BECKHOFF New Automation Technology

Documentation | EN

BX5200

Bus Terminal Controller for DeviceNet

2022-02-03 | Version: 2.3.0

Table of contents

1 Foreword

1.1 Notes on the documentation

Intended audience

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with the applicable national standards.
It is essential that the documentation and the following notes and explanations are followed when installing and commissioning these components.
It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

The responsible staff must ensure that the application or use of the products described satisfy all the requirements for safety, including all the relevant laws, regulations, guidelines and standards.

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.

We reserve the right to revise and change the documentation at any time and without prior announcement.

No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered trademarks of and licensed by Beckhoff Automation GmbH. Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Patent Pending

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents: EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702 with corresponding applications or registrations in various other countries.

EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.

Copyright

1.2 Safety instructions

Safety regulations

Please note the following safety instructions and explanations!
Product-specific safety instructions can be found on following pages or in the areas mounting, wiring, commissioning etc.

Exclusion of liability

All the components are supplied in particular hardware and software configurations appropriate for the application. Modifications to hardware or software configurations other than those described in the documentation are not permitted, and nullify the liability of Beckhoff Automation GmbH & Co. KG.

Personnel qualification

This description is only intended for trained specialists in control, automation and drive engineering who are familiar with the applicable national standards.

Description of instructions

In this documentation the following instructions are used.
These instructions must be read carefully and followed without fail!

⚠ DANGER

Serious risk of injury!

Failure to follow this safety instruction directly endangers the life and health of persons.

⚠ WARNING

Risk of injury!

Failure to follow this safety instruction endangers the life and health of persons.

⚠ CAUTION

Personal injuries!

Failure to follow this safety instruction can lead to injuries to persons.

NOTE

Damage to environment/equipment or data loss

Failure to follow this instruction can lead to environmental damage, equipment damage or data loss.

ℹ **Tip or pointer**

This symbol indicates information that contributes to better understanding.

1.3 Documentation issue status

Version	Comment
2.3.0	• Chapter *Technical data – BX* updated • Chapter *Instructions for ESD protection* added • Chapter *Disposal* added • New title page
2.2.0	• Update Technical data
2.1.0	• Download links updated • Design of safety instructions adapted to IEC 82079-1.
2.0.0	• Migration
1.2.0	• Update to firmware version 1.20
1.1.0	• Notes to meet the UL requirements added.
1.0.10	• First public issue

Firmware BX5200

The BX Controller displays its firmware version for about 3 seconds when it is switched on.
For updating the firmware you need a serial cable, the KS2000 configuration software, or the firmware update program.

Firmware	Description
1.20	• Firmware for BX Controllers from hardware version 3.5 • Switching of the COM 2 interface between RS232 and RS485 modified • New: Support for TwinSAFE Bus Terminals: maximally one logic terminal with a maximum of 7 connections allowed on the K-bus (further information) • New: 1000 bytes persistent data installed in addition to the 2 KB remanent data **Attention** Firmware version 1.20 does not run on older hardware versions (lower than 3.5). The hardware version of your BX controller can be found on its sticker.
1.10	First version

2 Product overview

2.1 Bus Terminal Controllers of the BX series

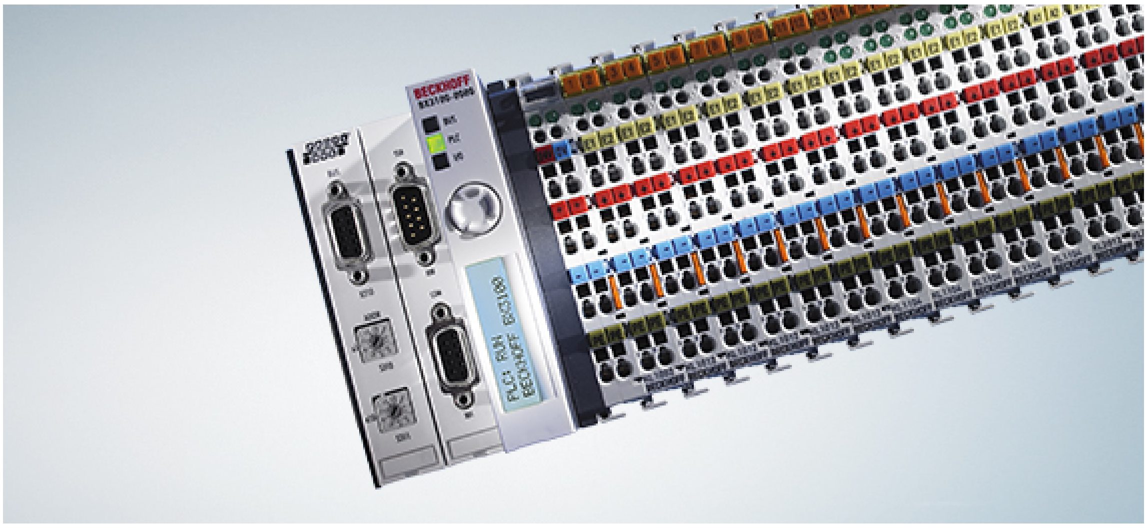


Fig. 1: Bus Terminal Controllers of the BX series

The Bus Terminal Controllers of the BX series (BX controllers) offer a high degree of flexibility. In terms of the equipment and performance range, the BX series is positioned between the BC series Bus Terminal controller and the CX1000 Embedded PC. The concept of a stand-alone controller in combination with a link to a higher-level fieldbus system is based on the BC series. The housing design originates from the CX1000. The main features distinguishing the BC and BX series are the larger memory and the expanded interfaces of the BX series.

The BX controllers consist of a programmable IEC 61131-3 controller, a connection to the higher-level fieldbus system and the K-bus interface for connecting the Beckhoff Bus Terminals. In addition, the BX controllers have two serial interfaces: one for programming, the other for free utilization. The device itself includes an illuminated LC display (2 rows with 16 characters each) with joystick switch and a real-time clock. Further peripheral devices, e.g. displays, can be connected via the integrated Beckhoff Smart System Bus (SSB).

The Bus Terminals are connected on the right side of the BX controller, as usual. The comprehensive range of different I/Os enables any input signal to be read and any output signal that may be required to be generated. The BX controllers can be used for a wide range of automation tasks, from garage door control to autonomous temperature control at injection molding machines. The BX controllers are also eminently suitable for a modular machine concept. Within a network, the BX controllers can exchange data with other machine components via the fieldbus interfaces. The real-time clock also enables decentralized applications, for which the day of the week or the time play an important role.

The areas of application of this series are similar to that of the BC series, but due to the larger memory the BX can execute significantly more complex and larger programs and can manage more data locally (e.g. history and trend data recording), which are then successively fetched over the fieldbus.

Bus Terminal and end terminal required

To operate a BX controller, at least one Bus Terminal with process image and the end terminal must be connected to its K-bus.

Fieldbus interface

The variants of the BX series Bus Terminal Controllers differ in terms of their fieldbus interfaces. Additionally, two serial interfaces are integrated for programming and for the connection of further serial devices. Five different versions cover the main fieldbus systems:

- BX3100: PROFIBUS DP

- BX5100: CANopen
- BX5200: DeviceNet
- BX8000: RS232 or RS485 (without fieldbus interface)
- BX9000: Ethernet ModbusTCP/ADS-TCP/UDP

Programming

The BX controllers are programmed based on the effective IEC 61131-3 standard. As with all other Beckhoff controllers, the TwinCAT automation software is the basis for parameterization and programming. Users therefore have the familiar TwinCAT tools available, e.g. PLC programming interface, System Manager and TwinCAT Scope. Data is exchanged optionally via the serial port (COM1) or via the fieldbus through Beckhoff PC FCxxxx fieldbus cards.

Configuration

The configuration is also carried out using TwinCAT. The fieldbus interface, the SSB bus and the real-time clock can be configured and parameterized via the System Manager. The System Manager can read all connected devices and Bus Terminals. After the parameterization, the configuration is saved on the BX via the serial interface. The configuration thus created can be accessed again later.

2.2 BX5200 - Introduction

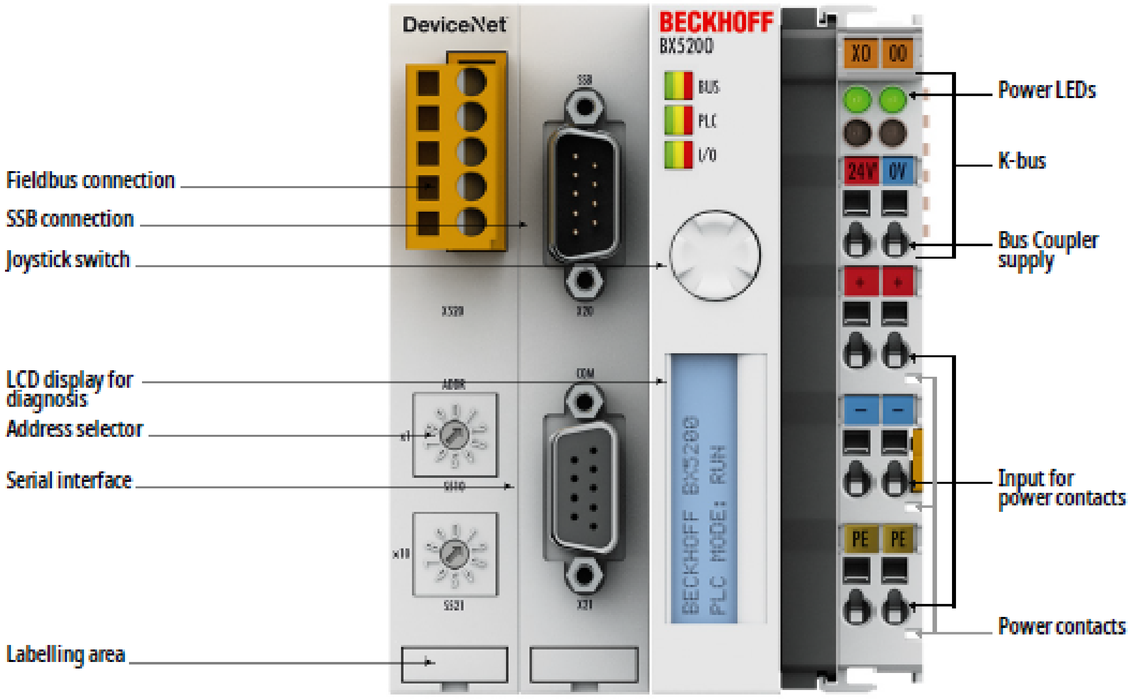


Fig. 2: BX5200

The BX5200 Bus Terminal Controller has a DeviceNet slave interface. It has automatic baud rate detection up to 500 kbaud and an address selection switch for address assignment. Up to 512 byte of input and 512 byte of output can be exchanged with the controller.

One unit consists of the BX5200 Bus Terminal Controller with up to 64 Bus Terminals and a bus end terminal. With the terminal bus extension system, the connection of up to 255 Bus Terminals is possible. The controller is programmed via the COM1 or via the DeviceNet interface of the FC510x PC Fieldbus Card.

2.3 Technical data

2.3.1 Technical data - BX

Technical data	BX3100	BX5100	BX5200	BX8000	BX9000
Processor	16 bit micro-controller				
Diagnostic LEDs	2 x power supply, 2 x K-Bus				
Display	FSTN 2 x 16 lines display for diagnosis or own texts, illuminated				
Switch	Joystick switch for parameterization and diagnosis				
Clock	battery-powered internal clock for time and date				
Configuration and programming software	TwinCAT PLC				
Fieldbus interface	PROFIBUS DP	CANopen	DeviceNet	-	Ethernet
Fieldbus connection	D-sub, 9-pin	Open style connector, 5 pin		-	RJ45
SSB	CANopen-based sub-bus interface				
Terminal Bus (K-Bus)	64 (255 with K-bus extension)				
Digital peripheral signals	2040 inputs/outputs				
Analog peripheral signals	1024 inputs/outputs				
Configuration possibility	via TwinCAT or the controller				
max. number of bytes, fieldbus	depending on fieldbus				
max. number of bytes, PLC	2048 bytes of input data, 2048 bytes of output data				

Supply	BX3100	BX5100	BX5200	BX8000	BX9000
Power supply (Us)	24 V_{DC} (-15% /+20%)				
Input current (Us)	180 mA + (total K-bus current)/4, see UL requirements				
Starting current (Us)	approx. 2.5 x continuous current				
K-bus current (5 V)	maximum 1450 mA				
Power contact voltage (Up)	24 V_{DC} max.				
Power contact current load (Up)	max. 10 A, see UL requirements				
Dielectric strength	500 V (power contact/supply voltage/Ethernet/fieldbus)				

⚠ CAUTION

UL requirements

For power supplies of the BX Controller (Us) und the Power Contacts (Up) use a 4 A fuse or an *NEC Class 2*-compliant power supply to meet the UL requirements!

Technical data	BX3100	BX5100	BX5200	BX8000	BX9000
Permissible ambient temperature range during operation	0°C ... +55 °C (bevore hardware version 4.4) -25°C ... +60 °C (from hardware version 4.4)				0°C ... +55 °C
Permissible ambient temperature range during storage	-20°C ... +85 °C (bevore hardware version 4.4) -40°C ... +85 °C (from hardware version 4.4)				-20°C ... +85 °C
Relative humidity	95 % no condensation				
Vibration / shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27				
EMC immunity / emission	conforms to EN 61000-6-2 / EN 61000-6-4				
Protection class	IP20				
Approvals/markings*	CE, UKCA, cULus, EAC				

*) Real applicable approvals/markings see type plate on the side (product marking).

Mechanical data	BX3100	BX5100	BX5200	BX8000	BX9000
Weight	app. 170 g				
Dimensions (W x H x D)	app. 83 mm x 100 mm x 90 mm (BX8000: app. 65 mm x 100 mm x 90 mm)				
Mounting	with latch, on mounting rail (35 mm DIN rail)				
Installation position	any				
Connection cross-section	0.08 mm² ... 2.5 mm² AWG 28 ... 14 8 ... 9 mm strip length				

2.3.2 Technical Data - DeviceNet

System data	DeviceNet (BX5200)		
Number of nodes	64		
Number of I/O points	depending on controller		
Data transfer medium	shielded, twisted copper wire with energy supply, 5-pin		
Cable length	500 m	250 m	100 m
Data transfer rate	125 kBaud	250 kBaud	500 kBaud
I/O communication types	Polling		
Max. number of bytes, fieldbus	255 bytes input and 255 bytes output		
Baud rate	Automatic baud rate detection (for possible baud rates see data transfer rate)		

2.3.3 Technical data - SSB Interface

System data	SSB Interface
Max. number of slaves	8
Max. number of PDOs	32 RxPODs / 32 TxPDOs
Baud rate	10 k ... 1 MBaud
Permitted slave addresses	1 to 64

2.3.4 Technical data - PLC

PLC data	BX3100	BX5100	BX5200	BX8000	BX9000
Programmability	via programming interface (COM1 or COM2) or via fieldbus				
Program memory	256 kbyte				
Source code memory	256 kbyte				
Data memory	256 kbyte				
Remanent flags	2 kbyte				
PLC cycle time	Approx. 0.85 ms for 1000 IL commands (without I/O cycle)				
Programming languages	IEC 6-3 (IL, LD, FBD, ST, SFC)				
Propagation delay	1 PLC task (second task in preparation)				
Online change	Yes				
Up/Down Load Code	Yes/Yes				

