unsigned8)
Data0	Daten Low-Low-Byte (LLSB)
Data3	Daten High-High-Byte (MMSB)

Optional ist es möglich, im ersten CAN-Datenbyte die Anzahl der gültigen Parameter-Datenbytes anzugeben

Anzahl Parameter-Bytes	1	2	3	4
Erstes CAN-Datenbyte	0x2F	0x2B	0x27	0x23

In der Regel ist das jedoch nicht erforderlich, da jeweils nur die niederwertigen Datenbytes bis zur Länge des zu beschreibenden Objektverzeichniseintrags ausgewertet werden. Ein Download von Daten bis zu 4 Byte Länge kann daher bei BECKHOFF Busknoten immer mit 22 h im ersten CAN-Datenbyte erfolgen.

**Client -> Server, Download Response**

11-bit Identifier	8 Byte Nutzdaten							
0x580 (=1408dez) + Node-ID	0x60	Index0	Index1	SubIdx	0x00	0x00	0x00	0x00

Parameter	Erläuterung
Index0	Index Low-Byte (Unsigned16, LSB)
Index1	Index High-Byte (Unsigned16, MSB)
SubIdx	Subindex (Unsigned8)

### Abbruch Parameterkommunikation

Im Falle einer fehlerhaften Parameterkommunikation wird diese abgebrochen. Client bzw. Server senden dazu ein SDO-Telegramm folgender Struktur:

11-bit Identifier	8 Byte Nutzdaten							
0x580 (Client) oder 0x600(Server) + Node-ID	0x80	Index0	Index1	SubIdx	Error0	Error1	Error2	Error3

Parameter	Erläuterung
Index0	Index Low-Byte (Unsigned16, LSB)
Index1	Index High-Byte (Unsigned16, MSB)
SubIdx	Subindex (Unsigned8)
Error0	SDO Fehler-Code Low-Low-Byte (LLSB)
Error3	SDO Fehler-Code High-High-Byte (MMSB)

Liste der SDO-Fehler-Codes (Abbruch-Grund des SDO-Transfers):

SDO Fehler-Code	Erläuterung
0x05 03 00 00	Toggle Bit nicht geändert
0x05 04 00 01	SDO Command Specifier ungültig oder unbekannt
0x06 01 00 00	Zugriff auf dieses Objekt wird nicht unterstützt
0x06 01 00 02	Versuch, auf einen Read_Only Parameter zu schreiben
0x06 02 00 00	Objekt nicht im Objektverzeichnis vorhanden
0x06 04 00 41	Objekt kann nicht ins PDO gemappt werden
0x06 04 00 42	Anzahl und/oder Länge der gemappten Objekte würde PDO Länge überschreiten
0x06 04 00 43	Allgemeine Parameter Inkompatibilität
0x06 04 00 47	Allgemeiner interner Fehler im Gerät
0x06 06 00 00	Zugriff wegen Hardware-Fehler abgebrochen
0x06 07 00 10	Datentyp oder Parameterlänge stimmen nicht überein oder sind unbekannt
0x06 07 00 12	Datentyp stimmt nicht überein, Parameterlänge zu groß
0x06 07 00 13	Datentyp stimmt nicht überein, Parameterlänge zu klein
0x06 09 00 11	Subindex nicht vorhanden
0x06 09 00 30	allgemeiner Wertebereich-Fehler
0x06 09 00 31	Wertebereich-Fehler: Parameter wert zu groß
0x06 09 00 32	Wertebereich-Fehler: Parameter wert zu klein
0x06 0A 00 23	Resource nicht verfügbar
0x08 00 00 21	Zugriff wegen lokaler Applikation nicht möglich
0x08 00 00 22	Zugriff wegen aktuellem Gerätestatus nicht möglich

Für die Register-Kommunikation (Index 0x4500, 0x4501) wurden weitere, herstellerspezifische Fehler-Codes eingeführt:

<b>SDO Fehler-Code</b>	<b>Erläuterung</b>
0x06 02 00 11	ungültige Tabelle: Tabelle oder Kanal nicht vorhanden
0x06 02 00 10	ungültiges Register: Tabelle nicht vorhanden
0x06 01 00 22	Schreibschutz noch gesetzt
0x06 07 00 43	fehlerhafte Anzahl Funktionsargumente
0x06 01 00 21	Funktion noch aktiv, später erneut versuchen
0x05 04 00 40	Allgemeiner Routing Fehler
0x06 06 00 21	Fehler Zugriff BC Tabelle
0x06 09 00 10	Allgemeiner Fehler bei Kommunikation mit Klemme
0x05 04 00 47	Time-out bei Kommunikation mit Klemme

## 6.2.5 Identifier-Verteilung

### Default Identifier

CANopen sieht für die wichtigsten Kommunikationsobjekte Default Identifier vor, die aus der 7-Bit Knotenadresse (Node-ID) und einem 4-Bit Function-Code nach folgendem Schema abgeleitet werden:

11 Bit Identifier

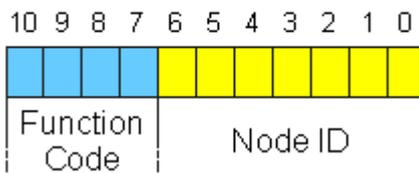


Abb. 140: Default Identifier

Für die Broadcast-Objekte wird die Node-ID 0 eingesetzt. Damit ergeben sich folgende Default Identifier:

### Broadcast-Objekte

Objekt	Funktion	Function Code	resultierende COB ID		Objekt für Comm. Parameter / Mapping
			hex	dez	
NMT	Boot-Up	0	0x00	0	- / -
SYNC	Synchronisation	1	0x80	128	0x1005 [▶ 162]+0x1006 [▶ 162] / -

### Peer-to-Peer-Objekte

Objekt	Funktion	Function Code	resultierende COB ID		Objekt für Comm. Parameter / Mapping
			hex	dez	
Emergency	Status / Fehler	1	0x81 - 0xFF	129 - 255	- / -
PDO1 (tx)	dig. Eingänge	11	0x181 - 0x1FF	385 - 511	0x1800 [▶ 168] / 0x1A00 [▶ 170]
PDO1 (rx)	digitale Ausgänge	100	0x201 - 0x27F	513 - 639	0x1400 [▶ 165] / 0x1600 [▶ 167]
PDO2 (tx)	analoge Eingänge	101	0x281 - 0x2FF	641 - 767	0x1801 [▶ 168] / 0x1A01 [▶ 170]
PDO2 (rx)	analoge Ausgänge	110	0x301 - 0x37F	769 - 895	0x1401 [▶ 166] / 0x1601 [▶ 167]
PDO3 (tx)	analoge Eingänge*	111	0x381 - 0x3FF	897 - 1023	0x1802 [▶ 169] / 0x1A02 [▶ 170]
PDO3 (rx)	analoge Ausgänge*	1000	0x401 - 0x47F	1025 - 1151	0x1402 [▶ 166] / 0x1602 [▶ 168]
PDO4 (tx)	analoge Eingänge*	1001	0x481 - 0x4FF	1153 - 1279	0x1803 [▶ 169] / 0x1A03 [▶ 170]
PDO4 (rx)	analoge Ausgänge*	1010	0x501 - 0x57F	1281 - 1407	0x1403 [▶ 166] / 0x1603 [▶ 168]
SDO (tx)	Parameter	1011	0x581 - 0x5FF	1409 - 1535	- / -
SDO (rx)	Parameter	1100	0x601 - 0x67F	1537 - 1663	- / -
Guarding	Life-/Node-guarding, Heartbeat, Boot-Up Nachricht	1110	0x701 - 0x77F	1793 - 1919	(0x100C [▶ 163], 0x100D [▶ 163], 0x100E [▶ 163], 0x1016 [▶ 164], 0x1017 [▶ 165])

\*) Für PDO3 und PDO4 gilt das Beckhoff Default Mapping. In den meisten Konfigurationen enthalten PDO 3+4 Daten von analogen Ein/Ausgängen, es können jedoch auch "überzählige" Daten von digitalen E/As oder Daten von Sonderklemmen sein. Details finden Sie im Abschnitt [PDO Mapping \[▶ 143\]](#).

Bis zur CANopen-Spezifikation Version 3 waren jeweils 2 PDOs mit Default-Identifiern versehen. Die BECKHOFF Buskoppler bis Firmwarestand BA entsprechen diesem Stand der Spezifikation. Ab Firmwarestand C0 (CANopen Version 4) sind Default Identifier für bis zu 4 PDOs vorgesehen.

**Herstellerspezifische Default Identifier für zusätzliche PDOs**

Default Identifier für zusätzliche PDOs

Den zusätzlichen PDOs, die von den Beckhoff Buskopplern nach dem Standardschema befüllt werden, wird kein Identifier zugeordnet. Der Anwender muss in das Objektverzeichnis einen Identifier für diese PDOs eintragen. Einfacher ist es, die belegten PDOs über das Objekt [0x5500](#) [► 179] aktivieren.

Über diesen Eintrag im Objektverzeichnis wird die Default-Identifier-Verteilung auf bis zu 11 PDOs ausgedehnt. Es ergeben sich folgende Identifier:

Objekt	Function Code	resultierende COB ID (hex)	resultierende COB ID (dez)
PDO5 (tx)	1101	0x681 - 0x6BF	1665 - 1727
PDO5 (rx)	1111	0x781 - 0x7BF	1921- 1983
PDO6 (tx)	111	0x1C1 - 0x1FF	449 - 511
PDO6 (rx)	1001	0x241 - 0x27F	577 - 639
PDO7 (tx)	1011	0x2C1 - 0x2FF	705 - 767
PDO7 (rx)	1101	0x341 - 0x37F	833 - 895
PDO8 (tx)	1111	0x3C1- 0x3FF	961 - 1023
PDO8 (rx)	10001	0x441 - 0x47F	1089 - 1151
PDO9 (tx)	10011	0x4C1 - 0x4FF	1217 - 1279
PDO9 (rx)	10101	0x541 - 0x57F	1345 - 1407
PDO10 (tx)	10111	0x5C1 - 0x5FF	1473 - 1535
PDO10 (rx)	11001	0x641 - 0x67F	1601- 1663
PDO11 (tx)	11011	0x6C1 - 0x6FF	1729 - 1791
PDO11 (rx)	11101	0x741 - 0x77F	1857 - 1919

<b>HINWEIS</b>
<p><b>Index 0x5500</b></p> <p>Der Index 0x5500 darf nicht genutzt werden, wenn Buskoppler mit mehr als 5 PDOs in Netzen mit Knotennummern über 64 vorhanden sind, da es sonst zu Identifier-Überschneidungen kommen kann.</p>

## 6.3 Objektverzeichnis

### 6.3.1 Objektverzeichnis - Struktur

Im CANopen-Objektverzeichnis werden alle für den Buskoppler relevanten CANopen-Objekte eingetragen. Das Objektverzeichnis ist in drei verschiedene Bereiche aufgeteilt:

1. Kommunikationsspezifischer Profilbereich (Index 0x1000 - 0x1FFF).  
Enthält die Beschreibung aller spezifischen Parameter für die Kommunikation.
2. Herstellerspezifischer Profilbereich (Index 0x2000 - 0x5FFF).  
Enthält die Beschreibung herstellerspezifischer Einträge.
3. Standardisierter Geräteprofilbereich (0x6000 - 0x9FFF).  
Enthält die Objekte für das Geräteprofil nach DS-401.

Jeder Eintrag im Objektverzeichnis ist durch einen 16-Bit-Index gekennzeichnet. Falls ein Objekt aus mehreren Komponenten besteht (z. B. Objekttyp Array oder Record), sind die Komponenten über einen 8-Bit-Subindex gekennzeichnet. Der Objektname beschreibt die Funktion eines Objekts, das Datentyp-Attribut spezifiziert den Datentyp des Eintrags. Über das Zugriffsattribut ist spezifiziert, ob ein Eintrag nur gelesen werden kann, nur geschrieben werden oder gelesen und geschrieben werden darf.

#### Kommunikationsspezifischer Bereich

In diesem Bereich des Objektverzeichnisses stehen alle für die Kommunikation des CANopen-Buskopplers notwendigen Parameter und Objekte. Im Bereich 0x1000 - 0x1018 stehen verschiedene, allgemeine kommunikationsspezifische Parameter (z. B. der Geräte name).

Die Kommunikationsparameter (z. B. Identifier) der Receive-PDOs stehen im Bereich 0x1400 - 0x140F (plus Subindex). Die Mapping-Parameter der Receive-PDOs stehen im Bereich von 0x1600 - 0x160F (plus Subindex). Die Mappingparameter enthalten die Querverweise auf die Applikationsobjekte, die in die PDOs gemappt sind und die Datenbreite des entsprechenden Objektes (siehe auch Abschnitt PDO-Mapping).

Die Kommunikations- und Mapping-Parameter der Transmit-PDOs stehen in den Bereichen 0x1800 - 0x180F bzw. 0x1A00 - 0x1A0F.

#### Herstellerspezifischer Bereich

In diesem Bereich finden sich Einträge, die BECKHOFF spezifisch sind, z. B.:

- Datenobjekte für Sonderklemmen
- Objekte für die Register-Kommunikation, über die auf alle internen Register der Buskoppler und Busklemmen zugegriffen werden kann.
- Objekte für die vereinfachte Konfiguration der PDOs

#### Standardisierter Geräteprofilbereich

Im Standardisierten Geräteprofilbereich wird das CANopen-Geräteprofil DS-401 Version 1 unterstützt. Für Analogeingänge stehen dabei Funktionen zur Verfügung, um die Kommunikation in der ereignisgesteuerten Betriebsart an die Applikationsanforderungen anzupassen und die Buslast zu minimieren:

- Grenzwertüberwachung
- Deltafunktion
- Ereignissteuerung aktivieren / deaktivieren

### 6.3.2 Objektliste

**i Erreichbarkeit der Objekte und Register**

Die Objekte aus dem Objektverzeichnis sind per SDO-Zugriff, jedoch nicht generell über die Konfigurations-Software KS2000 erreichbar. Dagegen sind alle Register, die per KS2000 konfiguriert werden können, auch per SDO-Zugriff auf das Objektverzeichnis (Objekte 0x4500 und 0x4501) erreichbar - wenn auch nicht mit dem gleichen Bedienungskomfort wie mit der Konfigurations-Software KS2000.

Parameter	Index	BK5120/BK515x	BK5110	LC5100	BX5100/BC5150	CX8051/B510
<a href="#">Gerätetyp [▶ 161]</a>	0x1000	x	x	x		x
<a href="#">Fehlerregister [▶ 161]</a>	0x1001	x	x	x	x	x *
<a href="#">Fehlerspeicher [▶ 161]</a>	0x1003	x	x	x		x *
<a href="#">Sync Identifier [▶ 162]</a>	0x1005	x	x	x	x	x
<a href="#">Sync Intervall [▶ 162]</a>	0x1006	x	x	x	x	x
<a href="#">Gerätename [▶ 162]</a>	0x1008	x	x	x	x	x *
<a href="#">Hardware-Version [▶ 162]</a>	0x1009	x	x	x		x *
<a href="#">Software-Version [▶ 163]</a>	0x100A	x	x	x	x	x *
<a href="#">Knotennummer [▶ 163]</a>	0x100B	x	x	x		
<a href="#">Guard Time [▶ 163]</a>	0x100C	x	x	x	x	x
<a href="#">Life Time Factor [▶ 163]</a>	0x100D	x	x	x	x	x
<a href="#">Guarding Identifier [▶ 163]</a>	0x100E	x	x	x		
<a href="#">Parameter speichern [▶ 163]</a>	0x1010	x	x	x		
<a href="#">Default-Werte laden [▶ 164]</a>	0x1011	x	x	x		
<a href="#">Emergency Identifier [▶ 164]</a>	0x1014	x	x	x		
<a href="#">Consumer Heartbeat Time [▶ 164]</a>	0x1016	x	x	x	x	x
<a href="#">Producer Heartbeat Time [▶ 165]</a>	0x1017	x	x	x	x	x
<a href="#">Geräteerkennung (Identity Object) [▶ 165]</a>	0x1018	x	x	x	x	x *
<a href="#">Server SDO Parameter [▶ 165]</a>	0x1200	x	x	x		
<a href="#">Kommunikationsparameter 1.-5. RxPDO [▶ 165]</a>	0x1400 - 0x1404	x	x	x	x	x
<a href="#">Kommunikationsparameter 6.-16. RxPDO [▶ 166]</a>	0x1405 - 0x140F	x			x	x
<a href="#">Kommunikationsparameter 17.-32. RxPDO [▶ 166]</a>	0x1410 - 0x141F				x nur BX5100	x
<a href="#">Mapping 1.-5. RxPDO [▶ 167]</a>	0x1600 - 0x1604	x	x	x	x	x
<a href="#">Mapping 6. -16. RxPDO [▶ 168]</a>	0x1605 - 0x160F	x			x	x
<a href="#">Mapping 17. -32. RxPDO [▶ 168]</a>	0x1610 - 0x161F				x nur BX5100	x
<a href="#">Kommunikationsparameter 1.-5. TxPDO [▶ 168]</a>	0x1800 - 0x1804	x	x	x	x	x
<a href="#">Kommunikationsparameter 6.-16. TxPDO [▶ 169]</a>	0x1805 - 0x180F	x			x	x
<a href="#">Kommunikationsparameter 17.-32. TxPDO [▶ 169]</a>	0x1810 - 0x181F				x nur BX5100	x
<a href="#">Mapping 1.-5. TxPDO [▶ 170]</a>	0x1A00 - 0x1A04	x	x	x	x	x
<a href="#">Mapping 6. -16. TxPDO [▶ 170]</a>	0x1A05 - 0x1A0F	x			x	x
<a href="#">Mapping 17. -32. TxPDO [▶ 170]</a>	0x1A10 - 0x1A1F				x nur BX5100	x

Parameter	Index	BK5120/BK515x	BK5110	LC5100	BX5100/BC5150	CX8051/B510
Merkerbereich %MB0-511	0x2F00				x	
Merkerbereich %MB511-1023	0x2F01				x	
Merkerbereich %MB1024-1535	0x2F02				x	
Merkerbereich %MB1536-2047	0x2F03				x	
Merkerbereich %MB2048-2559	0x2F04				x	
Merkerbereich %MB2560-3071	0x2F05				x	
Merkerbereich %MB3072-3584	0x2F06				x	
Merkerbereich %MB3585-4095	0x2F07				x	
3-Byte Sonderklemmen, Eingangsdaten <a href="#">[► 171]</a>	0x2600	x				
3-Byte Sonderklemmen, Ausgangsdaten <a href="#">[► 171]</a>	0x2700	x				
4-Byte Sonderklemmen, Eingangsdaten <a href="#">[► 171]</a>	0x2800	x				
4-Byte Sonderklemmen, Ausgangsdaten <a href="#">[► 172]</a>	0x2900	x				
5-Byte Sonderklemmen, Eingangsdaten <a href="#">[► 172]</a>	0x2A00	x				
5-Byte Sonderklemmen, Ausgangsdaten <a href="#">[► 172]</a>	0x2B00	x				
6-Byte Sonderklemmen, Eingangsdaten <a href="#">[► 172]</a>	0x2C00	x				
6-Byte Sonderklemmen, Ausgangsdaten <a href="#">[► 172]</a>	0x2D00	x				
8-Byte Sonderklemmen, Eingangsdaten <a href="#">[► 173]</a>	0x3000	x				
8-Byte Sonderklemmen, Ausgangsdaten <a href="#">[► 173]</a>	0x3100	x				
Register-Kommunikation, Busknoten <a href="#">[► 173]</a>	0x4500	x	x	x		
Register-Kommunikation, Busklemme/Erweiterungsbox <a href="#">[► 177]</a>	0x4501	x	x	x		
PDOs aktivieren <a href="#">[► 179]</a>	0x5500	x	x	x		
NetId <a href="#">[► 183]</a>	0x5FFE				x	
Digitale Eingänge <a href="#">[► 180]</a>	0x6000	x	x	x		
Interrupt-Maske <a href="#">[► 180]</a>	0x6126	x	x	x		
Digitale Ausgänge <a href="#">[► 181]</a>	0x6200	x	x	x		
Analoge Eingänge <a href="#">[► 181]</a>	0x6401	x				
Analoge Ausgänge <a href="#">[► 182]</a>	0x6411	x				
Ereignissteuerung, analoge Eingänge <a href="#">[► 182]</a>	0x6423	x				
Oberer Grenzwert, analoge Eingänge <a href="#">[► 182]</a>	0x6424	x				
Unterer Grenzwert, analoge Eingänge <a href="#">[► 182]</a>	0x6425	x				
Deltafunktion, analoge Eingänge <a href="#">[► 182]</a>	0x6426	x				

\*) Wird ein ADS-Server angemeldet, werden diese Objekte per ADS-Notification an die SPS weitergeleitet und müssen dort beantwortet werden.

### 6.3.3 Objekte und Daten

#### Gerätetyp

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1000	0	Device Type	Unsigned32	ro	N	0x00000000	Angabe des Gerätetyps

Der 32 Bit-Wert ist in zwei 16 Bit-Felder unterteilt:

MSB	LSB
Additional Information	Geräteprofil-Nummer
0000 0000 0000 wxyz	0x191 (401 <sub>dez</sub> )

Die *Additional Information* enthält Angaben über die Signalarten des E/A-Gerätes:

z=1 bedeutet digitale Eingänge,

y=1 bedeutet digitale Ausgänge,

x=1 bedeutet analoge Eingänge,

w=1 bedeutet analoge Ausgänge.

Ein BK5120 mit digitalen und analogen Eingängen, aber ohne Ausgänge, liefert also 0x00 05 01 91 zurück.

Sonderklemmen (z. B. serielle Schnittstellen, PWM-Ausgänge, Inkremental-Encoder-Eingänge) werden nicht berücksichtigt. Ein Koppler, der z. B. nur serielle Schnittstellenklemmen KL6001 bestückt hat, liefert also 0x00 00 01 91 zurück.

Der Gerätetyp liefert nur eine grobe Klassifizierung des Gerätes. Für die detaillierte Identifizierung des Buskopplers und der angesteckten Klemmen kann das Klemmenbezeichnungs-Register des Buskopplers gelesen werden (Details siehe Register-Kommunikation Index 0x4500).

#### Fehlerregister

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1001	0	Error Register	Unsigned8	ro	N	0x00	Fehlerregister

Der 8 Bit-Wert ist wie folgt kodiert:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ManSpec.	reserviert	reserviert	Comm.	reserviert	reserviert	reserviert	Generic

ManSpec.: Herstellerspezifischer Fehler, wird in Objekt 1003 genauer spezifiziert.

Comm.: Kommunikationsfehler (Overrun CAN)

Generic: Ein nicht näher spezifizierter Fehler ist aufgetreten (Flag ist bei jeder Fehlermeldung gesetzt)

#### Fehlerspeicher

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1003	0x00	Predefined error field (Fehlerspeicher)	Unsigned8	rw	N	0x00	Objekt 1003h enthält eine Beschreibung der im Gerät aufgetretenen Fehler - Subindex 0 die Anzahl der gespeicherten Fehlerzustände.
	1	Actual error	Unsigned32	ro	N	Keiner	Letzter aufgetretener Fehlerzustand
	...	...	...	--	...	...	...
	10	Standard error field	Unsigned32	ro	N	Keiner	Es werden maximal 10 Fehlerzustände gespeichert.

Der 32 Bit-Wert im Fehlerspeicher ist in zwei 16 Bit-Felder unterteilt:

MSB	LSB
Additional Code	Error Code

Der Additional Code enthält den Error Trigger (siehe Emergency-Objekt) und damit eine detaillierte Fehlerbeschreibung.

Neue Fehler werden jeweils an Subindex 1 gespeichert, alle anderen Sub-indices werden entsprechend inkrementiert. Durch Schreiben einer 0 auf Subindex 0 wird der gesamte Fehlerspeicher gelöscht.

Wenn kein Fehler seit dem Power-On aufgetreten ist, dann besteht Objekt 0x1003 nur aus Subindex 0 mit eingetragener 0. Durch einen Reset oder Power Cycle wird der Fehlerspeicher gelöscht.

Wie bei CANopen üblich wird das LSB zuerst und das MSB zuletzt übertragen.

### Sync Identifier

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1005	0	COB-ID Sync Message	Unsigned32	rw	N	0x80000080	Identifier der SYNC-Nachricht

Die unteren 11 Bit des 32-Bit Wertes enthalten den Identifier (0x80=128dez). Bit 30 gibt Auskunft, ob das Gerät das SYNC-Telegramm sendet (1) oder nicht (0). Die CANopen E/A Geräte empfangen das SYNC Telegramm, dementsprechend ist Bit 30=0. Bit 31 ist aus Gründen der Abwärtskompatibilität ohne Bedeutung.

### Sync Intervall

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1006	0	Communication cycle period	Unsigned32	rw	N	0x00000000	Länge des SYNC-Intervalls in $\mu$ s.

Wenn hier ein Wert ungleich Null eingetragen wird, so geht der Busknoten in den Fehlerzustand, wenn beim synchronen PDO-Betrieb innerhalb der Watchdog-Zeit kein SYNC-Telegramm empfangen wurde. Die Watchdog- Zeit entspricht hierbei dem 1,5-fachen der eingestellten communication cycle period - es kann also der vorgesehene SYNC-Abstand eingetragen werden.

Das E/A Update wird bei den Beckhoff CANopen Busknoten direkt nach Empfang des SYNC Telegramms durchgeführt, wenn folgende Voraussetzungen gegeben sind:

- Firmwarestand ab C0 (ab CANopen Version 4.01).
- alle PDOs, die über Daten verfügen, auf die synchrone Kommunikationsart eingestellt (0...240).
- Sync Intervall in Objekt 0x1006 eingetragen und (Sync Intervall x kleinste PDO Übertragungsart) kleiner als 90 ms.

Die Baugruppen sind dann durchsynchronisiert.

### Gerätename

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1008	0	Manufacturer Device Name	Visible String	ro	N	BK51x0, LC5100, IPxxxx-B510 od. ILxxxx- B510	Gerätename des Busknotens

Da der zurück gelieferte Wert größer als 4 Bytes ist, wird das segmentierte SDO-Protokoll zur Übertragung verwendet.

### Hardware-Version

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1009	0	Manufacturer Hardware-Version	Visible String	ro	N	-	Hardwareversionsnummer des Busknotens

Da der zurück gelieferte Wert größer als 4 Bytes ist, wird das segmentierte SDO-Protokoll zur Übertragung verwendet.

**Software-Version**

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x100A	0	Manufacturer Software-Version	Visible String	ro	N	-	Softwareversionsnummer des Busknotens

Da der zurück gelieferte Wert größer als 4 Bytes ist, wird das segmentierte SDO-Protokoll zur Übertragung verwendet.

**Knotennummer**

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x100B	0	Node-ID	Unsigned32	ro	N	keiner	eingestellte Knotennummer

Die Knotennummer wird aus Kompatibilitätsgründen unterstützt.

**Guard Time**

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x100C	0	Guard Time [ms]	Unsigned16	rw	N	0	Abstand zwischen zwei Guard Telegrammen. wird durch NMT-Master oder Konfigurationstool eingestellt.

**Life Time Factor**

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x100D	0	Life Time Factor	Unsigned8	rw	N	0	Life Time Factor x Guard Time = Life Time (Watchdog für Life Guarding)

Wenn innerhalb der Life Time kein Guarding-Telegramm empfangen wurde, geht der Knoten in den Fehlerzustand. Wenn Life Time Factor und/oder Guard Time = 0 sind, so führt der Knoten kein Lifeguarding durch, kann aber dennoch vom Master überwacht werden (Node Guarding).

**Guarding Identifier**

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x100E	0	COB-ID guarding protocol	Unsigned32	ro	N	0x000007xy, xy = NodeID	Identifier des Guarding Protokolls

Der Guarding Identifier wird aus Kompatibilitätsgründen unterstützt. Seit CANopen Version 4 darf der Guarding Identifier nicht mehr verändert werden.

**Parameter speichern**

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1010	0	Store Parameter	Unsigned8	ro	N	1	Anzahl der Speicheroptionen
	1	store all parameters	Unsigned32	rw	N	1	Speichert alle (speicherbaren) Parameter

Durch Schreiben der Signatur save im ASCII-Code (hexadezimal 0x65766173) auf Subindex 1 werden die aktuellen Parameter nichtflüchtig gespeichert. (Bytefolge auf dem Bus incl. SDO Protokoll: 0x23 0x10 0x10 0x01 0x73 0x61 0x76 0x65).

Der Speichervorgang dauert ca. drei Sekunden, bei Erfolg wird anschließend durch das entsprechende TxSDO (0x60 im ersten Byte) bestätigt. Da der Buskoppler während des Speichervorgangs keine CAN-Telegramme senden und empfangen kann, kann nur gespeichert werden, wenn der Knoten im Zustand Pre-Operational ist. Es wird empfohlen, vor dem Abspeichern das gesamte Netz in den Zustand Pre-Operational zu versetzen. Dadurch wird ein Puffer-Überlauf vermieden.

Gespeichert werden:

- Die aktuelle Klemmenbestückung (Anzahl jeder Klemmenkategorie)
- Alle PDO Parameter (Identifier, Transmission Type, Inhibit Zeit, Mapping).

## ● Gültige Identifier

**i** Anschließend gelten die gespeicherten Identifier, nicht mehr die aus der Knotenadresse abgeleiteten Default-Identifier. Änderungen der DIP-Schalterstellung beeinflussen die PDOs dann nicht mehr!

- Alle SYNC Parameter
- Alle Guarding Parameter
- Grenzwerte, Deltawerte und Interrupt Enable für Analogeingänge

Die in den Klemmen über Register-Kommunikation direkt gespeicherten Parameter werden dort sofort nichtflüchtig gespeichert.

### Default-Werte laden

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1011	0	Restore Parameter	Unsigned8	ro	N	4	Anzahl der Rücksetze-Optionen
	1	Restore all parameters	Unsigned32	rw	N	1	Setzt alle Parameter auf Default-Werte zurück
	4	Set manufacturer Defaults	Unsigned32	rw	N	1	Setzt alle Koppler-Parameter auf Hersteller-Einstellungen zurück (auch Register)

Durch Schreiben der Signatur *load* im ASCII-Code (hexadezimal 0x64616F6C) auf Subindex 1 werden alle Parameter **beim nächsten Booten (Reset)** auf Default-Werte (Auslieferungszustand) zurückgesetzt.