2.4 The principle of the Bus Terminal

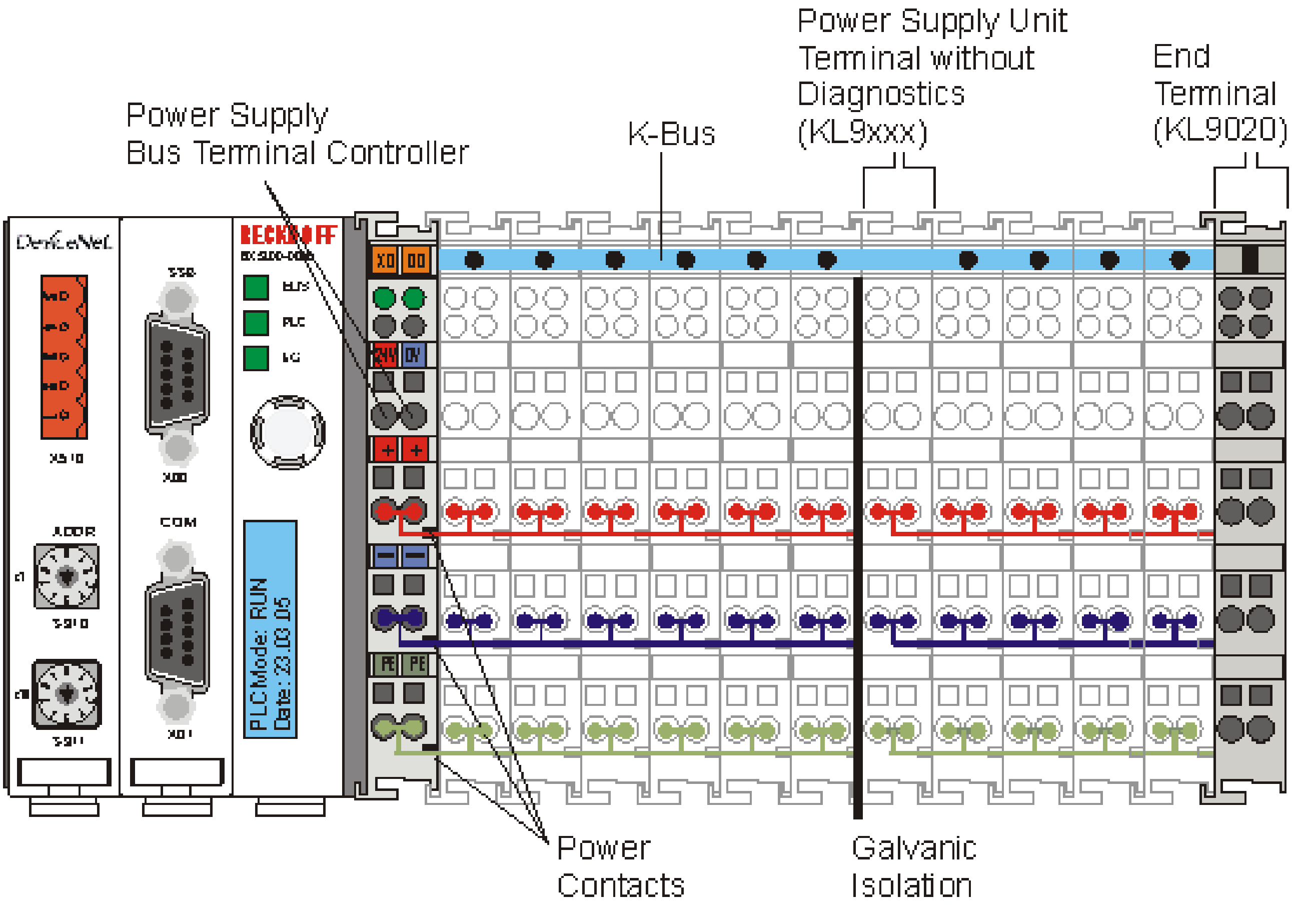


Fig. 3: The principle of the Bus Terminal

2.5 The Beckhoff Bus Terminal system

Up to 256 Bus Terminals, with 1 to 16 I/O channels per signal form

The Bus Terminal system is the universal interface between a fieldbus system and the sensor / actuator level. A unit consists of a Bus Coupler as the head station, and up to 64 electronic series terminals, the last one being an end terminal. Up to 255 Bus Terminals can be connected via the K-Bus extension. For each technical signal form, terminals are available with one, two, four or eight I/O channels, which can be mixed as required. All the terminal types have the same mechanical construction, so that difficulties of planning and design are minimized. The height and depth match the dimensions of compact terminal boxes.

Decentralized wiring of each I/O level

Fieldbus technology allows more compact forms of controller to be used. The I/O level does not have to be brought to the controller. The sensors and actuators can be wired decentrally, using minimum cable lengths. The controller can be installed at any location within the plant.

Industrial PCs as controllers

The use of an Industrial PC as the controller means that the operating and observing element can be implemented in the controller's hardware. The controller can therefore be located at an operating panel, in a control room, or at some similar place. The Bus Terminals form the decentralized input/output level of the controller in the control cabinet and the subsidiary terminal boxes. The power sector of the plant is also controlled over the bus system in addition to the sensor/actuator level. The Bus Terminal replaces the conventional series terminal as the wiring level in the control cabinet. The control cabinet can have smaller dimensions.

Bus Couplers for all usual bus systems

The Beckhoff Bus Terminal system unites the advantages of a bus system with the possibilities of the compact series terminal. Bus Terminals can be driven within all the usual bus systems, thus reducing the controller parts count. The Bus Terminals then behave like conventional connections for that bus system. All the performance features of the particular bus system are supported.

Mounting on standardized mounting rails

The installation is standardized thanks to the simple and space-saving mounting on a standardized mounting rail (EN 60715, 35 mm) and the direct wiring of actuators and sensors, without cross connections between the terminals. The consistent labelling scheme also contributes.

The small physical size and the great flexibility of the Bus Terminal system allow it to be used wherever a series terminal is also used. Every type of connection, such as analog, digital, serial or the direct connection of sensors can be implemented.

Modularity

The modular assembly of the terminal strip with Bus Terminals of various functions limits the number of unused channels to a maximum of one per function. The presence of two channels in one terminal is the optimum compromise of unused channels and the cost of each channel. The possibility of electrical isolation through potential feed terminals also helps to keep the number of unused channels low.

Display of the channel state

The integrated LEDs show the state of the channel at a location close to the sensors and actuators.

K-Bus

The K-Bus is the data path within a terminal strip. The K-Bus is led through from the Bus Coupler through all the terminals via six contacts on the terminals' side walls. The end terminal terminates the K-Bus. The user does not have to learn anything about the function of the K-Bus or about the internal workings of the terminals and the Bus Coupler. Many software tools that can be supplied make project planning, configuration and operation easy.

Potential feed terminals for isolated groups

The operating voltage is passed on to following terminals via three power contacts. You can divide the terminal strip into arbitrary isolated groups by means of potential feed terminals. The potential feed terminals play no part in the control of the terminals, and can be inserted at any locations within the terminal strip.

Up to 64 Bus Terminals can be used in a terminal block, with optional K-Bus extension for up to 256 Bus Terminals. This count does include potential feed terminals, but not the end terminal.

Bus Couplers for various fieldbus systems

Various Bus Couplers can be used to couple the electronic terminal strip quickly and easily to different fieldbus systems. It is also possible to convert to another fieldbus system at a later time. The Bus Coupler performs all the monitoring and control tasks that are necessary for operation of the connected Bus Terminals. The operation and configuration of the Bus Terminals is carried out exclusively by the Bus Coupler. Nevertheless, the parameters that have been set are stored in each Bus Terminal, and are retained in the event of voltage drop-out. Fieldbus, K-Bus and I/O level are electrically isolated.

If the exchange of data over the fieldbus is prone to errors or fails for a period of time, register contents (such as counter states) are retained, digital outputs are cleared, and analog outputs take a value that can be configured for each output when commissioning. The default setting for analog outputs is 0 V or 0 mA. Digital outputs return in the inactive state. The timeout periods for the Bus Couplers correspond to the usual settings for the fieldbus system. When converting to a different bus system it is necessary to bear in mind the need to change the timeout periods if the bus cycle time is longer.

The interfaces

A Bus Coupler has six different methods of connection. These interfaces are designed as plug connectors and as spring-loaded terminals.

3 Mounting and wiring

3.1 Instructions for ESD protection

NOTE

Destruction of the devices by electrostatic discharge possible!

The devices contain components at risk from electrostatic discharge caused by improper handling.

- Please ensure you are electrostatically discharged and avoid touching the contacts of the device directly.
- Avoid contact with highly insulating materials (synthetic fibers, plastic film etc.).
- Surroundings (working place, packaging and personnel) should by grounded probably, when handling with the devices.
- Each assembly must be terminated at the right hand end with a KL9010 bus end terminal, to ensure the protection class and ESD protection.

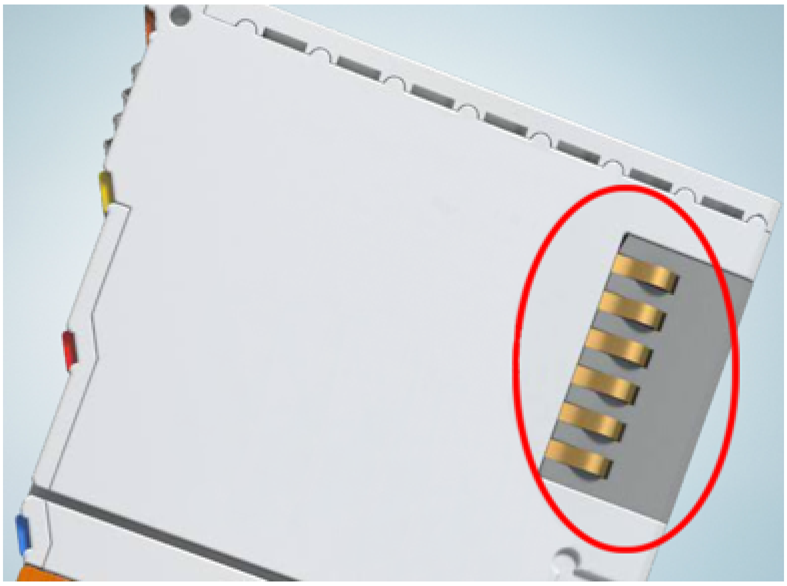


Fig. 4: Spring contacts of the Beckhoff I/O components

3.2 Mounting

⚠ **WARNING**

Risk of injury through electric shock and damage to the device!

De-energize the Bus Terminal I/O system before you start installation, disassembly or wiring of the components!

3.2.1 Dimensions

The Beckhoff Bus Terminal system is characterized by low physical volume and high modularity. When planning a project it must be assumed that at least one Bus Coupler and a number of Bus Terminals will be used. The dimensions of the Bus Terminal Controllers are independent of the fieldbus system.

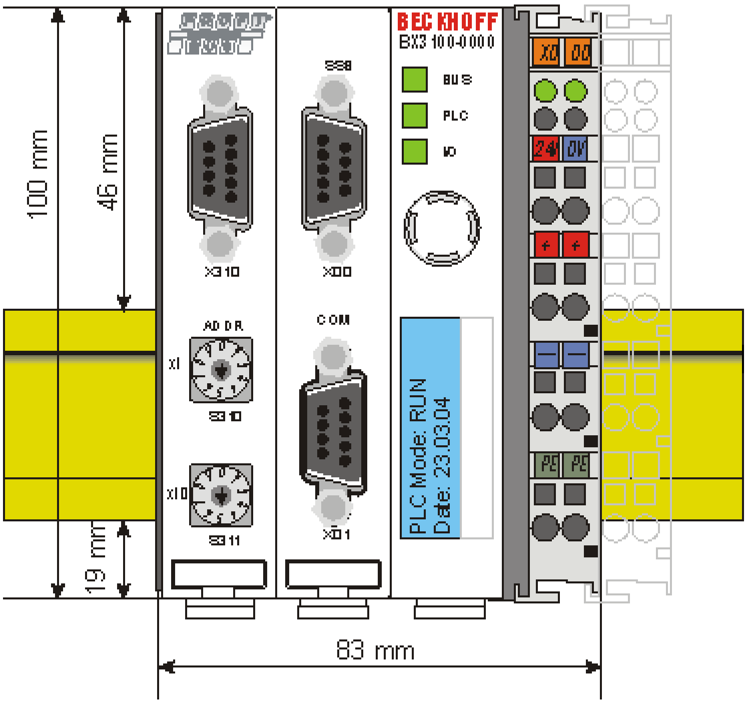


Fig. 5: BX3100, BX5100, BX5200, BX9000

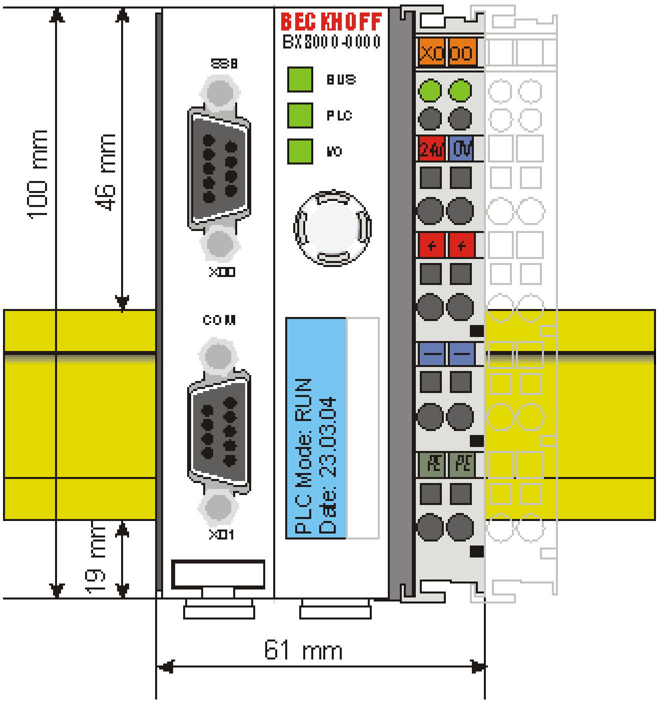


Fig. 6: BX8000

The overall width of the fieldbus station is the width of the Bus Terminal Controller plus the widths of the individual Bus Terminals (including the KL9010 bus end terminal). Depending on design, the Bus Terminals are 12 mm or 24 mm wide. The height is 100 mm.

The BX series Bus Terminal Controllers are up to 83 mm wide and 91 mm deep.

Pay attention to the total depth

Note that a Bus Terminal Controller with DIN rail and connected plug connectors is usually higher than the specified value of 91 mm. Example:
BX3100 + ZB3100 + DIN rail = 105 mm

3.2.2 Installation on mounting rails

Mounting

1. The white pull-tabs on the underside of the BX controller are connected to a latching mechanism. Pull the tabs downwards before pushing the BX controller onto the mounting rail.

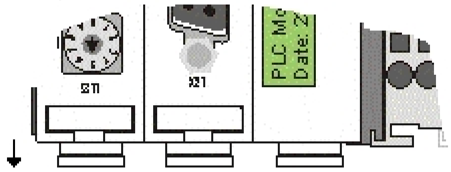


Fig. 7: Released BX controller

NOTE

Avoid damaging the display during the installation!

Avoid pressing on the display when you push the BX controller onto the mounting rail, in order to avoid damaging the display.

2. Now press the BX controller onto the mounting rail.
3. Once it has snapped onto the mounting rail, push the tabs back into their initial position.

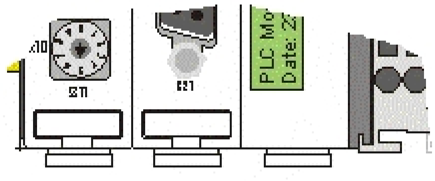


Fig. 8: Latched BX controller

Disassembly

1. First release all pull tabs on the underside of the BX controller.
2. Then pull the orange tab next to the power supply for the power contacts.

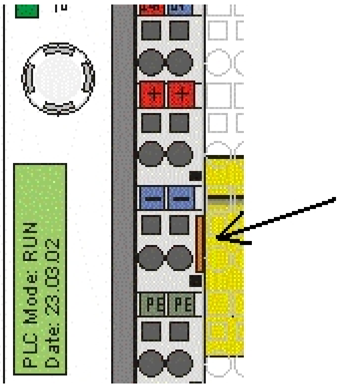


Fig. 9: Disassembly

3.3 Disposal

Products marked with a crossed-out wheeled bin shall not be discarded with the normal waste stream. The device is considered as waste electrical and electronic equipment. The national regulations for the disposal of waste electrical and electronic equipment must be observed.

3.4 Wiring

⚠ WARNING

Risk of injury through electric shock and damage to the device!

De-energize the Bus Terminal I/O system before you start installation, disassembly or wiring of the components!

3.4.1 Potential groups, insulation testing and PE

Potential groups

A Beckhoff Bus Terminal block usually has three different potential groups:

- The fieldbus interface is electrically isolated (except for individual Low Cost couplers) and forms the first potential group.
- Bus Coupler / Bus Terminal Controller logic, K-bus and terminal logic form a second electrically isolated potential group.
- The inputs and outputs are supplied via the power contacts and form further potential groups.

Groups of I/O terminals can be consolidated to further potential groups via potential supply terminals or separation terminals.

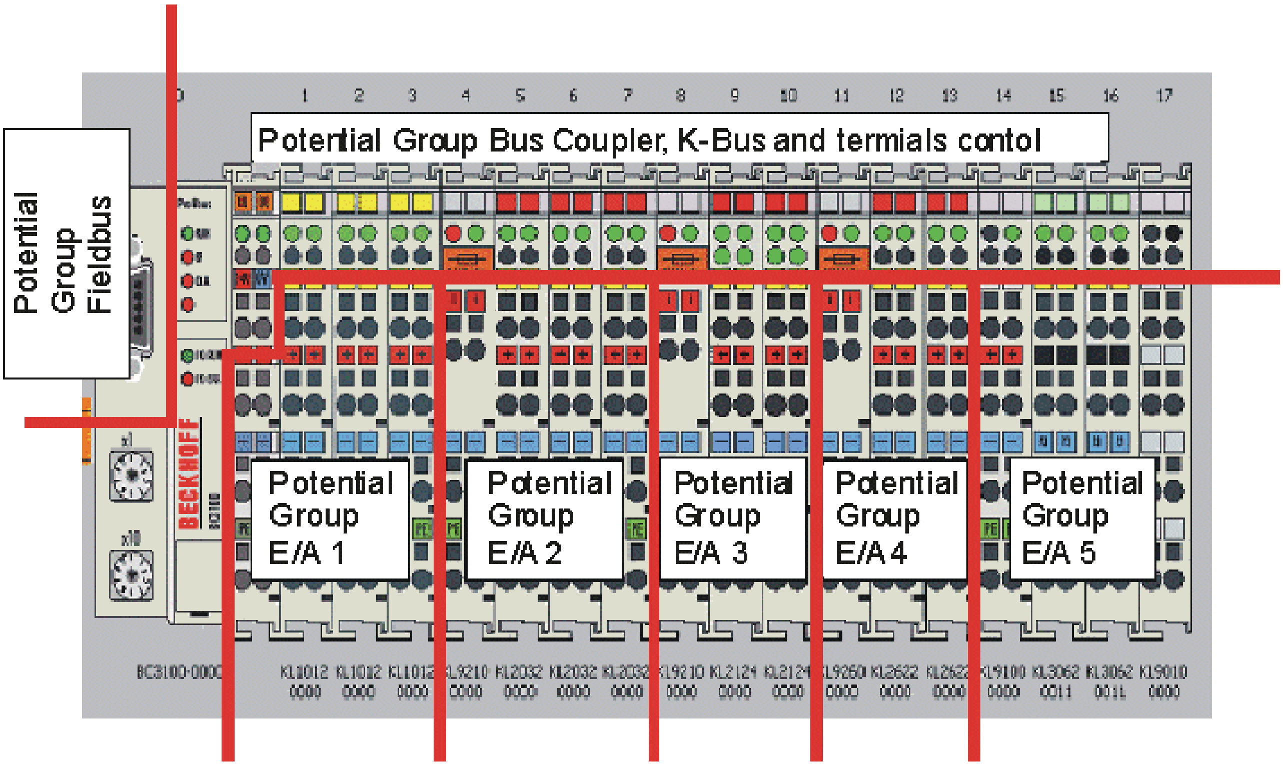


Fig. 10: Potential groups of a Bus Terminal block

Insulation testing

The connection between Bus Coupler / Bus Terminal Controller and Bus Terminals is realized automatically by latching the components. The transfer of the data and the supply voltage for the intelligent electronics in the Bus Terminals is performed by the K-bus. The supply of the field electronics is performed through the power contacts. Plugging together the power contacts creates a supply rail. Since some Bus Terminals (e.g. analog Bus Terminals or 4-channel digital Bus Terminals) are not looped through these power contacts or not completely the Bus Terminal contact assignments must be considered.

The potential feed terminals interrupt the power contacts, and represent the start of a new supply rail. The Bus Coupler / Bus Terminal Controller can also be used for supplying the power contacts.

PE power contacts

The power contact labelled PE can be used as a protective earth. For safety reasons this contact mates first when plugging together, and can ground short-circuit currents of up to 125 A.

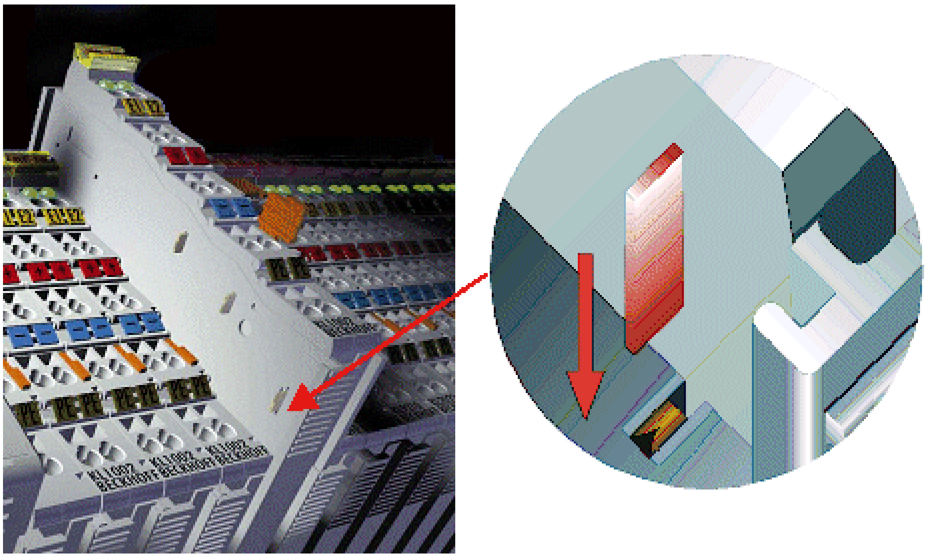


Fig. 11: Power contact on the left

It should be noted that, for reasons of electromagnetic compatibility, the PE contacts are capacitively coupled to the mounting rail. This can both lead to misleading results and to damaging the terminal during insulation testing (e.g. breakdown of the insulation from a 230 V power consuming device to the PE conductor). The PE supply line at the Bus Coupler / Bus Terminal Controller must be disconnected for an insulation test. In order to uncouple further feed locations for the purposes of testing, the feed terminals can be pulled at least 10 mm out from the connected group of other terminals. In that case, the PE conductors do not have to be disconnected.

The power contact with the label PE must not be used for other potentials.

3.4.2 Power supply

⚠ CAUTION

c UL US

Note the UL requirements for the power supply.

These UL requirements apply to all supply voltages of the BX controller (Us and Up)!

To comply with UL requirements, the BX controllers may only be connected to supply voltages (24 V_{DC}) that originate

- from an isolated source protected by a fuse of max. 4A (according to UL248) or
- from a voltage supply complying with NEC class 2.
  An NEC class 2 voltage source must not be connected in series or parallel with another NEC class 2 voltage source!

⚠ CAUTION

c UL US

No unlimited voltage sources!

In order to comply with UL requirements, the BX controllers must not be connected to unlimited voltage sources!

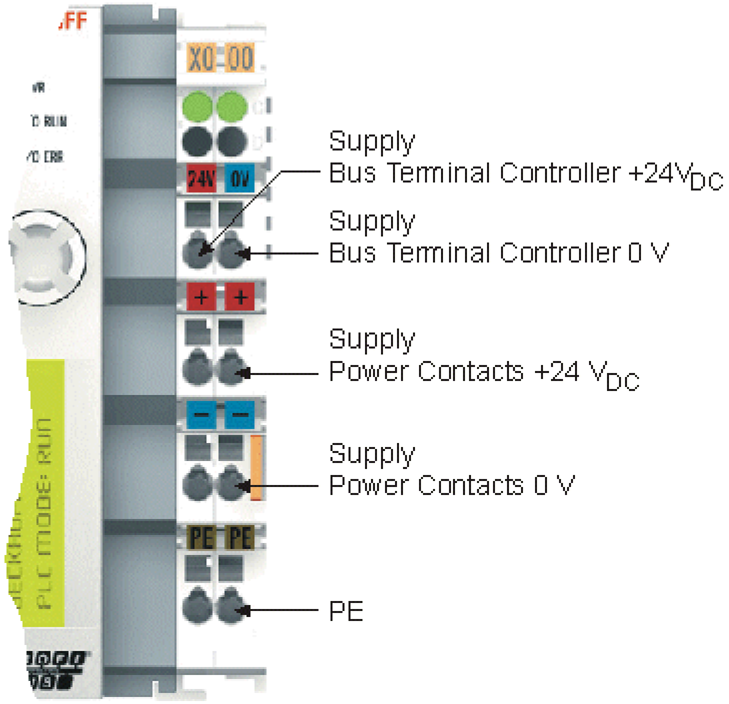
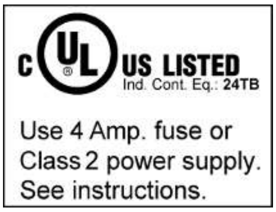

Supply
Bus Terminal Controller +24V$_{DC}$
Supply
Bus Terminal Controller 0 V
Supply
Power Contacts +24 V$_{DC}$
Supply
Power Contacts 0 V
PE
X0 00
24V
0V
+
+
–
–
PE
PE
PLC mode: RUN

Fig. 12: Terminal points for the Bus Terminal Controller supply


c UL US LISTED
Ind. Cont. Eq.: 24TB

Use 4 Amp. fuse or
Class 2 power supply.
See instructions.

Fig. 13: UL marking of the BX Controller

Supply of Bus Terminal Controller and Bus Terminals (Us)

The Bus Terminal Controller requires a supply voltage of 24 V$_{DC}$.

The BX controller is connected via the upper terminal points labelled *24 V* and *0 V*. This voltage supplies the Bus Terminal Controller electronics, and the Bus Terminal electronics via the K-bus. It is galvanically separated from the field level voltage.

Power contacts supply (Up)

The bottom six connections with spring-loaded terminals can be used to feed the supply for the peripherals. The spring-loaded terminals are joined in pairs to a power contact. The feed for the power contacts has no connection to the voltage supply for the BX electronics. The design of the feed permits voltages of up to 24 V.

The spring-loaded terminals are designed for wires with cross-sections from 0.08 mm^2 to 2.5 mm^2.

The assignment in pairs and the electrical connection between feed terminal contacts allows the connection wires to be looped through to various terminal points. The current load from the power contact must not exceed 10 A for long periods. The current carrying capacity between two spring-loaded terminals is identical to that of the connecting wires.

Power contacts

On the right hand face of the Bus Terminal Controller there are three spring contacts for the power contact connections. The spring contacts are hidden in slots so that they cannot be accidentally touched. By attaching a Bus Terminal the blade contacts on the left hand side of the Bus Terminal are connected to the spring contacts. The tongue and groove guides on the top and bottom of the Bus Terminal Controllers and of the Bus Terminals guarantees that the power contacts mate securely.

3.4.3 Programming cable for COM1

You can use a 1:1 cable for programming the BX Controllers (socket/plug, and only connect the pins listed below). On the BX side you need a nine-pin connector, and on the PC side usually a nine-pin socket. The wiring is 1:1, and the necessary pins can be found in the table below. The length of the cable should not exceed 5 meters!

Description	BX COM Port 1	PC COM port RS 232 serial interface
Cable	Plug connector, pin	Socket, pin
RS 232 RxD/TxD	2	2
RS 232 RxD/TxD	3	3
GND	5	5

NOTE

All pins that are not listed in the table are reserved

Please note that pins that are not listed are not freely available at this COM port [▶ 24], but are reserved for other signals.

ZK1000-0030

The programming cable can be used to program the BX controller via the COM 1 interface and connect another serial device at the COM 2 interface. Once installed, make sure the maximum overall height of the plug connector is not exceeded.

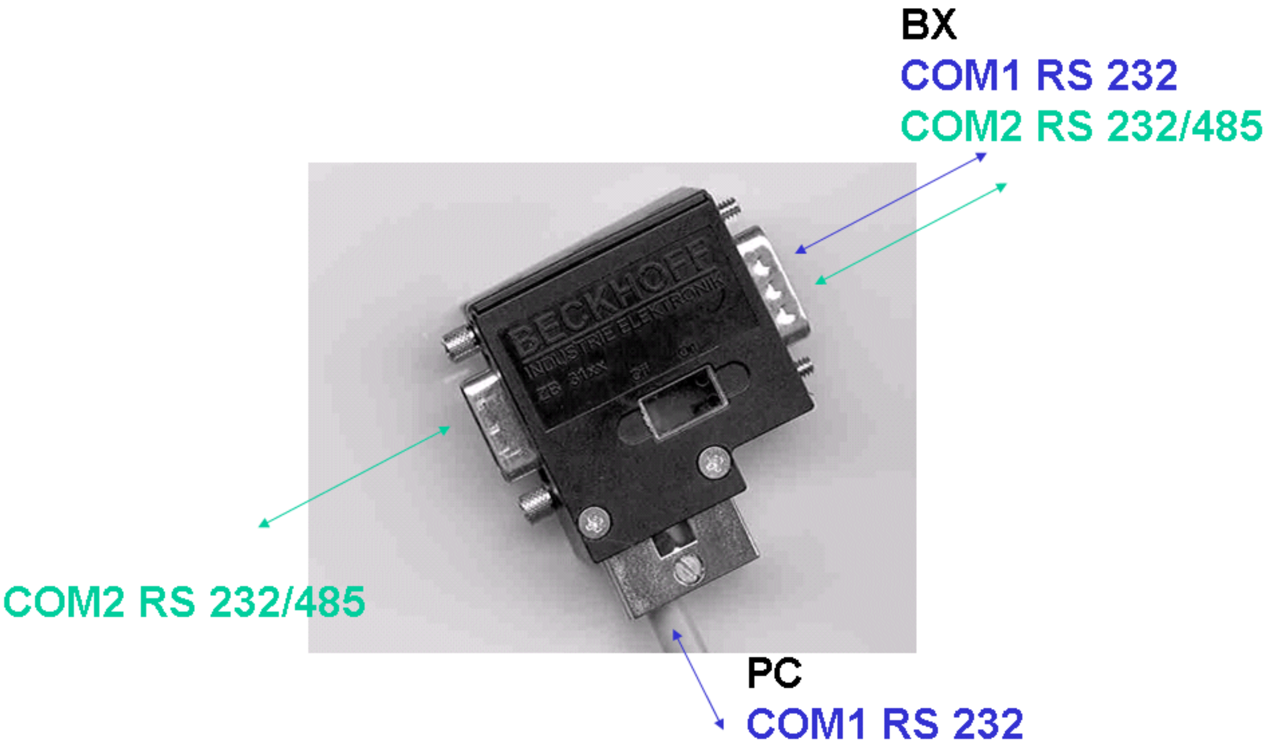


Fig. 14: Programming cable ZK1000-0030 - COM 1 and COM 2

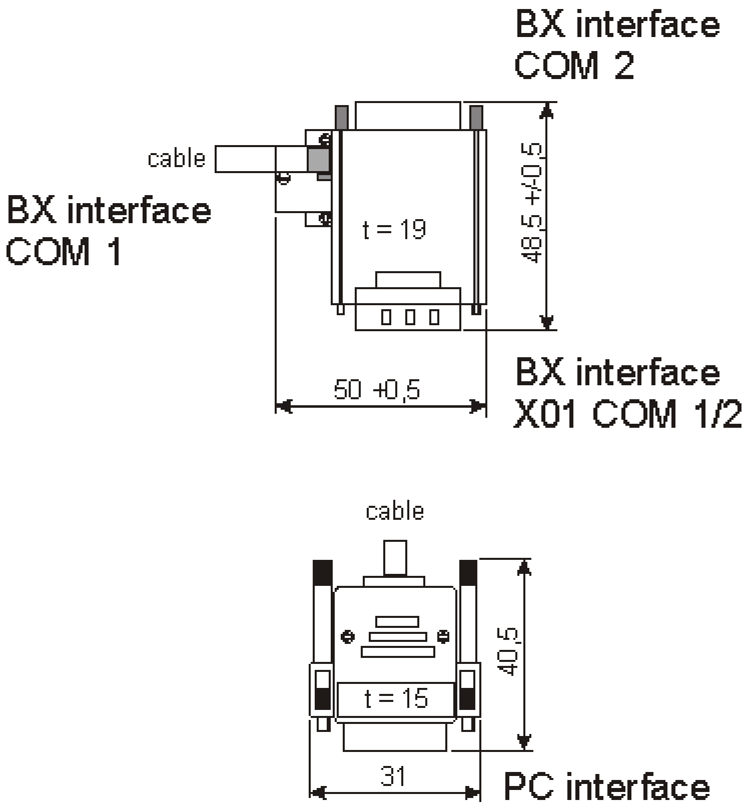


Fig. 15: Programming cable ZK1000-0030 - plug connector dimensions

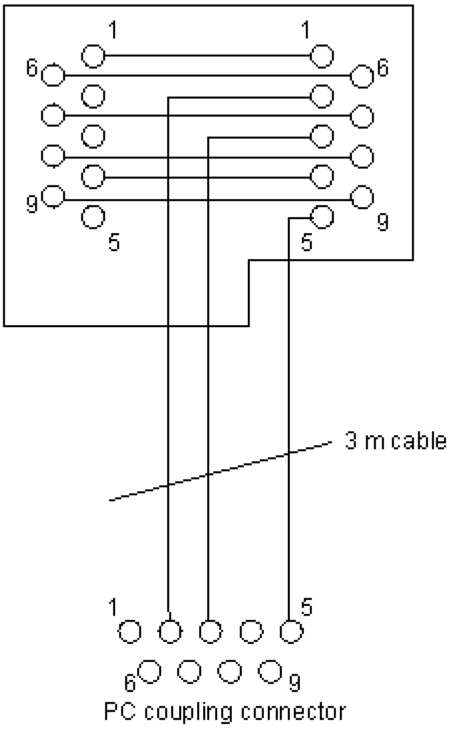


Fig. 16: Programming cable ZK1000-0030 - Pinning

3.4.4 SSB and COM interface

The basic BX controller module the COM1, COM2 and SSB (Smart System Bus) interfaces. A D-sub socket is used for COM1 and COM2. A special programming cable (ZK1000-0030) for connecting the two interfaces is available from Beckhoff. The COM2 interface is intended for the connection of serial devices. For the COM2 interface, you can choose between RS232 or RS485.
Libraries [▶ 113] are available for the serial COM2 interface.

SSB interface

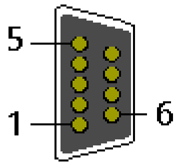


Fig. 17: SSB interface

SSB interface assignment (plug connector X00)

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

COM1 (RS 232) and COM2 (RS 232/485) interface

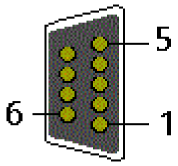


Fig. 18: COM1 (RS 232) and COM2 (RS 232/485) interface

COM interface assignment (socket X01)

PIN	Interface	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 cabling

Notes related to checking the CAN wiring can be found in the Trouble Shooting [▸ 66] section.