(Bytefolge auf dem Bus incl. SDO Protokoll: 0x23 0x11 0x10 0x01 0x6C 0x6F 0x61 0x64).

Hierdurch werden die Default-Identifier für die PDOs wieder aktiv.

### Emergency Identifier

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1014	0	COB-ID Emergency	Unsigned32	rw	N	0x00000080, + NodeID	Identifier des Emergency-Telegramms

Die unteren 11 Bit des 32-Bit Wertes enthalten den Identifier (0x80=128dez). Über das MSBit lässt sich einstellen ob das Gerät das Emergency-Telegramm sendet (1) oder nicht (0).

Alternativ lässt sich die Diagnose-Funktion der Busknoten auch durch das Bit *Gerätediagnose* in der K-Buskonfiguration (siehe Objekt 0x4500) abschalten.

### Consumer Heartbeat Time

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1016	0	Anzahl Elemente	Unsigned8	ro	N	2	Die Consumer Heartbeat Time beschreibt die erwartete Heartbeat-Zykluszeit sowie die Node-ID des überwachten Knotens
	1	Consumer Heartbeat Time	Unsigned32	rw	N	0	Watchdog Zeit in ms und Node-ID des überwachten Knotens

Der 32 Bit-Wert wird wie folgt verwendet:

MSB		LSB	
Bit 31...24	Bit 23...16	Bit 15...0	
reserviert (0)	Node-ID (Unsigned8)	heartbeat time in ms (Unsigned16)	

Aus der Node-ID ergibt sich der überwachte Identifier durch die Default-Identifier-Verteilung: Guard-ID = 0x700 + Node-ID.

Wie bei CANopen üblich wird das LSB zuerst und das MSB zuletzt übertragen.

**Producer Heartbeat Time**

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1017	0	Producer Heartbeat Time	Unsigned16	rw	N	0	Zeitspanne in ms zwischen zwei gesendeten Heartbeat-Telegrammen

**Geräteerkennung (Identity Object)**

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1018	0	Identity Object: Anzahl Elemente	Unsigned8	ro	N	4	Das Identity Objekt enthält allgemeine Angaben zu Art und Ausgabestand des Gerätes.
	1	Vendor ID	Unsigned32	ro	N	0x00000002	Herstellerkennung. Beckhoff hat die Vendor-ID 2
	2	Product Code	Unsigned32	ro	N	abhängig vom Produkt	Geräteerkennung
	3	Revision Number	Unsigned32	ro	N	-	Versionsnummer
	4	Serial Number	Unsigned32	ro	N	-	Produktionsdatum Low-Wort, High-Byte: Kalenderwoche (dez), Low-Wort, Low-Byte: Kalenderjahr

Produkt	Product Code
BK5120	0x11400
BK5110	0x113F6
LC5100	0x113EC
IPwxyz-B510	0x2wxyz
IL2301-B510	0x2008FD

**Server SDO Parameter**

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1200	0	Anzahl Elemente	Unsigned8	ro	N	2	Kommunikationsparameter des Server SDOs. Subindex 0: Anzahl der folgenden Parameter
	1	COB-ID Client ->Server	Unsigned32	ro	N	0x000006xy, xy=Node-ID	COB-ID RxSDO (Client -> Server)
	2	COB-ID Server ->Client	Unsigned32	ro	N	0x00000580 + Node-ID	COB-ID TxSDO (Client -> Server)

Aus Gründen der Abwärtskompatibilität im Objektverzeichnis enthalten.

**Kommunikationsparameter1. RxPDO**

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1400	0	Anzahl Elemente	Unsigned8	ro	N	5	Kommunikationsparameter des ersten Receive-PDOs. Subindex 0: Anzahl der folgenden Parameter
	1	COB-ID	Unsigned32	rw	N	0x000002xy, xy=Node-ID	COB-ID (Communication Object Identifier) RxPDO1
	2	Transmission Type	Unsigned8	rw	N	255	Übertragungsart des PDOs
	3	Inhibit Time	Unsigned16	rw	N	0	Aus Gründen der Abwärtskompatibilität vorhanden, im RxPDO nicht genutzt.
	4	CMS Priority Group	Unsigned8	rw	N	-	Aus Gründen der Abwärtskompatibilität vorhanden, nicht genutzt.
	5	Event Timer	Unsigned16	rw	N	0	Event-Timer. Definiert Watchdog Zeit für Empfangsüberwachung des PDOs.

Subindex 1 (COB-ID): Die unteren 11 Bit des 32-Bit Wertes (Bits 0-10) enthalten den CAN-Identifizier, das MSBit (Bit 31) gibt Auskunft, ob das PDO aktuell existiert (0) oder nicht (1), Bit 30 teilt mit, ob ein RTR-Zugriff auf dieses PDO zulässig ist (0) oder nicht (1). Es ist nicht erlaubt, den Identifizier (Bit 0-10) zu ändern, während das Objekt existiert (Bit 31=0). Der Subindex 2 enthält die Übertragungsart (siehe Einführung PDOs).

**Kommunikationsparameter2. RxPDO**

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1401	0	Anzahl Elemente	Unsigned8	ro	N	5	Kommunikationsparameter des zweiten Receive-PDOs.
	1	COB-ID	Unsigned32	rw	N	0x000003xy, xy=Node-ID	COB-ID (Communication Object Identifier) RxPDO2
	2	Transmission Type	Unsigned8	rw	N	255	Übertragungsart des PDOs
	3	Inhibit Time	Unsigned16	rw	N	0	Aus Gründen der Abwärtskompatibilität vorhanden, im RxPDO nicht genutzt.
	4	CMS Priority Group	Unsigned8	rw	N	-	Aus Gründen der Abwärtskompatibilität vorhanden, nicht genutzt.
	5	Event Timer	Unsigned16	rw	N	0	Event-Timer. Definiert Watchdog Zeit für Empfangsüberwachung des PDOs.

**Kommunikationsparameter3. RxPDO**

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1402	0	Anzahl Elemente	Unsigned8	ro	N	5	Kommunikationsparameter des dritten Receive-PDOs.
	1	COB-ID	Unsigned32	rw	N	0x000004xy, xy=Node-ID	COB-ID (Communication Object Identifier) RxPDO3
	2	Transmission Type	Unsigned8	rw	N	255	Übertragungsart des PDOs
	3	Inhibit Time	Unsigned16	rw	N	0	Aus Gründen der Abwärtskompatibilität vorhanden, im RxPDO nicht genutzt.
	4	CMS Priority Group	Unsigned8	rw	N	-	Aus Gründen der Abwärtskompatibilität vorhanden, nicht genutzt.
	5	Event Timer	Unsigned16	rw	N	0	Event-Timer. Definiert Watchdog Zeit für Empfangsüberwachung des PDOs.

**Kommunikationsparameter4. RxPDO**

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1403	0	Anzahl Elemente	Unsigned8	ro	N	5	Kommunikationsparameter des vierten Receive-PDOs.
	1	COB-ID	Unsigned32	rw	N	0x000005xy, xy=Node-ID	COB-ID (Communication Object Identifier) RxPDO4
	2	Transmission Type	Unsigned8	rw	N	255	Übertragungsart des PDOs
	3	Inhibit Time	Unsigned16	rw	N	0	Aus Gründen der Abwärtskompatibilität vorhanden, im RxPDO nicht genutzt.
	4	CMS Priority Group	Unsigned8	rw	N	-	Aus Gründen der Abwärtskompatibilität vorhanden, nicht genutzt.
	5	Event Timer	Unsigned16	rw	N	0	Event-Timer. Definiert Watchdog Zeit für Empfangsüberwachung des PDOs.

**Kommunikationsparameter5.-16. RxPDO**

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1404 - 0x140F (je nach Geräte Typ)	0	Anzahl Elemente	Unsigned8	ro	N	5	Kommunikationsparameter des 5. bis 16. Receive-PDOs.
	1	COB-ID	Unsigned32	rw	N	0x8000000	COB-ID (Communication Object Identifier) RxPDO5...16
	2	Transmission Type	Unsigned8	rw	N	255	Übertragungsart des PDOs
	3	Inhibit Time	Unsigned16	rw	N	0	Aus Gründen der Abwärtskompatibilität vorhanden, im RxPDO nicht genutzt.
	4	CMS Priority Group	Unsigned8	rw	N	-	Aus Gründen der Abwärtskompatibilität vorhanden, nicht genutzt.
	5	Event Timer	Unsigned16	rw	N	0	Event-Timer. Definiert Watchdog Zeit für Empfangsüberwachung des PDOs.

Die Anzahl der RxPDOs je Busknoten-Typ kann den technischen Daten entnommen werden.

**Mapping-Parameter1. RxPDO**

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1600	0	Anzahl Elemente	Unsigned8	rw	N	abhängig von Typ und Bestückung	Mapping-Parameter des ersten Receive-PDOs; Subindex 0: Anzahl der gemappten Objekte.
	1	1. gemapptes Objekt	Unsigned32	rw	N	0x62000108	1. gemapptes Applikationsobjekt (2 Byte Index, 1 Byte Subindex, 1 Byte Bitbreite)
	2	2. gemapptes Objekt	Unsigned32	rw	N	0x62000208	2. gemapptes Applikationsobjekt (2 Byte Index, 1 Byte Subindex, 1 Byte Bitbreite)
	...	...	...	...	...	...	...
	8	8. gemapptes Objekt	Unsigned32	rw	N	0x62000808	8. gemapptes Applikationsobjekt (2 Byte Index, 1 Byte Subindex, 1 Byte Bitbreite)

Das erste Empfangs-PDO (RxPDO1) ist per Default für digitale Ausgangsdaten vorgesehen. Je nach Anzahl der bestückten Ausgänge wird automatisch die erforderliche Länge des PDOs bestimmt und die entsprechenden Objekte gemappt. Da die digitalen Ausgänge byteweise organisiert sind, kann die Länge des PDOs in Bytes direkt dem Subindex 0 entnommen werden.

**Mapping-Änderungen**

Um das Mapping zu verändern muss folgende Reihenfolge eingehalten werden (ab CANopen Version 4 vorgeschrieben):

1. PDO löschen (Bit 31 im Identifier-Eintrag (Subindex1) des Kommunikations-Parameters auf 1 setzen)
2. Mapping deaktivieren (Subindex 0 des Mapping Eintrages auf 0 setzen)
3. Mapping Einträge ändern (Subindices 1...8)
4. Mapping aktivieren (Subindex 0 des Mapping Eintrages auf die korrekte Anzahl der gemappten Objekte setzen)
5. PDO anlegen (Bit 31 im Identifier-Eintrag (Subindex 1) des Kommunikations-Parameters auf 0 setzen)

**Mapping-Parameter2. RxPDO**

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1601	0	Anzahl Elemente	Unsigned8	rw	N	abhängig von Typ und Bestückung	Mapping-Parameter des zweiten Receive-PDOs; Subindex 0: Anzahl der gemappten Objekte.
	1	1. gemapptes Objekt	Unsigned32	rw	N	0x64110110	1. gemapptes Applikationsobjekt (2 Byte Index, 1 Byte Subindex, 1 Byte Bitbreite)
	2	2. gemapptes Objekt	Unsigned32	rw	N	0x64110210	2. gemapptes Applikationsobjekt (2 Byte Index, 1 Byte Subindex, 1 Byte Bitbreite)
	...	...	...	...	...	...	...
	8	8. gemapptes Objekt	Unsigned32	rw	N	0x00000000	8. gemapptes Applikationsobjekt (2 Byte Index, 1 Byte Subindex, 1 Byte Bitbreite)

Das zweite Empfangs-PDO (RxPDO2) ist per Default für analoge Ausgänge vorgesehen. Je nach Anzahl der bestückten Ausgänge wird automatisch die erforderliche Länge des PDOs bestimmt und die entsprechenden Objekte gemappt. Da die analogen Ausgänge wortweise organisiert sind, kann die Länge des PDOs in Bytes direkt dem Subindex 0 entnommen werden.

Um das Mapping zu verändern muss eine bestimmte Reihenfolge eingehalten werden (siehe Objekt Index 0x1600).

### Mapping-Parameter3.-16. RxPDO

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1602-0x160F (je nach Geräte Typ)	0	Anzahl Elemente	Unsigned8	rw	N	abhängig von Typ und Bestückung	Mapping-Parameter des 3.-16. Receive-PDOs; Subindex 0: Anzahl der gemappten Objekte.
	1	1. gemapptes Objekt	Unsigned32	rw	N	0x00000000 (Siehe Text)	1. gemapptes Applikationsobjekt (2 Byte Index, 1 Byte Subindex, 1 Byte Bitbreite)
	2	2. gemapptes Objekt	Unsigned32	rw	N	0x00000000 (Siehe Text)	2. gemapptes Applikationsobjekt (2 Byte Index, 1 Byte Subindex, 1 Byte Bitbreite)
	...	...	...	...	...	...	...
	8	8. gemapptes Objekt	Unsigned32	rw	N	0x00000000 (Siehe Text)	8. gemapptes Applikationsobjekt (2 Byte Index, 1 Byte Subindex, 1 Byte Bitbreite)

Das 3. bis 16. Empfangs-PDO (RxPDO3ff) wird vom Busknoten je nach Klemmen-Bestückung (bzw. je nach Erweiterungs-Modulen) automatisch mit einem Default Mapping versehen. Die Vorgehensweise ist im Kapitel PDO-Mapping beschrieben.

Um das Mapping zu verändern muss eine bestimmte Reihenfolge eingehalten werden (siehe Objekt Index 0x1600).

#### **i** Beckhoff Default Mapping

DS401 V2 schreibt für die PDOs 3 und 4 als Default Mapping analoge Ein- bzw. Ausgangsdaten vor. Das entspricht dann dem Beckhoff Default Mapping, wenn weniger als 65 digitale Ein- bzw. Ausgänge vorhanden sind. Um die Abwärtskompatibilität zu gewährleisten wird das Beckhoff Default Mapping beibehalten - die Geräte entsprechen damit in ihrem Mapping-Verhalten DS401 V1, in allen anderen Belangen DS401 V2.

### Kommunikationsparameter1. TxPDO

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1800	0	Anzahl Elemente	Unsigned8	ro	N	5	Kommunikationsparameter des ersten SendepDOs. Subindex 0: Anzahl der folgenden Parameter
	1	COB-ID	Unsigned32	rw	N	0x00000180 + Node-ID	COB-ID (Communication Object Identifier) TxPDO1
	2	Transmission Type	Unsigned8	rw	N	255	Übertragungsart des PDOs
	3	Inhibit Time	Unsigned16	rw	N	0	Wiederholungsverzögerung [Wert x 100 µs]
	4	CMS Priority Group	Unsigned8	rw	N	-	Aus Gründen der Abwärtskompatibilität vorhanden, nicht genutzt.
	5	Event Timer	Unsigned16	rw	N	0	Event-Timer

Subindex 1 (COB-ID): Die unteren 11 Bit des 32-Bit Wertes (Bits 0-10) enthalten den CAN-Identifizier, das MSBit (Bit 31) gibt Auskunft, ob das PDO aktuell existiert (0) oder nicht (1), Bit 30 teilt mit, ob ein RTR-Zugriff auf dieses PDO zulässig ist (0) oder nicht (1). Es ist nicht erlaubt, den Identifizier (Bit 0-10) zu ändern, während das Objekt existiert (Bit 31=0). Der Subindex 2 enthält die Übertragungsart, Subindex 3 die Wiederholungsverzögerung zwischen zwei gleichen PDOs, Subindex 5 enthält den Event Timer. Subindex 4 ist aus Kompatibilitätsgründen vorhanden, wird aber nicht genutzt. (siehe auch Einführung PDOs).

### Kommunikationsparameter2. TxPDO

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1801	0	Anzahl Elemente	Unsigned8	ro	N	5	Kommunikationsparameter des zweiten SendepDOs. Subindex 0: Anzahl der folgenden Parameter
	1	COB-ID	Unsigned32	rw	N	0x00000280 + Node-ID	COB-ID (Communication Object Identifier) TxPDO1
	2	Transmission Type	Unsigned8	rw	N	255	Übertragungsart des PDOs
	3	Inhibit Time	Unsigned16	rw	N	0	Wiederholungsverzögerung [Wert x 100 µs]
	4	CMS Priority Group	Unsigned8	rw	N	-	Aus Gründen der Abwärtskompatibilität vorhanden, nicht genutzt.
	5	Event Timer	Unsigned16	rw	N	0	Event-Timer

Das zweite Sende-PDO ist per Default für analoge Eingänge vorgesehen und für ereignisgesteuerte Übertragung konfiguriert (Transmission Type 255). Die Ereignissteuerung muss zunächst aktiviert werden (siehe Objekt 0x6423), ansonsten können die Eingänge nur per Remote Transmission Request (RTR) abgefragt (gepollt) werden.

**Kommunikationsparameter3. TxPDO**

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1802	0	Anzahl Elemente	Unsigned8	ro	N	5	Kommunikationsparameter des dritten Sende-PDOs. Subindex 0: Anzahl der folgenden Parameter
	1	COB-ID	Unsigned32	rw	N	0x00000380 + Node-ID	COB-ID (Communication Object Identifier) TxPDO1
	2	Transmission Type	Unsigned8	rw	N	255	Übertragungsart des PDOs
	3	Inhibit Time	Unsigned16	rw	N	0	Wiederholungsverzögerung [Wert x 100 µs]
	4	CMS Priority Group	Unsigned8	rw	N	-	Aus Gründen der Abwärtskompatibilität vorhanden, nicht genutzt.
5	Event Timer	Unsigned16	rw	N	0	Event-Timer	

Das dritte Sende-PDO wird in der Regel analoge Eingangsdaten enthalten (siehe Mapping). Es ist für ereignisgesteuerte Übertragung konfiguriert (Transmission Type 255). Die Ereignissteuerung muss zunächst aktiviert werden (siehe Objekt 0x6423), ansonsten können die Eingänge nur per Remote Transmission Request (RTR) abgefragt (gepollt) werden.

**Kommunikationsparameter4. TxPDO**

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1803	0	Anzahl Elemente	Unsigned8	ro	N	5	Kommunikationsparameter des vierten Sende-PDOs. Subindex 0: Anzahl der folgenden Parameter
	1	COB-ID	Unsigned32	rw	N	0x00000480 + Node-ID	COB-ID (Communication Object Identifier) TxPDO1
	2	Transmission Type	Unsigned8	rw	N	255	Übertragungsart des PDOs
	3	Inhibit Time	Unsigned16	rw	N	0	Wiederholungsverzögerung [Wert x 100 µs]
	4	CMS Priority Group	Unsigned8	rw	N	-	Aus Gründen der Abwärtskompatibilität vorhanden, nicht genutzt.
5	Event Timer	Unsigned16	rw	N	0	Event-Timer	

Das vierte Sende-PDO wird in der Regel analoge Eingangsdaten enthalten (siehe Mapping). Es ist für ereignisgesteuerte Übertragung konfiguriert (Transmission Type 255). Die Ereignissteuerung muss zunächst aktiviert werden (siehe Objekt 0x6423), ansonsten können die Eingänge nur per Remote Transmission Request (RTR) abgefragt (gepollt) werden.

**Kommunikationsparameter5.-16. TxPDO**

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1804-0x180F (je nach Gerätetyp)	0	Anzahl Elemente	Unsigned8	ro	N	5	Kommunikationsparameter des 5.-16. Sende-PDOs. Subindex 0: Anzahl der folgenden Parameter
	1	COB-ID	Unsigned32	rw	N	0x00000000	COB-ID (Communication Object Identifier) TxPDO1
	2	Transmission Type	Unsigned8	rw	N	255	Übertragungsart des PDOs
	3	Inhibit Time	Unsigned16	rw	N	0	Wiederholungsverzögerung [Wert x 100 µs]
	4	CMS Priority Group	Unsigned8	rw	N	-	Aus Gründen der Abwärtskompatibilität vorhanden, nicht genutzt.
5	Event Timer	Unsigned16	rw	N	0	Event-Timer	

**Mapping 1. TxPDO**

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1A00	0	Anzahl Elemente	Unsigned8	rw	N	abhängig von Typ und Bestückung	Mapping-Parameter des ersten Transmit PDOs; Subindex 0: Anzahl der gemappten Objekte.
	1	1. gemapptes Objekt	Unsigned32	rw	N	0x60000108	1. gemapptes Applikationsobjekt (2 Byte Index, 1 Byte Subindex, 1 Byte Bitbreite)
	2	2. gemapptes Objekt	Unsigned32	rw	N	0x60000208	2. gemapptes Applikationsobjekt (2 Byte Index, 1 Byte Subindex, 1 Byte Bitbreite)
	...	...	...	...	...	...	...
	8	8. gemapptes Objekt	Unsigned32	rw	N	0x60000808	8. gemapptes Applikationsobjekt (2 Byte Index, 1 Byte Subindex, 1 Byte Bitbreite)

Das erste Sende-PDO (TxPDO1) ist per Default für digitale Eingangsdaten vorgesehen. Je nach Anzahl der bestückten Eingänge wird automatisch die erforderliche Länge des PDOs bestimmt und die entsprechenden Objekte gemappt. Da die digitalen Eingänge byteweise organisiert sind, kann die Länge des PDOs in Bytes direkt dem Subindex 0 entnommen werden.

Um das Mapping zu verändern muss eine bestimmte Reihenfolge eingehalten werden (siehe Objekt Index 0x1600).

**Mapping 2. TxPDO**

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1A01	0	Anzahl Elemente	Unsigned8	rw	N	abhängig von Typ und Bestückung	Mapping-Parameter des zweiten Transmit PDOs; Subindex 0: Anzahl der gemappten Objekte.
	1	1. gemapptes Objekt	Unsigned32	rw	N	0x64010110	1. gemapptes Applikationsobjekt (2 Byte Index, 1 Byte Subindex, 1 Byte Bitbreite)
	2	2. gemapptes Objekt	Unsigned32	rw	N	0x64010210	2. gemapptes Applikationsobjekt (2 Byte Index, 1 Byte Subindex, 1 Byte Bitbreite)
	...	...	...	...	...	...	...
	8	8. gemapptes Objekt	Unsigned32	rw	N		8. gemapptes Applikationsobjekt (2 Byte Index, 1 Byte Subindex, 1 Byte Bitbreite)

Das zweite Sende-PDO (TxPDO2) ist per Default für analoge Eingangsdaten vorgesehen. Je nach Anzahl der bestückten Eingänge wird automatisch die erforderliche Länge des PDOs bestimmt und die entsprechenden Objekte gemappt. Da die analogen Eingänge wortweise organisiert sind, kann die Länge des PDOs in Bytes direkt dem Subindex 0 entnommen werden.

Um das Mapping zu verändern muss eine bestimmte Reihenfolge eingehalten werden (siehe Objekt Index 0x1600).

**Mapping 3.-16. TxPDO**

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1A02-0x1A0F (je nach Geräte Typ)	0	Anzahl Elemente	Unsigned8	rw	N	abhängig von Typ und Bestückung	Mapping-Parameter des 3.-16. Transmit PDOs; Subindex 0: Anzahl der gemappten Objekte.
	1	1. gemapptes Objekt	Unsigned32	rw	N	0x00000000 (Siehe Text)	1. gemapptes Applikationsobjekt (2 Byte Index, 1 Byte Subindex, 1 Byte Bitbreite)
	2	2. gemapptes Objekt	Unsigned32	rw	N	0x00000000 (Siehe Text)	2. gemapptes Applikationsobjekt (2 Byte Index, 1 Byte Subindex, 1 Byte Bitbreite)
	...	...	...	...	...	...	...
	8	8. gemapptes Objekt	Unsigned32	rw	N	0x00000000 (Siehe Text)	8. gemapptes Applikationsobjekt (2 Byte Index, 1 Byte Subindex, 1 Byte Bitbreite)

Das 3. bis 16. Sende-PDO (TxPDO3ff) wird vom Busknoten je nach Klemmen-Bestückung (bzw. je nach Erweiterungs-Modulen) automatisch mit einem Default Mapping versehen. Die Vorgehensweise ist im Kapitel PDO-Mapping beschrieben.

Um das Mapping zu verändern muss eine bestimmte Reihenfolge eingehalten werden (siehe Objekt Index 0x1600).

**i Beckhoff Default Mapping**

DS401 V2 schreibt für die PDOs 3+4 als Default Mapping analoge Ein- bzw. Ausgangsdaten vor. Das entspricht dem Beckhoff Default Mapping dann, wenn weniger als 65 digitale Ein- bzw. Ausgänge vorhanden sind. Um die Abwärtskompatibilität zu gewährleisten wird das Beckhoff Default Mapping beibehalten - die Geräte entsprechen damit in ihrem Mapping-Verhalten DS401 V1, in allen anderen Belangen DS401 V2.

Im Objektverzeichnis (und damit auch im eds File) sind der Vollständigkeit halber zusätzlich folgende Objekteinträge vorhanden:

Index	Bedeutung
0x2000	Digitale Eingänge (Funktion identisch mit Objekt 0x6000)
0x2100	Digitale Ausgänge (Funktion identisch mit Objekt 0x6100)
0x2200	1-Byte Sonderklemmen, Eingänge (derzeit keine entsprechenden Klemmen im Produktprogramm vorhanden)
0x2300	1-Byte Sonderklemmen, Ausgänge (derzeit keine entsprechenden Klemmen im Produktprogramm vorhanden)
0x2400	2-Byte Sonderklemmen, Eingänge (derzeit keine entsprechenden Klemmen im Produktprogramm vorhanden)
0x2500	2-Byte Sonderklemmen, Ausgänge (derzeit keine entsprechenden Klemmen im Produktprogramm vorhanden)
0x2E00	7-Byte Sonderklemmen, Eingänge (derzeit keine entsprechenden Klemmen im Produktprogramm vorhanden)
0x2F00	7-Byte Sonderklemmen, Ausgänge (derzeit keine entsprechenden Klemmen im Produktprogramm vorhanden)

**3-Byte Sonderklemmen, Eingangsdaten**

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x2600	0	Anzahl Elemente	Unsigned8	ro	N	abhängig von Typ und Bestückung	Anzahl verfügbarer 3-Byte Sonderkanäle, Eingänge
	1	1st input block	Unsigned24	ro	Y	0x000000	1. Eingangskanal
	...	...	...	...	...	...	...
	0X80	128. input block	Unsigned24	ro	Y	0x000000	128. Eingangskanal

Beispiel für Sonderklemmen mit 3-Byte Eingangsdaten (in Default-Einstellung): KL2502 (PWM Ausgänge, 2 x 3 Bytes)

**3-Byte Sonderklemmen, Ausgangsdaten**

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x2700	0	Anzahl Elemente	Unsigned8	ro	N	abhängig von Typ und Bestückung	Anzahl verfügbarer 3-Byte Sonderkanäle, Ausgänge
	1	1st output block	Unsigned24	rww	Y	0x000000	1. Ausgangskanal
	...	...	...	...	...	...	...
	0X80	128. output block	Unsigned24	rww	Y	0x000000	128. Ausgangskanal

Beispiel für Sonderklemmen mit 3-Byte Ausgangsdaten (in der Default-Einstellung): KL2502 (PWM Ausgänge, 2 x 3 Bytes)

**4-Byte Sonderklemmen, Eingangsdaten**

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x2800	0	Anzahl Elemente	Unsigned8	ro	N	abhängig von Typ und Bestückung	Anzahl verfügbarer 4-Byte Sonderkanäle, Eingänge
	1	1st input block	Unsigned32	ro	Y	0x00000000	1. Eingangskanal
	...	...	...	...	...	...	...
	0X80	128. input block	Unsigned32	ro	Y	0x00000000	128. Eingangskanal

Beispiele für Sonderklemmen mit 4-Byte Eingangsdaten (in der Default-Einstellung): KL5001, KL6001, KL6021, KL6051

**4-Byte Sonderklemmen, Ausgangsdaten**

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x2900	0	Anzahl Elemente	Unsigned8	ro	N	abhängig von Typ und Bestückung	Anzahl verfügbarer 4-Byte Sonderkanäle, Ausgänge
	1	1st output block	Unsigned32	rww	Y	0x00000000	1. Ausgangskanal
	...	...	...	...	...	...	...
	0X80	128. output block	Unsigned32	rww	Y	0x00000000	128. Ausgangskanal

Beispiele für Sonderklemmen mit 4-Byte Ausgangsdaten (in der Default-Einstellung): KL5001, KL6001, KL6021, KL6051

**5-Byte Sonderklemmen, Eingangsdaten**

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x2A00	0	Anzahl Elemente	Unsigned8	ro	N	abhängig von Typ und Bestückung	Anzahl verfügbarer 5-Byte Sonderkanäle, Eingänge
	1	1st input block	Unsigned40	ro	Y	0x0000000000	1. Eingangskanal
	...	...	...	...	...	...	...
	0X40	64. input block	Unsigned40	ro	Y	0x0000000000	64. Eingangskanal

Beispiel für Sonderklemmen mit 5-Byte Eingangsdaten (in der Default-Einstellung): KL1501

**5-Byte Sonderklemmen, Ausgangsdaten**

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x2B00	0	Anzahl Elemente	Unsigned8	ro	N	abhängig von Typ und Bestückung	Anzahl verfügbarer 5-Byte Sonderkanäle, Ausgänge
	1	1st output block	Unsigned40	rww	Y	0x0000000000	1. Ausgangskanal
	...	...	...	...	...	...	...
	0X40	64. output block	Unsigned40	rww	Y	0x0000000000	64. Ausgangskanal

Beispiel für Sonderklemmen mit 5-Byte Ausgangsdaten (in der Default-Einstellung): KL1501

**6-Byte Sonderklemmen, Eingangsdaten**

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x2C00	0	Anzahl Elemente	Unsigned8	ro	N	abhängig von Typ und Bestückung	Anzahl verfügbarer 6-Byte Sonderkanäle, Eingänge
	1	1st input block	Unsigned48	ro	Y	0x0000000000	1. Eingangskanal
	...	...	...	...	...	...	...
	0X40	64. input block	Unsigned48	ro	Y	0x0000000000	64. Eingangskanal

Beispiel für Sonderklemmen mit 6-Byte Eingangsdaten (in der Default-Einstellung): KL5051, KL5101, KL5111

**6-Byte Sonderklemmen, Ausgangsdaten**

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x2D00	0	Anzahl Elemente	Unsigned8	ro	N	abhängig von Typ und Bestückung	Anzahl verfügbarer 6-Byte Sonderkanäle, Ausgänge
	1	1st output block	Unsigned48	rww	Y	0x0000000000	1. Ausgangskanal
	...	...	...	...	...	...	...
	0X40	64. output block	Unsigned48	rww	Y	0x0000000000	64. Ausgangskanal

Beispiel für Sonderklemmen mit 6-Byte Ausgangsdaten (in der Default-Einstellung): KL5051, KL5101, KL5111

**8-Byte Sonderklemmen, Eingangsdaten**

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x3000	0	Anzahl Elemente	Unsigned8	ro	N	abhängig von Typ und Bestückung	Anzahl verfügbarer 6-Byte Sonderkanäle, Eingänge
	1	1st input block	Unsigned64	ro	Y	0x0000000000	1. Eingangskanal
	...	...	...	...	...	...	...
	0x40	64. input block	Unsigned64	ro	Y	0x0000000000	64. Eingangskanal

Beispiel für Sonderklemmen mit 8-Byte Eingangsdaten: KL5101 (mit Word-Alignment, nicht in der Default-Einstellung)

**8-Byte Sonderklemmen, Ausgangsdaten**

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x3100	0	Anzahl Elemente	Unsigned8	ro	N	abhängig von Typ und Bestückung	Anzahl verfügbarer 6-Byte Sonderkanäle, Ausgänge
	1	1st output block	Unsigned64	rww	Y	0x0000000000	1. Ausgangskanal
	...	...	...	...	...	...	...
	0x40	64. output block	Unsigned64	rww	Y	0x0000000000	64. Ausgangskanal

Beispiel für Sonderklemmen mit 8-Byte Ausgangsdaten: KL5101 (mit Word-Alignment, nicht in der Default-Einstellung)

**Register-Kommunikation Busknoten**

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x4500	0	Register Access	Unsigned32	rw	N	keiner	Zugriff interne Register Busknoten

Der 32 Bit-Wert ist wie folgt aufgebaut:

MSB		LSB	
Zugriff (Bit7) + Tabellenummer (Bit 6...0)	Registernummer	High-Byte Registerwert	Low-Byte Registerwert
[0..1] + [0...0x7F]	[0...0xFF]	[0...0xFF]	[0...0xFF]

Wie bei CANopen üblich wird das LSB zuerst und das MSB zuletzt übertragen.

Durch Zugriff auf Index 0x4500 können beliebige Register der Busstation beschrieben oder gelesen werden. Die Kanalnummer und Register werden hierbei im 32 Bit-Datenwert adressiert.

**Registerwert lesen**

Zunächst muss dem Koppler mitgeteilt werden, welches Register gelesen werden soll. Hierzu muss ein SDO-Schreibzugriff auf die entsprechende Index/Subindex-Kombination erfolgen mit:

- Tabellenummer (Zugriffs-Bit=0) in Byte 3
- Registeradresse in Byte 2 des 32 Bit Datenwertes.

Bytes 1 und 0 werden nicht ausgewertet, wenn das Zugriffs-Bit (MSB in Byte 3) = 0 ist. Anschließend kann der Registerwert auf derselben Index/Subindex-Kombination gelesen werden.

Der Koppler setzt das Zugriffs-Bit nach dem Schreiben der auszulesenden Registeradresse so lange auf 1, bis der korrekte Wert zur Verfügung steht. Beim SDO-Lesezugriff ist also zu überprüfen, dass die Tabellenummer im Wertebereich 0...0x7F liegt.

Ein Zugriffsfehler bei der Register-Kommunikation wird durch entsprechende Rückgabewerte des SDO-Protokolls angezeigt (siehe Kapitel SDO, Abbruch Parameterkommunikation).

### Beispiel Registerwert lesen

Es soll festgestellt werden, welcher Baud-Ratenindex der Schalterstellung 1,1 (DIP 7,8) zugeordnet ist (siehe Kapitel *Netzwerkadresse und Baud-Raten*). Hierzu muss der Wert in Tabelle 100, Register 3 gelesen werden. Es müssen also folgende SDO Telegramme gesendet werden:

Schreibzugriff (Download Request) auf Index 4500, Subindex 0 mit 32 Bit Datenwert 0x64 03 00 00.

Id=0x600+Node-ID DLC=8; Data=23 00 45 00 00 00 03 64

Anschließend Lesezugriff (Upload Request) auf den gleichen Index/Subindex, hierbei ist der Datenwert beliebig (hier 00).

Id=0x600+Node-ID DLC=8; Data=40 00 45 00 00 00 00 00

Der Koppler antwortet mit dem Upload Response Telegramm:

Id=0x580+Node-ID DLC=8; Data=43 00 45 00 04 00 03 64

Es steht hier also der Wert 4 in diesem Register, dieser Baud-Ratenindex entspricht 125 kBit/s (Default-Wert).

### Registerwert Schreiben

SDO-Schreibzugriff auf die entsprechende Index/Subindex-Kombination mit:

- Tabellenummer + 0x80 (Zugriffs-Bit=1) in Byte 3
- Registeradresse in Byte 2
- High-Byte Registerwert in Byte 1
- Low-Byte Registerwert in Byte 0 des 32 Bit Datenwertes

### Koppler-Schreibschutz aufheben

Bevor die Register des Buskopplers beschrieben werden können muss zunächst der Schreibschutz aufgehoben werden. Hierzu müssen die folgenden Werte in der angegebenen Reihenfolge auf die entsprechenden Register geschrieben werden:

Arbeitsschritt	Tabelle	Register	Wert	entsprechender SDO Download-Wert (0x4500/0)
1.	99	2	45054 (0xAFFE)	0xE3 02 AF FE (0xE3=0x63(=99)+0x80)
2.	99	1	1 (0x0001)	0xE3 01 00 01
3.	99	0	257 (0x0101)	0xE3 00 01 01

### Koppler-Schreibschutz aufheben (CAN Darstellung)

Um den Koppler-Schreibschutz aufzuheben müssen also folgende SDO-Telegramme (Download Requests) an den Koppler geschickt werden:

Id=0x600+Node-ID DLC=8; Data=23 00 45 00 FE AF 02 E3

Id=0x600+Node-ID DLC=8; Data=23 00 45 00 01 00 01 E3

Id=0x600+Node-ID DLC=8; Data=23 00 45 00 01 01 00 E3

### Beispiel Registerwert Schreiben

Nachdem der Schreibschutz aufgehoben wurde, soll nun der Baud-Ratenindex für die DIP-Schalterstellung 1,1 auf den Wert 7 gesetzt werden. Damit wird dieser Schalterstellung die Baud-Rate 20 kBaud zugeordnet.

Hierzu muss Tabelle 100, Register 3 mit dem Wert 7 beschrieben werden, das erfolgt durch SDO-Schreibzugriff (Download Request) auf Index 0x4500, Subindex 0 mit dem 32 Bit-Wert E4 03 00 07 (0xE4 = 0x64+0x80):

Id=0x600+Node-ID DLC=8; Data=23 00 45 00 07 00 03 E4

**Klemmen identifizieren**

Über die Tabelle 9 des Buskopplers kann die Kennung des Kopplers (bzw. der Busstation) und der angesteckten Busklemmen gelesen werden. Dabei enthält Register 0 die Kennung des Buskopplers selbst, Register 1 die Kennung der ersten Klemme und Register n die Kennung der n-ten Klemme:

Tabellenummer	Registernummer	Beschreibung	Wertebereich
9	0	Busstation-Kennung	0 - 65535
9	1-255	Kennung Erweiterungsmodul/Busklemme	0 - 65535

Die Buskopplerbeschreibung in Registernummer 0 enthält 5120 = 0x1400 beim BK5120, 5110 = 0x13F6 beim BK5110 und 5100 = 0x13EC beim LC5100. Bei den Feldbus Box Baugruppen steht in Register 0 die Kennung 510dez = 0x1FE bzw. 518dez = 0x206.

Die Kennung der Erweiterungsmodule bzw. Klemmenbeschreibung enthält bei analogen und Sonderklemmen die Klemmenbezeichnung (Dez);  
 Beispiel: ist als dritte Klemme eine KL3042 gesteckt, so enthält Register 3 den Wert 3042<sub>dez</sub> (0x0BE2).

Bei digitalen Klemmen wird folgende Bit-Kennung verwendet:

MSB								LSB							
1	s6	s5	s4	s3	s2	s1	s0	0	0	0	0	0	0	a	e

s6...s1: Datenbreite in Bit; a=1: Ausgangsklemme; e=1: Eingangsklemme

Diese Kennung führt zu den unten aufgeführten Klemmenbeschreibungen bei den Klemmen:

Kennung Klemmen	Bedeutung
0x8201	2 Bit digitale Eingangsklemme, z. B. KL1002, KL1052, KI9110, KL9260
0x8202	2 Bit digitale Ausgangsklemme, z. B. KL2034, KL2612, KL2702
0x8401	4 Bit digitale Eingangsklemme, z. B. KL1104, KL1124, KL1194
0x8402	4 Bit digitale Ausgangsklemme, z. B. KL2124, KL2134, KL2184
0x8403	4 Bit digitale Ein/Ausgangsklemme, z. B. KL2212

und folgende Kennung bei den Erweiterungs-Box Modulen:

Kennung Erweiterungs-Box Modulen	Bedeutung
0x000A	4 Bit Eingangs- und 4 Bit Ausgangsmodul
0x0011	8 Bit Eingangs- und 8 Bit Ausgangsmodul
0x0014	8 Bit digitales Eingangsmodul
0x0015	8 Bit digitales Ausgangsmodul

**Allgemeine Koppler-Konfiguration (Tabelle 0)**

Die Tabelle 0 des Buskopplers enthält die Daten für die allgemeine Kopplerkonfiguration. In der Regel muss diese nicht verändert werden; für besondere Anwendungsfälle können die Einstellungen jedoch über die KS2000 Konfigurations-Software oder den direkten Zugriff über die Register-Kommunikation verändert werden. Hierzu muss zunächst der Schreibschutz aufgehoben werden (siehe oben).

Im Folgenden werden die relevanten Registereinträge beschrieben:

**K-Buskonfiguration**

Tabelle 0, Register 2 enthält die K-Buskonfiguration und ist wie folgt codiert (Default-Wert: 0x0006):

MSB								LSB								
0	0	0	0	0	0	0	0	0	0	0	0	0	0	D	G	A

**A: Autoreset**

Bei K-Bus-Fehler wird zyklisch versucht, den K-Bus durch Reset wieder zu aufzustarten. Wenn Emergencies und Guarding nicht ausgewertet werden, so kann es bei aktiviertem Autoreset vorkommen, dass Aus- und Eingangsinformation unerkant verloren geht.

- 0: kein Autoreset (Default)
- 1: Autoreset aktiv

### G: Gerätediagnose

Meldung (über Emergency), z. B. dass  
 - Drahtbruch bei Stromeingängen (mit Diagnose)  
 - 10 V überschritten bei 1-10V Eingangsklemme

- 0: Gerätediagnose abgeschaltet
- 1: Gerätediagnose aktiv (Default)

### D: Diagnosedaten

Digitaler Klemmen ins Prozessabbild einblenden (z. B. KL2212). Diese Flag wird nur ausgewertet, wenn die Gerätediagnose aktiv ist (siehe oben).

- 0: Nicht einblenden
- 1: Einblenden (Default)

### Prozessabbildbeschreibung

Tabelle 0, Register 3 enthält die Prozessabbildbeschreibung und ist wie folgt codiert (Default-Wert: 0x0903):

MSB								LSB							
0	0	0	0	k1	k0	f1	f0	0	0	a	0	d	k	1	1

#### k0...k1: Reaktion auf K-Bus-Fehler

- 0,2: Eingänge bleiben unverändert (Default=2);
- 1: Eingänge auf 0 setzen (TxPDO mit Nullen wird verschickt)

#### f0...f1: Reaktion auf Feldbusfehler

0: Stoppen der K-Bus Zyklen, Watchdog auf Klemmen spricht an, Fehlerausgangswerte werden aktiv. Beim Neustart werden zunächst die alten Ausgangswerte gesetzt.

- 1: Ausgänge auf 0 setzen, Stoppen der K-Bus Zyklen (Default). 2: Ausgänge bleiben unverändert.

#### a: Word-Alignment von Analog- und Sonderklemmen

- 0: kein Alignment (Default)
- 1: Daten auf Wortgrenzen mappen (Prozessdatum beginnt stets auf gerader Adresse im PDO)

#### d: Datenformat komplexe Klemmen (Analog- und Sonderklemmen)

- 0: Intel-Format (Default)
- 1: Motorola-Format

#### k: Auswertung komplexe Klemmen (Analog- und Sonderklemmen)

- 0: nur Nutzdaten (Default)
- 1: komplette Auswertung (Achtung: Analogkanäle benötigen dann statt z. B. 2 Eingangsbytes je 3 Eingangs- und 3 Ausgangsbytes; statt 4 Kanäle je PDO werden für 2 Kanäle je ein Rx- und ein TxPDO benötigt)

**Register-Kommunikation Busklemme/Erweiterungsbox**

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x4501	0	Access Terminal Register	Unsigned8	ro	N	keiner	Index 0x4501 ermöglicht den Zugriff auf alle Register der Busklemmen bzw. Erweiterungs-module. Subindex 0 enthält die Anzahl der gesteckten Busklemmen.
	1	Access Reg. Terminal 1	Unsigned32	rw	N	keiner	Zugriff Register Busklemme bzw. E-Modul 1
	...	...	...	...	...	...	...
	0xFE	Access Reg. Terminal 254	Unsigned32	rw	N	keiner	Zugriff Register Busklemme bzw. E-Modul 254

Der 32Bit-Wert ist wie folgt aufgebaut:

MSB		LSB	
Zugriff (Bit7) + Kanalnummer (Bit 6...0)	Registernummer	High-Byte Registerwert	Low-Byte Registerwert
[0..1] + [0...0x7F]	[0...0xFF]	[0...0xFF]	[0...0xFF]

Wie bei CANopen üblich wird das LSB zuerst und das MSB zuletzt übertragen.

Durch Zugriff auf Index 0x4501 können die Anwenderregister der Busklemmen bzw. Erweiterungs-module beschrieben oder gelesen werden. Die Baugruppen verfügen über einen Registersatz je Ein- bzw. Ausgangskanal. Die Adressierung der Baugruppen erfolgt über den Subindex, die Kanalnummer und Register werden im 32 Bit-Datenwert adressiert. Hierbei entspricht die Kanalnummer 0 dem ersten Kanal, 1 dem zweiten Kanal etc.

**Registerwert lesen**

Zunächst muss dem Koppler mitgeteilt werden, welches Register gelesen werden soll. Hierzu muss ein SDO-Schreibzugriff auf die entsprechende Index/Subindex-Kombination erfolgen mit:

Kanalnummer (Zugriffs-Bit=0) in Byte 3

Registeradresse in Byte 2 des 32 Bit Datenwertes.

Bytes 1 und 0 werden nicht ausgewertet, wenn das Zugriffs-Bit (MSB in Byte 3) = 0 ist. Anschließend kann der Registerwert auf derselben Index/Subindex-Kombination gelesen werden.

Der Koppler setzt das Zugriffs-Bit nach dem Schreiben der auszulesenden Registeradresse so lange auf 1, bis der korrekte Wert zur Verfügung steht. Beim SDO-Lesezugriff ist also zu überprüfen, dass die Tabellennummer im Wertebereich 0...0x7F liegt.

Ein Zugriffsfehler bei der Register-Kommunikation wird durch entsprechende Rückgabewerte des SDO-Protokolls angezeigt (siehe Kapitel SDO, Abbruch Parameterkommunikation).

**Beispiel Registerwert lesen**

Bei einer Thermoelement-Eingangsklemme KL3202 soll festgestellt werden, auf welchen Thermoelement-Typ der zweite Eingangs-Kanal eingestellt ist. Hierzu muss das Feature-Register 32 gelesen werden. Die Klemme befindet sich am fünften Steckplatz neben dem Buskoppler. Es müssen also folgende SDO-Telegramme gesendet werden:

Schreibzugriff (Download Request) auf Index 0x4501, Subindex 5 mit 32 Bit Datenwert 01 20 00 00 (0x01 = 2. Kanal, 0x20 = Register 32)

Id=0x600+Node-ID DLC=8; Data=23 01 45 05 00 00 20 01

Anschließend Lesezugriff (Upload Request) auf den gleichen Index/Subindex, hierbei ist der Datenwert beliebig (hier: 0x00).

Id=0x600+Node-ID DLC=8; Data=40 01 45 05 00 00 00 00

Der Koppler antwortet mit dem Upload Response Telegramm:

Id=0x580+Node-ID DLC=8; Data=43 01 45 05 06 31 20 01

Es steht hier also der Wert 31 06 im Feature-Register. Die obersten 4 Bit kennzeichnen den Thermoelement-Typ. Sie sind hier 3, demnach ist der eingestellte Typ für diesen Kanal PT500 (siehe Dokumentation KL3202).

### Registerwert Schreiben

SDO-Schreibzugriff auf die entsprechende Index/Subindex-Kombination mit:

- Kanalnummer + 0x80 (Zugriffs-Bit=1) in Byte 3
- Registeradresse in Byte 2
- High-Byte Registerwert in Byte 1
- Low-Byte Registerwert in Byte 0 des 32 Bit Datenwertes

### HINWEIS

#### Überprüfen Sie den geschriebenen Wert

Wenn der Schreibschutz nicht aufgehoben wurde (z. B. fehlerhaftes Codewort), so wird ein Schreibzugriff auf die Klemmenregister zwar bestätigt (SDO Download Response), der Wert jedoch nicht in das Register übernommen. Es wird deshalb empfohlen, den geschriebenen Wert anschließend auszulesen und zu vergleichen.

### Klemmen-Schreibschutz aufheben

Bevor die Anwender-Register der Busklemmen (Register 32-xx, je nach Klemmentyp bzw. Erweiterungsmodul) beschrieben werden können muss zunächst der Schreibschutz aufgehoben werden. Hierzu wird das folgende Codewort in das Register 31 des entsprechenden Kanals geschrieben:

Schreibschutz	Kanal	Register	Wert	entsprechender SDO Download-Wert (0x4500/0)
	1,2, 3 oder 4	31 (0x1F)	4661 (0x1235)	8y 1F 12 35 (y=Kanalnummer)

### Klemmen-Schreibschutz aufheben (CAN Darstellung)

Um den Klemmen-Schreibschutz aufzuheben muss also das folgende SDO-Telegramm an den Koppler geschickt werden:

Id=600 + Node-ID DLC=8; Data=23 01 45 xx 35 12 1F 8y

wobei xx den Steckplatz der Klemme und y den Kanal kennzeichnen.

### Beispiel Schreibschutz aufheben

Steckt also beispielsweise an einem BK5120 mit der Knotenadresse 3 eine Thermoelement-Eingangsklemme KL3202 an Steckplatz 5, so ist der Schreibschutz für den ersten Kanal wie folgt aufzuheben:

Id=0x603 DLC=8; Data=23 01 45 05 35 12 1F 80

Für den zweiten Kanal ist folgendes Telegramm zu senden:

Id=0x603 DLC=8; Data=23 01 45 05 35 12 1F 81

### Beispiel Registerwert Schreiben

Der Thermoelement-Typ des zweiten Kanals der KL3202 Klemme an Steckplatz 5 soll nun auf PT1000 umgestellt werden. Hierzu müssen die obersten 4 Bits (oberstes Nibble) im Feature-Register mit dem Wert 2 beschrieben werden. Es wird davon ausgegangen, dass für alle anderen Bits des Feature-Registers die Default-Werte übernommen werden sollen. Nachdem der Schreibschutz aufgehoben wurde, ist per SDO Schreibzugriff (Download Request) der folgende 32 Bit-Wert auf Index 0x4501, Subindex 05 zu schreiben: 81 20 21 06 (0x81=01+0x80; 0x20=32;0x2106 = Registerwert).

Das entsprechende Telegramm sieht auf dem Bus wie folgt aus:

Id=0x600+Node-ID DLC=8; Data=23 01 45 05 06 21 20 81

**PDOs aktivieren**

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x5500	0	Activate PDO Defaults	Unsigned32	rw	N	0x00000000	setzt PDO Communication Parameter für PDO 2...11

CANopen definiert Default-Identifizier für jeweils 4 Sende (Tx) und Empfangs (Rx) PDOs, alle anderen PDOs sind nach dem Aufstarten der Knoten zunächst deaktiviert. Über den Index 0x5500 lassen sich alle PDOs aktivieren, die gemäß Klemmenbestückung mit Prozessdaten vorbelegt sind (herstellerspezifisches Default Mapping). Dabei wird für PDO5...11 eine herstellereigene Default-Identifizier-Verteilung vorgenommen sowie für PDO 2...11 der Transmission Type und eine einheitliche Inhibit Zeit eingestellt. Nicht mit Prozessdaten versehene (also in der aktuellen Konfiguration überzählige) PDOs werden nicht aktiviert.

**● Pre-Operational erforderlich**

**i** Dieses Objekt kann nur im Pre-Operational Zustand beschrieben werden!

Der 32 Bit-Wert wird wie folgt verwendet:

MSB		LSB	
Transmission Type RxPDOs	Transmission Type TxPDOs	High-Byte Inhibit Zeit	Low-Byte Inhibit Zeit

Wie bei CANopen üblich wird das LSB zuerst und das MSB zuletzt übertragen.

**Beispiel**

PDOs aktivieren für Busknoten Nummer 1, Inhibit Zeit auf 10 ms (=100 x 100 µs) setzen, Transmission Type TxPDOs auf 255 setzen, Transmission Type RxPDOs auf 1 setzen. Folgendes Telegramm ist zu senden:  
Id=0x601 DLC=8; Data=23 00 55 00 64 00 FF 01

Der Knoten antwortet mit folgendem Telegramm:  
Id=0x601 DLC=8; Data=60 00 55 00 00 00 00 00

**Verwendete Identifizier**

Die Default-Identifizier-Verteilung für die zusätzlichen PDOs lässt die vordefinierten Bereiche für Guarding, SDOs etc. frei, geht ab PDO6 von maximal 64 Knoten im Netz aus und erfolgt nach folgendem Schema:

Objekt	Function Code	resultierende COB-ID (hex)	resultierende COB-ID (dez)
TxPDO5	1101	0x681 - 0x6BF	1665 - 1727
RxPDO5	1111	0x781 - 0x7BF	1921 - 1983
TxPDO6	00111	0x1C1 - 0x1FF	449 - 511
RxPDO6	01001	0x241 - 0x27F	577 - 639
TxDPO7	01011	0x2C1 - 0x2FF	705 - 767
RxPDO7	01101	0x341 - 0x37F	833 - 895
TxPDO8	01111	0x3C1 - 0x3FF	961 - 1023
RxPDO8	10001	0x441 - 0x47F	1089 - 1151
TxPDO9	10011	0x4C1 - 0x4FF	1217 - 1279
RxPDO9	10101	0x541 - 0x57F	1345 - 1407
TxDPO10	10111	0x5C1 - 0x5FF	1473 - 1535
RxPDO10	11001	0x641 - 0x67F	1601 - 1663
TxPDO11	11011	0x6C1 - 0x6FF	1729 - 1791
RxPDO11	11101	0x741 - 0x77F	1857 - 1919

**HINWEIS**

**Index 0x5500**

Es ist darauf zu achten, dass der Index 0x5500 nicht genutzt wird, wenn Buskoppler mit mehr als 5 PDOs in Netzen mit Knoten-Adressen >64 vorhanden sind, da es sonst zu Identifizier-Überschneidungen kommen kann. In diesem Fall müssen die PDO Identifizier individuell eingestellt werden.

Der Übersichtlichkeit halber sind die nach CANopen definierten Default-Identifizier hier ebenfalls aufgeführt:

Objekt	Function Code	resultierende COB-ID (hex)	resultierende COB-ID (dez)
Emergency	0001	0x81 - 0xBF [0xFF]	129 - 191 [255]
TxPDO1	0011	0x181 - 0x1BF [0x1FF]	385 - 447 [511]
RxPDO1	0100	0x201 - 0x23F [0x27F]	513 - 575 [639]
TxPDO2	0101	0x281 - 0x2BF [0x2FF]	641 - 676 [767]
RxPDO2	0110	0x301 - 0x33F [0x37F]	769 - 831 [895]
TxDPO3	0111	0x381 - 0x3BF [0x3FF]	897 - 959 [1023]
RxPDO3	1000	0x401 - 0x43F [0x47F]	1025 - 1087 [1151]
TxPDO4	1001	0x481 - 0x4BF [0x4FF]	1153 - 1215 [1279]
RxPDO4	1010	0x501 - 0x53F [0x57F]	1281 - 1343 [1407]
SDO (Tx)	1011	0x581 - 0x5BF [0x5FF]	1409 - 1471 [1535]
SDO (Rx)	1100	0x601 - 0x63F [0x67F]	1537 - 1599 [1663]
Guarding / Heartbeat/ Bootup	1110	0x701 - 0x73F [0x77F]	1793 - 1855 [1919]

Angegeben sind die Identifier, die sich aus den DIP-Schalter-Einstellungen am Koppler ergeben, sowie in eckigen Klammern der Identifier-Bereich für die Knotenadressen 64...127 (am Buskoppler BK5110, BK5120 und LC5100 nicht einstellbar). Bei den Feldbus Box-Modulen und dem Buskoppler BK515x lassen sich die Adressen 1...99 einstellen.

Eine tabellarische Übersicht über alle Identifier findet sich im Anhang.

### Digitale Eingänge

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x6000	0	Anzahl Elemente	Unsigned8	ro	N	abhängig von Typ und Bestückung	Anzahl verfügbarer digitaler 8-Bit Eingangdatenblöcke
	1	1st input block	Unsigned8	ro	Y	0x00	1. Eingangskanal
	...	...	...	...	...	...	...
	0xFE	254. input block	Unsigned8	ro	Y	0x00	254. Eingangskanal

### Interrupt Maske

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x6126	0	Anzahl Elemente	Unsigned8	ro	N	abhängig von Typ	Anzahl der 32-Bit Interrupt Masken = 2 x Anzahl TxPDOs
	1	IR-Mask0 TxPDO1	Unsigned32	rw	N	0xFFFFFFFF	IR-Maske Bytes 0...3 TxPDO1
	2	IR-Mask1 TxPDO1	Unsigned32	rw	N	0xFFFFFFFF	IR-Maske Bytes 4...7 TxPDO1
	3	IR-Mask0 TxPDO2	Unsigned32	rw	N	0xFFFFFFFF	IR-Maske Bytes 0...3 TxPDO2
...	...	...	...	...	...	...	...
	0x20	IR-Mask1 TxPDO16	Unsigned32	rw	N	0xFFFFFFFF	IR-Maske Bytes 4...7 TxPDO16

Per Default führt jede Änderung eines Wertes im ereignisgesteuerten PDO zum Versenden des Telegramms. Mit der Interrupt Maske kann bestimmt werden, welche Daten-Änderungen hierfür ausgewertet werden. Durch Nullen der entsprechenden Bereiche innerhalb der PDOs werden diese bei der Ereignissteuerung ("Interrupt-Steuerung") ausmaskiert. Die Interrupt Maske umfasst nicht nur die PDOs mit digitalen Eingängen, sondern alle vorhandenen TxPDOs. Falls die TxPDOs kürzer als 8 Bytes sind, wird der überzählige Teil der IR-Maske nicht ausgewertet.

Die Interrupt Maske beeinflusst nur TxPDOs mit Transmission Type 254 und 255. Sie wird nicht auf dem Gerät gespeichert (auch nicht durch das Objekt 0x1010). Änderungen der Maske zur Laufzeit (im Operational Status) sind möglich und werden bei der nächsten Eingangsdaten-Änderung ausgewertet.

Die Interrupt Maske wird für TxPDOs mit analogen Eingangsdaten nicht ausgewertet, wenn für die Eingänge Grenzwerte (0x6424, 0x6425) oder die Delta Funktion (0x6426) aktiviert wurden.

Dieser Eintrag ist ab Firmware Stand C3 implementiert.

**Beispiel für die Zuordnung der Daten**

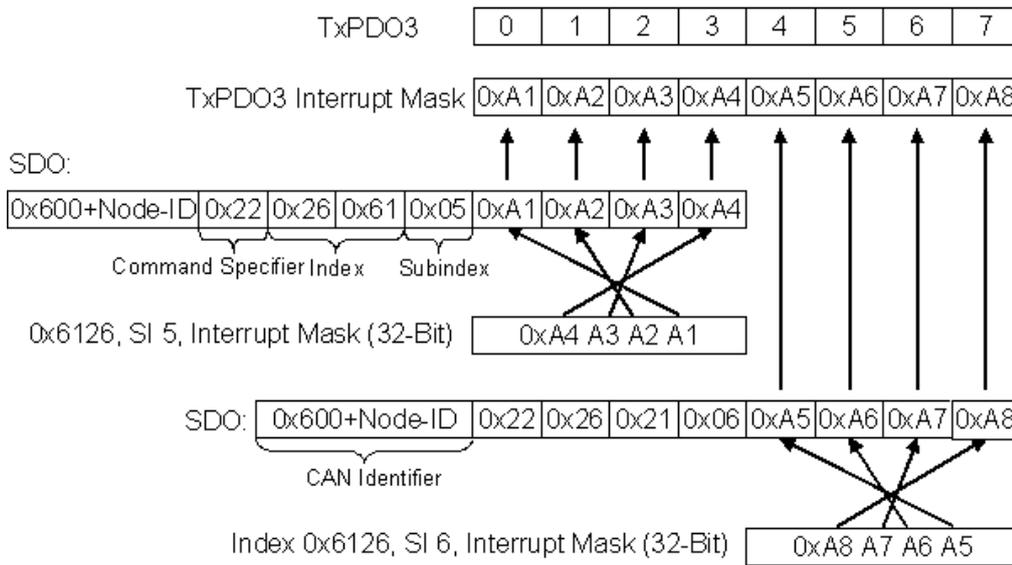


Abb. 141: Beispiel für die Zuordnung der Daten

**Anwendungsbeispiel**

Der Zählerwert eines schnellen Zählereingangs soll nur übertragen werden, sobald sich Bits im Statuswort (z. B. der Latch-Eingang) geändert haben. Hierzu muss der 32-Bit Zählerwert in der Interrupt Maske ausmaskiert (=genullt) werden. Der Status befindet sich im Byte 0, der Zählerwert liegt per Default in den Bytes 1...4 des entsprechenden PDOs (im Beispiel TxPDO3, da <65 digitale und <5 analoge Eingänge vorhanden sind).