3.4.5.1 CAN topology

CAN is a 2-wire bus system, to which all participating devices are connected in parallel (i.e. using short drop lines). The bus must be terminated at each end with a 120 (or 121) Ohm terminating resistor to prevent reflections. This is also necessary even if the cable lengths are very short!

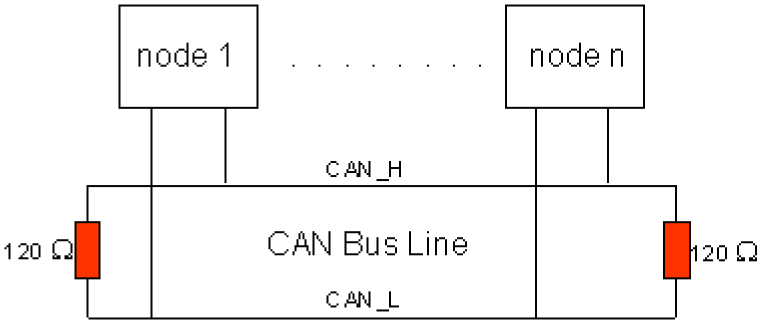


Fig. 19: Termination of the bus with a 120 Ohm termination resistor

Since the CAN signals are represented on the bus as the difference between the two levels, the CAN leads are not very sensitive to incoming interference (EMI): Both leads are affected, so the interference has very little effect on the difference.

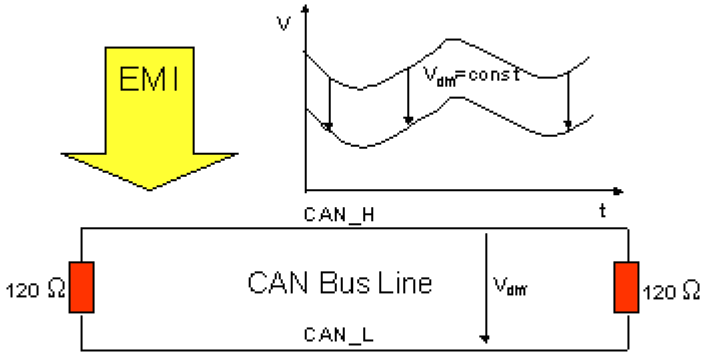


Fig. 20: Insensitivity to incoming interference

3.4.5.2 Bus length

The maximum length of a CAN bus is primarily limited by the signal propagation delay. The multi-master bus access procedure (arbitration) requires signals to reach all the nodes at effectively the same time (before the sampling within a bit period). Since the signal propagation delays in the CAN connecting equipment (transceivers, opto-couplers, CAN controllers) are almost constant, the line length must be chosen in accordance with the baud rate:

Baud rate	Bus length
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

*) A figure of 40 m at 1 Mbit/s is often found in the CAN literature. This does not, however, apply to networks with optically isolated CAN controllers. The worst case calculation for opto-couplers yields a figure 5 m at 1 Mbit/s - in practice, however, 20 m can be reached without difficulty.

It may be necessary to use repeaters for bus lengths greater than 1000 m.

3.4.5.3 Drop lines

Drop lines must always be avoided as far as possible, since they inevitably cause reflections. The reflections caused by drop lines are not however usually critical, provided they have decayed fully before the sampling time. In the case of the bit timing settings selected in the Bus Couplers it can be assumed that this is the case, provided the following drop line lengths are not exceeded:

Baud rate	Drop line length	Total length of all drop lines
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

Drop lines must not have terminating resistors.

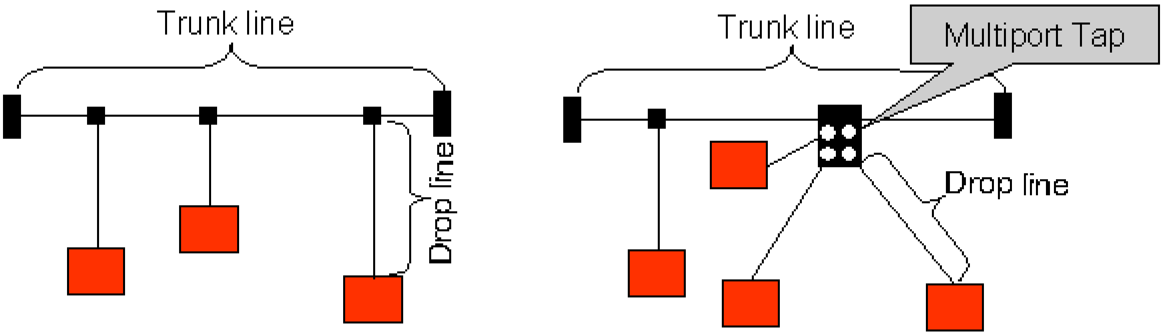


Fig. 21: Sample topology of drop lines

3.4.5.4 Star Hub (Multiport Tap)

Shorter drop line lengths must be maintained when passive distributors ("multiport taps"), such as the Beckhoff ZS5052-4500 Distributor Box. The following table indicates the maximum drop line lengths and the maximum length of the trunk line (without the drop lines):

Baud rate	Drop line length with multiport topology	Trunk line length (without drop lines)
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 cable

Screened twisted-pair cables (2x2) with a characteristic impedance of between 108 and 132 Ohm is recommended for the CAN wiring. If the CAN transceiver’s reference potential (CAN ground) is not to be connected, the second pair of conductors can be omitted. (This is only recommended for networks of small physical size with a common power supply for all the participating devices).

ZB5100 CAN Cable

A high quality CAN cable with the following properties is included in Beckhoff's range:

- 2 x 2 x 0.25 mm² (AWG 24) twisted pairs, cable colors: red/black + white/black
- double screened
- braided screen with filler strand (can be attached directly to pin 3 of the 5-pin connection terminal)
- flexible (minimum bending radius 35 mm when bent once, 70 mm for repeated bending)
- characteristic impedance (60 kHz): 120 ohm
- conductor resistance < 80 Ohm/km
- sheath: grey PVC, outside diameter 7.3 +/- 0.4 mm
- Weight: 64 kg/km.
- printed with "Beckhoff ZB5100 CAN-BUS 2x2x0.25" and meter marking (length data every 20cm)

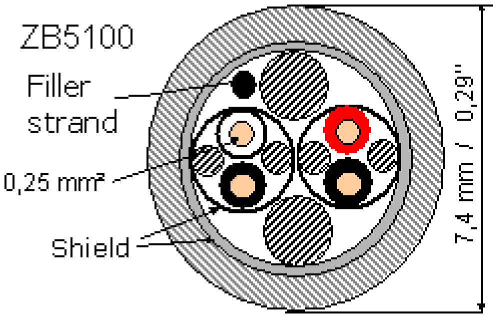


Fig. 22: Structure of CAN cable ZB5100

ZB5200 CAN/DeviceNet Cable

The ZB5200 cable material corresponds to the DeviceNet specification, and is also suitable for CANopen systems. The ready-made ZK1052-xxxx-xxxx bus cables for the Fieldbus Box modules are made from this cable material. It has the following specification:

- 2 x 2 x 0.34 mm² (AWG 22) twisted pairs
- double screened, braided screen with filler strand
- characteristic impedance (1 MHz): 126 ohm
- Conductor resistance 54 Ohm/km
- sheath: grey PVC, outside diameter 7.3 mm
- printed with "InterlinkBT DeviceNet Type 572" as well as UL and CSA ratings
- stranded wire colors correspond to the DeviceNet specification
- UL recognized AWM Type 2476 rating
- CSA AWM I/II A/B 80°C 300V FT1
- corresponds to the DeviceNet "Thin Cable" specification

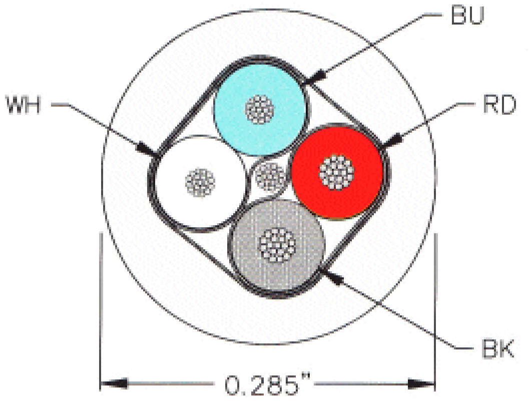


Fig. 23: Structure of CAN/DeviceNet cable ZB5200

3.4.5.6 Shielding

The screen is to be connected over the entire length of the bus cable, and only galvanically grounded at one point, in order to avoid ground loops.
The design of the screening, in which HF interference is diverted through R/C elements to the mounting rail assumes that the rail is appropriately earthed and free from interference. If this is not the case, it is possible that HF interference will be transmitted from the mounting rail to the screen of the bus cable. In that case the screen should not be attached to the couplers - it should nevertheless still be fully connected through.

Notes related to checking the CAN wiring can be found in the Trouble Shooting [▶ 66] section.

3.4.5.7 Cable colors

Suggested method of using the Beckhoff CAN cable on Bus Terminal and Fieldbus Box:

BK51x0 pin PIN BX5100 (X510)	Pin BK5151 CX8050, CX8051, CXxxxx-B510/M510	Fieldbus Box pin	Pin FC51xx	Function	ZB5100 cable color	ZB5200 cable color
1	3	3	3	CAN Ground	**black/** (red)	**black**
2	2	5	2	CAN Low	**black**	**blue**
3	5	1	5	Shield	Filler strand	Filler strand
4	7	4	7	CAN high	**white**	**white**
5	9	2	9	not used	**(red)**	**(red)**

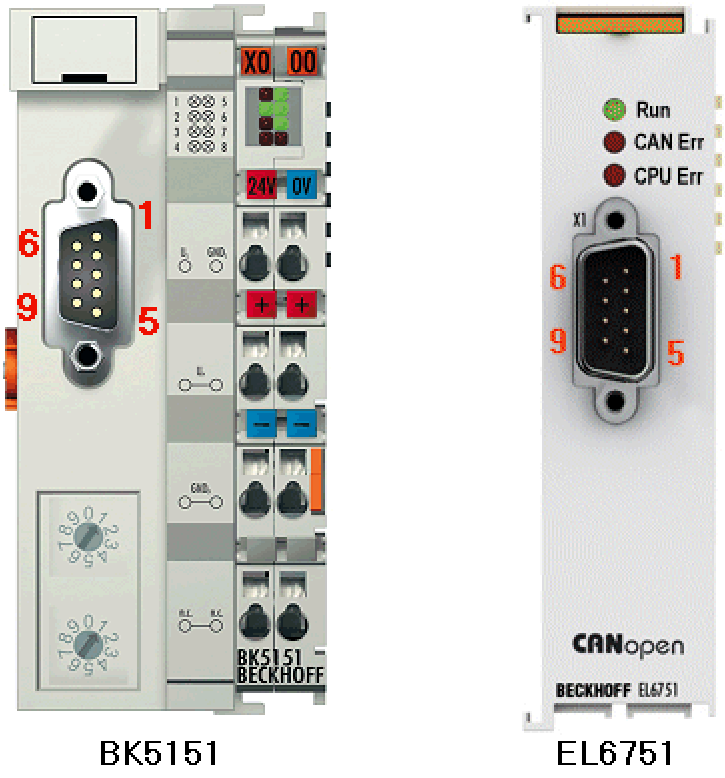
3.4.5.8 BK5151, FC51xx, CX with CAN interface and EL6751: D-sub, 9 pin

The CANbus cable is connected to the FC51x1, FC51x2 CANopen cards and in the case of the EL6751 CANopen master/slave terminal via 9-pin Sub-D sockets with the following pin assignment.

Pin	Assignment
2	CAN low (CAN-)
3	CAN ground (internally connected to pin 6)
6	CAN ground (internally connected to pin 3)
7	CAN high (CAN+)

The unlisted pins are not connected.
The mounting rail contact spring and the plug shield are connected together.

Note: an auxiliary voltage of up to 30 V_{DC} may be connected to pin 9. Some CAN devices use this to supply the transceiver.



BK5151 EL6751

Fig. 24: BK5151, EL6751 pin assignment

FC51x2:

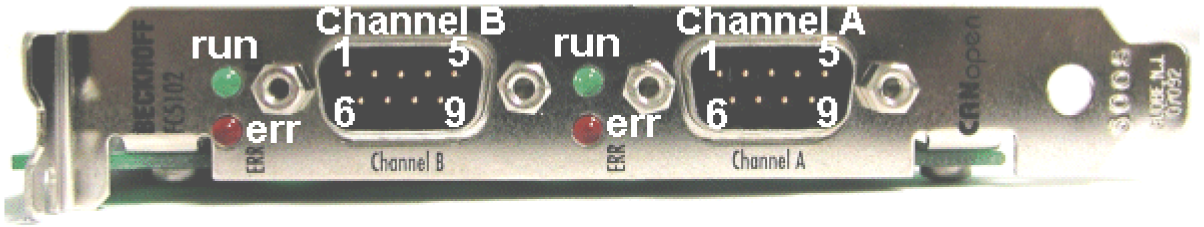


Fig. 25: FC51x2

3.4.5.9 BK51x0/BX5100: 5-pin open style connector

The BK51x0/BX5100 (X510) Bus Couplers have a recessed front surface on the left hand side with a five pin connector.
The supplied CANopen socket can be inserted here.

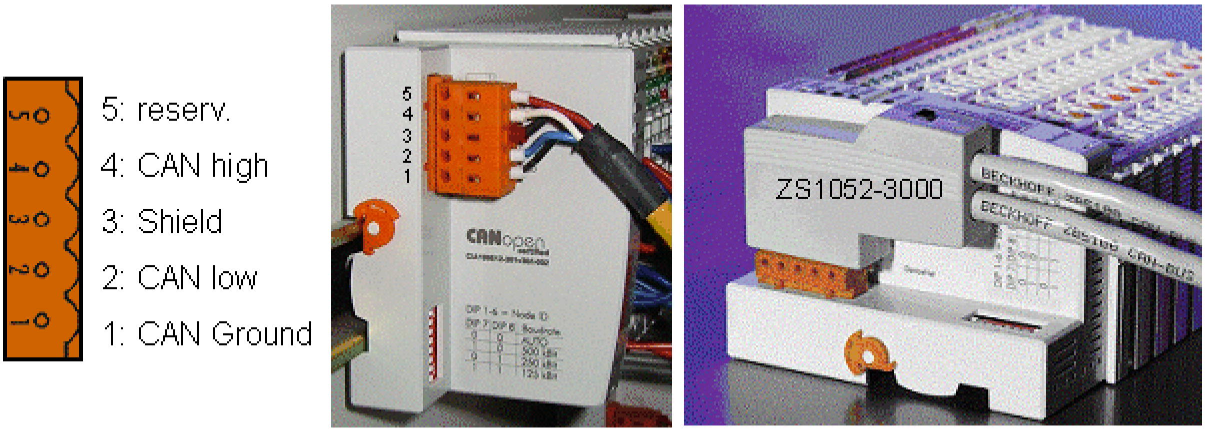


Fig. 26: BK51x0/BX5100 socket assignment

The left figure shows the socket in the BK51x0/BX5100 Bus Coupler. Pin 5 is the connection strip's top most pin. Pin 5 is not used. Pin 4 is the CAN high connection, pin 2 is the CAN low connection, and the screen is connected to pin 3 (which is connected to the mounting rail via an R/C network). CAN-GND can optionally be connected to pin 1. If all the CAN ground pins are connected, this provides a common reference potential for the CAN transceivers in the network. It is recommended that the CAN GND be connected to earth at one location, so that the common CAN reference potential is close to the supply potential. Since the CANopen BK51X0/BX5100 Bus Couplers provide full electrical isolation of the bus connection, it may in appropriate cases be possible to omit wiring up the CAN ground.

ZS1052-3000 Bus Interface Connector

The ZS1052-3000 CAN Interface Connector can be used as an alternative to the supplied connector. This makes the wiring significantly easier. There are separate terminals for incoming and outgoing leads and a large area of the screen is connected via the strain relief. The integrated terminating resistor can be switched externally. When it is switched on, the outgoing bus lead is electrically isolated - this allows rapid wiring fault location and guarantees that no more than two resistors are active in the network.

3.4.5.10 LC5100: Bus connection via spring-loaded terminals

In the low cost LC5100 Coupler, the CAN wires are connected directly to the contact points 1 (CAN-H, marked with C+) and 5 (CAN-L, marked with C-). The screen can optionally be connected to contact points 4 or 8, which are connected to the mounting rail via an R/C network.