Also muss in Index 0x6126, Subindex5 der Wert 0x0000 00FF und in Subindex6 der Wert 0xFFFF FF00 eingetragen werden.

Die entsprechenden SDOs sehen demnach wie folgt aus:

11-bit Identifier	8 Byte Nutzdaten							
0x600+ Node-ID	0x22	0x26	0x61	0x05	0xFF	0x00	0x00	0x00
11-bit Identifier	8 Byte Nutzdaten							
0x600+ Node-ID	0x22	0x26	0x61	0x06	0x00	0xFF	0xFF	0xFF

**Digitale Ausgänge**

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x6200	0	Anzahl Elemente	Unsigned8	ro	N	abhängig von Typ und Bestückung	Anzahl verfügbarer digitaler 8-Bit Ausgangsdatenblöcke
	1	1st input block	Unsigned8	rw	Y	0x00	1. Ausgangskanal
	...	...	...	...	...	...	...
	0XFE	254. input block	Unsigned8	rw	Y	0x00	254. Ausgangskanal

**Analoge Eingänge**

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x6401	0	Anzahl Elemente	Unsigned8	ro	N	abhängig von Typ und Bestückung	Anzahl verfügbarer analoger Eingangskanäle
	1	1st input	Unsigned16	ro	Y	0x0000	1. Eingangskanal
	...	...	...	...	...	...	...
	0XFE	254. input	Unsigned16	ro	Y	0x0000	254. Eingangskanal

Die analogen Signale werden linksbündig dargestellt. Damit wird die Darstellung im Prozessabbild unabhängig von der tatsächlichen Auflösung. Details zum Datenformat finden sich beim jeweiligen Signaltyp.

## Analoge Ausgänge

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x6411	0	Anzahl Elemente	Unsigned8	ro	N	abhängig von Typ und Bestückung	Anzahl verfügbarer analoger Ausgangskanäle
	1	1st input block	Unsigned16	rw	Y	0x0000	1. Ausgangskanal
	...	...	...	...	...	...	...
	0XFE	254. input block	Unsigned16	rw	Y	0x0000	254. Ausgangskanal

Die analogen Signale werden linksbündig dargestellt. Damit wird die Darstellung im Prozessabbild unabhängig von der tatsächlichen Auflösung. Details zum Datenformat finden sich beim jeweiligen Signaltyp.

## Ereignissteuerung Analoge Eingänge

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x6423	0	Global Interrupt Enable	Boolean	rw	N	FALSE (0)	Aktiviert das ereignisgesteuerte Senden von PDOs mit Analogeingängen.

Nach CANopen sind die Analogeingänge in TxPDO2...4 zwar per Default auf den Transmission Type 255 (ereignisgesteuert) eingestellt, jedoch ist das Ereignis (die Änderung eines Eingangswertes) über die Ereignissteuerung im Objekt 0x6423 deaktiviert, um ein Überfluten des Busses mit Analogsignalen zu verhindern. Es empfiehlt sich, das Datenaufkommen der Analog-PDOs entweder durch synchrone Kommunikation oder durch Verwendung des Event Timers zu kontrollieren. Im ereignisgesteuerten Betrieb kann das Sendeverhalten der Analog-PDOs vor dem Aktivieren durch Einstellen von Inhibit-Zeit (Objekt 0x1800ff, Subindex 3) und/oder Grenzwertüberwachung (Objekt 0x6424 + 0x6425) und/oder Deltafunktion (Objekt 0x6426) parametrieren werden.

## Oberer Grenzwert Analoge Eingänge

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x6424	0	Anzahl Elemente	Unsigned8	ro	N	abhängig von Typ und Bestückung	Anzahl verfügbarer analoger Eingangskanäle
	1	upper limit 1st input	Unsigned16	rw	Y	0x0000	Oberer Grenzwert 1. Eingangskanal
	...	...	...	...	...	...	...
	0XFE	upper limit 254. input	Unsigned16	rw	Y	0x0000	Oberer Grenzwert 254. Eingangskanal

Werte ungleich 0 aktivieren den oberen Grenzwert für diesen Kanal. Ein PDO wird dann abgesetzt wenn dieser Grenzwert überschritten wird. Zusätzlich muss die Ereignissteuerung aktiviert sein (Objekt 0x6423). Das Datenformat entspricht dem der Analogeingänge.

## Unterer Grenzwert Analoge Eingänge

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x6425	0	Anzahl Elemente	Unsigned8	ro	N	abhängig von Typ und Bestückung	Anzahl verfügbarer analoger Eingangskanäle
	1	lower limit 1st input	Unsigned16	rw	Y	0x0000	Unterer Grenzwert 1. Eingangskanal
	...	...	...	...	...	...	...
	0XFE	lower limit 254. input	Unsigned16	rw	Y	0x0000	Unterer Grenzwert 254. Eingangskanal

Werte ungleich 0 aktivieren den unteren Grenzwert für diesen Kanal. Ein PDO wird dann abgesetzt wenn dieser Grenzwert unterschritten wird. Zusätzlich muss die Ereignissteuerung aktiviert sein (Objekt 0x6423). Das Datenformat entspricht dem der Analogeingänge.

## Deltafunktion Analoge Eingänge

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x6426	0	Anzahl Elemente	Unsigned8	ro	N	abhängig von Typ und Bestückung	Anzahl verfügbarer analoger Eingangskanäle
	1	delta value 1st input	Unsigned16	rw	Y	0x0000	Deltawert 1. Eingangskanal
	...	...	...	...	...	...	...
	0XFE	delta value 254. input	Unsigned16	rw	Y	0x0000	Deltawert 254. Eingangskanal

Werte ungleich 0 aktivieren die Deltafunktion für diesen Kanal. Ein PDO wird dann abgesetzt wenn sich der Wert seit dem letzten Senden um mehr als den Deltawert verändert hat. Zusätzlich muss die Ereignissteuerung aktiviert sein (Objekt 0x6423). Das Datenformat entspricht dem der Analogeingänge (Deltawert: nur positive Werte).

### 6.3.4 Objekte und Daten der BX5100/BC5150

#### Zugriff auf lokierte Merker

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x2F00	0	Anzahl des Subindex	Unsigned8	ro	N	128	Anzahl des Subindex
	1	Merker	Unsigned32	rw	N	Keiner	Merker %MB0..3
	2	Merker	Unsigned32	rw	N	Keiner	Merker %MB4..7
	...	...	...	...	...	...	...
	128	Merker	Unsigned32	rw	N	Keiner	Merker %MB508..511

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x2F01-0x2F07	0	Anzahl des Subindex	Unsigned8	ro	N	128	Anzahl des Subindex
	1	Merker	Unsigned32	rw	N	Keiner	Merker %MBx..x+4
	2	Merker	Unsigned32	rw	N	Keiner	Merker %MBy..y+4
	...	...	...	...	...	...	...
	128	Merker	Unsigned32	rw	N	Keiner	Merker %MBz...z+14

#### AMS NetId

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x5FFE	0	NetId	String	rw	N	1.1.1.1.1	AMS Net Id

## **6.4 ADS-Kommunikation**

### **6.4.1 ADS-Dienste**

Siehe Programmierung [▶ 102]

# 7 Fehlerbehandlung und Diagnose

## 7.1 Diagnose

### Zustand des CANopen

In vielen Fällen ist es wichtig zu wissen, ob die Kommunikation mit dem übergeordneten Master noch funktioniert. Verknüpfen Sie hierfür die Variable *NodeState* mit Ihrem SPS-Programm. Hierfür ist eine TwinCAT-Konfiguration erforderlich.

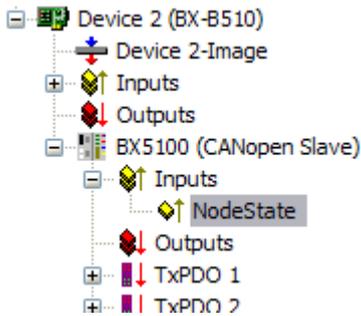


Abb. 142: CANopen Diagnose-Byte im System Manager

Fehlernummer	Beschreibung	Abhilfe
0	kein Fehler	-
2	Guarding-Fehler	Überprüfen Sie die Verbindung
20	Zu wenig PDOs erhalten (nur TwinCAT Config)	Überprüfen Sie die Konfiguration
22	Sync-Fehler	Überprüfen Sie die Verbindung
129	Knoten ist pre-operational	Schalten Sie den Knoten operational
130	Knoten ist gestoppt	Starten des Knoten

### Beispiel

Wenn der CANopen unterbrochen wird (z. B. weil das Kabel abgezogen oder die SPS ausgeschaltet wurde), dann meldet das der Busklemmen-Controller mit ??? im NodeState.

### Auslesen des Feldbusstatus per ADS

In der Default-Konfiguration oder der TwinCAT Konfiguration können Sie den Feldbusstatus über ADSREAD auslesen.

Parameter ADSREAD Baustein	Beschreibung
NetID	lokal - Leerstring
Port	1
IndexGroup	16#0006
IndexOffset	16#000C_AE00
LEN	1

### Zustand des K-Bus

Sollte der interne K-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 Variable *K-Bus-State* mit Ihrem SPS-Programm.

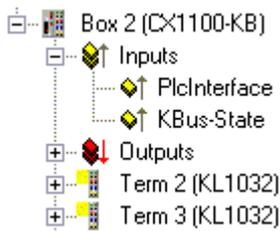


Abb. 143: Zustand des K-Bus

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 der TwinCAT Konfiguration können Sie den Feldbusstatus über ADSREAD auslesen.

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 ein BX-Controller zwei Gruppen von LEDs:

- Auf der rechten oberen Seite des BX-Controllers befinden sich zwei grüne Power LEDs. Diese dienen zur Diagnose der Spannungsversorgung des BX-Controllers und der Powerkontakte.
- Die DIAG LEDs befinden sich über der linken Seite des Navigationsschalters. Sie zeigen den Zustand des Feldbusses, der PLC und des K-Busses an.

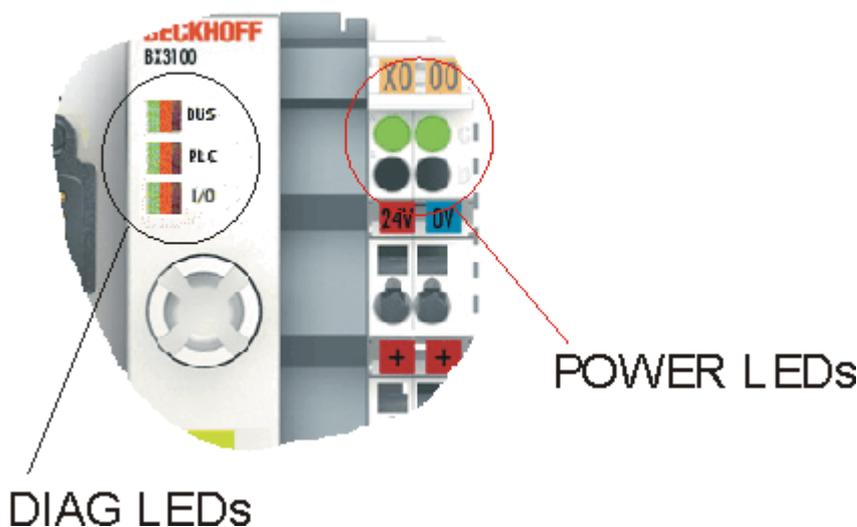


Abb. 144: Diagnose LEDs für den Feldbus, die SPS, den K-Bus und die Spannungsversorgung

**POWER LEDs zur Diagnose der Spannungsversorgung**

Power LEDs	Bedeutung
Linke LED aus	BX-Controller hat keine Spannungsversorgung
Rechte LED aus	Powerkontakte haben keine Spannungsversorgung

**DIAG LEDs**

Die DIAG LEDs sind unterteilt in:

- Bus: Feldbusdiagnose
- PLC: SPS-Diagnose
- I/O: K-Bus-Diagnose

Die LEDs können die Zustände aus, grün, orange und rot annehmen.



Abb. 145: Diagnose LEDs für den Feldbus, die SPS und den K-Bus

Nach dem Einschalten überprüft der BX-Controller sofort die angeschlossene Konfiguration. Der fehlerfreie Hochlauf wird durch das Verlöschen der LED I/O signalisiert. Das rote Leuchten der I/O LED zeigt einen Fehler im Bereich der Busklemmen an. Die Fehlerart wird im Display angezeigt. Das ermöglicht eine schnelle Fehlerbeseitigung.

**LEDs zur Diagnose des Feldbusses**

LED Bus	Bedeutung
aus	- kein Feldbus angeschlossen - Bauratensuche aktiviert: BX-Controller sucht die Baudrate
leuchtet rot	Baudrate gefunden und CAN Controller im Warning Limit (z. B. Stecker gezogen)
blinkt rot	Baudrate gefunden und CAN Error im Bus-Off
leuchtet orange	BX im Zustand PRE-Operational oder Stopped (kein Fehler)
blinkt orange	BX im Zustand PRE-Operational und Fehler Schnelles Blinken gefolgt von 1 x langsames Blinken - Guarding-Timeout 2 x langsames Blinken - SYNC-Timeout 3 x langsames Blinken - Event-Timeout
leuchtet grün	Buskommunikation ist in Ordnung, BX-Controller ist im Datenaustausch

**LED zur SPS-Diagnose**

<b>LED PLC</b>	<b>Bedeutung</b>
aus	SPS im Stop oder kein Programm vorhanden
blinkt red	Die eingestellte Task-Zeit wird ab und zu überschritten (siehe auch Kapitel Deaktivieren der LED für Zykluszeit-Überschreitung).
leuchtet rot	Die eingestellte Task-Zeit wird immer überschritten.
leuchtet orange	PLC läuft ohne Bootprojekt (leuchtet nur während des Zyklus), beim Erzeugen des Bootprojektes blinkt die LED orange
leuchtet grün	Bootprojekt - PLC läuft (leuchtet nur während des Zyklus)

**LED zur K-Bus Diagnose**

<b>LED I/O</b>	<b>Bedeutung</b>
aus	über den K-Bus werden keine Daten ausgetauscht
leuchtet rot	Fehler Blinken - Fehlerart - Anzeige Display
leuchtet orange	Register- bzw. KS2000-Online-Zugriff
leuchtet grün	K-Bus ist in Ordnung und läuft

**Fehlercode 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 Koppler) der Fehler lokalisiert werden</li> </ul>
1	0	EEPROM-Prüfsummenfehler	Herstellereinstellung mit der Konfigurationssoftware KS2000 setzen
	1	Überlauf im Code Buffer	Weniger Busklemmen stecken. Die programmierte Konfiguration hat zu viele Einträge in der Tabelle
	2	Unbekannter Datentyp	Software Update des BX-Controllers notwendig
2	-	reserviert	-
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-Controller	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-Controller tauschen
	1	Interner Datenfehler	Hardware-Reset des BX-Controllers (aus - und wieder einschalten)
	2	DIP-Schalter nach einem Software-Reset verändert	Hardware-Reset des BX-Controllers (aus - und wieder einschalten)
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 übertragen
	1	Falsche oder fehlerhaft Lib implementiert	Entfernen Sie die fehlerhaft Lib
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, Herstellereinstellung mit der Konfigurationssoftware KS2000 setzen
16	n	Länge der K-Bus-Daten stimmt nicht mehr	BX-Controller erneut Starten, falls der Fehler erneut auftritt, Herstellereinstellung mit der Konfigurationssoftware KS2000 setzen

## 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

# 8 Anhang

## 8.1 Schnelleinstieg für erfahrene Anwender

### Zielgruppe

Diese Kurz-Übersicht wendet sich an Anwender, die CAN bereits kennen. Sie zeigt, welche CAN-Nachrichten erforderlich sind, um mit BECKHOFF CANopen Ein-/Ausgabebaugruppen in der Ausgangskonfiguration (mit Default-Einstellungen) zu arbeiten.

In jedem Fall ist die ausführliche Dokumentation zu lesen und zu berücksichtigen!

### Hardware-Konfiguration

Über den DIP-Schalter sind auf den Buskopplern eine einheitliche Übertragungsrate sowie jeweils unterschiedliche Knotenadressen (Node-ID) einzustellen. Die Belegung des Schalters ist auf den Baugruppen aufgedruckt. Dabei ist zu beachten, dass CANopen die Adresse "0" zum Ansprechen aller Baugruppen (Broadcast) verwendet, diese also nicht auf der Baugruppe eingestellt werden darf.

### Start der Baugruppen

CANopen ermöglicht es, die Baugruppen mit einem einzigen Netzwerkmanagement [[▶ 138](#)]-Telegramm zu starten:

11-bit Identifier	2 Byte Nutzdaten						
0x00	0x01	0x00					

Dabei enthält das erste Datenbyte das Start-Kommando (Start\_Remote\_Node), das zweite Datenbyte die Knotenadresse (hier: 0, damit werden alle Knoten angesprochen).

Nachdem die Baugruppen gestartet wurden, sind die Ein- und Ausgänge freigeschaltet. In der Grundkonfiguration (Default-Einstellung) kommunizieren die Baugruppen im ereignisgesteuerten Modus, d.h. Änderungen der digitalen Eingänge werden sofort übermittelt, nach dem Empfang von Telegrammen mit Ausgangsdaten werden die Ausgänge sofort entsprechend gesetzt

### CAN-Identifizier

Die CAN-Identifizier für die Ein- und Ausgangsdaten werden aus der Knotenadresse (1-63) abgeleitet:

Datenart	Default CAN-Identifizier
digitale Eingänge 1...64	0x180 (=384 <sub>dez</sub> ) + Knotenadresse
digitale Ausgänge 1...64	0x200 (=512 <sub>dez</sub> ) + Knotenadresse
analoge Eingänge 1...4	0x280 (=640 <sub>dez</sub> ) + Knotenadresse
analoge Ausgänge 1...4	0x300 (=768 <sub>dez</sub> ) + Knotenadresse
analoge Eingänge 5...8*	0x380 (=896 <sub>dez</sub> ) + Knotenadresse
analoge Ausgänge 5...8*	0x400 (=1024 <sub>dez</sub> ) + Knotenadresse
analoge Eingänge 9...12*	0x480 (=1152 <sub>dez</sub> ) + Knotenadresse
analoge Ausgänge 9...12*	0x500 (=1280 <sub>dez</sub> ) + Knotenadresse

\* falls mehr als 64 digitale Ein- bzw. Ausgänge vorhanden sind verschiebt sich der Bereich entsprechend (siehe Abschnitt Default Mapping).

### Digitale Eingänge

Die CAN-Nachrichten mit digitalen Eingangsdaten stellen sich wie folgt dar:

11-bit Identifier	1-8 Byte Nutzdaten (je nach Zahl der Eingangsklemmen bzw. Erweiterungsbaugruppen)							
0x180(=384 <sub>dez</sub> ) + Node-ID	E0	E1	E2	E3	E4	E5	E6	E7

E0: Eingangsbytes auf Eingangsklemmen (bzw. Feldbus Box Baugruppen), von links nach rechts.

### Digitale Ausgänge

Die CAN-Nachrichten mit digitalen Ausgangsdaten haben folgende Struktur:

11-bit Identifier	1-8 Byte Nutzdaten (je nach Zahl der Ausgangsklemmen bzw. Erweiterungsbaugruppen)							
0x200(=512 <sub>dez</sub> ) + Node-ID	A0	A1	A2	A3	A4	A5	A6	A7

A0: Ausgangsbytes auf Ausgangsklemmen (bzw. Feldbus Box Baugruppen), von links nach rechts.

### Analoge Eingänge

So sehen CAN-Nachrichten mit analogen Eingangsdaten aus:

11-bit Identifier	4-8 Byte Nutzdaten (je nach Zahl der analogen Eingänge)							
0x280(640 <sub>dez</sub> ) + Node-ID	E0.0	E0.1	E1.0	E1.1	E2.0	E2.1	E3.0	E3.1

E x.0...E x.1: Analogeingang x. Die detaillierte Beschreibung des Datenformats findet sich im Objektverzeichnis bei Objekt [0x6401](#) [[▶ 168](#)].

### Sendeverhalten Analogeingänge

Um ein "Überfluten" des Busses mit sich ständig ändernden Analogeingangswerten zu vermeiden, erzeugen die Analogeingangs-Baugruppen nach CANopen im Defaultzustand keine Datentelegramme. Die Analogdaten lassen sich über Remote-Zugriff (Remote Transmit Request, CAN-Nachricht ohne Daten mit gesetztem RTR-Bit) auf die Analogeingangs-Telegramme auslesen. Alternativ kann selbstverständlich auch die Baugruppe so umkonfiguriert werden, dass eine Änderung des Eingangswerts zum Absetzen eines Datentelegramms führt. Hierzu wird ein Wert > 0 auf Index [0x6423](#) [[▶ 168](#)] des Objektverzeichnisses geschrieben. Das entsprechende SDO-Telegramm sieht wie folgt aus:

11-bit Identifier	8 Byte Nutzdaten							
0x600(=768 <sub>dez</sub> ) + Node-ID	0x22	0x23	0x64	0x00	0x01	0x00	0x00	0x00

Es empfiehlt sich, analoge Eingangsdaten statt per Ereignissteuerung (jede Änderung des LSB ist ein Ereignis und führt zum Versenden des entsprechenden Telegramms) zyklisch synchron oder per event timer zu verschicken. Falls die Ereignissteuerung beibehalten werden soll, so sollte die Datenmenge durch Setzen eines Delta-Wertes (Objektverzeichnis Index [0x6426](#) [[▶ 168](#)]), von Grenzwerten ([0x6424](#) [[▶ 168](#)] + [0x6425](#) [[▶ 168](#)]) oder einer Inhibit-Time (erneute Datenübertragung frühestens nach Ablauf der Inhibit-Time, 0x1801ff) reduziert werden. Details zur Parameterkommunikation finden sich im Kapitel [Servicedaten: SDO](#) [[▶ 152](#)].

### Analoge Ausgänge

So sehen CAN-Nachrichten mit analogen Ausgangsdaten aus:

11-bit Identifier	4-8 Byte Nutzdaten (je nach Zahl der analogen Ausgänge)							
0x300(=768 <sub>dez</sub> ) + Node-ID	A0.0	A0.1	A1.0	A1.1	A2.0	A2.1	A3.0	A3.1

A x.0...A x.1: Analogausgang x. Die detaillierte Beschreibung des Datenformats findet sich im Objektverzeichnis bei Objekt [0x6411](#) [[▶ 161](#)].

**Default Identifier**

Eine tabellarische Übersicht über alle Default-Identifizier findet sich im Anhang. Mit Hilfe dieser Übersicht lassen sich die von einem CAN-Monitor angezeigten CAN-Nachrichten schnell und einfach identifizieren.

**Stop der Baugruppen**

Falls erforderlich kann die Prozessdaten-Kommunikation der Baugruppen mit folgendem Telegramm gestoppt werden:

11-bit Identifier	2 Byte Nutzdaten							
0x00	0x80	0xYZ						

0xXX: Knotenadresse; 0xYZ=0x00 adressiert alle Baugruppen

**Guarding**

Die oben beschriebenen Telegramme reichen für viele Anwendungen bereits aus. Da die Baugruppen per Default im ereignisgesteuerten Modus arbeiten (kein zyklischer Datenaustausch), wird ein Ausfall einer Baugruppe jedoch nicht unbedingt erkannt. Abhilfe schafft hier die Überwachung der Baugruppen durch zyklisches Abfragen des Status, das sogenannte Node-Guarding.

Hierzu wird zyklisch ein Statustelegamm über Remote-Transmit-Request (RTR) angefordert:

11-bit Identifier	keine Nutzdaten im Anforderungstelegamm (RTR)
0x700(=1792 <sub>dez</sub> ) + Node-ID	(RTR-Bit im Header gesetzt)

Die Baugruppen antworten mit einem Telegramm, das ein Status-Byte enthält.

11-bit Identifier	1 Byte Nutzdaten
0x700(=1792 <sub>dez</sub> ) + Node-ID	0xYZ

0xYZ: Status-Byte:

Bits 6...0 enthalten den Knoten-Status (0x7F=127:Pre-Operational, 0x05=Operational; 0x04= Stopped bzw. Prepared).

Bit 7 = Toggle Bit (kippt nach jedem Senden).

Damit der Buskoppler einen Ausfall des Netzwerk-Masters erkennt (Watchdog-Funktion), müssen noch die Guard-Time [▶ 163] (Objekt 0x100C) und der Life-Time-Factor [▶ 163] (Objekt 0x100D) auf Werte ungleich 0 gesetzt werden. (Reaktionszeit bei Ausfall: Guard-Time x Life Time Factor).

**Heartbeat**

Alternativ zum Guarding kann die Baugruppe auch über den sogenannten Heartbeat überwacht werden. Hierbei setzt der Knoten zyklisch sein Statustelegamm (Heartbeat) ab. Datenanforderungstelegamme (Remote Frames) sind nicht erforderlich.

Um das Heartbeat-Telegramm zu aktivieren muss die Producer Heartbeat [▶ 165] Time gesetzt werden. Dies erfolgt mit folgendem SDO [▶ 152]-Telegramm:

11-bit Identifier	8 Byte Nutzdaten							
0x600(=768 <sub>dez</sub> ) + Node-ID	0x22	0x17	0x10	0x00	0xcd	0xab	0x00	0x00

wobei 0xabcd die gewünschte Heartbeat-Zykluszeit in Millisekunden ist.

Mit den beschriebenen Telegrammen sind Sie nun in der Lage, die Baugruppen zu starten und zu stoppen, Eingänge zu lesen, Ausgänge zu schreiben und die Baugruppen zu überwachen. Versäumen Sie nicht, das Handbuch sorgfältig zu lesen. Nur so können Sie die vielfältigen Features der BECKHOFF CANopen Buskoppler wirklich nutzen.

## 8.2 CAN Identifier-Liste

Die hier aufgeführte Liste soll bei der Identifizierung und Zuordnung von CANopen Nachrichten helfen. Aufgeführt sind alle von der CANopen Default Identifier Verteilung zugeordneten Identifier, sowie die von BECKHOFF via Objekt [0x5500 \[▶ 179\]](#) vergebenen herstellerspezifischen Default Identifier (nur in Netzen mit Knotenadressen <64 zu verwenden).

In der \*chm-Ausgabe der Dokumentation dienen die folgenden Werte als Suchhilfe und "Einsprungpunkte" in die umfangreiche Identifier-Tabelle:

Dezimal: [400 \[▶ 195\]](#), [500 \[▶ 200\]](#), [600 \[▶ 201\]](#), [700 \[▶ 196\]](#), [800 \[▶ 197\]](#), [900 \[▶ 197\]](#), [1000 \[▶ 202\]](#), [1100 \[▶ 203\]](#), [1200 \[▶ 198\]](#), [1300 \[▶ 199\]](#), [1400 \[▶ 204\]](#), [1500 \[▶ 204\]](#), [1600 \[▶ 205\]](#), [1700 \[▶ 199\]](#), [1800 \[▶ 207\]](#), [1900 \[▶ 206\]](#)

Hexadezimal: [0x181 \[▶ 195\]](#), [0x1C1 \[▶ 200\]](#), [0x201 \[▶ 196\]](#), [0x301 \[▶ 197\]](#), [0x401 \[▶ 198\]](#), [0x501 \[▶ 199\]](#), [0x601 \[▶ 207\]](#), [0x701 \[▶ 207\]](#)

Die Identifier-Verteilung via Objekt [0x5500 \[▶ 179\]](#) folgt diesem Schema:

Objekt	resultierende COB-ID (dez)	resultierende COB-ID (hex)
<a href="#">Emergency [▶ 195]</a>	129 bis 191 [255]	0x81 bis 0xBF [0xFF]
<a href="#">TxPDO1 [▶ 195]</a>	385 bis 447 [511]	0x181 bis 0x1BF [0x1FF]
<a href="#">RxPDO1 [▶ 196]</a>	513 bis 575 [639]	0x201 bis 0x23F [0x27F]
<a href="#">TxPDO2 [▶ 196]</a>	641 bis 676 [767]	0x281 bis 0x2BF [0x2FF]
<a href="#">RxPDO2 [▶ 197]</a>	769 bis 831 [895]	0x301 bis 0x33F [0x37F]
<a href="#">TxPDO3 [▶ 197]</a>	897 bis 959 [1023]	0x381 bis 0x3BF [0x3FF]
<a href="#">RxPDO3 [▶ 198]</a>	1025 bis 1087 [1151]	0x401 bis 0x43F [0x47F]
<a href="#">TxPDO4 [▶ 198]</a>	1153 bis 1215 [1279]	0x481 bis 0x4BF [0x4FF]
<a href="#">RxPDO4 [▶ 199]</a>	1281 bis 1343 [1407]	0x501 bis 0x53F [0x57F]
<a href="#">TxPDO5 [▶ 199]</a>	1665 bis 1727	0x681 bis 0x6BF
<a href="#">RxPDO5 [▶ 200]</a>	1921 bis 1983	0x781 bis 0x7BF
<a href="#">TxPDO6 [▶ 200]</a>	449 bis 511	0x1C1 bis 0x1FF
<a href="#">RxPDO6 [▶ 201]</a>	577 bis 639	0x241 bis 0x27F
<a href="#">TxPDO7 [▶ 201]</a>	705 bis 767	0x2C1 bis 0x2FF
<a href="#">RxPDO7 [▶ 202]</a>	833 bis 895	0x341 bis 0x37F
<a href="#">TxPDO8 [▶ 202]</a>	961 bis 1023	0x3C1 bis 0x3FF
<a href="#">RxPDO8 [▶ 203]</a>	1089 bis 1151	0x441 bis 0x47F
<a href="#">TxPDO9 [▶ 203]</a>	1217 bis 1279	0x4C1 bis 0x4FF
<a href="#">RxPDO9 [▶ 204]</a>	1345 bis 1407	0x541 bis 0x57F
<a href="#">TxPDO10 [▶ 204]</a>	1473 bis 1535	0x5C1 bis 0x5FF
<a href="#">RxPDO10 [▶ 205]</a>	1601 bis 1663	0x641 bis 0x67F
<a href="#">TxPDO11 [▶ 205]</a>	1729 bis 1791	0x6C1 bis 0x6FF
<a href="#">RxPDO11 [▶ 206]</a>	1857 bis 1919	0x741 bis 0x77F
<a href="#">SDO (Tx) [▶ 206]</a>	1409 bis 1471 [1535]	0x581 bis 0x5BF [0x5FF]
<a href="#">SDO (Rx) [▶ 207]</a>	1537 bis 1599 [1663]	0x601 bis 0x63F [0x67F]
<a href="#">Guarding / Heartbeat / Bootup [▶ 207]</a>	1793 bis 1855 [1919]	0x701 bis 0x73F [0x77F]

### Identifierliste

Mit \* gekennzeichnete Identifier werden auf den Buskopplern nach Beschreiben von Index 0x5500 herstellerspezifisch vergeben.

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
0	0x00	NMT	149	0x95	EMCY Nd.21	171	0xAB	EMCY Nd.43
128	0x80	SYNC	150	0x96	EMCY Nd.22	172	0xAC	EMCY Nd.44
129	0x81	EMCY Nd.1	151	0x97	EMCY Nd.23	173	0xAD	EMCY Nd.45
130	0x82	EMCY Nd.2	152	0x98	EMCY Nd.24	174	0xAE	EMCY Nd.46
131	0x83	EMCY Nd.3	153	0x99	EMCY Nd.25	175	0xAF	EMCY Nd.47
132	0x84	EMCY Nd.4	154	0x9A	EMCY Nd.26	176	0xB0	EMCY Nd.48
133	0x85	EMCY Nd.5	155	0x9B	EMCY Nd.27	177	0xB1	EMCY Nd.49
134	0x86	EMCY Nd.6	156	0x9C	EMCY Nd.28	178	0xB2	EMCY Nd.50
135	0x87	EMCY Nd.7	157	0x9D	EMCY Nd.29	179	0xB3	EMCY Nd.51
136	0x88	EMCY Nd.8	158	0x9E	EMCY Nd.30	180	0xB4	EMCY Nd.52
137	0x89	EMCY Nd.9	159	0x9F	EMCY Nd.31	181	0xB5	EMCY Nd.53
138	0x8A	EMCY Nd.10	160	0xA0	EMCY Nd.32	182	0xB6	EMCY Nd.54
139	0x8B	EMCY Nd.11	161	0xA1	EMCY Nd.33	183	0xB7	EMCY Nd.55
140	0x8C	EMCY Nd.12	162	0xA2	EMCY Nd.34	184	0xB8	EMCY Nd.56
141	0x8D	EMCY Nd.13	163	0xA3	EMCY Nd.35	185	0xB9	EMCY Nd.57
142	0x8E	EMCY Nd.14	164	0xA4	EMCY Nd.36	186	0xBA	EMCY Nd.58
143	0x8F	EMCY Nd.15	165	0xA5	EMCY Nd.37	187	0xBB	EMCY Nd.59
144	0x90	EMCY Nd.16	166	0xA6	EMCY Nd.38	188	0xBC	EMCY Nd.60
145	0x91	EMCY Nd.17	167	0xA7	EMCY Nd.39	189	0xBD	EMCY Nd.61
146	0x92	EMCY Nd.18	168	0xA8	EMCY Nd.40	190	0xBE	EMCY Nd.62
147	0x93	EMCY Nd.19	169	0xA9	EMCY Nd.41	191	0xBF	EMCY Nd.63
148	0x94	EMCY Nd.20	170	0xAA	EMCY Nd.42			