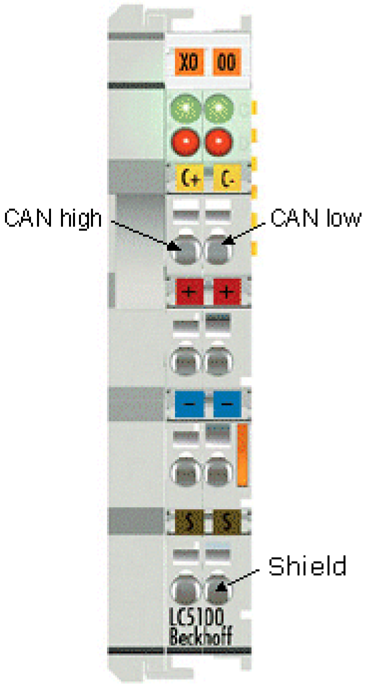


Fig. 27: LC5100

NOTE

Risk of device damage!

On account of the lack of electrical isolation, the CAN driver can be destroyed or damaged due to incorrect cabling. Always carry out the cabling in the switched-off condition.
First connect the power supply and then the CAN. Check the cabling and only then switch on the voltage.

3.4.5.11 Fieldbus Box: M12 CAN socket

The IPxxxx-B510, IL230x-B510 and IL230x-C510 Fieldbus Boxes are connected to the bus using 5-pin M12 plug-in connectors.

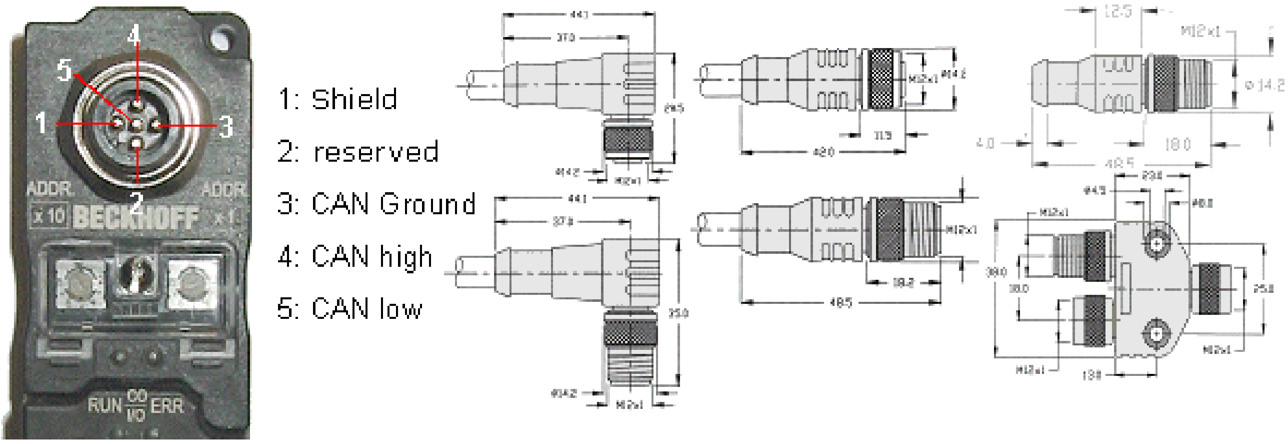


Fig. 28: Pin assignment: M12 plug, fieldbus box

Beckhoff offer plugs for field assembly, passive distributor's, terminating resistors and a wide range of pre-assembled cables for the Fieldbus Box system. Details be found in the catalogue, or under www.beckhoff.de.

4 Parameterization and Commissioning

4.1 Start-up behavior of the Bus Terminal Controller

When the Bus Terminal Controller is switched on it checks its state, configures the K-bus, creates a configuration list based on the connected Bus Terminals and starts its local PLC.
The I/O LEDs flash when the Bus Terminal Controller starts up. If the system is in an error-free state, the I/O LEDs should stop flashing after approx. 2-3 seconds. In the event of a fault the error type determines which LED flashes (see chapter *Diagnostic LEDs*).

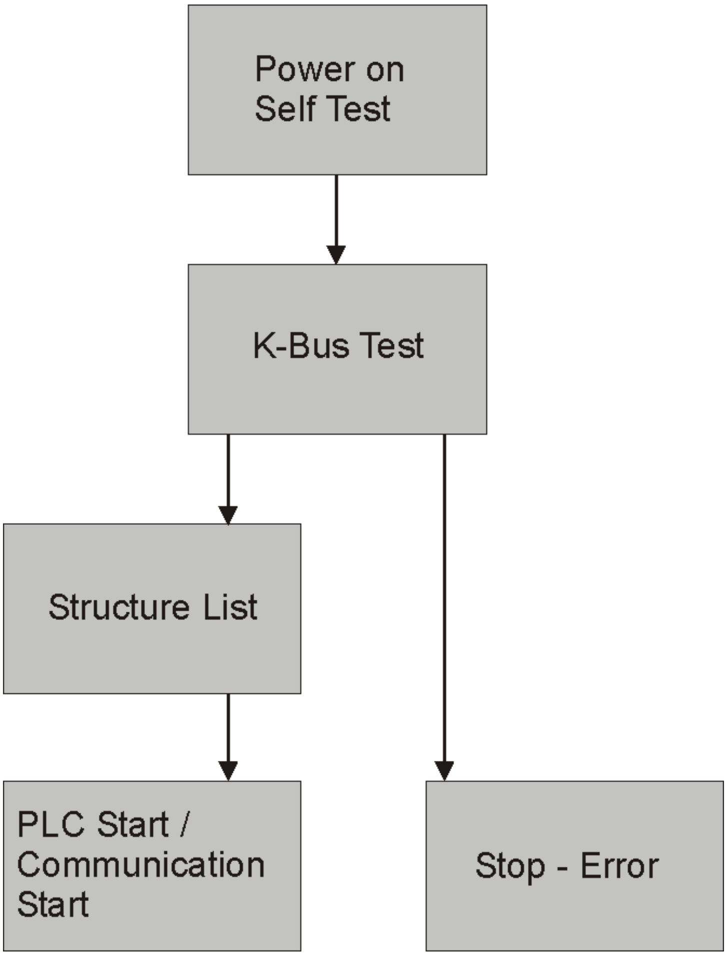


Fig. 29: **Start-up behavior of the Bus Terminal Controller**

4.2 Configuration

4.2.1 Overview

Configuration types

The Bus Terminal controllers of the BCxx50, BCxx20 and BXxx00 series can be configured in two different ways: DEFAULT CONFIG or TwinCAT CONFIG.

DEFAULT-CONFIG

Bus Terminals are mapped in the order they are inserted, i.e. first the complex Bus Terminals followed by the digital Bus Terminals.

The complex Bus Terminals are mapped as follows:

- Word Alignment
- complex representation

⚠ CAUTION

The process image depends on the connected terminals!

The process image changes when a terminal is added or removed!

The data of the fieldbus slaves interface are referred to as PLC variables. The PLC variables have addresses from %QB1000 and %IB1000

The DEFAULT CONFIG (without PLC program) can also be used for writing and testing of the Connected Bus Terminals. To this end, the Bus Terminal Controller must be scanned in the System Manager, and FreeRun mode must be enabled (to use this function, no PLC program may be active on the Bus Terminal Controller).

TWINCAT-CONFIG

In the TwinCAT CONFIG the Bus Terminals and PLC variables can be freely linked as required (TwinCAT System Manager file required). The configuration is transferred to the coupler via the System Manager and ADS.

The following is required for the TwinCAT configuration (TC file):

- Via the fieldbus (PROFIBUS, CANopen, Ethernet)
  PROFIBUS: (BC3150, BX3100)
  - PC with FC310x from version 2.0 and TwinCAT 2.9 build 1000
  - BX3100 with CIF60 or CP5412
  - TwinCAT 2.9 build 946
    (**NOTE:** with PROFIBUS cards from Hilscher only one ADS communication is permitted, i.e. either System Manager or PLC Control)
    CANopen: (BC5150, BX5100)
  - PC with FC510x from version 1.76 TwinCAT build 1030
    DeviceNet: (BC5250, BX5200)
  - on request
    Ethernet: (BC9050, BC9020, BC9120, BX9000)
  - PC with TwinCAT 2.10 build 1322
- Via the serial 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 from firmware B0
  - For BC8150 from TwinCAT 2.10 build 1243

BCxx50 and BXxx00 can be parameterized via the System Manager of the TwinCAT program.

- Variable I/O mapping
- Type-specific PROFIBUS data (BC3150 and BX3100 only)
- RTC (real-time clock) (BX series only)
- SSB (Smart System Bus) (BX series only)
- PLC settings
- K-Bus settings

The configuration can be transferred to the BCxx50 or BXxx00 via fieldbus ADS protocol or serial ADS protocol.

The TwinCAT configuration can be used to link variables, I/Os and data. The following is possible:

- PLC - K-BUS
- PLC fieldbus (e.g. PROFIBUS slave interface to PLC)
- K-bus fieldbus (only for BX controllers)
- Support for TwinSAFE terminals (only BX controllers from firmware 1.17)

In addition, the TwinCAT configuration can be used to parameterize special behavior, for example whether data are preserved or set to "0" in the event of a fieldbus error.
The real-time clock can be set via a tab in the system manager.

Work steps

1. Setting the fieldbus address
2. Open the System Manager and create a TC file
3. Configure fieldbus data in the TC file
4. Save the TC file
5. Opening a new system manager, creating a PC file and reading in saved TX file
6. Creating a link to a PLC task
7. Saving the configuration
8. Starting the TwinCAT system
9. Open the TC file in the System Manager, complete the configuration and transfer it to the BCxx50, BCxx20 or BXxx00
10. Transfer the program to BCxx50, BCxx20 or BXxx00
11. Creating a boot project

4.2.2 Creating a TwinCAT configuration

In order to configure a Bus Terminal Controller of the BCxx50, BCxx20, BXxx00 or BC9191 series, create a BX file in the System Manager. To simplify matters, files for the basic units have already been prepared. Open the corresponding Bus Terminal Controller with *New from Template*.

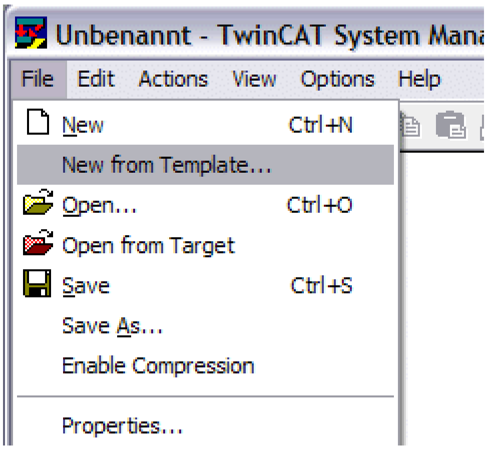
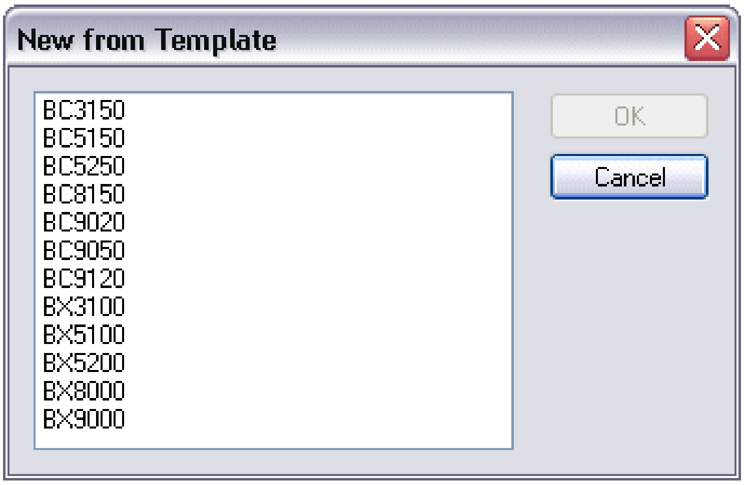


Fig. 30: Creating a TwinCAT configuration

Select the corresponding Bus Terminal Controller.



BC3150
BC5150
BC5250
BC8150
BC9020
BC9050
BC9120
BX3100
BX5100
BX5200
BX8000
BX9000

Fig. 31: Selecting the Bus Terminal Controller

All Bus Terminal Controller components are now available:

- Fieldbus interface
- K-bus interface [▶ 50]
- PLC program [▶ 52]
- SSB [▶ 55] (only Bus Terminal Controllers of the BX series)

Please refer to the relevant chapter for device configuration.

4.2.3 Downloading a TwinCAT configuration

The TwinCAT configuration is loaded into the Bus Terminal Controller via ADS protocol.

Serial ADS protocol

(all Bus Terminal Controllers of the BXxx00 and BCxx50 series)

Enter the serial ADS connection, as described in the chapter Serial ADS [▶ 41].

ADS protocol via the fieldbus

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

A prerequisite is that TwinCAT operates as master and is engaged in data exchange, i.e. the physical and fieldbus configuration must be complete, and data exchange must take place between the master (e.g. fieldbus master card) and the Bus Terminal Controller.

Choose Target System

Select the Bus Terminal Controller onto which the configuration is to be loaded. Use the function key F8 to open the dialog for downloading your file to the corresponding device.

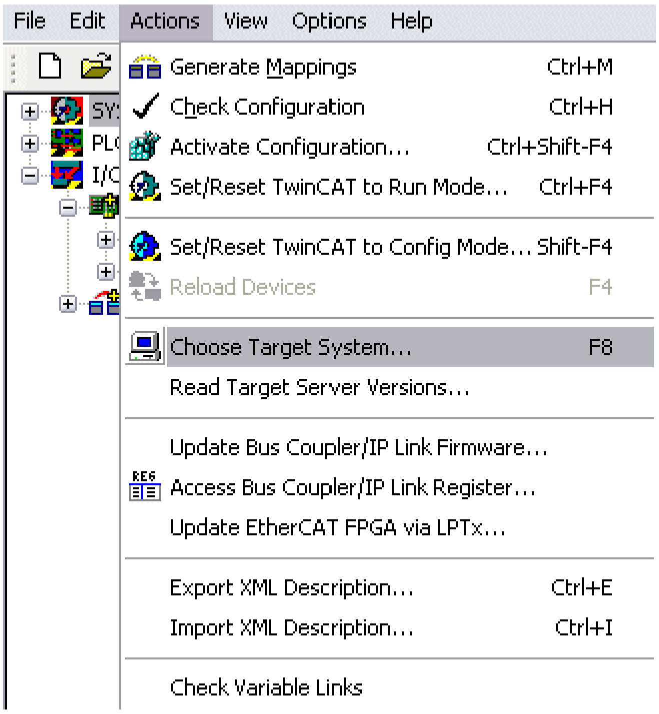

File
Edit
Actions
View
Options
Help
SY
PL
I/C
Generate Mappings Ctrl+M
Check Configuration Ctrl+H
Activate Configuration... Ctrl+Shift-F4
Set/Reset TwinCAT to Run Mode... Ctrl+F4
Set/Reset TwinCAT to Config Mode... Shift-F4
Reload Devices F4
Choose Target System... F8
Read Target Server Versions...
Update Bus Coupler/IP Link Firmware...
Access Bus Coupler/IP Link Register...
Update EtherCAT FPGA via LPTx...
Export XML Description... Ctrl+E
Import XML Description... Ctrl+I
Check Variable Links

Fig. 32: Downloading a TwinCAT configuration

Select the corresponding Bus Terminal Controller.

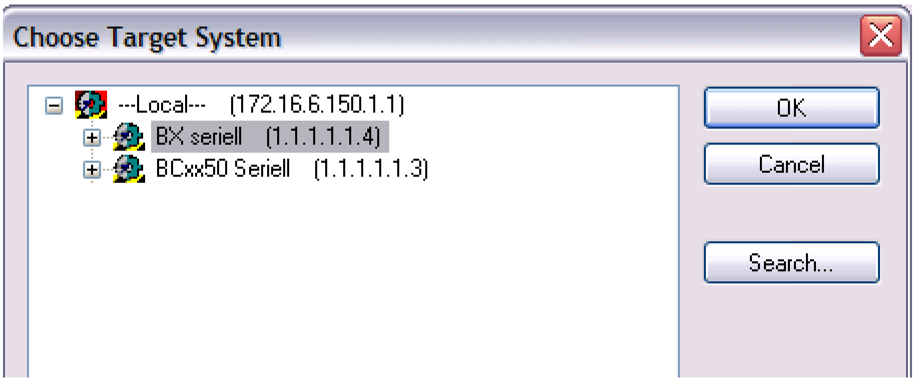


Fig. 33: Selecting the Bus Terminal Controller

The state of the Bus Terminal Controller is shown at the bottom right of the System Manager.

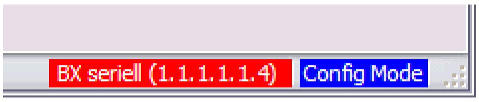


Fig. 34: State of the Bus Terminal Controller

In *Config mode / FreeRun* the configuration can now be downloaded to Bus Terminal Controller. If the Bus Terminal Controller is in *Stop mode*, ADS communication is not yet activated. In this case, it is not possible to download the configuration.

To activate the TwinCAT configuration select Ctrl+Shift+F4 or *Activate Configuration*.