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
385	0x181	TxPDO1, DI, Nd.1	406	0x196	TxPDO1, DI, Nd.22	427	0x1AB	TxPDO1, DI, Nd.43
386	0x182	TxPDO1, DI, Nd.2	407	0x197	TxPDO1, DI, Nd.23	428	0x1AC	TxPDO1, DI, Nd.44
387	0x183	TxPDO1, DI, Nd.3	408	0x198	TxPDO1, DI, Nd.24	429	0x1AD	TxPDO1, DI, Nd.45
388	0x184	TxPDO1, DI, Nd.4	409	0x199	TxPDO1, DI, Nd.25	430	0x1AE	TxPDO1, DI, Nd.46
389	0x185	TxPDO1, DI, Nd.5	410	0x19A	TxPDO1, DI, Nd.26	431	0x1AF	TxPDO1, DI, Nd.47
390	0x186	TxPDO1, DI, Nd.6	411	0x19B	TxPDO1, DI, Nd.27	432	0x1B0	TxPDO1, DI, Nd.48
391	0x187	TxPDO1, DI, Nd.7	412	0x19C	TxPDO1, DI, Nd.28	433	0x1B1	TxPDO1, DI, Nd.49
392	0x188	TxPDO1, DI, Nd.8	413	0x19D	TxPDO1, DI, Nd.29	434	0x1B2	TxPDO1, DI, Nd.50
393	0x189	TxPDO1, DI, Nd.9	414	0x19E	TxPDO1, DI, Nd.30	435	0x1B3	TxPDO1, DI, Nd.51
394	0x18A	TxPDO1, DI, Nd.10	415	0x19F	TxPDO1, DI, Nd.31	436	0x1B4	TxPDO1, DI, Nd.52
395	0x18B	TxPDO1, DI, Nd.11	416	0x1A0	TxPDO1, DI, Nd.32	437	0x1B5	TxPDO1, DI, Nd.53
396	0x18C	TxPDO1, DI, Nd.12	417	0x1A1	TxPDO1, DI, Nd.33	438	0x1B6	TxPDO1, DI, Nd.54
397	0x18D	TxPDO1, DI, Nd.13	418	0x1A2	TxPDO1, DI, Nd.34	439	0x1B7	TxPDO1, DI, Nd.55
398	0x18E	TxPDO1, DI, Nd.14	419	0x1A3	TxPDO1, DI, Nd.35	440	0x1B8	TxPDO1, DI, Nd.56
399	0x18F	TxPDO1, DI, Nd.15	420	0x1A4	TxPDO1, DI, Nd.36	441	0x1B9	TxPDO1, DI, Nd.57
400	0x190	TxPDO1, DI, Nd.16	421	0x1A5	TxPDO1, DI, Nd.37	442	0x1BA	TxPDO1, DI, Nd.58
401	0x191	TxPDO1, DI, Nd.17	422	0x1A6	TxPDO1, DI, Nd.38	443	0x1BB	TxPDO1, DI, Nd.59
402	0x192	TxPDO1, DI, Nd.18	423	0x1A7	TxPDO1, DI, Nd.39	444	0x1BC	TxPDO1, DI, Nd.60
403	0x193	TxPDO1, DI, Nd.19	424	0x1A8	TxPDO1, DI, Nd.40	445	0x1BD	TxPDO1, DI, Nd.61
404	0x194	TxPDO1, DI, Nd.20	425	0x1A9	TxPDO1, DI, Nd.41	446	0x1BE	TxPDO1, DI, Nd.62
405	0x195	TxPDO1, DI, Nd.21	426	0x1AA	TxPDO1, DI, Nd.42	447	0x1BF	TxPDO1, DI, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
513	0x201	RxPDO1, DO, Nd.1	534	0x216	RxPDO1, DO, Nd.22	555	0x22B	RxPDO1, DO, Nd.43
514	0x202	RxPDO1, DO, Nd.2	535	0x217	RxPDO1, DO, Nd.23	556	0x22C	RxPDO1, DO, Nd.44
515	0x203	RxPDO1, DO, Nd.3	536	0x218	RxPDO1, DO, Nd.24	557	0x22D	RxPDO1, DO, Nd.45
516	0x204	RxPDO1, DO, Nd.4	537	0x219	RxPDO1, DO, Nd.25	558	0x22E	RxPDO1, DO, Nd.46
517	0x205	RxPDO1, DO, Nd.5	538	0x21A	RxPDO1, DO, Nd.26	559	0x22F	RxPDO1, DO, Nd.47
518	0x206	RxPDO1, DO, Nd.6	539	0x21B	RxPDO1, DO, Nd.27	560	0x230	RxPDO1, DO, Nd.48
519	0x207	RxPDO1, DO, Nd.7	540	0x21C	RxPDO1, DO, Nd.28	561	0x231	RxPDO1, DO, Nd.49
520	0x208	RxPDO1, DO, Nd.8	541	0x21D	RxPDO1, DO, Nd.29	562	0x232	RxPDO1, DO, Nd.50
521	0x209	RxPDO1, DO, Nd.9	542	0x21E	RxPDO1, DO, Nd.30	563	0x233	RxPDO1, DO, Nd.51
522	0x20A	RxPDO1, DO, Nd.10	543	0x21F	RxPDO1, DO, Nd.31	564	0x234	RxPDO1, DO, Nd.52
523	0x20B	RxPDO1, DO, Nd.11	544	0x220	RxPDO1, DO, Nd.32	565	0x235	RxPDO1, DO, Nd.53
524	0x20C	RxPDO1, DO, Nd.12	545	0x221	RxPDO1, DO, Nd.33	566	0x236	RxPDO1, DO, Nd.54
525	0x20D	RxPDO1, DO, Nd.13	546	0x222	RxPDO1, DO, Nd.34	567	0x237	RxPDO1, DO, Nd.55
526	0x20E	RxPDO1, DO, Nd.14	547	0x223	RxPDO1, DO, Nd.35	568	0x238	RxPDO1, DO, Nd.56
527	0x20F	RxPDO1, DO, Nd.15	548	0x224	RxPDO1, DO, Nd.36	569	0x239	RxPDO1, DO, Nd.57
528	0x210	RxPDO1, DO, Nd.16	549	0x225	RxPDO1, DO, Nd.37	570	0x23A	RxPDO1, DO, Nd.58
529	0x211	RxPDO1, DO, Nd.17	550	0x226	RxPDO1, DO, Nd.38	571	0x23B	RxPDO1, DO, Nd.59
530	0x212	RxPDO1, DO, Nd.18	551	0x227	RxPDO1, DO, Nd.39	572	0x23C	RxPDO1, DO, Nd.60
531	0x213	RxPDO1, DO, Nd.19	552	0x228	RxPDO1, DO, Nd.40	573	0x23D	RxPDO1, DO, Nd.61
532	0x214	RxPDO1, DO, Nd.20	553	0x229	RxPDO1, DO, Nd.41	574	0x23E	RxPDO1, DO, Nd.62
533	0x215	RxPDO1, DO, Nd.21	554	0x22A	RxPDO1, DO, Nd.42	575	0x23F	RxPDO1, DO, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
641	0x281	TxPDO2, AI, Nd.1	662	0x296	TxPDO2, AI, Nd.22	683	0x2AB	TxPDO2, AI, Nd.43
642	0x282	TxPDO2, AI, Nd.2	663	0x297	TxPDO2, AI, Nd.23	684	0x2AC	TxPDO2, AI, Nd.44
643	0x283	TxPDO2, AI, Nd.3	664	0x298	TxPDO2, AI, Nd.24	685	0x2AD	TxPDO2, AI, Nd.45
644	0x284	TxPDO2, AI, Nd.4	665	0x299	TxPDO2, AI, Nd.25	686	0x2AE	TxPDO2, AI, Nd.46
645	0x285	TxPDO2, AI, Nd.5	666	0x29A	TxPDO2, AI, Nd.26	687	0x2AF	TxPDO2, AI, Nd.47
646	0x286	TxPDO2, AI, Nd.6	667	0x29B	TxPDO2, AI, Nd.27	688	0x2B0	TxPDO2, AI, Nd.48
647	0x287	TxPDO2, AI, Nd.7	668	0x29C	TxPDO2, AI, Nd.28	689	0x2B1	TxPDO2, AI, Nd.49
648	0x288	TxPDO2, AI, Nd.8	669	0x29D	TxPDO2, AI, Nd.29	690	0x2B2	TxPDO2, AI, Nd.50
649	0x289	TxPDO2, AI, Nd.9	670	0x29E	TxPDO2, AI, Nd.30	691	0x2B3	TxPDO2, AI, Nd.51
650	0x28A	TxPDO2, AI, Nd.10	671	0x29F	TxPDO2, AI, Nd.31	692	0x2B4	TxPDO2, AI, Nd.52
651	0x28B	TxPDO2, AI, Nd.11	672	0x2A0	TxPDO2, AI, Nd.32	693	0x2B5	TxPDO2, AI, Nd.53
652	0x28C	TxPDO2, AI, Nd.12	673	0x2A1	TxPDO2, AI, Nd.33	694	0x2B6	TxPDO2, AI, Nd.54
653	0x28D	TxPDO2, AI, Nd.13	674	0x2A2	TxPDO2, AI, Nd.34	695	0x2B7	TxPDO2, AI, Nd.55
654	0x28E	TxPDO2, AI, Nd.14	675	0x2A3	TxPDO2, AI, Nd.35	696	0x2B8	TxPDO2, AI, Nd.56
655	0x28F	TxPDO2, AI, Nd.15	676	0x2A4	TxPDO2, AI, Nd.36	697	0x2B9	TxPDO2, AI, Nd.57
656	0x290	TxPDO2, AI, Nd.16	677	0x2A5	TxPDO2, AI, Nd.37	698	0x2BA	TxPDO2, AI, Nd.58
657	0x291	TxPDO2, AI, Nd.17	678	0x2A6	TxPDO2, AI, Nd.38	699	0x2BB	TxPDO2, AI, Nd.59
658	0x292	TxPDO2, AI, Nd.18	679	0x2A7	TxPDO2, AI, Nd.39	700	0x2BC	TxPDO2, AI, Nd.60
659	0x293	TxPDO2, AI, Nd.19	680	0x2A8	TxPDO2, AI, Nd.40	701	0x2BD	TxPDO2, AI, Nd.61
660	0x294	TxPDO2, AI, Nd.20	681	0x2A9	TxPDO2, AI, Nd.41	702	0x2BE	TxPDO2, AI, Nd.62
661	0x295	TxPDO2, AI, Nd.21	682	0x2AA	TxPDO2, AI, Nd.42	703	0x2BF	TxPDO2, AI, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
769	0x301	RxPDO2, AO, Nd.1	790	0x316	RxPDO2, AO, Nd.22	811	0x32B	RxPDO2, AO, Nd.43
770	0x302	RxPDO2, AO, Nd.2	791	0x317	RxPDO2, AO, Nd.23	812	0x32C	RxPDO2, AO, Nd.44
771	0x303	RxPDO2, AO, Nd.3	792	0x318	RxPDO2, AO, Nd.24	813	0x32D	RxPDO2, AO, Nd.45
772	0x304	RxPDO2, AO, Nd.4	793	0x319	RxPDO2, AO, Nd.25	814	0x32E	RxPDO2, AO, Nd.46
773	0x305	RxPDO2, AO, Nd.5	794	0x31A	RxPDO2, AO, Nd.26	815	0x32F	RxPDO2, AO, Nd.47
774	0x306	RxPDO2, AO, Nd.6	795	0x31B	RxPDO2, AO, Nd.27	816	0x330	RxPDO2, AO, Nd.48
775	0x307	RxPDO2, AO, Nd.7	796	0x31C	RxPDO2, AO, Nd.28	817	0x331	RxPDO2, AO, Nd.49
776	0x308	RxPDO2, AO, Nd.8	797	0x31D	RxPDO2, AO, Nd.29	818	0x332	RxPDO2, AO, Nd.50
777	0x309	RxPDO2, AO, Nd.9	798	0x31E	RxPDO2, AO, Nd.30	819	0x333	RxPDO2, AO, Nd.51
778	0x30A	RxPDO2, AO, Nd.10	799	0x31F	RxPDO2, AO, Nd.31	820	0x334	RxPDO2, AO, Nd.52
779	0x30B	RxPDO2, AO, Nd.11	800	0x320	RxPDO2, AO, Nd.32	821	0x335	RxPDO2, AO, Nd.53
780	0x30C	RxPDO2, AO, Nd.12	801	0x321	RxPDO2, AO, Nd.33	822	0x336	RxPDO2, AO, Nd.54
781	0x30D	RxPDO2, AO, Nd.13	802	0x322	RxPDO2, AO, Nd.34	823	0x337	RxPDO2, AO, Nd.55
782	0x30E	RxPDO2, AO, Nd.14	803	0x323	RxPDO2, AO, Nd.35	824	0x338	RxPDO2, AO, Nd.56
783	0x30F	RxPDO2, AO, Nd.15	804	0x324	RxPDO2, AO, Nd.36	825	0x339	RxPDO2, AO, Nd.57
784	0x310	RxPDO2, AO, Nd.16	805	0x325	RxPDO2, AO, Nd.37	826	0x33A	RxPDO2, AO, Nd.58
785	0x311	RxPDO2, AO, Nd.17	806	0x326	RxPDO2, AO, Nd.38	827	0x33B	RxPDO2, AO, Nd.59
786	0x312	RxPDO2, AO, Nd.18	807	0x327	RxPDO2, AO, Nd.39	828	0x33C	RxPDO2, AO, Nd.60
787	0x313	RxPDO2, AO, Nd.19	808	0x328	RxPDO2, AO, Nd.40	829	0x33D	RxPDO2, AO, Nd.61
788	0x314	RxPDO2, AO, Nd.20	809	0x329	RxPDO2, AO, Nd.41	830	0x33E	RxPDO2, AO, Nd.62
789	0x315	RxPDO2, AO, Nd.21	810	0x32A	RxPDO2, AO, Nd.42	831	0x33F	RxPDO2, AO, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
897	0x381	TxPDO3*, Nd.1	918	0x396	TxPDO3*, Nd.22	939	0x3AB	TxPDO3*, Nd.43
898	0x382	TxPDO3*, Nd.2	919	0x397	TxPDO3*, Nd.23	940	0x3AC	TxPDO3*, Nd.44
899	0x383	TxPDO3*, Nd.3	920	0x398	TxPDO3*, Nd.24	941	0x3AD	TxPDO3*, Nd.45
900	0x384	TxPDO3*, Nd.4	921	0x399	TxPDO3*, Nd.25	942	0x3AE	TxPDO3*, Nd.46
901	0x385	TxPDO3*, Nd.5	922	0x39A	TxPDO3*, Nd.26	943	0x3AF	TxPDO3*, Nd.47
902	0x386	TxPDO3*, Nd.6	923	0x39B	TxPDO3*, Nd.27	944	0x3B0	TxPDO3*, Nd.48
903	0x387	TxPDO3*, Nd.7	924	0x39C	TxPDO3*, Nd.28	945	0x3B1	TxPDO3*, Nd.49
904	0x388	TxPDO3*, Nd.8	925	0x39D	TxPDO3*, Nd.29	946	0x3B2	TxPDO3*, Nd.50
905	0x389	TxPDO3*, Nd.9	926	0x39E	TxPDO3*, Nd.30	947	0x3B3	TxPDO3*, Nd.51
906	0x38A	TxPDO3*, Nd.10	927	0x39F	TxPDO3*, Nd.31	948	0x3B4	TxPDO3*, Nd.52
907	0x38B	TxPDO3*, Nd.11	928	0x3A0	TxPDO3*, Nd.32	949	0x3B5	TxPDO3*, Nd.53
908	0x38C	TxPDO3*, Nd.12	929	0x3A1	TxPDO3*, Nd.33	950	0x3B6	TxPDO3*, Nd.54
909	0x38D	TxPDO3*, Nd.13	930	0x3A2	TxPDO3*, Nd.34	951	0x3B7	TxPDO3*, Nd.55
910	0x38E	TxPDO3*, Nd.14	931	0x3A3	TxPDO3*, Nd.35	952	0x3B8	TxPDO3*, Nd.56
911	0x38F	TxPDO3*, Nd.15	932	0x3A4	TxPDO3*, Nd.36	953	0x3B9	TxPDO3*, Nd.57
912	0x390	TxPDO3*, Nd.16	933	0x3A5	TxPDO3*, Nd.37	954	0x3BA	TxPDO3*, Nd.58
913	0x391	TxPDO3*, Nd.17	934	0x3A6	TxPDO3*, Nd.38	955	0x3BB	TxPDO3*, Nd.59
914	0x392	TxPDO3*, Nd.18	935	0x3A7	TxPDO3*, Nd.39	956	0x3BC	TxPDO3*, Nd.60
915	0x393	TxPDO3*, Nd.19	936	0x3A8	TxPDO3*, Nd.40	957	0x3BD	TxPDO3*, Nd.61
916	0x394	TxPDO3*, Nd.20	937	0x3A9	TxPDO3*, Nd.41	958	0x3BE	TxPDO3*, Nd.62
917	0x395	TxPDO3*, Nd.21	938	0x3AA	TxPDO3*, Nd.42	959	0x3BF	TxPDO3*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1025	0x401	RxPDO3*, Nd.1	1046	0x416	RxPDO3*, Nd.22	1067	0x42B	RxPDO3*, Nd.43
1026	0x402	RxPDO3*, Nd.2	1047	0x417	RxPDO3*, Nd.23	1068	0x42C	RxPDO3*, Nd.44
1027	0x403	RxPDO3*, Nd.3	1048	0x418	RxPDO3*, Nd.24	1069	0x42D	RxPDO3*, Nd.45
1028	0x404	RxPDO3*, Nd.4	1049	0x419	RxPDO3*, Nd.25	1070	0x42E	RxPDO3*, Nd.46
1029	0x405	RxPDO3*, Nd.5	1050	0x41A	RxPDO3*, Nd.26	1071	0x42F	RxPDO3*, Nd.47
1030	0x406	RxPDO3*, Nd.6	1051	0x41B	RxPDO3*, Nd.27	1072	0x430	RxPDO3*, Nd.48
1031	0x407	RxPDO3*, Nd.7	1052	0x41C	RxPDO3*, Nd.28	1073	0x431	RxPDO3*, Nd.49
1032	0x408	RxPDO3*, Nd.8	1053	0x41D	RxPDO3*, Nd.29	1074	0x432	RxPDO3*, Nd.50
1033	0x409	RxPDO3*, Nd.9	1054	0x41E	RxPDO3*, Nd.30	1075	0x433	RxPDO3*, Nd.51
1034	0x40A	RxPDO3*, Nd.10	1055	0x41F	RxPDO3*, Nd.31	1076	0x434	RxPDO3*, Nd.52
1035	0x40B	RxPDO3*, Nd.11	1056	0x420	RxPDO3*, Nd.32	1077	0x435	RxPDO3*, Nd.53
1036	0x40C	RxPDO3*, Nd.12	1057	0x421	RxPDO3*, Nd.33	1078	0x436	RxPDO3*, Nd.54
1037	0x40D	RxPDO3*, Nd.13	1058	0x422	RxPDO3*, Nd.34	1079	0x437	RxPDO3*, Nd.55
1038	0x40E	RxPDO3*, Nd.14	1059	0x423	RxPDO3*, Nd.35	1080	0x438	RxPDO3*, Nd.56
1039	0x40F	RxPDO3*, Nd.15	1060	0x424	RxPDO3*, Nd.36	1081	0x439	RxPDO3*, Nd.57
1040	0x410	RxPDO3*, Nd.16	1061	0x425	RxPDO3*, Nd.37	1082	0x43A	RxPDO3*, Nd.58
1041	0x411	RxPDO3*, Nd.17	1062	0x426	RxPDO3*, Nd.38	1083	0x43B	RxPDO3*, Nd.59
1042	0x412	RxPDO3*, Nd.18	1063	0x427	RxPDO3*, Nd.39	1084	0x43C	RxPDO3*, Nd.60
1043	0x413	RxPDO3*, Nd.19	1064	0x428	RxPDO3*, Nd.40	1085	0x43D	RxPDO3*, Nd.61
1044	0x414	RxPDO3*, Nd.20	1065	0x429	RxPDO3*, Nd.41	1086	0x43E	RxPDO3*, Nd.62
1045	0x415	RxPDO3*, Nd.21	1066	0x42A	RxPDO3*, Nd.42	1087	0x43F	RxPDO3*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1153	0x481	TxPDO4*, Nd.1	1174	0x496	TxPDO4*, Nd.22	1195	0x4AB	TxPDO4*, Nd.43
1154	0x482	TxPDO4*, Nd.2	1175	0x497	TxPDO4*, Nd.23	1196	0x4AC	TxPDO4*, Nd.44
1155	0x483	TxPDO4*, Nd.3	1176	0x498	TxPDO4*, Nd.24	1197	0x4AD	TxPDO4*, Nd.45
1156	0x484	TxPDO4*, Nd.4	1177	0x499	TxPDO4*, Nd.25	1198	0x4AE	TxPDO4*, Nd.46
1157	0x485	TxPDO4*, Nd.5	1178	0x49A	TxPDO4*, Nd.26	1199	0x4AF	TxPDO4*, Nd.47
1158	0x486	TxPDO4*, Nd.6	1179	0x49B	TxPDO4*, Nd.27	1200	0x4B0	TxPDO4*, Nd.48
1159	0x487	TxPDO4*, Nd.7	1180	0x49C	TxPDO4*, Nd.28	1201	0x4B1	TxPDO4*, Nd.49
1160	0x488	TxPDO4*, Nd.8	1181	0x49D	TxPDO4*, Nd.29	1202	0x4B2	TxPDO4*, Nd.50
1161	0x489	TxPDO4*, Nd.9	1182	0x49E	TxPDO4*, Nd.30	1203	0x4B3	TxPDO4*, Nd.51
1162	0x48A	TxPDO4*, Nd.10	1183	0x49F	TxPDO4*, Nd.31	1204	0x4B4	TxPDO4*, Nd.52
1163	0x48B	TxPDO4*, Nd.11	1184	0x4A0	TxPDO4*, Nd.32	1205	0x4B5	TxPDO4*, Nd.53
1164	0x48C	TxPDO4*, Nd.12	1185	0x4A1	TxPDO4*, Nd.33	1206	0x4B6	TxPDO4*, Nd.54
1165	0x48D	TxPDO4*, Nd.13	1186	0x4A2	TxPDO4*, Nd.34	1207	0x4B7	TxPDO4*, Nd.55
1166	0x48E	TxPDO4*, Nd.14	1187	0x4A3	TxPDO4*, Nd.35	1208	0x4B8	TxPDO4*, Nd.56
1167	0x48F	TxPDO4*, Nd.15	1188	0x4A4	TxPDO4*, Nd.36	1209	0x4B9	TxPDO4*, Nd.57
1168	0x490	TxPDO4*, Nd.16	1189	0x4A5	TxPDO4*, Nd.37	1210	0x4BA	TxPDO4*, Nd.58
1169	0x491	TxPDO4*, Nd.17	1190	0x4A6	TxPDO4*, Nd.48	1211	0x4BB	TxPDO4*, Nd.59
1170	0x492	TxPDO4*, Nd.18	1191	0x4A7	TxPDO4*, Nd.49	1212	0x4BC	TxPDO4*, Nd.60
1171	0x493	TxPDO4*, Nd.19	1192	0x4A8	TxPDO4*, Nd.40	1213	0x4BD	TxPDO4*, Nd.61
1172	0x494	TxPDO4*, Nd.20	1193	0x4A9	TxPDO4*, Nd.41	1214	0x4BE	TxPDO4*, Nd.62
1173	0x495	TxPDO4*, Nd.21	1194	0x4AA	TxPDO4*, Nd.42	1215	0x4BF	TxPDO4*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1281	0x501	RxPDO4*, Nd.1	1302	0x516	RxPDO4*, Nd.22	1323	0x52B	RxPDO4*, Nd.43
1282	0x502	RxPDO4*, Nd.2	1303	0x517	RxPDO4*, Nd.23	1324	0x52C	RxPDO4*, Nd.44
1283	0x503	RxPDO4*, Nd.3	1304	0x518	RxPDO4*, Nd.24	1325	0x52D	RxPDO4*, Nd.45
1284	0x504	RxPDO4*, Nd.4	1305	0x519	RxPDO4*, Nd.25	1326	0x52E	RxPDO4*, Nd.46
1285	0x505	RxPDO4*, Nd.5	1306	0x51A	RxPDO4*, Nd.26	1327	0x52F	RxPDO4*, Nd.47
1286	0x506	RxPDO4*, Nd.6	1307	0x51B	RxPDO4*, Nd.27	1328	0x530	RxPDO4*, Nd.48
1287	0x507	RxPDO4*, Nd.7	1308	0x51C	RxPDO4*, Nd.28	1329	0x531	RxPDO4*, Nd.49
1288	0x508	RxPDO4*, Nd.8	1309	0x51D	RxPDO4*, Nd.29	1330	0x532	RxPDO4*, Nd.50
1289	0x509	RxPDO4*, Nd.9	1310	0x51E	RxPDO4*, Nd.30	1331	0x533	RxPDO4*, Nd.51
1290	0x50A	RxPDO4*, Nd.10	1311	0x51F	RxPDO4*, Nd.31	1332	0x534	RxPDO4*, Nd.52
1291	0x50B	RxPDO4*, Nd.11	1312	0x520	RxPDO4*, Nd.32	1333	0x535	RxPDO4*, Nd.53
1292	0x50C	RxPDO4*, Nd.12	1313	0x521	RxPDO4*, Nd.33	1334	0x536	RxPDO4*, Nd.54
1293	0x50D	RxPDO4*, Nd.13	1314	0x522	RxPDO4*, Nd.34	1335	0x537	RxPDO4*, Nd.55
1294	0x50E	RxPDO4*, Nd.14	1315	0x523	RxPDO4*, Nd.35	1336	0x538	RxPDO4*, Nd.56
1295	0x50F	RxPDO4*, Nd.15	1316	0x524	RxPDO4*, Nd.36	1337	0x539	RxPDO4*, Nd.57
1296	0x510	RxPDO4*, Nd.16	1317	0x525	RxPDO4*, Nd.37	1338	0x53A	RxPDO4*, Nd.58
1297	0x511	RxPDO4*, Nd.17	1318	0x526	RxPDO4*, Nd.38	1339	0x53B	RxPDO4*, Nd.59
1298	0x512	RxPDO4*, Nd.18	1319	0x527	RxPDO4*, Nd.39	1340	0x53C	RxPDO4*, Nd.60
1299	0x513	RxPDO4*, Nd.19	1320	0x528	RxPDO4*, Nd.40	1341	0x53D	RxPDO4*, Nd.61
1300	0x514	RxPDO4*, Nd.20	1321	0x529	RxPDO4*, Nd.41	1342	0x53E	RxPDO4*, Nd.62
1301	0x515	RxPDO4*, Nd.21	1322	0x52A	RxPDO4*, Nd.42	1343	0x53F	RxPDO4*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1665	0x681	TxPDO5*, Nd.1	1686	0x696	TxPDO5*, Nd.22	1707	0x6AB	TxPDO5*, Nd.43
1666	0x682	TxPDO5*, Nd.2	1687	0x697	TxPDO5*, Nd.23	1708	0x6AC	TxPDO5*, Nd.44
1667	0x683	TxPDO5*, Nd.3	1688	0x698	TxPDO5*, Nd.24	1709	0x6AD	TxPDO5*, Nd.45
1668	0x684	TxPDO5*, Nd.4	1689	0x699	TxPDO5*, Nd.25	1710	0x6AE	TxPDO5*, Nd.46
1669	0x685	TxPDO5*, Nd.5	1690	0x69A	TxPDO5*, Nd.26	1711	0x6AF	TxPDO5*, Nd.47
1670	0x686	TxPDO5*, Nd.6	1691	0x69B	TxPDO5*, Nd.27	1712	0x6B0	TxPDO5*, Nd.48
1671	0x687	TxPDO5*, Nd.7	1692	0x69C	TxPDO5*, Nd.28	1713	0x6B1	TxPDO5*, Nd.49
1672	0x688	TxPDO5*, Nd.8	1693	0x69D	TxPDO5*, Nd.29	1714	0x6B2	TxPDO5*, Nd.50
1673	0x689	TxPDO5*, Nd.9	1694	0x69E	TxPDO5*, Nd.30	1715	0x6B3	TxPDO5*, Nd.51
1674	0x68A	TxPDO5*, Nd.10	1695	0x69F	TxPDO5*, Nd.31	1716	0x6B4	TxPDO5*, Nd.52
1675	0x68B	TxPDO5*, Nd.11	1696	0x6A0	TxPDO5*, Nd.32	1717	0x6B5	TxPDO5*, Nd.53
1676	0x68C	TxPDO5*, Nd.12	1697	0x6A1	TxPDO5*, Nd.33	1718	0x6B6	TxPDO5*, Nd.54
1677	0x68D	TxPDO5*, Nd.13	1698	0x6A2	TxPDO5*, Nd.34	1719	0x6B7	TxPDO5*, Nd.55
1678	0x68E	TxPDO5*, Nd.14	1699	0x6A3	TxPDO5*, Nd.35	1720	0x6B8	TxPDO5*, Nd.56
1679	0x68F	TxPDO5*, Nd.15	1700	0x6A4	TxPDO5*, Nd.36	1721	0x6B9	TxPDO5*, Nd.57
1680	0x690	TxPDO5*, Nd.16	1701	0x6A5	TxPDO5*, Nd.37	1722	0x6BA	TxPDO5*, Nd.58
1681	0x691	TxPDO5*, Nd.17	1702	0x6A6	TxPDO5*, Nd.38	1723	0x6BB	TxPDO5*, Nd.59
1682	0x692	TxPDO5*, Nd.18	1703	0x6A7	TxPDO5*, Nd.39	1724	0x6BC	TxPDO5*, Nd.60
1683	0x693	TxPDO5*, Nd.19	1704	0x6A8	TxPDO5*, Nd.40	1725	0x6BD	TxPDO5*, Nd.61
1684	0x694	TxPDO5*, Nd.20	1705	0x6A9	TxPDO5*, Nd.41	1726	0x6BE	TxPDO5*, Nd.62
1685	0x695	TxPDO5*, Nd.21	1706	0x6AA	TxPDO5*, Nd.42	1727	0x6BF	TxPDO5*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1921	0x781	RxPDO5*, Nd.1	1942	0x796	RxPDO5*, Nd.22	1963	0x7AB	RxPDO5*, Nd.43
1922	0x782	RxPDO5*, Nd.2	1943	0x797	RxPDO5*, Nd.23	1964	0x7AC	RxPDO5*, Nd.44
1923	0x783	RxPDO5*, Nd.3	1944	0x798	RxPDO5*, Nd.24	1965	0x7AD	RxPDO5*, Nd.45
1924	0x784	RxPDO5*, Nd.4	1945	0x799	RxPDO5*, Nd.25	1966	0x7AE	RxPDO5*, Nd.46
1925	0x785	RxPDO5*, Nd.5	1946	0x79A	RxPDO5*, Nd.26	1967	0x7AF	RxPDO5*, Nd.47
1926	0x786	RxPDO5*, Nd.6	1947	0x79B	RxPDO5*, Nd.27	1968	0x7B0	RxPDO5*, Nd.48
1927	0x787	RxPDO5*, Nd.7	1948	0x79C	RxPDO5*, Nd.28	1969	0x7B1	RxPDO5*, Nd.49
1928	0x788	RxPDO5*, Nd.8	1949	0x79D	RxPDO5*, Nd.29	1970	0x7B2	RxPDO5*, Nd.50
1929	0x789	RxPDO5*, Nd.9	1950	0x79E	RxPDO5*, Nd.30	1971	0x7B3	RxPDO5*, Nd.51
1930	0x78A	RxPDO5*, Nd.10	1951	0x79F	RxPDO5*, Nd.31	1972	0x7B4	RxPDO5*, Nd.52
1931	0x78B	RxPDO5*, Nd.11	1952	0x7A0	RxPDO5*, Nd.32	1973	0x7B5	RxPDO5*, Nd.53
1932	0x78C	RxPDO5*, Nd.12	1953	0x7A1	RxPDO5*, Nd.33	1974	0x7B6	RxPDO5*, Nd.54
1933	0x78D	RxPDO5*, Nd.13	1954	0x7A2	RxPDO5*, Nd.34	1975	0x7B7	RxPDO5*, Nd.55
1934	0x78E	RxPDO5*, Nd.14	1955	0x7A3	RxPDO5*, Nd.35	1976	0x7B8	RxPDO5*, Nd.56
1935	0x78F	RxPDO5*, Nd.15	1956	0x7A4	RxPDO5*, Nd.36	1977	0x7B9	RxPDO5*, Nd.57
1936	0x790	RxPDO5*, Nd.16	1957	0x7A5	RxPDO5*, Nd.37	1978	0x7BA	RxPDO5*, Nd.58
1937	0x791	RxPDO5*, Nd.17	1958	0x7A6	RxPDO5*, Nd.38	1979	0x7BB	RxPDO5*, Nd.59
1938	0x792	RxPDO5*, Nd.18	1959	0x7A7	RxPDO5*, Nd.39	1980	0x7BC	RxPDO5*, Nd.60
1939	0x793	RxPDO5*, Nd.19	1960	0x7A8	RxPDO5*, Nd.40	1981	0x7BD	RxPDO5*, Nd.61
1940	0x794	RxPDO5*, Nd.20	1961	0x7A9	RxPDO5*, Nd.41	1982	0x7BE	RxPDO5*, Nd.62
1941	0x795	RxPDO5*, Nd.21	1962	0x7AA	RxPDO5*, Nd.42	1983	0x7BF	RxPDO5*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
449	0x1C1	TxPDO6*, Nd.1	470	0x1D6	TxPDO6*, Nd.22	491	0x1EB	TxPDO6*, Nd.43
450	0x1C2	TxPDO6*, Nd.2	471	0x1D7	TxPDO6*, Nd.23	492	0x1EC	TxPDO6*, Nd.44
451	0x1C3	TxPDO6*, Nd.3	472	0x1D8	TxPDO6*, Nd.24	493	0x1ED	TxPDO6*, Nd.45
452	0x1C4	TxPDO6*, Nd.4	473	0x1D9	TxPDO6*, Nd.25	494	0x1EE	TxPDO6*, Nd.46
453	0x1C5	TxPDO6*, Nd.5	474	0x1DA	TxPDO6*, Nd.26	495	0x1EF	TxPDO6*, Nd.47
454	0x1C6	TxPDO6*, Nd.6	475	0x1DB	TxPDO6*, Nd.27	496	0x1F0	TxPDO6*, Nd.48
455	0x1C7	TxPDO6*, Nd.7	476	0x1DC	TxPDO6*, Nd.28	497	0x1F1	TxPDO6*, Nd.49
456	0x1C8	TxPDO6*, Nd.8	477	0x1DD	TxPDO6*, Nd.29	498	0x1F2	TxPDO6*, Nd.50
457	0x1C9	TxPDO6*, Nd.9	478	0x1DE	TxPDO6*, Nd.30	499	0x1F3	TxPDO6*, Nd.51
458	0x1CA	TxPDO6*, Nd.10	479	0x1DF	TxPDO6*, Nd.31	500	0x1F4	TxPDO6*, Nd.52
459	0x1CB	TxPDO6*, Nd.11	480	0x1E0	TxPDO6*, Nd.32	501	0x1F5	TxPDO6*, Nd.53
460	0x1CC	TxPDO6*, Nd.12	481	0x1E1	TxPDO6*, Nd.33	502	0x1F6	TxPDO6*, Nd.54
461	0x1CD	TxPDO6*, Nd.13	482	0x1E2	TxPDO6*, Nd.34	503	0x1F7	TxPDO6*, Nd.55
462	0x1CE	TxPDO6*, Nd.14	483	0x1E3	TxPDO6*, Nd.35	504	0x1F8	TxPDO6*, Nd.56
463	0x1CF	TxPDO6*, Nd.15	484	0x1E4	TxPDO6*, Nd.36	505	0x1F9	TxPDO6*, Nd.57
464	0x1D0	TxPDO6*, Nd.16	485	0x1E5	TxPDO6*, Nd.37	506	0x1FA	TxPDO6*, Nd.58
465	0x1D1	TxPDO6*, Nd.17	486	0x1E6	TxPDO6*, Nd.38	507	0x1FB	TxPDO6*, Nd.59
466	0x1D2	TxPDO6*, Nd.18	487	0x1E7	TxPDO6*, Nd.39	508	0x1FC	TxPDO6*, Nd.60
467	0x1D3	TxPDO6*, Nd.19	488	0x1E8	TxPDO6*, Nd.40	509	0x1FD	TxPDO6*, Nd.61
468	0x1D4	TxPDO6*, Nd.20	489	0x1E9	TxPDO6*, Nd.41	510	0x1FE	TxPDO6*, Nd.62
469	0x1D5	TxPDO6*, Nd.21	490	0x1EA	TxPDO6*, Nd.42	511	0x1FF	TxPDO6*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
577	0x241	RxPDO6*, Nd.1	598	0x256	RxPDO6*, Nd.22	619	0x26B	RxPDO6* Nd.43
578	0x242	RxPDO6*, Nd.2	599	0x257	RxPDO6*, Nd.23	620	0x26C	RxPDO6, Nd.44
579	0x243	RxPDO6*, Nd.3	600	0x258	RxPDO6*, Nd.24	621	0x26D	RxPDO6*, Nd.45
580	0x244	RxPDO6*, Nd.4	601	0x259	RxPDO6*, Nd.25	622	0x26E	RxPDO6*, Nd.46
581	0x245	RxPDO6*, Nd.5	602	0x25A	RxPDO6*, Nd.26	623	0x26F	RxPDO6*, Nd.47
582	0x246	RxPDO6*, Nd.6	603	0x25B	RxPDO6*, Nd.27	624	0x270	RxPDO6*, Nd.48
583	0x247	RxPDO6*, Nd.7	604	0x25C	RxPDO6*, Nd.28	625	0x271	RxPDO6*, Nd.49
584	0x248	RxPDO6*, Nd.8	605	0x25D	RxPDO6*, Nd.29	626	0x272	RxPDO6*, Nd.50
585	0x249	RxPDO6*, Nd.9	606	0x25E	RxPDO6*, Nd.30	627	0x273	RxPDO6*, Nd.51
586	0x24A	RxPDO6*, Nd.10	607	0x25F	RxPDO6*, Nd.31	628	0x274	RxPDO6*, Nd.52
587	0x24B	RxPDO6*, Nd.11	608	0x260	RxPDO6*, Nd.32	629	0x275	RxPDO6*, Nd.53
588	0x24C	RxPDO6*, Nd.12	609	0x261	RxPDO6*, Nd.33	630	0x276	RxPDO6*, Nd.54
589	0x24D	RxPDO6*, Nd.13	610	0x262	RxPDO6*, Nd.34	631	0x277	RxPDO6*, Nd.55
590	0x24E	RxPDO6*, Nd.14	611	0x263	RxPDO6*, Nd.35	632	0x278	RxPDO6*, Nd.56
591	0x24F	RxPDO6*, Nd.15	612	0x264	RxPDO6*, Nd.36	633	0x279	RxPDO6*, Nd.57
592	0x250	RxPDO6*, Nd.16	613	0x265	RxPDO6*, Nd.3	634	0x27A	RxPDO6*, Nd.58
593	0x251	RxPDO6*, Nd.17	614	0x266	RxPDO6*, Nd.8	635	0x27B	RxPDO6*, Nd.59
594	0x252	RxPDO6*, Nd.18	615	0x267	RxPDO6*, Nd.39	636	0x27C	RxPDO6*, Nd.60
595	0x253	RxPDO6*, Nd.19	616	0x268	RxPDO6*, N.40	637	0x27D	RxPDO6*, Nd.61
596	0x254	RxPDO6*, Nd.20	617	0x269	RxPDO6*, d.41	638	0x27E	RxPDO6*, Nd.62
597	0x255	RxPDO6*, Nd.21	618	0x26A	RxPDO6*,Nd.42	639	0x27F	RxPDO6*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
705	0x2C1	TxPDO7*, Nd.1	726	0x2D6	TxPDO7*, Nd.22	747	0x2EB	TxPDO7*, Nd.43
706	0x2C2	TxPDO7*, Nd.2	727	0x2D7	TxPDO7*, Nd.23	748	0x2EC	TxPDO7*, Nd.44
707	0x2C3	TxPDO7*, Nd.3	728	0x2D8	TxPDO7*, Nd.24	749	0x2ED	TxPDO7*, Nd.45
708	0x2C4	TxPDO7*, Nd.4	729	0x2D9	TxPDO7*, Nd.25	750	0x2EE	TxPDO7*, Nd.46
709	0x2C5	TxPDO7*, Nd.5	730	0x2DA	TxPDO7*, Nd.26	751	0x2EF	TxPDO7*, Nd.47
710	0x2C6	TxPDO7*, Nd.6	731	0x2DB	TxPDO7*, Nd.27	752	0x2F0	TxPDO7*, Nd.48
711	0x2C7	TxPDO7*, Nd.7	732	0x2DC	TxPDO7*, Nd.28	753	0x2F1	TxPDO7*, Nd.49
712	0x2C8	TxPDO7*, Nd.8	733	0x2DD	TxPDO7*, Nd.29	754	0x2F2	TxPDO7*, Nd.50
713	0x2C9	TxPDO7*, Nd.9	734	0x2DE	TxPDO7*, Nd.30	755	0x2F3	TxPDO7*, Nd.51
714	0x2CA	TxPDO7*, Nd.10	735	0x2DF	TxPDO7*, Nd.31	756	0x2F4	TxPDO7*, Nd.52
715	0x2CB	TxPDO7*, Nd.11	736	0x2E0	TxPDO7*, Nd.32	757	0x2F5	TxPDO7*, Nd.53
716	0x2CC	TxPDO7*, Nd.12	737	0x2E1	TxPDO7*, Nd.33	758	0x2F6	TxPDO7*, Nd.54
717	0x2CD	TxPDO7*, Nd.13	738	0x2E2	TxPDO7*, Nd.34	759	0x2F7	TxPDO7*, Nd.55
718	0x2CE	TxPDO7*, Nd.14	739	0x2E3	TxPDO7*, Nd.35	760	0x2F8	TxPDO7*, Nd.56
719	0x2CF	TxPDO7*, Nd.15	740	0x2E4	TxPDO7*, Nd.36	761	0x2F9	TxPDO7*, Nd.57
720	0x2D0	TxPDO7*, Nd.16	741	0x2E5	TxPDO7*, Nd.37	762	0x2FA	TxPDO7*, Nd.58
721	0x2D1	TxPDO7*, Nd.17	742	0x2E6	TxPDO7*, Nd.38	763	0x2FB	TxPDO7*, Nd.59
722	0x2D2	TxPDO7*, Nd.18	743	0x2E7	TxPDO7*, Nd.39	764	0x2FC	TxPDO7*, Nd.60
723	0x2D3	TxPDO7*, Nd.19	744	0x2E8	TxPDO7*, Nd.40	765	0x2FD	TxPDO7*, Nd.61
724	0x2D4	TxPDO7*, Nd.20	745	0x2E9	TxPDO7*, Nd.41	766	0x2FE	TxPDO7*, Nd.62
725	0x2D5	TxPDO7*, Nd.21	746	0x2EA	TxPDO7*, Nd.42	767	0x2FF	TxPDO7*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
833	0x341	RxPDO7*, Nd.1	854	0x356	RxPDO7*, Nd.22	875	0x36B	RxPDO7*, Nd.43
834	0x342	RxPDO7*, Nd.2	855	0x357	RxPDO7*, Nd.23	876	0x36C	RxPDO7*, Nd.44
835	0x343	RxPDO7*, Nd.3	856	0x358	RxPDO7*, Nd.24	877	0x36D	RxPDO7*, Nd.45
836	0x344	RxPDO7*, Nd.4	857	0x359	RxPDO7*, Nd.25	878	0x36E	RxPDO7*, Nd.46
837	0x345	RxPDO7*, Nd.5	858	0x35A	RxPDO7*, Nd.26	879	0x36F	RxPDO7*, Nd.47
838	0x346	RxPDO7*, Nd.6	859	0x35B	RxPDO7*, Nd.27	880	0x370	RxPDO7*, Nd.48
839	0x347	RxPDO7*, Nd.7	860	0x35C	RxPDO7*, Nd.28	881	0x371	RxPDO7*, Nd.49
840	0x348	RxPDO7*, Nd.8	861	0x35D	RxPDO7*, Nd.29	882	0x372	RxPDO7*, Nd.50
841	0x349	RxPDO7*, Nd.9	862	0x35E	RxPDO7*, Nd.30	883	0x373	RxPDO7*, Nd.51
842	0x34A	RxPDO7*, Nd.10	863	0x35F	RxPDO7*, Nd.31	884	0x374	RxPDO7*, Nd.52
843	0x34B	RxPDO7*, Nd.11	864	0x360	RxPDO7*, Nd.32	885	0x375	RxPDO7*, Nd.53
844	0x34C	RxPDO7*, Nd.12	865	0x361	RxPDO7*, Nd.33	886	0x376	RxPDO7*, Nd.54
845	0x34D	RxPDO7*, Nd.13	866	0x362	RxPDO7*, Nd.34	887	0x377	RxPDO7*, Nd.55
846	0x34E	RxPDO7*, Nd.14	867	0x363	RxPDO7*, Nd.35	888	0x378	RxPDO7*, Nd.56
847	0x34F	RxPDO7*, Nd.15	868	0x364	RxPDO7*, Nd.36	889	0x379	RxPDO7*, Nd.57
848	0x350	RxPDO7*, Nd.16	869	0x365	RxPDO7*, Nd.37	890	0x37A	RxPDO7*, Nd.58
849	0x351	RxPDO7*, Nd.17	870	0x366	RxPDO7*, Nd.38	891	0x37B	RxPDO7*, Nd.59
850	0x352	RxPDO7*, Nd.18	871	0x367	RxPDO7*, Nd.39	892	0x37C	RxPDO7*, Nd.60
851	0x353	RxPDO7*, Nd.19	872	0x368	RxPDO7*, Nd.40	893	0x37D	RxPDO7*, Nd.61
852	0x354	RxPDO7*, Nd.20	873	0x369	RxPDO7*, Nd.41	894	0x37E	RxPDO7*, Nd.62
853	0x355	RxPDO7*, Nd.21	874	0x36A	RxPDO7*, Nd.42	895	0x37F	RxPDO7*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
961	0x3C1	TxPDO8*, Nd.1	982	0x3D6	TxPDO8*, Nd.22	1003	0x3EB	TxPDO8*, Nd.43
962	0x3C2	TxPDO8*, Nd.2	983	0x3D7	TxPDO8*, Nd.23	1004	0x3EC	TxPDO8*, Nd.44
963	0x3C3	TxPDO8*, Nd.3	984	0x3D8	TxPDO8*, Nd.24	1005	0x3ED	TxPDO8*, Nd.45
964	0x3C4	TxPDO8*, Nd.4	985	0x3D9	TxPDO8*, Nd.25	1006	0x3EE	TxPDO8*, Nd.46
965	0x3C5	TxPDO8*, Nd.5	986	0x3DA	TxPDO8*, Nd.26	1007	0x3EF	TxPDO8*, Nd.47
966	0x3C6	TxPDO8*, Nd.6	987	0x3DB	TxPDO8*, Nd.27	1008	0x3F0	TxPDO8*, Nd.48
967	0x3C7	TxPDO8*, Nd.7	988	0x3DC	TxPDO8*, Nd.28	1009	0x3F1	TxPDO8*, Nd.49
968	0x3C8	TxPDO8*, Nd.8	989	0x3DD	TxPDO8*, Nd.29	1010	0x3F2	TxPDO8*, Nd.50
969	0x3C9	TxPDO8*, Nd.9	990	0x3DE	TxPDO8*, Nd.30	1011	0x3F3	TxPDO8*, Nd.51
970	0x3CA	TxPDO8*, Nd.10	991	0x3DF	TxPDO8*, Nd.31	1012	0x3F4	TxPDO8*, Nd.52
971	0x3CB	TxPDO8*, Nd.11	992	0x3E0	TxPDO8*, Nd.32	1013	0x3F5	TxPDO8*, Nd.53
972	0x3CC	TxPDO8*, Nd.12	993	0x3E1	TxPDO8*, Nd.33	1014	0x3F6	TxPDO8*, Nd.54
973	0x3CD	TxPDO8*, Nd.13	994	0x3E2	TxPDO8*, Nd.34	1015	0x3F7	TxPDO8*, Nd.55
974	0x3CE	TxPDO8*, Nd.14	995	0x3E3	TxPDO8*, Nd.35	1016	0x3F8	TxPDO8*, Nd.56
975	0x3CF	TxPDO8*, Nd.15	996	0x3E4	TxPDO8*, Nd.36	1017	0x3F9	TxPDO8*, Nd.57
976	0x3D0	TxPDO8*, Nd.16	997	0x3E5	TxPDO8*, Nd.37	1018	0x3FA	TxPDO8*, Nd.58
977	0x3D1	TxPDO8*, Nd.17	998	0x3E6	TxPDO8*, Nd.38	1019	0x3FB	TxPDO8*, Nd.59
978	0x3D2	TxPDO8*, Nd.18	999	0x3E7	TxPDO8*, Nd.39	1020	0x3FC	TxPDO8*, Nd.60
979	0x3D3	TxPDO8*, Nd.19	1000	0x3E8	TxPDO8*, Nd.40	1021	0x3FD	TxPDO8*, Nd.61
980	0x3D4	TxPDO8*, Nd.20	1001	0x3E9	TxPDO8*, Nd.41	1022	0x3FE	TxPDO8*, Nd.62
981	0x3D5	TxPDO8*, Nd.21	1002	0x3EA	TxPDO8*, Nd.42	1023	0x3FF	TxPDO8*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1089	0x441	RxPDO8*, Nd.1	1110	0x456	RxPDO8*, Nd.22	1131	0x46B	RxPDO8*, Nd.43
1090	0x442	RxPDO8*, Nd.2	1111	0x457	RxPDO8*, Nd.23	1132	0x46C	RxPDO8*, Nd.44
1091	0x443	RxPDO8*, Nd.3	1112	0x458	RxPDO8*, Nd.24	1133	0x46D	RxPDO8*, Nd.45
1092	0x444	RxPDO8*, Nd.4	1113	0x459	RxPDO8*, Nd.25	1134	0x46E	RxPDO8*, Nd.46
1093	0x445	RxPDO8*, Nd.5	1114	0x45A	RxPDO8*, Nd.26	1135	0x46F	RxPDO8*, Nd.47
1094	0x446	RxPDO8*, Nd.6	1115	0x45B	RxPDO8*, Nd.27	1136	0x470	RxPDO8*, Nd.48
1095	0x447	RxPDO8*, Nd.7	1116	0x45C	RxPDO8*, Nd.28	1137	0x471	RxPDO8*, Nd.49
1096	0x448	RxPDO8*, Nd.8	1117	0x45D	RxPDO8*, Nd.29	1138	0x472	RxPDO8*, Nd.50
1097	0x449	RxPDO8*, Nd.9	1118	0x45E	RxPDO8*, Nd.30	1139	0x473	RxPDO8*, Nd.51
1098	0x44A	RxPDO8*, Nd.10	1119	0x45F	RxPDO8*, Nd.31	1140	0x474	RxPDO8*, Nd.52
1099	0x44B	RxPDO8*, Nd.11	1120	0x460	RxPDO8*, Nd.32	1141	0x475	RxPDO8*, Nd.53
1100	0x44C	RxPDO8*, Nd.12	1121	0x461	RxPDO8*, Nd.33	1142	0x476	RxPDO8*, Nd.54
1101	0x44D	RxPDO8*, Nd.13	1122	0x462	RxPDO8*, Nd.34	1143	0x477	RxPDO8*, Nd.55
1102	0x44E	RxPDO8*, Nd.14	1123	0x463	RxPDO8*, Nd.35	1144	0x478	RxPDO8*, Nd.56
1103	0x44F	RxPDO8*, Nd.15	1124	0x464	RxPDO8*, Nd.36	1145	0x479	RxPDO8*, Nd.57
1104	0x450	RxPDO8*, Nd.16	1125	0x465	RxPDO8*, Nd.37	1146	0x47A	RxPDO8*, Nd.58
1105	0x451	RxPDO8*, Nd.17	1126	0x466	RxPDO8*, Nd.38	1147	0x47B	RxPDO8*, Nd.59
1106	0x452	RxPDO8*, Nd.18	1127	0x467	RxPDO8*, Nd.39	1148	0x47C	RxPDO8*, Nd.60
1107	0x453	RxPDO8*, Nd.19	1128	0x468	RxPDO8*, Nd.40	1149	0x47D	RxPDO8*, Nd.61
1108	0x454	RxPDO8*, Nd.20	1129	0x469	RxPDO8*, Nd.41	1150	0x47E	RxPDO8*, Nd.62
1109	0x455	RxPDO8*, Nd.21	1130	0x46A	RxPDO8*, Nd.42	1151	0x47F	RxPDO8*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1217	0x4C1	TxPDO9*, Nd.1	1238	0x4D6	TxPDO9*, Nd.22	1259	0x4EB	TxPDO9*, Nd.43
1218	0x4C2	TxPDO9*, Nd.2	1239	0x4D7	TxPDO9*, Nd.23	1260	0x4EC	TxPDO9*, Nd.44
1219	0x4C3	TxPDO9*, Nd.3	1240	0x4D8	TxPDO9*, Nd.24	1261	0x4ED	TxPDO9*, Nd.45
1220	0x4C4	TxPDO9*, Nd.4	1241	0x4D9	TxPDO9*, Nd.25	1262	0x4EE	TxPDO9*, Nd.46
1221	0x4C5	TxPDO9*, Nd.5	1242	0x4DA	TxPDO9*, Nd.26	1263	0x4EF	TxPDO9*, Nd.47
1222	0x4C6	TxPDO9*, Nd.6	1243	0x4DB	TxPDO9*, Nd.27	1264	0x4F0	TxPDO9*, Nd.48
1223	0x4C7	TxPDO9*, Nd.7	1244	0x4DC	TxPDO9*, Nd.28	1265	0x4F1	TxPDO9*, Nd.49
1224	0x4C8	TxPDO9*, Nd.8	1245	0x4DD	TxPDO9*, Nd.29	1266	0x4F2	TxPDO9*, Nd.50
1225	0x4C9	TxPDO9*, Nd.9	1246	0x4DE	TxPDO9*, Nd.30	1267	0x4F3	TxPDO9*, Nd.51
1226	0x4CA	TxPDO9*, Nd.10	1247	0x4DF	TxPDO9*, Nd.31	1268	0x4F4	TxPDO9*, Nd.52
1227	0x4CB	TxPDO9*, Nd.11	1248	0x4E0	TxPDO9*, Nd.32	1269	0x4F5	TxPDO9*, Nd.53
1228	0x4CC	TxPDO9*, Nd.12	1249	0x4E1	TxPDO9*, Nd.33	1270	0x4F6	TxPDO9*, Nd.54
1229	0x4CD	TxPDO9*, Nd.13	1250	0x4E2	TxPDO9*, Nd.34	1271	0x4F7	TxPDO9*, Nd.55
1230	0x4CE	TxPDO9*, Nd.14	1251	0x4E3	TxPDO9*, Nd.35	1272	0x4F8	TxPDO9*, Nd.56
1231	0x4CF	TxPDO9*, Nd.15	1252	0x4E4	TxPDO9*, Nd.36	1273	0x4F9	TxPDO9*, Nd.57
1232	0x4D0	TxPDO9*, Nd.16	1253	0x4E5	TxPDO9*, Nd.37	1274	0x4FA	TxPDO9*, Nd.58
1233	0x4D1	TxPDO9*, Nd.17	1254	0x4E6	TxPDO9*, Nd.38	1275	0x4FB	TxPDO9*, Nd.59
1234	0x4D2	TxPDO9*, Nd.18	1255	0x4E7	TxPDO9*, Nd.39	1276	0x4FC	TxPDO9*, Nd.60
1235	0x4D3	TxPDO9*, Nd.19	1256	0x4E8	TxPDO9*, Nd.40	1277	0x4FD	TxPDO9*, Nd.61
1236	0x4D4	TxPDO9*, Nd.20	1257	0x4E9	TxPDO9*, Nd.41	1278	0x4FE	TxPDO9*, Nd.62
1237	0x4D5	TxPDO9*, Nd.21	1258	0x4EA	TxPDO9*, Nd.42	1279	0x4FF	TxPDO9*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1345	0x541	RxPDO9*, Nd.1	1366	0x556	RxPDO9*, Nd.22	1387	0x56B	RxPDO9*, Nd.43
1346	0x542	RxPDO9*, Nd.2	1367	0x557	RxPDO9*, Nd.23	1388	0x56C	RxPDO9*, Nd.44
1347	0x543	RxPDO9*, Nd.3	1368	0x558	RxPDO9*, Nd.24	1389	0x56D	RxPDO9*, Nd.45
1348	0x544	RxPDO9*, Nd.4	1369	0x559	RxPDO9*, Nd.25	1390	0x56E	RxPDO9*, Nd.46
1349	0x545	RxPDO9*, Nd.5	1370	0x55A	RxPDO9*, Nd.26	1391	0x56F	RxPDO9*, Nd.47
1350	0x546	RxPDO9*, Nd.6	1371	0x55B	RxPDO9*, Nd.27	1392	0x570	RxPDO9*, Nd.48
1351	0x547	RxPDO9*, Nd.7	1372	0x55C	RxPDO9*, Nd.28	1393	0x571	RxPDO9*, Nd.49
1352	0x548	RxPDO9*, Nd.8	1373	0x55D	RxPDO9*, Nd.29	1394	0x572	RxPDO9*, Nd.50
1353	0x549	RxPDO9*, Nd.9	1374	0x55E	RxPDO9*, Nd.30	1395	0x573	RxPDO9*, Nd.51
1354	0x54A	RxPDO9*, Nd.10	1375	0x55F	RxPDO9*, Nd.31	1396	0x574	RxPDO9*, Nd.52
1355	0x54B	RxPDO9*, Nd.11	1376	0x560	RxPDO9*, Nd.32	1397	0x575	RxPDO9*, Nd.53
1356	0x54C	RxPDO9*, Nd.12	1377	0x561	RxPDO9*, Nd.33	1398	0x576	RxPDO9*, Nd.54
1357	0x54D	RxPDO9*, Nd.13	1378	0x562	RxPDO9*, Nd.34	1399	0x577	RxPDO9*, Nd.55
1358	0x54E	RxPDO9*, Nd.14	1379	0x563	RxPDO9*, Nd.35	1400	0x578	RxPDO9*, Nd.56
1359	0x54F	RxPDO9*, Nd.15	1380	0x564	RxPDO9*, Nd.36	1401	0x579	RxPDO9*, Nd.57
1360	0x550	RxPDO9*, Nd.16	1381	0x565	RxPDO9*, Nd.37	1402	0x57A	RxPDO9*, Nd.58
1361	0x551	RxPDO9*, Nd.17	1382	0x566	RxPDO9*, Nd.38	1403	0x57B	RxPDO9*, Nd.59
1362	0x552	RxPDO9*, Nd.18	1383	0x567	RxPDO9*, Nd.39	1404	0x57C	RxPDO9*, Nd.60
1363	0x553	RxPDO9*, Nd.19	1384	0x568	RxPDO9*, Nd.40	1405	0x57D	RxPDO9*, Nd.61
1364	0x554	RxPDO9*, Nd.20	1385	0x569	RxPDO9*, Nd.41	1406	0x57E	RxPDO9*, Nd.62
1365	0x555	RxPDO9*, Nd.21	1386	0x56A	RxPDO9*, Nd.42	1407	0x57F	RxPDO9*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1473	0x5C1	TxPDO10*, Nd.1	1494	0x5D6	TxPDO10*, Nd.22	1515	0x5EB	TxPDO10*, Nd.43
1474	0x5C2	TxPDO10*, Nd.2	1495	0x5D7	TxPDO10*, Nd.23	1516	0x5EC	TxPDO10*, Nd.44
1475	0x5C3	TxPDO10*, Nd.3	1496	0x5D8	TxPDO10*, Nd.24	1517	0x5ED	TxPDO10*, Nd.45
1476	0x5C4	TxPDO10*, Nd.4	1497	0x5D9	TxPDO10*, Nd.25	1518	0x5EE	TxPDO10*, Nd.46
1477	0x5C5	TxPDO10*, Nd.5	1498	0x5DA	TxPDO10*, Nd.26	1519	0x5EF	TxPDO10*, Nd.47
1478	0x5C6	TxPDO10*, Nd.6	1499	0x5DB	TxPDO10*, Nd.27	1520	0x5F0	TxPDO10*, Nd.48
1479	0x5C7	TxPDO10*, Nd.7	1500	0x5DC	TxPDO10*, Nd.28	1521	0x5F1	TxPDO10*, Nd.49
1480	0x5C8	TxPDO10*, Nd.8	1501	0x5DE	TxPDO10*, Nd.29	1522	0x5F2	TxPDO10*, Nd.50
1481	0x5C9	TxPDO10*, Nd.9	1502	0x5DE	TxPDO10*, Nd.30	1523	0x5F3	TxPDO10*, Nd.51
1482	0x5CA	TxPDO10*, Nd.10	1503	0x5DF	TxPDO10*, Nd.31	1524	0x5F4	TxPDO10*, Nd.52
1483	0x5CB	TxPDO10*, Nd.11	1504	0x5E0	TxPDO10*, Nd.32	1525	0x5F5	TxPDO10*, Nd.53
1484	0x5CC	TxPDO10*, Nd.12	1505	0x5E1	TxPDO10*, Nd.33	1526	0x5F6	TxPDO10*, Nd.54
1485	0x5CD	TxPDO10*, Nd.13	1506	0x5E2	TxPDO10*, Nd.34	1527	0x5F7	TxPDO10*, Nd.55
1486	0x5CE	TxPDO10*, Nd.14	1507	0x5E3	TxPDO10*, Nd.35	1528	0x5F8	TxPDO10*, Nd.56
1487	0x5CF	TxPDO10*, Nd.15	1508	0x5E4	TxPDO10*, Nd.36	1529	0x5F9	TxPDO10*, Nd.57
1488	0x5D0	TxPDO10*, Nd.16	1509	0x5E5	TxPDO10*, Nd.37	1530	0x5FA	TxPDO10*, Nd.58
1489	0x5D1	TxPDO10*, Nd.17	1510	0x5E6	TxPDO10*, Nd.38	1531	0x5FB	TxPDO10*, Nd.59
1490	0x5D2	TxPDO10*, Nd.18	1511	0x5E7	TxPDO10*, Nd.39	1532	0x5FC	TxPDO10*, Nd.60
1491	0x5D3	TxPDO10*, Nd.19	1512	0x5E8	TxPDO10*, Nd.40	1533	0x5FD	TxPDO10*, Nd.61
1492	0x5D4	TxPDO10*, Nd.20	1513	0x5E9	TxPDO10*, Nd.41	1534	0x5FE	TxPDO10*, Nd.62
1493	0x5D5	TxPDO10*, Nd.21	1514	0x5EA	TxPDO10*, Nd.42	1535	0x5FF	TxPDO10*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1601	0x641	RxPDO10*, Nd.1	1622	0x656	RxPDO10*, Nd.22	1643	0x66B	RxPDO10*, Nd.43
1602	0x642	RxPDO10*, Nd.2	1623	0x657	RxPDO10*, Nd.23	1644	0x66C	RxPDO10*, Nd.44
1603	0x643	RxPDO10*, Nd.3	1624	0x658	RxPDO10*, Nd.24	1645	0x66D	RxPDO10*, Nd.45
1604	0x644	RxPDO10*, Nd.4	1625	0x659	RxPDO10*, Nd.25	1646	0x66E	RxPDO10*, Nd.46
1605	0x645	RxPDO10*, Nd.5	1626	0x65A	RxPDO10*, Nd.26	1647	0x66F	RxPDO10*, Nd.47
1606	0x646	RxPDO10*, Nd.6	1627	0x65B	RxPDO10*, Nd.27	1648	0x670	RxPDO10*, Nd.48
1607	0x647	RxPDO10*, Nd.7	1628	0x65C	RxPDO10*, Nd.28	1649	0x671	RxPDO10*, Nd.49
1608	0x648	RxPDO10*, Nd.8	1629	0x65D	RxPDO10*, Nd.29	1650	0x672	RxPDO10*, Nd.50
1609	0x649	RxPDO10*, Nd.9	1630	0x65E	RxPDO10*, Nd.30	1651	0x673	RxPDO10*, Nd.51
1610	0x64A	RxPDO10*, Nd.10	1631	0x65F	RxPDO10*, Nd.31	1652	0x674	RxPDO10*, Nd.52
1611	0x64B	RxPDO10*, Nd.11	1632	0x660	RxPDO10*, Nd.32	1653	0x675	RxPDO10*, Nd.53
1612	0x64C	RxPDO10*, Nd.12	1633	0x661	RxPDO10*, Nd.33	1654	0x676	RxPDO10*, Nd.54
1613	0x64D	RxPDO10*, Nd.13	1634	0x662	RxPDO10*, Nd.34	1655	0x677	RxPDO10*, Nd.55
1614	0x64E	RxPDO10*, Nd.14	1635	0x663	RxPDO10*, Nd.35	1656	0x678	RxPDO10*, Nd.56
1615	0x64F	RxPDO10*, Nd.15	1636	0x664	RxPDO10*, Nd.36	1657	0x679	RxPDO10*, Nd.57
1616	0x650	RxPDO10*, Nd.16	1637	0x665	RxPDO10*, Nd.37	1658	0x67A	RxPDO10*, Nd.58
1617	0x651	RxPDO10*, Nd.17	1638	0x666	RxPDO10*, Nd.38	1659	0x67B	RxPDO10*, Nd.59
1618	0x652	RxPDO10*, Nd.18	1639	0x667	RxPDO10*, Nd.39	1660	0x67C	RxPDO10*, Nd.60
1619	0x653	RxPDO10*, Nd.19	1640	0x668	RxPDO10*, Nd.40	1661	0x67D	RxPDO10*, Nd.61
1620	0x654	RxPDO10*, Nd.20	1641	0x669	RxPDO10*, Nd.41	1662	0x67E	RxPDO10*, Nd.62
1621	0x655	RxPDO10*, Nd.21	1642	0x66A	RxPDO10*, Nd.42	1663	0x67F	RxPDO10*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1729	0x6C1	TxPDO11*, Nd.1	1750	0x6D6	TxPDO11*, Nd.22	1771	0x6EB	TxPDO11*, Nd.43
1730	0x6C2	TxPDO11*, Nd.2	1751	0x6D7	TxPDO11*, Nd.23	1772	0x6EC	TxPDO11*, Nd.44
1731	0x6C3	TxPDO11*, Nd.3	1752	0x6D8	TxPDO11*, Nd.24	1773	0x6ED	TxPDO11*, Nd.45
1732	0x6C4	TxPDO11*, Nd.4	1753	0x6D9	TxPDO11*, Nd.25	1774	0x6EE	TxPDO11*, Nd.46
1733	0x6C5	TxPDO11*, Nd.5	1754	0x6DA	TxPDO11*, Nd.26	1775	0x6EF	TxPDO11*, Nd.47
1734	0x6C6	TxPDO11*, Nd.6	1755	0x6DB	TxPDO11*, Nd.27	1776	0x6F0	TxPDO11*, Nd.48
1735	0x6C7	TxPDO11*, Nd.7	1756	0x6DC	TxPDO11*, Nd.28	1777	0x6F1	TxPDO11*, Nd.49
1736	0x6C8	TxPDO11*, Nd.8	1757	0x6DD	TxPDO11*, Nd.29	1778	0x6F2	TxPDO11*, Nd.50
1737	0x6C9	TxPDO11*, Nd.9	1758	0x6DE	TxPDO11*, Nd.30	1779	0x6F3	TxPDO11*, Nd.51
1738	0x6CA	TxPDO11*, Nd.10	1759	0x6DF	TxPDO11*, Nd.31	1780	0x6F4	TxPDO11*, Nd.52
1739	0x6CB	TxPDO11*, Nd.11	1760	0x6E0	TxPDO11*, Nd.32	1781	0x6F5	TxPDO11*, Nd.53
1740	0x6CC	TxPDO11*, Nd.12	1761	0x6E1	TxPDO11*, Nd.33	1782	0x6F6	TxPDO11*, Nd.54
1741	0x6CD	TxPDO11*, Nd.13	1762	0x6E2	TxPDO11*, Nd.34	1783	0x6F7	TxPDO11*, Nd.55
1742	0x6CE	TxPDO11*, Nd.14	1763	0x6E3	TxPDO11*, Nd.35	1784	0x6F8	TxPDO11*, Nd.56
1743	0x6CF	TxPDO11*, Nd.15	1764	0x6E4	TxPDO11*, Nd.36	1785	0x6F9	TxPDO11*, Nd.57
1744	0x6D0	TxPDO11*, Nd.16	1765	0x6E5	TxPDO11*, Nd.37	1786	0x6FA	TxPDO11*, Nd.58
1745	0x6D1	TxPDO11*, Nd.17	1766	0x6E6	TxPDO11*, Nd.38	1787	0x6FB	TxPDO11*, Nd.59
1746	0x6D2	TxPDO11*, Nd.18	1767	0x6E7	TxPDO11*, Nd.39	1788	0x6FC	TxPDO11*, Nd.60
1747	0x6D3	TxPDO11*, Nd.19	1768	0x6E8	TxPDO11*, Nd.40	1789	0x6FD	TxPDO11*, Nd.61
1748	0x6D4	TxPDO11*, Nd.20	1769	0x6E9	TxPDO11*, Nd.41	1790	0x6FE	TxPDO11*, Nd.62
1749	0x6D5	TxPDO11*, Nd.21	1770	0x6EA	TxPDO11*, Nd.42	1791	0x6FF	TxPDO11*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1857	0x741	RxPDO11*, Nd.1	1878	0x756	RxPDO11*, Nd.22	1899	0x76B	RxPDO11*, Nd.43
1858	0x742	RxPDO11*, Nd.2	1879	0x757	RxPDO11*, Nd.23	1900	0x76C	RxPDO11*, Nd.44
1859	0x743	RxPDO11*, Nd.3	1880	0x758	RxPDO11*, Nd.24	1901	0x76D	RxPDO11*, Nd.45
1860	0x744	RxPDO11*, Nd.4	1881	0x759	RxPDO11*, Nd.25	1902	0x76E	RxPDO11*, Nd.46
1861	0x745	RxPDO11*, Nd.5	1882	0x75A	RxPDO11*, Nd.26	1903	0x76F	RxPDO11*, Nd.47
1862	0x746	RxPDO11*, Nd.6	1883	0x75B	RxPDO11*, Nd.27	1904	0x770	RxPDO11*, Nd.48
1863	0x747	RxPDO11*, Nd.7	1884	0x75C	RxPDO11*, Nd.28	1905	0x771	RxPDO11*, Nd.49
1864	0x748	RxPDO11*, Nd.8	1885	0x75D	RxPDO11*, Nd.29	1906	0x772	RxPDO11*, Nd.50
1865	0x749	RxPDO11*, Nd.9	1886	0x75E	RxPDO11*, Nd.30	1907	0x773	RxPDO11*, Nd.51
1866	0x74A	RxPDO11*, Nd.10	1887	0x75F	RxPDO11*, Nd.31	1908	0x774	RxPDO11*, Nd.52
1867	0x74B	RxPDO11*, Nd.11	1888	0x760	RxPDO11*, Nd.32	1909	0x775	RxPDO11*, Nd.53
1868	0x74C	RxPDO11*, Nd.12	1889	0x761	RxPDO11*, Nd.33	1910	0x776	RxPDO11*, Nd.54
1869	0x74D	RxPDO11*, Nd.13	1890	0x762	RxPDO11*, Nd.34	1911	0x777	RxPDO11*, Nd.55
1870	0x74E	RxPDO11*, Nd.14	1891	0x763	RxPDO11*, Nd.35	1912	0x778	RxPDO11*, Nd.56
1871	0x74F	RxPDO11*, Nd.15	1892	0x764	RxPDO11*, Nd.36	1913	0x779	RxPDO11*, Nd.57
1872	0x750	RxPDO11*, Nd.16	1893	0x765	RxPDO11*, Nd.37	1914	0x77A	RxPDO11*, Nd.58
1873	0x751	RxPDO11*, Nd.17	1894	0x766	RxPDO11*, Nd.38	1915	0x77B	RxPDO11*, Nd.59
1874	0x752	RxPDO11*, Nd.18	1895	0x767	RxPDO11*, Nd.39	1916	0x77C	RxPDO11*, Nd.60
1875	0x753	RxPDO11*, Nd.19	1896	0x768	RxPDO11*, Nd.40	1917	0x77D	RxPDO11*, Nd.61
1876	0x754	RxPDO11*, Nd.20	1897	0x769	RxPDO11*, Nd.41	1918	0x77E	RxPDO11*, Nd.62
1877	0x755	RxPDO11*, Nd.21	1898	0x76A	RxPDO11*, Nd.42	1919	0x77F	RxPDO11*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1409	0x581	SDO Tx Nd.1	1430	0x596	SDO Tx Nd.22	1451	0x5AB	SDO Tx Nd.43
1410	0x582	SDO Tx Nd.2	1431	0x597	SDO Tx Nd.23	1452	0x5AC	SDO Tx Nd.44
1411	0x583	SDO Tx Nd.3	1432	0x598	SDO Tx Nd.24	1453	0x5AD	SDO Tx Nd.45
1412	0x584	SDO Tx Nd.4	1433	0x599	SDO Tx Nd.25	1454	0x5AE	SDO Tx Nd.46
1413	0x585	SDO Tx Nd.5	1434	0x59A	SDO Tx Nd.26	1455	0x5AF	SDO Tx Nd.47
1414	0x586	SDO Tx Nd.6	1435	0x59B	SDO Tx Nd.27	1456	0x5B0	SDO Tx Nd.48
1415	0x587	SDO Tx Nd.7	1436	0x59C	SDO Tx Nd.28	1457	0x5B1	SDO Tx Nd.49
1416	0x588	SDO Tx Nd.8	1437	0x59D	SDO Tx Nd.29	1458	0x5B2	SDO Tx Nd.50
1417	0x589	SDO Tx Nd.9	1438	0x59E	SDO Tx Nd.30	1459	0x5B3	SDO Tx Nd.51
1418	0x58A	SDO Tx Nd.10	1439	0x59F	SDO Tx Nd.31	1460	0x5B4	SDO Tx Nd.52
1419	0x58B	SDO Tx Nd.11	1440	0x5A0	SDO Tx Nd.32	1461	0x5B5	SDO Tx Nd.53
1420	0x58C	SDO Tx Nd.12	1441	0x5A1	SDO Tx Nd.33	1462	0x5B6	SDO Tx Nd.54
1421	0x58D	SDO Tx Nd.13	1442	0x5A2	SDO Tx Nd.34	1463	0x5B7	SDO Tx Nd.55
1422	0x58E	SDO Tx Nd.14	1443	0x5A3	SDO Tx Nd.35	1464	0x5B8	SDO Tx Nd.56
1423	0x58F	SDO Tx Nd.15	1444	0x5A4	SDO Tx Nd.36	1465	0x5B9	SDO Tx Nd.57
1424	0x590	SDO Tx Nd.16	1445	0x5A5	SDO Tx Nd.37	1466	0x5BA	SDO Tx Nd.58
1425	0x591	SDO Tx Nd.17	1446	0x5A6	SDO Tx Nd.38	1467	0x5BB	SDO Tx Nd.59
1426	0x592	SDO Tx Nd.18	1447	0x5A7	SDO Tx Nd.39	1468	0x5BC	SDO Tx Nd.60
1427	0x593	SDO Tx Nd.19	1448	0x5A8	SDO Tx Nd.40	1469	0x5BD	SDO Tx Nd.61
1428	0x594	SDO Tx Nd.20	1449	0x5A9	SDO Tx Nd.41	1470	0x5BE	SDO Tx Nd.62
1429	0x595	SDO Tx Nd.21	1450	0x5AA	SDO Tx Nd.42	1471	0x5BF	SDO Tx Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1537	0x601	SDO Rx Nd.1	1558	0x616	SDO Rx Nd.22	1579	0x62B	SDO Rx Nd.43
1538	0x602	SDO Rx Nd.2	1559	0x617	SDO Rx Nd.23	1580	0x62C	SDO Rx Nd.44
1539	0x603	SDO Rx Nd.3	1560	0x618	SDO Rx Nd.24	1581	0x62D	SDO Rx Nd.45
1540	0x604	SDO Rx Nd.4	1561	0x619	SDO Rx Nd.25	1582	0x62E	SDO Rx Nd.46
1541	0x605	SDO Rx Nd.5	1562	0x61A	SDO Rx Nd.26	1583	0x62F	SDO Rx Nd.47
1542	0x606	SDO Rx Nd.6	1563	0x61B	SDO Rx Nd.27	1584	0x630	SDO Rx Nd.48
1543	0x607	SDO Rx Nd.7	1564	0x61C	SDO Rx Nd.28	1585	0x631	SDO Rx Nd.49
1544	0x608	SDO Rx Nd.8	1565	0x61D	SDO Rx Nd.29	1586	0x632	SDO Rx Nd.50
1545	0x609	SDO Rx Nd.9	1566	0x61E	SDO Rx Nd.30	1587	0x633	SDO Rx Nd.51
1546	0x60A	SDO Rx Nd.10	1567	0x61F	SDO Rx Nd.31	1588	0x634	SDO Rx Nd.52
1547	0x60B	SDO Rx Nd.11	1568	0x620	SDO Rx Nd.32	1589	0x635	SDO Rx Nd.53
1548	0x60C	SDO Rx Nd.12	1569	0x621	SDO Rx Nd.33	1590	0x636	SDO Rx Nd.54
1549	0x60D	SDO Rx Nd.13	1570	0x622	SDO Rx Nd.34	1591	0x637	SDO Rx Nd.55
1550	0x60E	SDO Rx Nd.14	1571	0x623	SDO Rx Nd.35	1592	0x638	SDO Rx Nd.56
1551	0x60F	SDO Rx Nd.15	1572	0x624	SDO Rx Nd.36	1593	0x639	SDO Rx Nd.57
1552	0x610	SDO Rx Nd.16	1573	0x625	SDO Rx Nd.37	1594	0x63A	SDO Rx Nd.58
1553	0x611	SDO Rx Nd.17	1574	0x626	SDO Rx Nd.38	1595	0x63B	SDO Rx Nd.59
1554	0x612	SDO Rx Nd.18	1575	0x627	SDO Rx Nd.39	1596	0x63C	SDO Rx Nd.60
1555	0x613	SDO Rx Nd.19	1576	0x628	SDO Rx Nd.40	1597	0x63D	SDO Rx Nd.61
1556	0x614	SDO Rx Nd.20	1577	0x629	SDO Rx Nd.41	1598	0x63E	SDO Rx Nd.62
1557	0x615	SDO Rx Nd.21	1578	0x62A	SDO Rx Nd.42	1599	0x63F	SDO Rx Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1793	0x701	Guarding Nd.1	1814	0x716	Guarding Nd.22	1835	0x72B	Guarding Nd.43
1794	0x702	Guarding Nd.2	1815	0x717	Guarding Nd.23	1836	0x72C	Guarding Nd.44
1795	0x703	Guarding Nd.3	1816	0x718	Guarding Nd.24	1837	0x72D	Guarding Nd.45
1796	0x704	Guarding Nd.4	1817	0x719	Guarding Nd.25	1838	0x72E	Guarding Nd.46
1797	0x705	Guarding Nd.5	1818	0x71A	Guarding Nd.26	1839	0x72F	Guarding Nd.47
1798	0x706	Guarding Nd.6	1819	0x71B	Guarding Nd.27	1840	0x730	Guarding Nd.48
1799	0x707	Guarding Nd.7	1820	0x71C	Guarding Nd.28	1841	0x731	Guarding Nd.49
1800	0x708	Guarding Nd.8	1821	0x71D	Guarding Nd.29	1842	0x732	Guarding Nd.50
1801	0x709	Guarding Nd.9	1822	0x71E	Guarding Nd.30	1843	0x733	Guarding Nd.51
1802	0x70A	Guarding Nd.10	1823	0x71F	Guarding Nd.31	1844	0x734	Guarding Nd.52
1803	0x70B	Guarding Nd.11	1824	0x720	Guarding Nd.32	1845	0x735	Guarding Nd.53
1804	0x70C	Guarding Nd.12	1825	0x721	Guarding Nd.33	1846	0x736	Guarding Nd.54
1805	0x70D	Guarding Nd.13	1826	0x722	Guarding Nd.34	1847	0x737	Guarding Nd.55
1806	0x70E	Guarding Nd.14	1827	0x723	Guarding Nd.35	1848	0x738	Guarding Nd.56
1807	0x70F	Guarding Nd.15	1828	0x724	Guarding Nd.36	1849	0x739	Guarding Nd.57
1808	0x710	Guarding Nd.16	1829	0x725	Guarding Nd.37	1850	0x73A	Guarding Nd.58
1809	0x711	Guarding Nd.17	1830	0x726	Guarding Nd.38	1851	0x73B	Guarding Nd.59
1810	0x712	Guarding Nd.18	1831	0x727	Guarding Nd.39	1852	0x73C	Guarding Nd.60
1811	0x713	Guarding Nd.19	1832	0x728	Guarding Nd.40	1853	0x73D	Guarding Nd.61
1812	0x714	Guarding Nd.20	1833	0x729	Guarding Nd.41	1854	0x73E	Guarding Nd.62
1813	0x715	Guarding Nd.21	1834	0x72A	Guarding Nd.42	1855	0x73F	Guarding Nd.63

## 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/bx5100/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)".

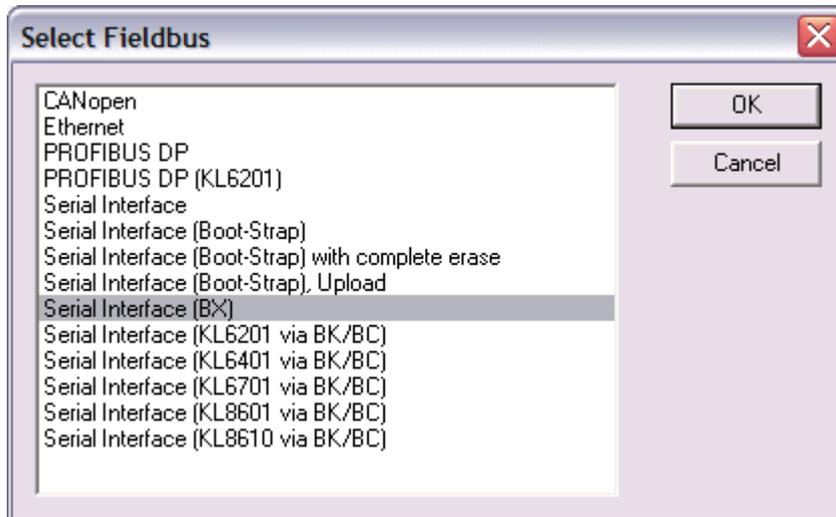


Abb. 146: Auswahl des Busklemmen-Controllers der BX-Serie

#### BCxx50 Serie

Wählen sie das entsprechende Gerät aus - hier "Serial Interface".

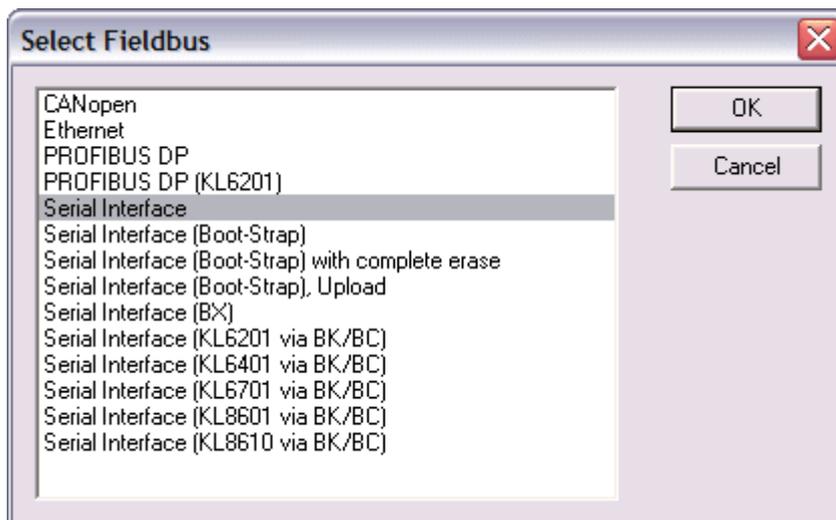


Abb. 147: Auswahl des Busklemmen-Controllers der BC-Serie

**BX- und BCxx50-Serie**

Anschließen wählen Sie den COM-Port aus.



Abb. 148: Auswahl des COM-Ports

Öffnen Sie das entsprechende File, das Sie herunterladen möchten.

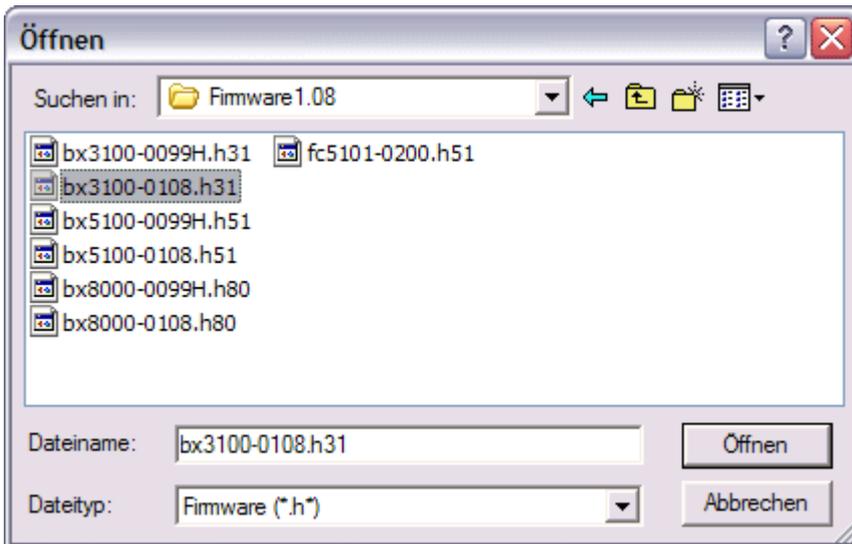


Abb. 149: Öffnen des Firmware-Files

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.