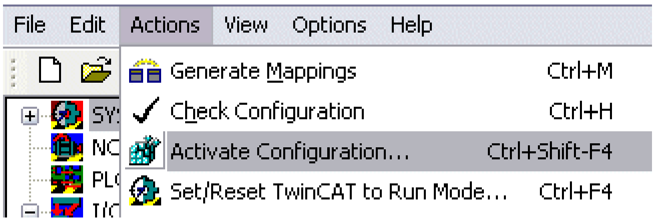

File
Edit
Actions
View
Options
Help
SY
NC
PL
I/C
Generate Mappings Ctrl+M
Check Configuration Ctrl+H
Activate Configuration... Ctrl+Shift-F4
Set/Reset TwinCAT to Run Mode... Ctrl+F4

Fig. 35: Activating the TwinCAT configuration

The current configuration is loaded onto the Bus Terminal Controller. The display will show *Store Config*, and the BUS and I/O LED will flash. Once the configuration is successfully loaded onto Bus Terminal Controller, *TwinCAT Config* should appear in the display of a BXxx00. The corresponding program can now be transferred to the Bus Terminal Controller (program-download via the fieldbus).

4.2.4 Uploading a TwinCAT configuration

The TwinCAT configuration is loaded into the Bus Terminal Controller via ADS protocol.

Serial ADS protocol

(all Bus Terminal Controllers of the BCxx50, BCxx20 and BXxx00 series)

Enter the serial ADS connection, as described in the chapter Serial ADS [▶ 41].

ADS protocol via the fieldbus

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

A prerequisite is that TwinCAT operates as master and is engaged in data exchange, i.e. the physical and fieldbus configuration must be complete, and data exchange must take place between the master (e.g. fieldbus card) and the Bus Terminal Controller.

Choose Target System

Select the Bus Terminal Controller onto which the configuration is to be loaded. Use the function key [F8] to open the dialog for downloading your file to the corresponding device.

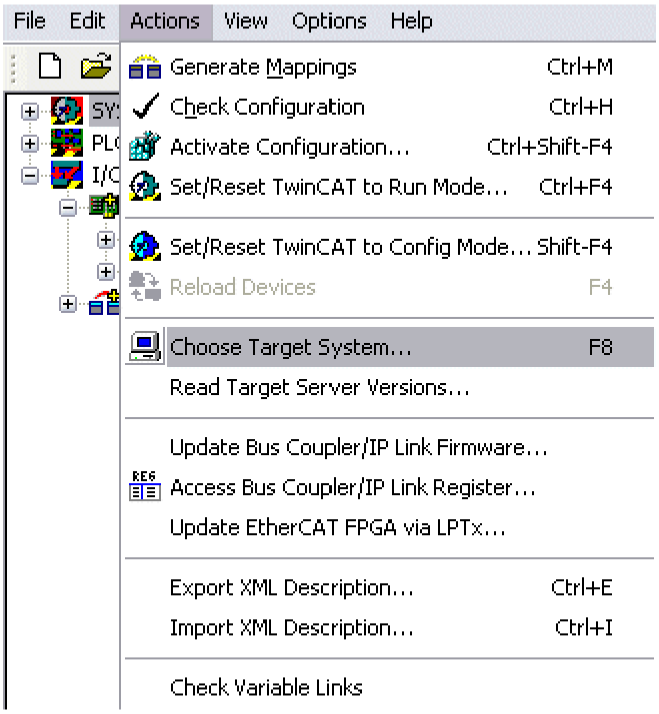

File
Edit
Actions
View
Options
Help
SYS
PLC
I/O
Generate Mappings Ctrl+M
Check Configuration Ctrl+H
Activate Configuration... Ctrl+Shift-F4
Set/Reset TwinCAT to Run Mode... Ctrl+F4
Set/Reset TwinCAT to Config Mode... Shift-F4
Reload Devices F4
Choose Target System... F8
Read Target Server Versions...
Update Bus Coupler/IP Link Firmware...
Access Bus Coupler/IP Link Register...
Update EtherCAT FPGA via LPTx...
Export XML Description... Ctrl+E
Import XML Description... Ctrl+I
Check Variable Links

Fig. 36: Choose Target System

Select the corresponding Bus Terminal Controller.

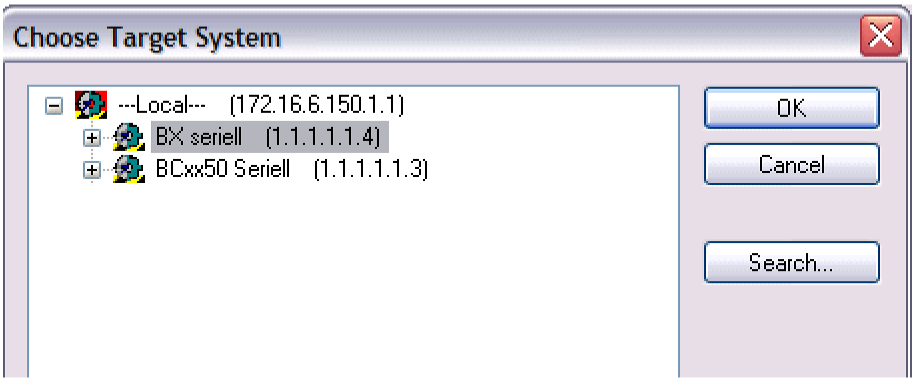


Fig. 37: Selecting the Bus Terminal Controller

The state of the Bus Terminal Controller is shown at the bottom right of the System Manager.

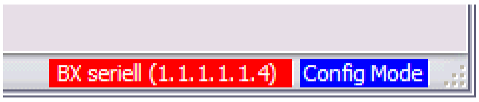


Fig. 38: State of the Bus Terminal Controller

Click on the red folder. The TwinCAT configuration will now be uploaded.

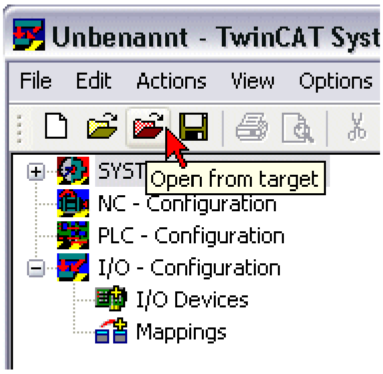


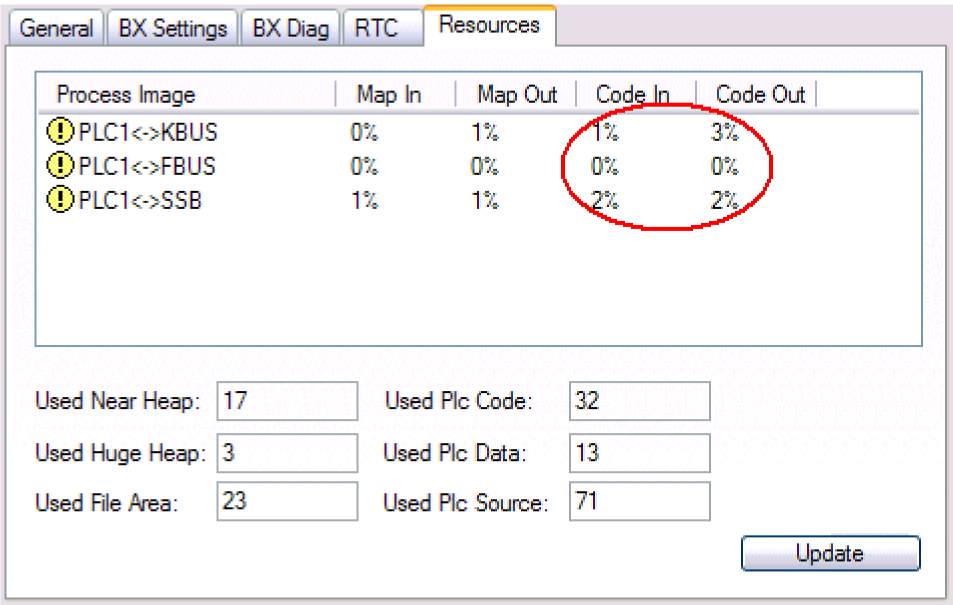
Fig. 39: Uploading the TwinCAT configuration

4.2.5 Resources in the Bus Terminal Controller

The memory resources assigned in the Bus Terminal Controller are shown in the System Manager in the *Resources* tab of the Bus Terminal Controller.

Mapping code

The mapping code is required for calculating the TwinCAT configuration (see Figure *Memory for the code mapping*). The percentages are added here. In the example from Fig. *Memory for code mapping*, 8% of the memory is allocated to the mapping calculation.



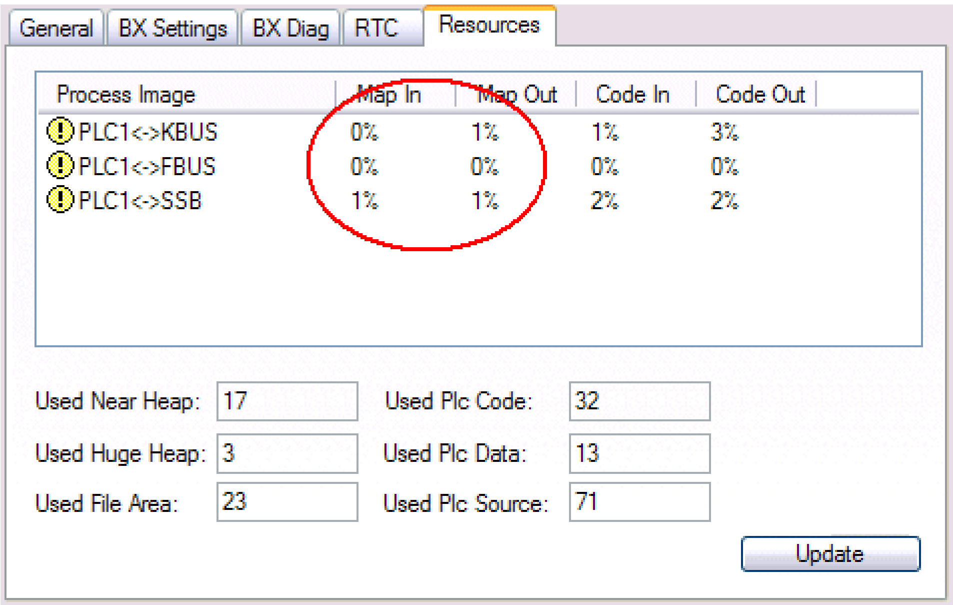
Process Image	Map In	Map Out	Code In	Code Out
PLC1<->KBUS	0%	1%	1%	3%
PLC1<->FBUS	0%	0%	0%	0%
PLC1<->SSB	1%	1%	2%	2%

Used Near Heap: 17
Used Huge Heap: 3
Used File Area: 23
Used Plc Code: 32
Used Plc Data: 13
Used Plc Source: 71

Fig. 40: Memory for code mapping

Data memory mapping

Data memory for mapping. The values are to be considered individually, i.e. each value can be up to 100%.



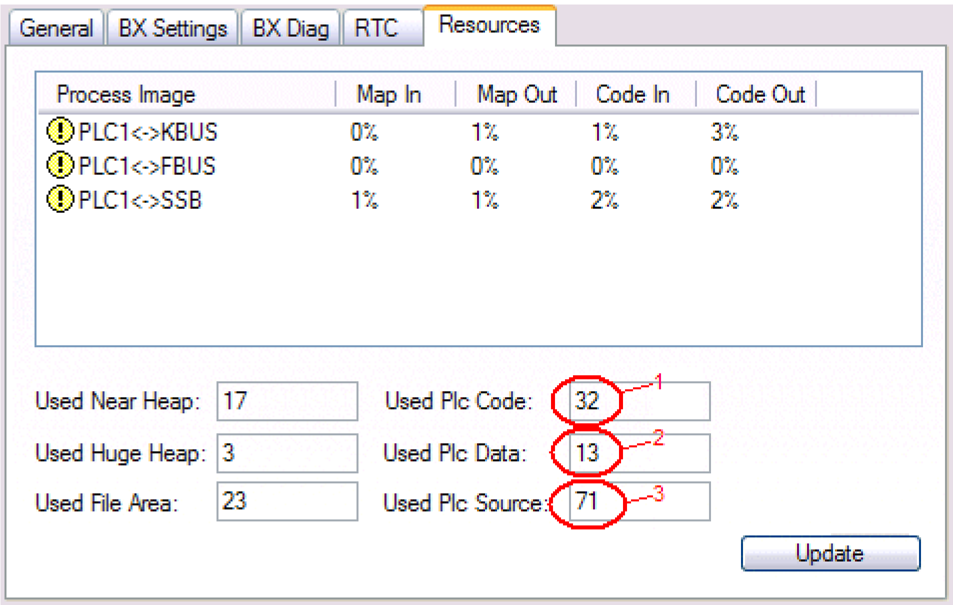
Process Image	Map In	Map Out	Code In	Code Out
PLC1<->KBUS	0%	1%	1%	3%
PLC1<->FBUS	0%	0%	0%	0%
PLC1<->SSB	1%	1%	2%	2%

Used Near Heap: 17
Used Huge Heap: 3
Used File Area: 23
Used Plc Code: 32
Used Plc Data: 13
Used Plc Source: 71

Fig. 41: Data memory mapping

Used code and data memory

Fig. *Code and data memory* (1) "Used PLC code" in %.
Fig. *Code and data memory* (2) "Used PLC data" in %.
Fig. *Code and data memory* (3) "Used PLC source" in %.



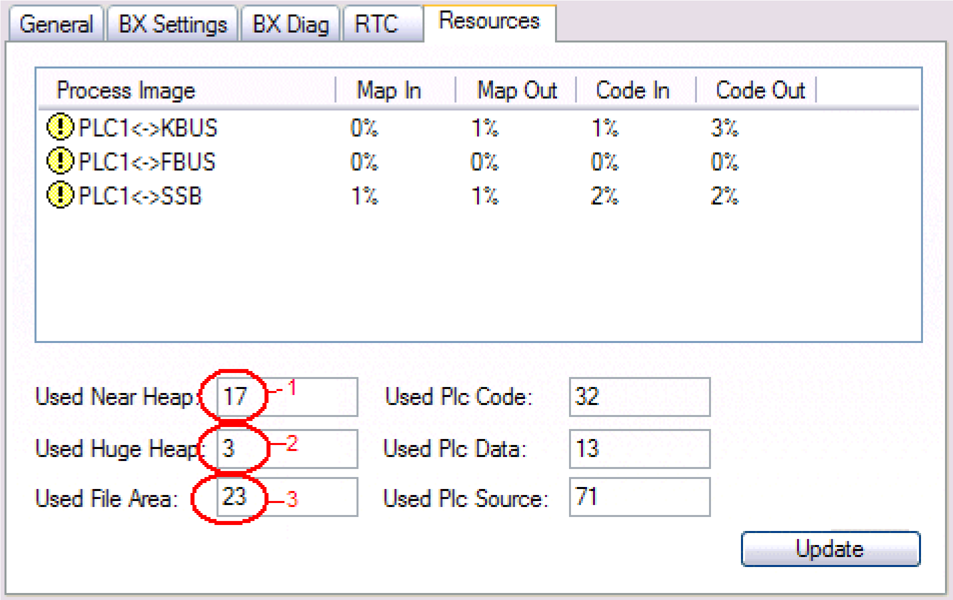
Process Image	Map In	Map Out	Code In	Code Out
PLC1<->KBUS	0%	1%	1%	3%
PLC1<->FBUS	0%	0%	0%	0%
PLC1<->SSB	1%	1%	2%	2%

Used Near Heap: 17
Used Huge Heap: 3
Used File Area: 23
Used Plc Code: 32
Used Plc Data: 13
Used Plc Source: 71

Fig. 42: Code and data memory

Other memory

Fig. *Other Memory* (1) "Used Near Heap" is required for the COM interface and SSB. % values.
Fig. *Other Memory* (2) "Used Huge Heap" is required for the ADS communication. % values. This value should be less than 30 %.
Fig. *Other Memory* (3) "Used File Area" is required for the TwinCAT configuration, the TSM file and the 16 kbyte flash access. % values.



Process Image	Map In	Map Out	Code In	Code Out
PLC1<->KBUS	0%	1%	1%	3%
PLC1<->FBUS	0%	0%	0%	0%
PLC1<->SSB	1%	1%	2%	2%

Used Near Heap: 17
Used Huge Heap: 3
Used File Area: 23
Used Plc Code: 32
Used Plc Data: 13
Used Plc Source: 71

Fig. 43: Other memory

4.2.6 ADS connection via serial interface

(from firmware version 1.xx or 0.99x, Bus Terminal Controllers of the BX series and for all BCxx50)

From TwinCAT 2.9 build 1020 (TwnCAT level PLC, NC or NCI)

Use only a serial connection

To ensure trouble-free operation of the ADS link via the serial interface, only a serial connection to the BX controller is allowed.
After successful configuration via the System Manager, close the System Manager before starting programming.

AMS Net ID in delivery state (default)

For BX9000

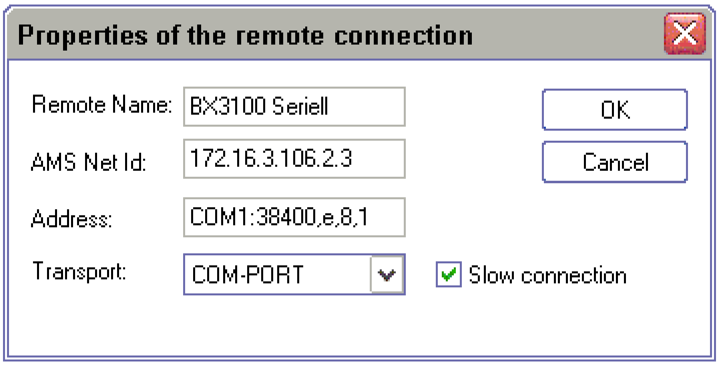
The default AMS Net ID is 172.16.21.20.1.1. If the IP address of the BX9000 is changed, the AMS Net ID of the BX9000 also changes. There is a menu option for displaying the current AMS Net ID. Example: If you change the IP address to 10.2.3.7, the AMS Net ID changes to 10.2.3.7.1.1.

For BC9050, BC9020, BC9120

The default AMS Net ID is 172.16.xxx.[DIP switch].1.1. If the IP address of the BX9xxx is changed, the AMS Net ID of the BX9xxx also changes.
Example: If you change the IP address to 10.2.3.7, the AMS Net ID changes to 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

Initializing the ADS connection

Enter the Bus Terminal Controller in the remote connection under TwinCAT. Click on the TwinCAT icon and open the features menu. The following settings can be made under the >AMS Remote< tab.



Properties of the remote connection

Remote Name: BX3100 Seriell
AMS Net Id: 172.16.3.106.2.3
Address: COM1:38400,e,8,1
Transport: COM-PORT
☑ Slow connection

Fig. 44: Properties of the remote connection

Remote Name: Any
AMS-Net-ID: 1.1.1.1.1.1 (Default)
Address: COM Port: Baud rate, parity, data bits, stop bits
Transport: Select "COM port"

When the Bus Terminal Controller is switched on, the default AMS Net ID is always "1.1.1.1.1.1" (except all Ethernet Controllers).
The AMS Net ID can be changed as required. Please note that the new AMS Net ID cannot be changed again in this way.

If you need to change the new AMS Net ID again, you have to restart the Bus Terminal Controller, so that the AMS Net ID is reset to the default AMS Net ID, "1.1.1.1.1.1".
You can now change the AMS Net ID again.

Strings can only be entered at the second call

No strings can be entered under address when the dialog is first called (see above). Enter the name, AMS Net ID and transport type and close the dialog. With the second call you can enter your COM port.

The communication starts when TwinCAT is in Config mode (TwinCAT icon is blue) or RUN mode (TwinCAT icon is green). The COM interface remains open until a TwinCAT stop occurs (TwinCAT icon is red). It is then available again for other programs. No error message is issued if the COM interface is used by another program during a TwinCAT restart (e.g. by the KS2000 configuration software).

AMS Net ID after ADS connection via the fieldbus

If you have addressed the Bus Terminal Controller with an ADS connection via the fieldbus before the serial ADS was used, the AMS Net ID was automatically changed by the System Manager. In this case a new serial ADS connection is only possible, if the AMS Net ID is adjusted.

BX series: reading the AMS Net ID

The current AMS Net ID can be read from the menu via the display of BX series Bus Terminal Controller.

AMS 1.1.1.1.1.1	AMS Net ID

4.2.7 DeviceNet slave interface

4.2.7.1 DeviceNet slave interface EDS file

BX5200

EDS file BX5200.eds (for controllers other than TwinCAT) (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3602701067.zip)

EDS file BX5200S.eds (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3602705419.zip)

For TwinCAT both EDS files must be copied into the directory *TwinCAT\IO\DeviceNet* .

Image for BX5200.ico (https://infosys.beckhoff.com/content/1033/bx5200/Resources/ico/4007122315.ico)

BC5250

EDS file BC5250.eds (for controllers other than TwinCAT) (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3602707595.zip)

EDS file BC5250S.eds (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/4007126667.zip)

For TwinCAT both EDS files must be copied into the directory *TwinCAT\IO\DeviceNet* .

Image for BC5250.ico (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3602707595.zip)

4.2.7.2 DeviceNet slave interface

There are two types of configuration. In the default configuration (delivery state) the DeviceNet data of the DeviceNet Slave interface map from the address 1000 of the BX5200/BC5250 and the first 8 bytes are activated. The length of the DeviceNet data and the characteristics of the local controller can be edited via the function blocks.
In the TwinCAT configuration, any required configuration can be created via the System Manager and variables can be connected in any required combination via the DeviceNet slave interface.

Default Configuration

In this configuration, the DeviceNet data are mapped as follows:

DeviceNet data

Number	Read/Write	BX5200/BC5250 process image
8 bytes (Polling Mode)	Rx/Tx	%IB1000...%IB1007/QB1000...%QB1007

If it is required to send or receive more/less than 8 bytes of data, the default setting must be changed by means of the function block (see DeviceNet-specific function blocks).

Characteristics of DeviceNet communication

The following characteristics can be set. Mixed operation of the DeviceNet communication types is not possible.

Polling

The master module ("scanner") sends the output data cyclically to the assigned devices and receives the input data in a response telegram.

Change-of-State

Telegrams are sent as soon as their contents have changed.

Cyclic

The modules send the data automatically after a cycle time has elapsed.

Strobed

The scanner requests the input data using a broadcast telegram to all the devices.

4.2.7.3 TwinCAT configuration

The TwinCAT configuration enables free, address-independent mapping of the PLC data to the CAN slave interface.

TwinCAT configuration

For the configuration you need the TwinCAT System Manager and an ADS connection to the BX/BCxx50.

The ADS connection can be established via the serial interface (see serial ADS).

4.2.7.4 Address

The address of the DeviceNet Slave interface is set via two rotary selection switches. The default setting is 11. All addresses from 0 to 63 are permitted. Each address may only occur once in the network. The address is edited when the BX Controller is switched off. The switches can be set to the required position using a screwdriver. Ensure that the switches engage correctly. The lower switch is the 10-multiplier, the upper switch is the 1-multiplier. The changed address is active as soon as the BX Controller is switched on again.

Example

You want to set address 34.
- lower rotary selection switch S311: 3
- upper rotary selection switch S310: 4

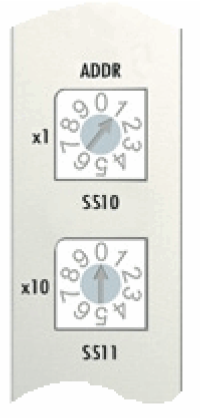


Fig. 45: Setting the node ID

4.2.7.5 Baud rate

Auto-Baud-Rate

In order for automatic baud rate detection to function, it is necessary for a number of valid telegrams to be present on the bus at the desired baud rate. The RUN and CAN ERR LEDs blink in rapid alternation while the baud rate search is in progress. Once a baud rate was detected and accepted, the Bus Terminal Controller continues with the initialization.

A software reset does not lead to a new baud rate search. The previously active baud rate is maintained.

4.2.7.6 Master parameterization

Configuration at a Rockwell/Allen Bradley controller

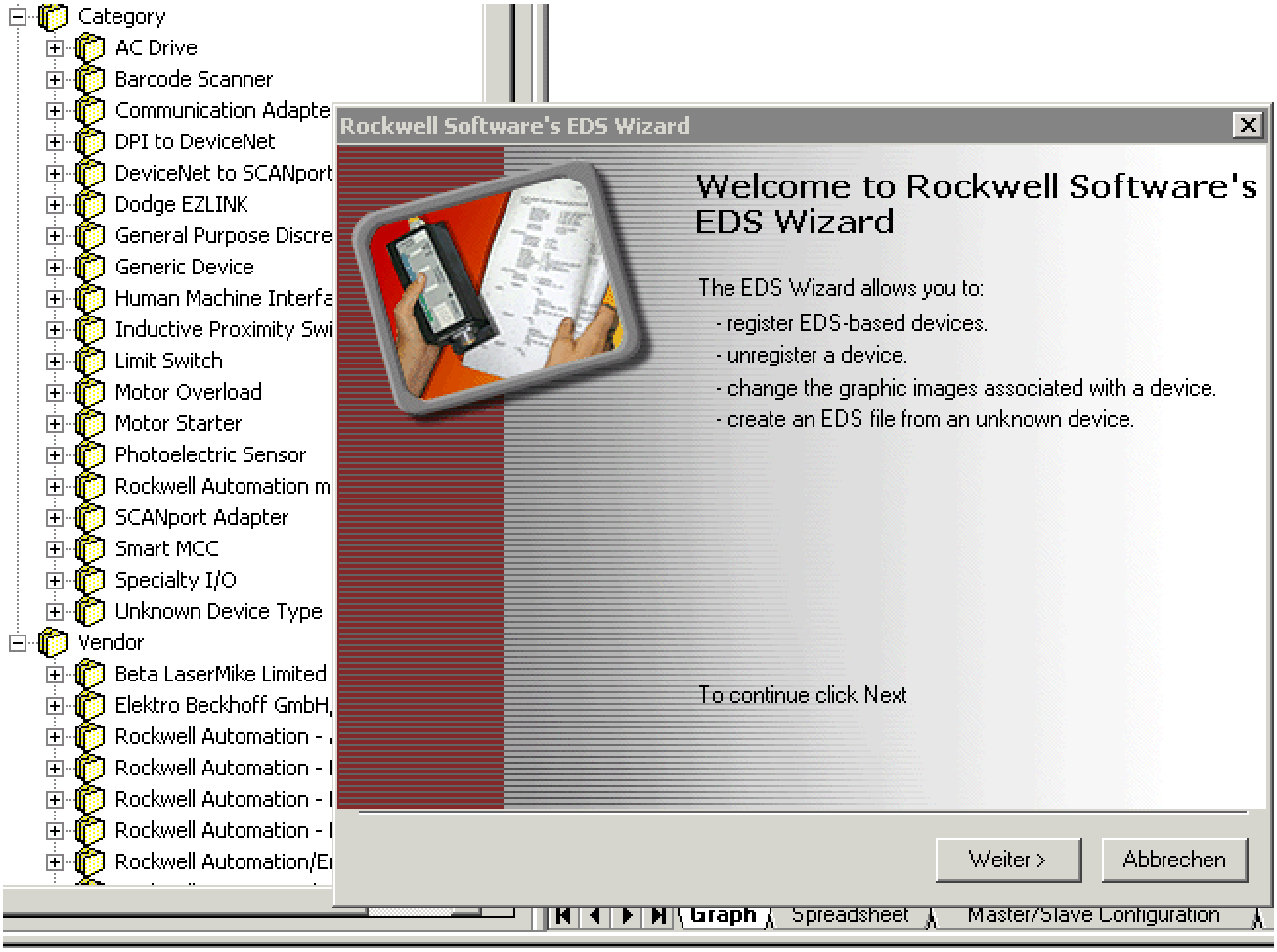
The BX5200 and BC5250 have two configuration types. In the default configuration, the coupler is supplied and has the following DeviceNet characteristics:

- AutoBaud
- 8 bytes inputs DN (from address %IB1000) - these are then the outputs on the master
- 8 bytes outputs DN (from address %QB1000) - these are then the inputs on the master

If it is necessary to change the preset data, a special library is required.

In the TwinCAT configuration, it is possible with the aid of the System Manager to set the DN Slave-specific characteristics. The library is not to be used for the default configuration.

The EDS file and the icon file are always integrated in the RSNetworx with the EDS wizard:



Rockwell Software's EDS Wizard

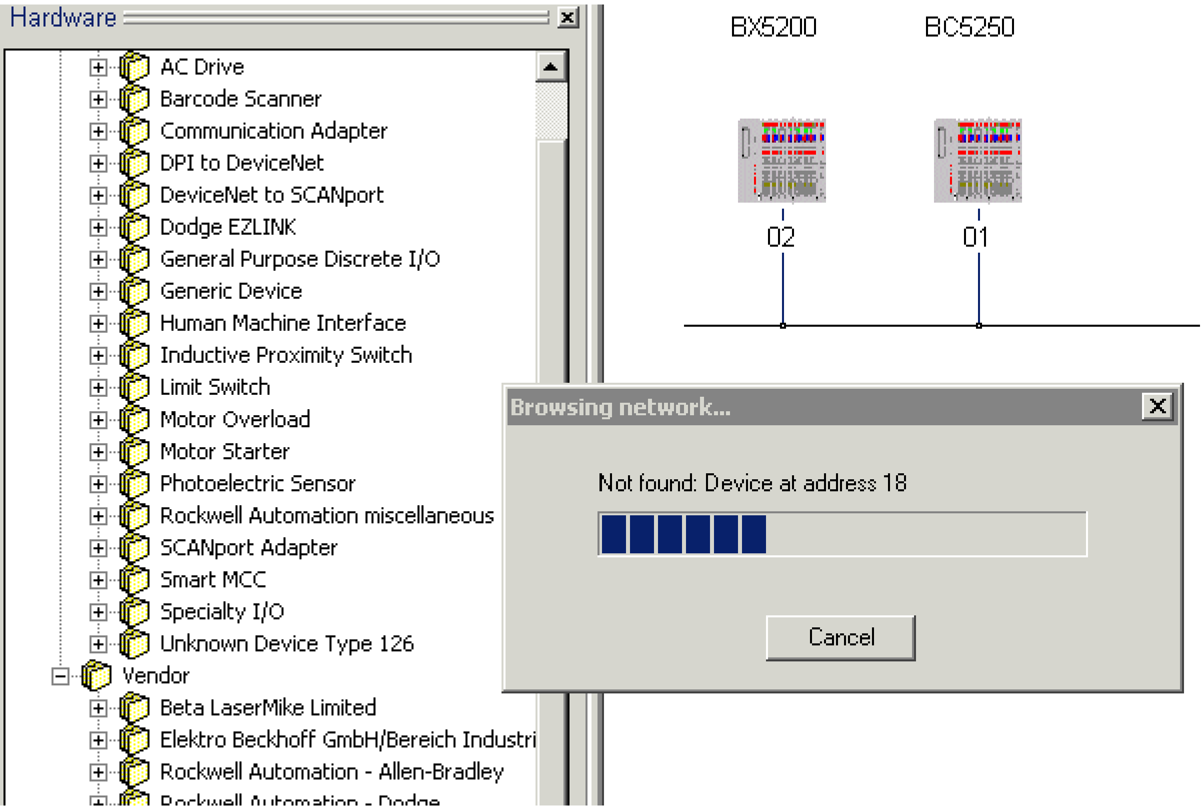
Welcome to Rockwell Software's EDS Wizard

The EDS Wizard allows you to:
- register EDS-based devices.
- unregister a device.
- change the graphic images associated with a device.
- create an EDS file from an unknown device.

To continue click Next

Fig. 46: EDS Wizard

Finally the DeviceNet network is searched with the Network-Browse for the nodes within the network. If the BC/BX are correctly connected to the network with the correct baud rate, they are recognized and correspondingly displayed:

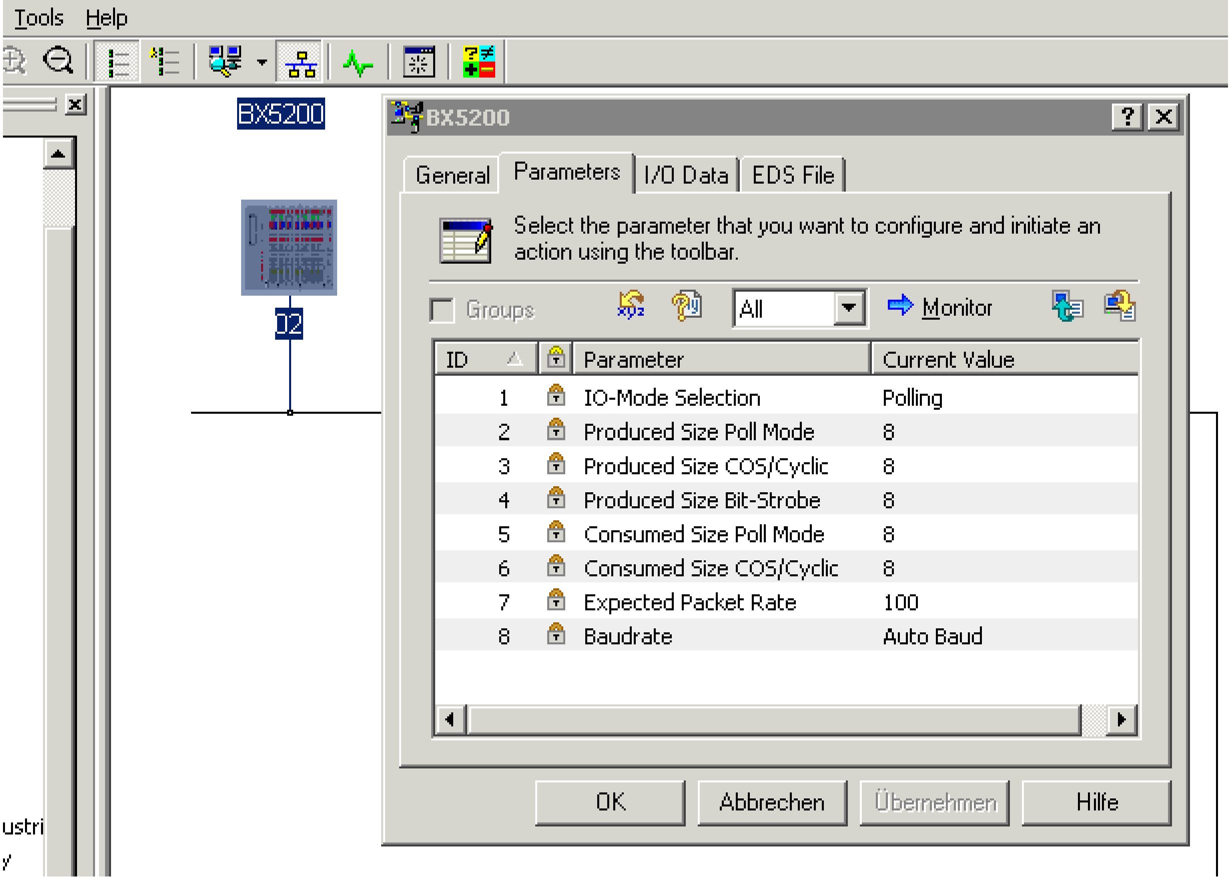


Not found: Device at address 18

Fig. 47: Network Browse

Double-clicking the icon opens the configuration dialog for the units.