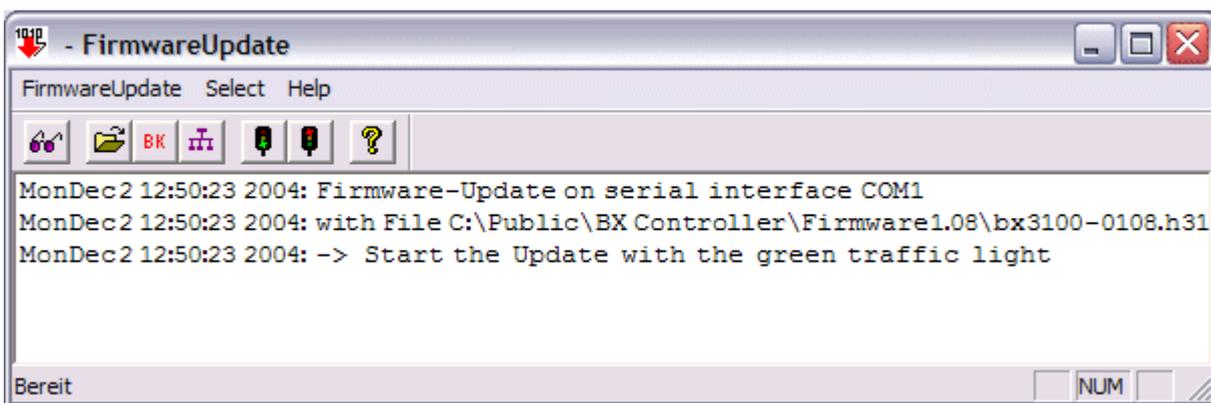


Abb. 150: Statusmeldungen des Firmware-Updates

**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/bx5100/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.

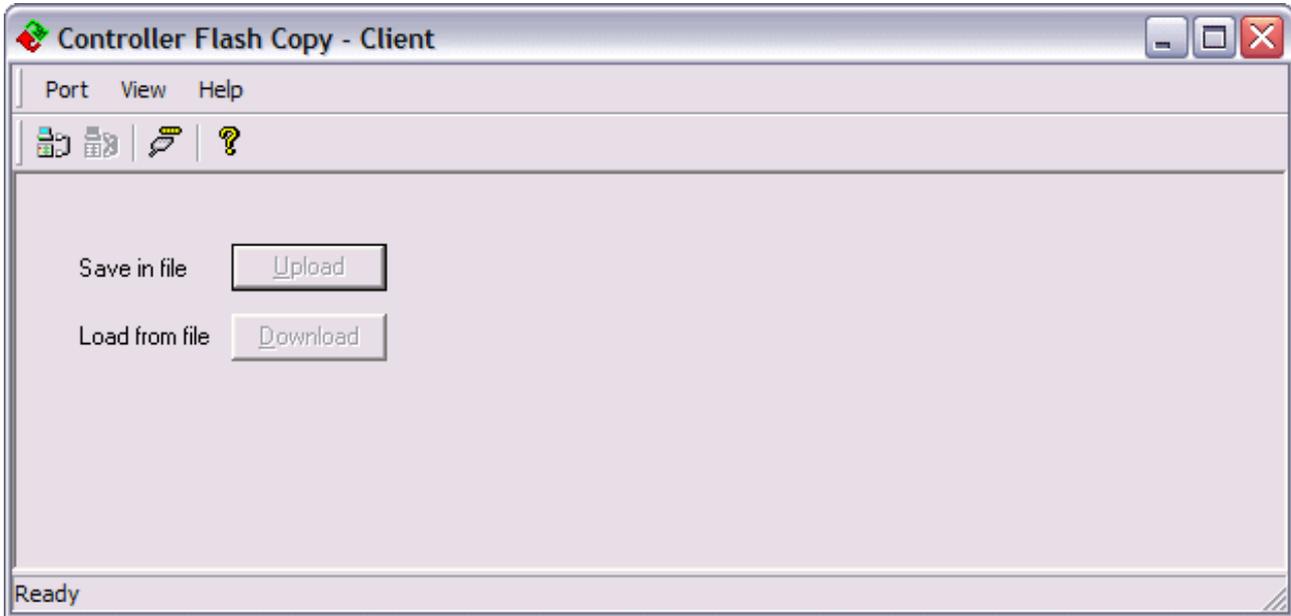


Abb. 151: CFC-Client

Port/Port Setup...

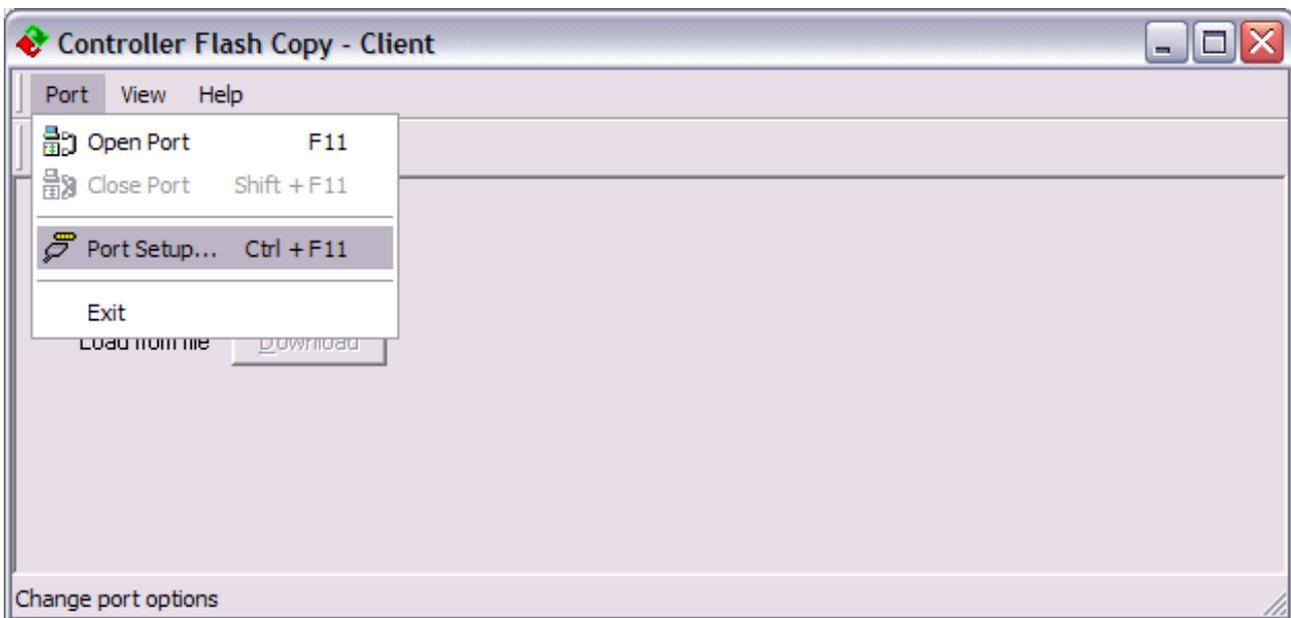


Abb. 152: Aufruf des Port-Setups

Wählen Sie die Einstellung aus.

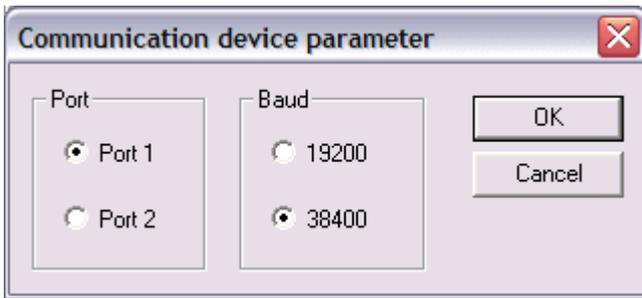


Abb. 153: Einstellen der COM-Parameter

Bevor Sie den COM-Port öffnen müssen Sie eine physikalische Verbindung zum BX-Controller herstellen. Öffnen Sie dann den COM Port.

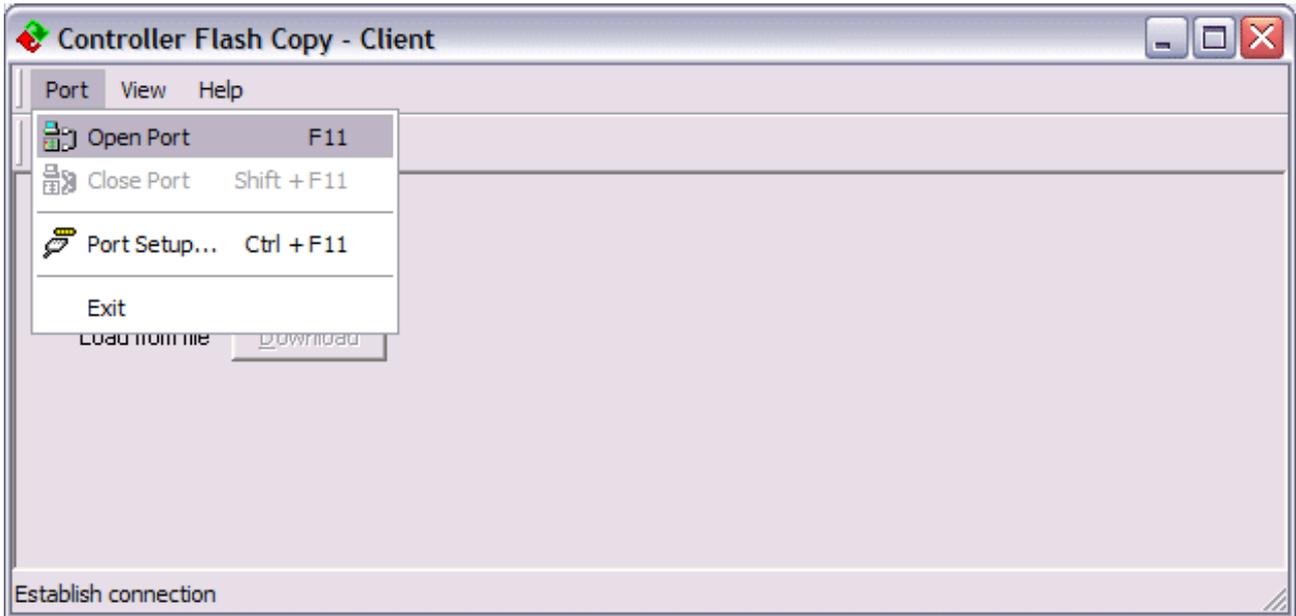


Abb. 154: Öffnen des COM-Ports

Mit Upload erstellen Sie eine Kopie des BX-Speichers speichern diese als BIN-Datei.

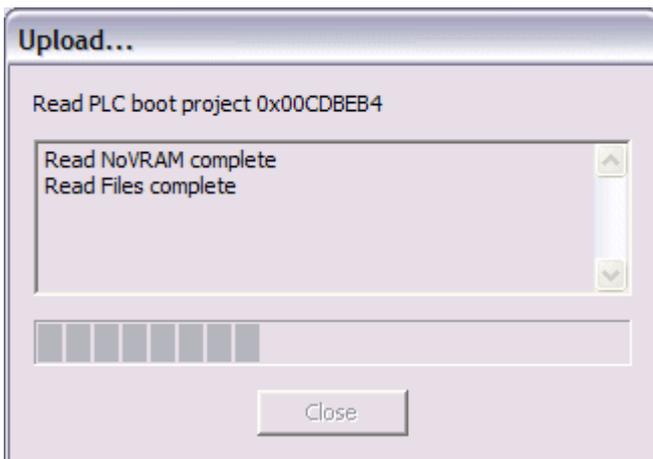


Abb. 155: Upload des BX-Speicherinhalts in ein BIN-File

Mit Download wird das BIN-File zum BX-Controller geladen. Der Controller startet automatisch nach erfolgreichem Download.

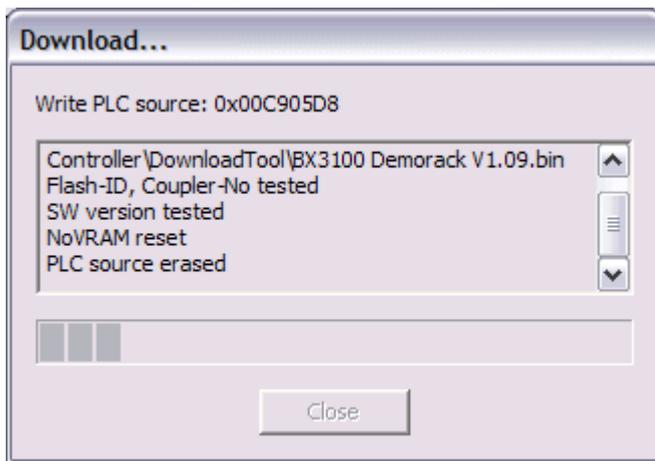


Abb. 156: Download eines BIN-Files auf den BX-Controller

\* ab Firmware 1.09 ist der CFC-Server auf der BX-Controller Reihe verfügbar

## 8.5 Beispielprogramme - Übersicht

Benennung	Beschreibung
<a href="#">Display [▶ 111]</a>	Beispiel zum Ansteuern des Displays
<a href="#">Navigationsschalter [▶ 110]</a>	Auslesen des Navigationsschalter aus der PLC heraus
<a href="#">Menü [▶ 81]</a>	Beispiel für ein eignes Menü mit Navigations-Schalter und Display
<a href="#">RTC [▶ 75]</a>	Beispiel zum Auslesen der Real Time Clock (RTC) über Funktionsbausteine
<a href="#">COM Port - BK/BC8x00 Master Interface [▶ 116]</a>	COM1- oder COM2-Schnittstelle als Master mit dem BK8x00 Protokoll
<a href="#">COM Port - BK8x00 Slave Interface [▶ 117]</a>	COM1- oder COM2-Schnittstelle als Slave mit dem BK8x00 Protokoll
<a href="#">COM Port - Cimrex 12 [▶ 123]</a>	Beispiel zum Ansteuern eines Cimrex 12 Displays über ModbusRTU
<a href="#">COM Port - ModbusRTU Slave [▶ 120]</a>	Verknüpfung der ModbusRTU Lib mit der COM 1 oder COM 2 Schnittstelle des BXes
<a href="#">COM Port - ModbusRTU Master [▶ 122]</a>	Verknüpfung der ModbusRTU Lib mit der COM 1 oder COM 2 Schnittstelle des BXes
<a href="#">COM Port - RK512 Protokoll [▶ 125]</a>	RK512-Protokoll über die COM 1 oder COM 2
<a href="#">COM Port - SMS über COM Port [▶ 126]</a>	Anschluss eines Siemens S35 Mobiltelefons an COM-Schnittstelle zum Versenden von SMS-Nachrichten
<a href="#">COM Port - COMlibV2 [▶ 123]</a>	String verschicken und empfangen mit der COMlibV2
<a href="#">SSB - Display [▶ 65]</a>	Cimrex Panel am SSB
<a href="#">SSB - AX2000 [▶ 63]</a>	AX2000 am SSB
<a href="#">SSB - BK51x0 [▶ 61]</a>	BK5120 am SSB
<a href="#">SSB - BX / BX Kommunikation [▶ 62]</a>	Kommunikation von BX zu BX (über SSB)
<a href="#">SSB - IclA Drive [▶ 66]</a>	IclA Drive am SSB
<a href="#">SSB - Lenze Drive [▶ 72]</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
zulässige Umgebungstemperatur im Betrieb	-25°C ... +85°C
Einbaulage	beliebig
Vibrationsfestigkeit	gemäß EN 60068-2-6
Schockfestigkeit	gemäß EN 60068-2-27, EN 60068-2-29
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.



Abb. 157: Typenschild eines BX-Controllers

Auf dem Aufkleber sind folgende Informationen abzulesen:

Aufdruck	Bedeutung für diesen Aufkleber
genaue Produktbezeichnung	Model BX
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!)
Hersteller	Beckhoff Automation GmbH
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	Serial No.: Seriennummer HW: Hardware Stand Date: Herstelldatum optional nur beim BX9000 MAC-ID

## 8.7 Zulassungen

### Konformitätskennzeichnung

CE

### Schutzart

IP20 gemäß EN60529

## 8.8 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

## 8.9 Literaturverzeichnis

### Deutsche Bücher

- Holger Zeltwanger (Hrsg.):  
**CANopen**,  
VDE Verlag, 2001. 197 Seiten,  
ISBN 3-800-72448-0

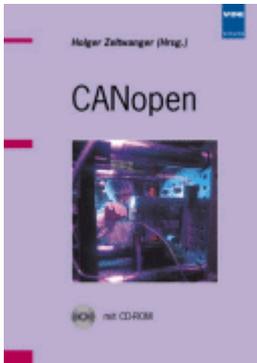


Abb. 158: CANopen

- Konrad Etschberger:  
**Controller Area Network**, Grundlagen, Protokolle, Bausteine, Anwendungen.  
Hanser Verlag, 2000. 431 Seiten.  
ISBN 3-446-19431-2

### 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

- Konrad Etschberger:  
**Controller Area Network**,  
Ixxat Press, 2001. 440 Seiten.  
ISBN 3-00-007376-0
- M. Farsi, M. Barbosa:  
**CANopen Implementation**,  
RSP 2000. 210 Seiten.  
ISBN 0-86380-247-8

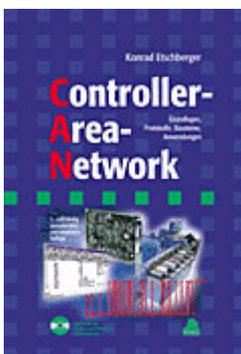


Abb. 159: CAN - Controller-Area-Network

**Standards**

- ISO 11898:  
Road Vehicles - Interchange of digital information - Controller Area Network (CAN) for high speed communication.
- CiA DS 301:  
CANopen Application Layer and Communication Profile.  
Erhältlich beim Verband CAN in Automation.
- CiA DS 401:  
CANopen Device Profile for Generic E/A Modules.  
Erhältlich beim Verband CAN in Automation.

## 8.10 Abkürzungsverzeichnis

**CAN**

Controller Area Network. In ISO 11898 standardisiertes serielles Bussystem, das als Basistechnologie für CANopen dient.

**CiA**

CAN in Automation e.V.. Internationaler Hersteller- und Nutzerverband mit Sitz in Erlangen/Deutschland.

**COB**

Communication Object. CAN-Telegramm mit bis zu 8 Datenbytes.

**COB-ID**

Communication Object Identifier. Telegrammadresse (nicht zu verwechseln mit Knotenadresse). CANopen verwendet die 11-Bit Identifier nach CAN 2.0A.

**NMT**

Network Management. Eines der Dienstelemente der CANopen-Spezifikation. Das Netzwerkmanagement dient zur Netzwerkinitialisierung und zur Knotenüberwachung.

**PDO**

Process Data Object oder Prozessdatenobjekt. CAN-Telegramm zur Übertragung von Prozessdaten (z. B. E/A-Daten).

**RxPDO**

Empfangs-PDO. PDOs werden immer aus Sicht des jeweiligen Gerätes bezeichnet. So wird ein TxPDO mit Eingangsdaten einer E/A Baugruppe zum RxPDO aus Sicht der Steuerung.

**SDO**

Service Data Object oder Servicedatenobjekt. CAN-Telegramm mit Protokoll zur Kommunikation mit Daten des Objektverzeichnisses (typisch Parameterdaten).

**TxPDO**

Sende-PDO (aus Sicht des CAN-Knotens bezeichnet).

## 8.11 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	BX5100 .....	13
Abb. 3	Federkontakte der Beckhoff I/O-Komponenten .....	19
Abb. 4	BX3100, BX5100, BX5200, BX9000.....	20
Abb. 5	BX8000 .....	20
Abb. 6	Enriegelter BX-Controller.....	21
Abb. 7	Verriegelter BX-Controller.....	21
Abb. 8	Demontage .....	21
Abb. 9	Potentialgruppen eines Busklemmenblocks .....	22
Abb. 10	Linksseitiger Powerkontakt .....	23
Abb. 11	Klemmstellen zur Versorgung des Busklemmen-Controllers .....	24
Abb. 12	UL-Kennzeichnung der BX-Controller .....	24
Abb. 13	Programmierkabel ZK1000-0030 - COM 1 und COM 2.....	25
Abb. 14	Programmierkabel ZK1000-0030 - Abmessungen der Stecker .....	26
Abb. 15	Programmierkabel ZK1000-0030 - Pinning .....	26
Abb. 16	SSB-Schnittstelle .....	27
Abb. 17	COM1- (RS 232) und COM2-Schnittstelle (RS 232/485) .....	27
Abb. 18	Abschluss des Busses mit Abschlusswiderstand 120 Ohm .....	28
Abb. 19	Unempfindlichkeit gegen eingeprägte Störungen.....	28
Abb. 20	Beispieltopologie Stichleitungen .....	29
Abb. 21	Aufbau CAN-Kabel ZB5100 .....	30
Abb. 22	Aufbau CAN-/DeviceNet-Kabel ZB5200 .....	30
Abb. 23	Pinbelegung BK5151, EL6751.....	32
Abb. 24	FC51x2 .....	32
Abb. 25	Belegung Verbindungsbuchse BK51x0/BX5100 .....	33
Abb. 26	LC5100 .....	34
Abb. 27	Pinbelegung M12 Stecker Feldbus Box.....	34
Abb. 28	Anlaufverhalten des Busklemmen-Controllers.....	35
Abb. 29	Anlegen einer TwinCAT-Konfiguration .....	37
Abb. 30	Auswahl des Busklemmen-Controllers .....	38
Abb. 31	Download einer TwinCAT-Konfiguration.....	39
Abb. 32	Auswahl des Busklemmen-Controllers .....	39
Abb. 33	Zustand des Busklemmen-Controllers.....	39
Abb. 34	Aktivieren der TwinCAT-Konfiguration.....	39
Abb. 35	Auswahl des Zielsystems .....	40
Abb. 36	Auswahl des Busklemmen-Controllers .....	41
Abb. 37	Zustand des Busklemmen-Controllers.....	41
Abb. 38	Hochladen der TwinCAT-Konfiguration .....	41
Abb. 39	Speicher für das Code Mapping .....	42
Abb. 40	Daten Speicher Mapping .....	42
Abb. 41	Code und Daten Speicher .....	43
Abb. 42	Sonstiger Speicher .....	43
Abb. 43	Eigenschaften der Remote-Verbindung.....	44
Abb. 44	Sendeverzögerungszeit (Inhibit Time).....	47

Abb. 45	Event Timer (Ereignis-Timer).....	47
Abb. 46	Einstellen der Node ID.....	48
Abb. 47	Karteireiter BX Settings .....	49
Abb. 48	Karteireiter BX Diag .....	50
Abb. 49	Auswahl des PLC-Projekts .....	51
Abb. 50	Verbinden vom PLC-Variable und Hardware.....	52
Abb. 51	Anzeige des Ziel-Systems .....	52
Abb. 52	Einstellen der Task-Zeit.....	53
Abb. 53	Anzeige der PLC-Zykluszeit .....	54
Abb. 54	Anfügen eines weiteren Gerätes .....	55
Abb. 55	Auswahl des CANopen Masters SSB.....	55
Abb. 56	Anfügen eines CANopen Geräts .....	56
Abb. 57	Auswahl eines CANopen-Knotens.....	56
Abb. 58	Anfügen/Bearbeiten von Objektverzeichniseinträgen.....	57
Abb. 59	NodeState, DiagFlag und EmergencyCounter .....	58
Abb. 60	Kommunikation von BX-Controller zu BX-Controller (über SSB) .....	62
Abb. 61	AX2000 .....	63
Abb. 62	CANopen Interface (X6) .....	65
Abb. 63	Cimrex-Panel am SSB des BX-Controllers.....	65
Abb. 64	IclA-Drive am SSB .....	66
Abb. 65	Anschlüsse des IclA-Drives .....	67
Abb. 66	Signalschnittstelle .....	68
Abb. 67	Feldbusanschluss .....	68
Abb. 68	Referenzierungsbereiche.....	70
Abb. 69	Listing der Referenzierungswerte .....	71
Abb. 70	Frequenzumrichter der Fa. Lenze .....	72
Abb. 71	Externe Spannungsversorgung - Interne Spannungsversorgung(Auslieferungszustand).....	73
Abb. 72	DIP-Schalter .....	73
Abb. 73	Freigabe des Kommunikationsmoduls.....	74
Abb. 74	Einstellen der Echtzeit-Uhr (RTC) .....	75
Abb. 75	Navigationsschalter der BX-Controller.....	77
Abb. 76	Schalterbelegung.....	78
Abb. 77	Maximale Anzahl der POU's überschritten.....	84
Abb. 78	Menüpfad Projekte / Optionen / Controller Settings .....	84
Abb. 79	Controller Settings .....	85
Abb. 80	Globaler Speicher nicht ausreichend.....	85
Abb. 81	Menüpfad Projekte / Optionen / Build .....	85
Abb. 82	Build.....	86
Abb. 83	Ändern der Verknüpfung von Variablen.....	92
Abb. 84	Verknüpfen einer Variable mit einem Eingang .....	92
Abb. 85	Öffnen des Optionsmenüs .....	94
Abb. 86	Auswahl des Source Downloads .....	95
Abb. 87	Download des Programm Codes.....	96
Abb. 88	Vorschritt des Downloads .....	96
Abb. 89	Upload eines Programms .....	97
Abb. 90	Auswahl des Datenübertragungswegs .....	97

Abb. 91	Auswahl des Gerätes.....	97
Abb. 92	Funktionsbaustein F_COMPORTREAD .....	105
Abb. 93	Funktionsbaustein F_COMPORTWRITE .....	106
Abb. 94	Funktionsbaustein FB_COMPORTOPEN .....	106
Abb. 95	Funktionsbaustein FB_COMPORTCLOSE .....	107
Abb. 96	Funktionsbaustein F_STARTDEBUGTIMER.....	110
Abb. 97	Funktionsbaustein F_READDEBUGTIMER .....	110
Abb. 98	Funktionsbaustein F_GETNAVSWITCH .....	110
Abb. 99	Funktionsbaustein FB_DISPWRITE .....	111
Abb. 100	Funktionsbaustein RTC .....	113
Abb. 101	Funktionsbaustein fb_ReadWriteFile.....	114
Abb. 102	Funktionsbaustein FB_BX_BK8X00_MASTER .....	116
Abb. 103	Funktionsbaustein FB_BX_BK8X00_SLAVE .....	118
Abb. 104	Kommunikationseigenschaften.....	119
Abb. 105	Funktionsbaustein FB_BX_COM_5.....	120
Abb. 106	Funktionsbaustein FB_BX_COM_64.....	120
Abb. 107	Funktionsbaustein FB_BX_COM_64EX.....	121
Abb. 108	Cimrex-Panel am COM-Port des BX-Controllers.....	123
Abb. 109	Mobiltelefon am COM-Port des BX-Controllers .....	126
Abb. 110	Funktionsbaustein F_GETVERSIONTCTWINSAFE .....	127
Abb. 111	Funktionsbaustein FB_TWINSAFE_KLX904_INPUT.....	127
Abb. 112	Funktionsbaustein FB_TWINSAFE_KLX904_input.....	128
Abb. 113	Verknüpfen der Eingangsdaten .....	129
Abb. 114	Auswahl der SafetyIn-Variable .....	129
Abb. 115	Funktionsbaustein FB_TWINSAFE_KLX904_OUTPUT .....	130
Abb. 116	Aufruf des Funktionsbausteins FB_TWINSAFE_KLX904_OUTPUT.....	131
Abb. 117	Aufruf des Funktionsbausteins FB_TWINSAFE_KLX904_OUTPUT.....	131
Abb. 118	Verknüpfen der Eingangsdaten .....	132
Abb. 119	Auswahl der entsprechenden SafetyQBx-Variable.....	132
Abb. 120	Auswahl des Datenübertragungswegs - Serielle Schnittstelle.....	133
Abb. 121	Parametrierung der seriellen Schnittstelle .....	133
Abb. 122	Auswahl des Datenübertragungswegs - AMS .....	134
Abb. 123	Auswahl des Gerätes.....	134
Abb. 124	Darstellung des BX5100 im TwinCAT System Manager .....	135
Abb. 125	Auswahl des Zielsystems .....	136
Abb. 126	Auswahl des Zielsystems - Box 2, Run-Time 1 des Busklemmen-Controllers .....	136
Abb. 127	CANopenLogo .....	137
Abb. 128	CANopen Gerätemodell.....	137
Abb. 129	Zustandsdiagramm CANopen Boot-up.....	139
Abb. 130	Schematische Darstellung „Guarding-Verfahren“ .....	141
Abb. 131	Schematische Darstellung „Heartbeat-Verfahren“ .....	143
Abb. 132	Default Identifier-Verteilung: Master/Slave .....	144
Abb. 133	PDO Linking: Peer to Peer .....	145
Abb. 134	Darstellung Übertragung CAN-Prozessdaten.....	145
Abb. 135	Darstellung CAN Telegramm „SYNC“ .....	146
Abb. 136	Zeitl. Diagramm „Inhibit-Time“ .....	148

Abb. 137 Zeitliche Darstellung des Event-Timers .....	148
Abb. 138 Darstellung Mapping .....	149
Abb. 139 SDO-Protokoll: Zugriff auf Objektverzeichnis .....	152
Abb. 140 Default Identifier .....	156
Abb. 141 Beispiel für die Zuordnung der Daten .....	181
Abb. 142 CANopen Diagnose-Byte im System Manager .....	185
Abb. 143 Zustand des K-Bus .....	186
Abb. 144 Diagnose LEDs für den Feldbus, die SPS, den K-Bus und die Spannungsversorgung .....	186
Abb. 145 Diagnose LEDs für den Feldbus, die SPS und den K-Bus .....	187
Abb. 146 Auswahl des Busklemmen-Controllers der BX-Serie .....	208
Abb. 147 Auswahl des Busklemmen-Controllers der BC-Serie .....	208
Abb. 148 Auswahl des COM-Ports .....	209
Abb. 149 Öffnen des Firmware-Files .....	209
Abb. 150 Statusmeldungen des Firmware-Updates .....	209
Abb. 151 CFC-Client .....	210
Abb. 152 Aufruf des Port-Setups .....	210
Abb. 153 Einstellen der COM-Parameter .....	211
Abb. 154 Öffnen des COM-Ports .....	211
Abb. 155 Upload des BX-Speicherinhalts in ein BIN-File .....	211
Abb. 156 Download eines BIN-Files auf den BX-Controller .....	212
Abb. 157 Typenschild eines BX-Controllers .....	213
Abb. 158 CANopen .....	215
Abb. 159 CAN - Controller-Area-Network .....	215



Mehr Informationen:  
[www.beckhoff.de/BX5100](http://www.beckhoff.de/BX5100)

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)