Below Parameter appear the I/O data lengths that are currently set at the BC/BX.



Select the parameter that you want to configure and initiate an action using the toolbar.

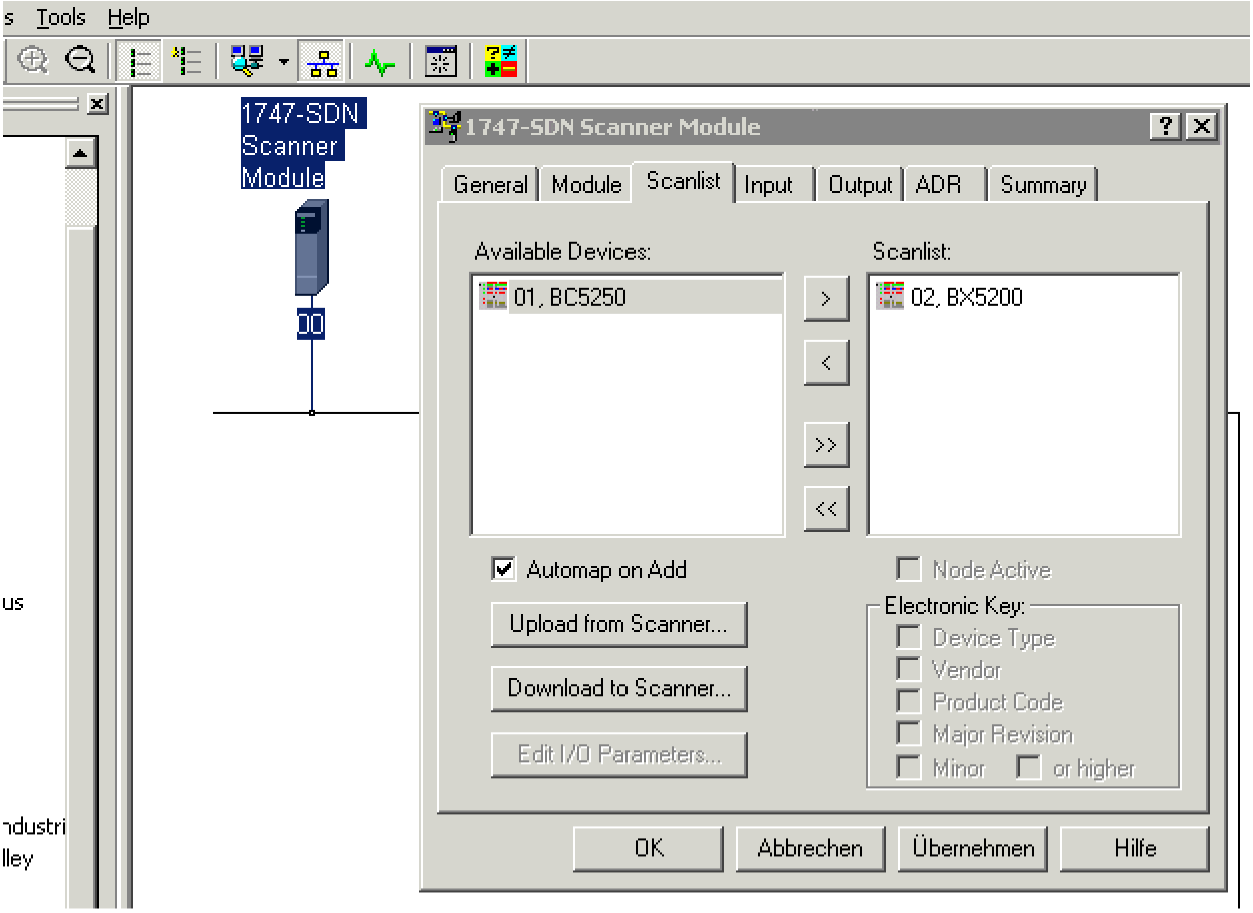
ID		Parameter	Current Value
1		IO-Mode Selection	Polling
2		Produced Size Poll Mode	8
3		Produced Size COS/Cyclic	8
4		Produced Size Bit-Strobe	8
5		Consumed Size Poll Mode	8
6		Consumed Size COS/Cyclic	8
7		Expected Packet Rate	100
8		Baudrate	Auto Baud

Fig. 48: I/O data lengths set on the BC/BX

The produced and consumed data should always be considered from the point of view of the network, i.e. BC/BX produce data for DeviceNet and consume data from DeviceNet.

- The data produced are thus the input data of the master, i.e. data that are sent from the BC/BX to the master: BC/BX Produced Size = Input Size Scanner
- The data consumed are the output data of the master, i.e. data that are sent from the master to the slave: BC/BX Consumed Size = Output Size Scanner

A double-click on the Scanner/DeviceNet Master icon opens the configuration dialog. The recognized modules appear in the *Available Devices* list and must now be added to the Scanlist.



Available Devices:
01, BC5250

Scanlist:
02, BX5200

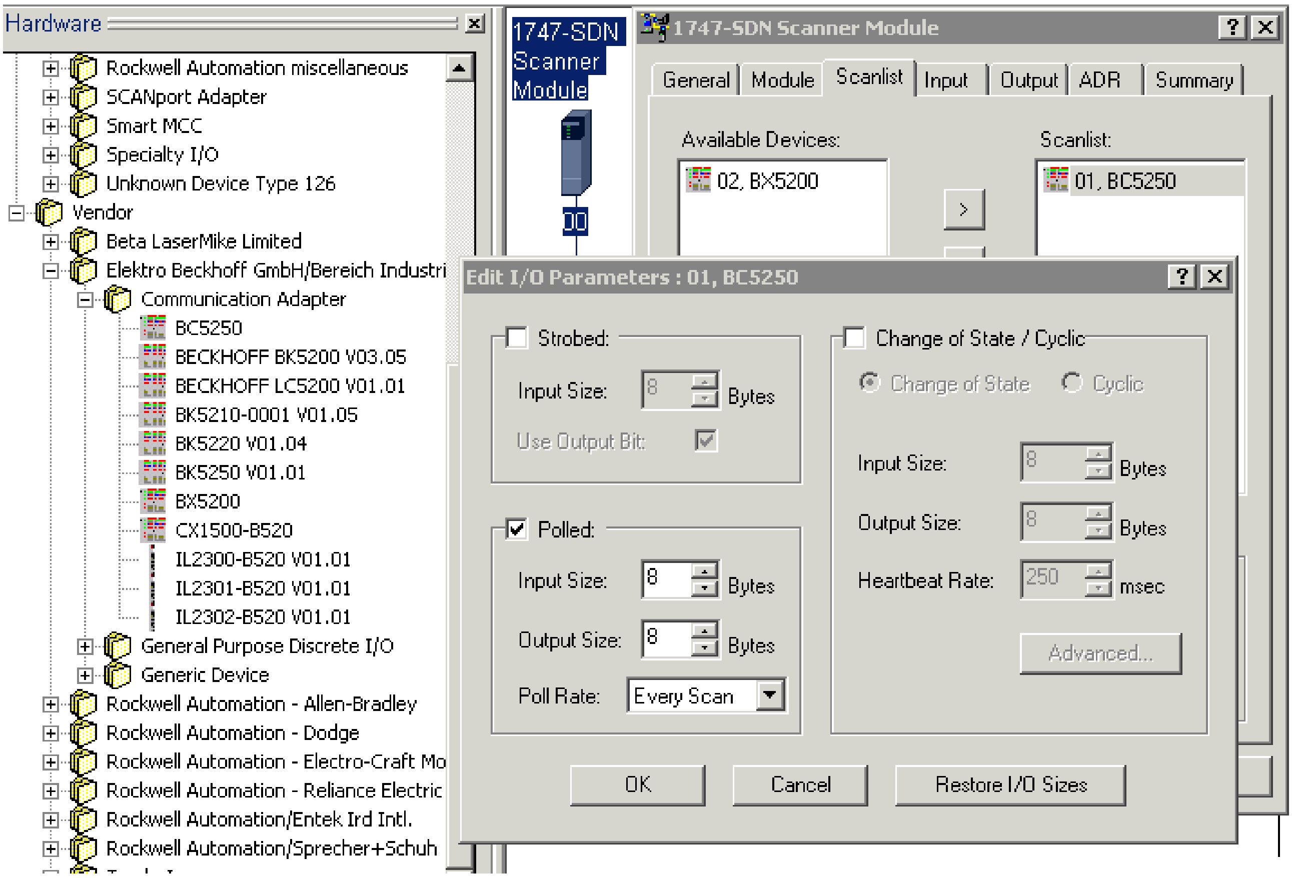
☑ Automap on Add

□ Node Active

Electronic Key:
□ Device Type
□ Vendor
□ Product Code
□ Major Revision
□ Minor □ or higher

Fig. 49: Adoption of the detected modules into the scan list

A double-click on the icon of the device to be configured opens the dialog for setting the I/O data lengths



Edit I/O Parameters : 01, BC5250

□ Strobed:
Input Size: 8 Bytes
Use Output Bit: ☑

☑ Polled:
Input Size: 8 Bytes
Output Size: 8 Bytes
Poll Rate: Every Scan

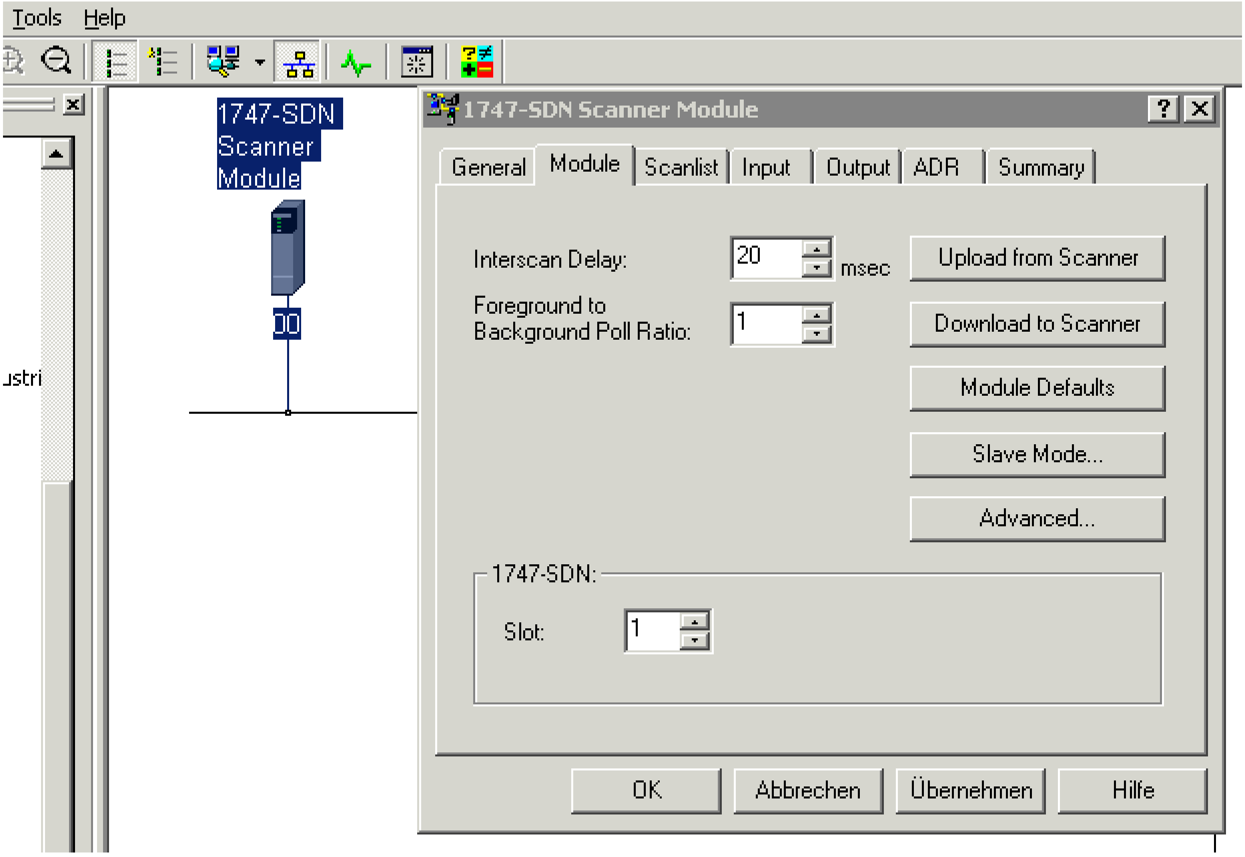
□ Change of State / Cyclic
● Change of State ○ Cyclic
Input Size: 8 Bytes
Output Size: 8 Bytes
Heartbeat Rate: 250 msec

Fig. 50: Setting of the I/O data lengths

- BC/BX Produced-Size = Input-Size
- BC/BX Consumed-Size = Output-Size

Furthermore, it must be ensured that the I/O operation mode selected here corresponds to the I/O mode selection.

The Interscan Delay, that is the time between each I/O cycle, must be selected according to the requirements of the BC/BX with respect to the size of the PLC program and the I/O data lengths.


1747-SDN Scanner Module

Interscan Delay: 20 msec

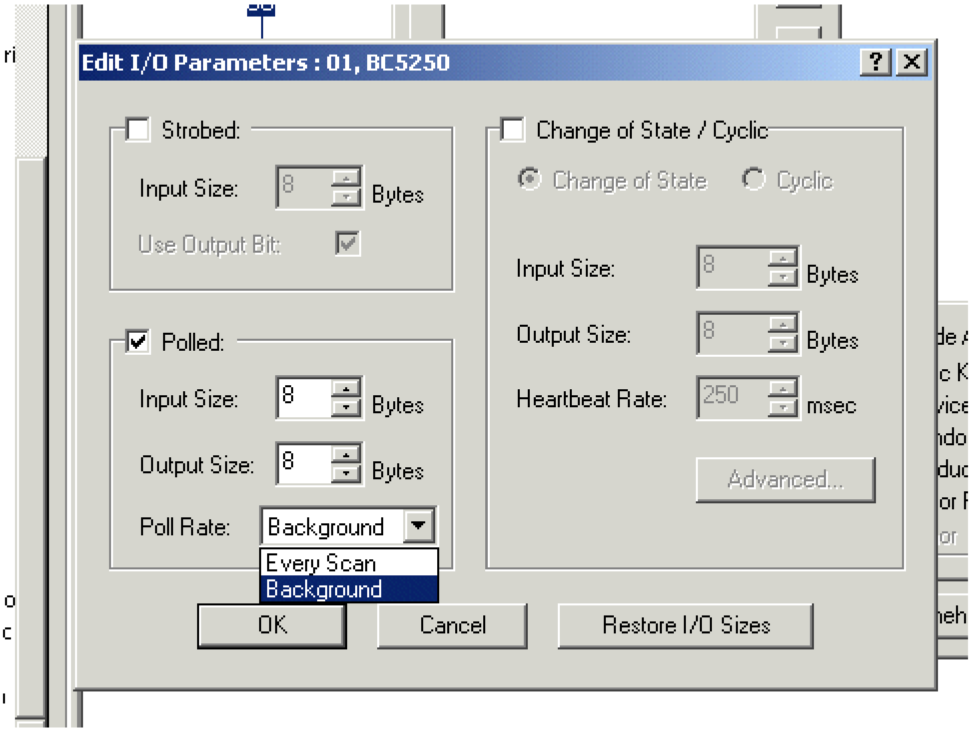
Foreground to Background Poll Ratio: 1

1747-SDN:

Slot: 1

Fig. 51: Selection of the interscan delay

If there are components in the network that require a fast I/O cycle, the I/O cycle can be set to Background under Poll Rate in the I/O parameter settings. More detailed information about this can be obtained from the scanner manual.


Edit I/O Parameters : 01, BC5250

□ Strobed:

Input Size: 8 Bytes

Use Output Bit: ☑

☑ Polled:

Input Size: 8 Bytes

Output Size: 8 Bytes

Poll Rate: Background

□ Change of State / Cyclic

● Change of State ○ Cyclic

Input Size: 8 Bytes

Output Size: 8 Bytes

Heartbeat Rate: 250 msec

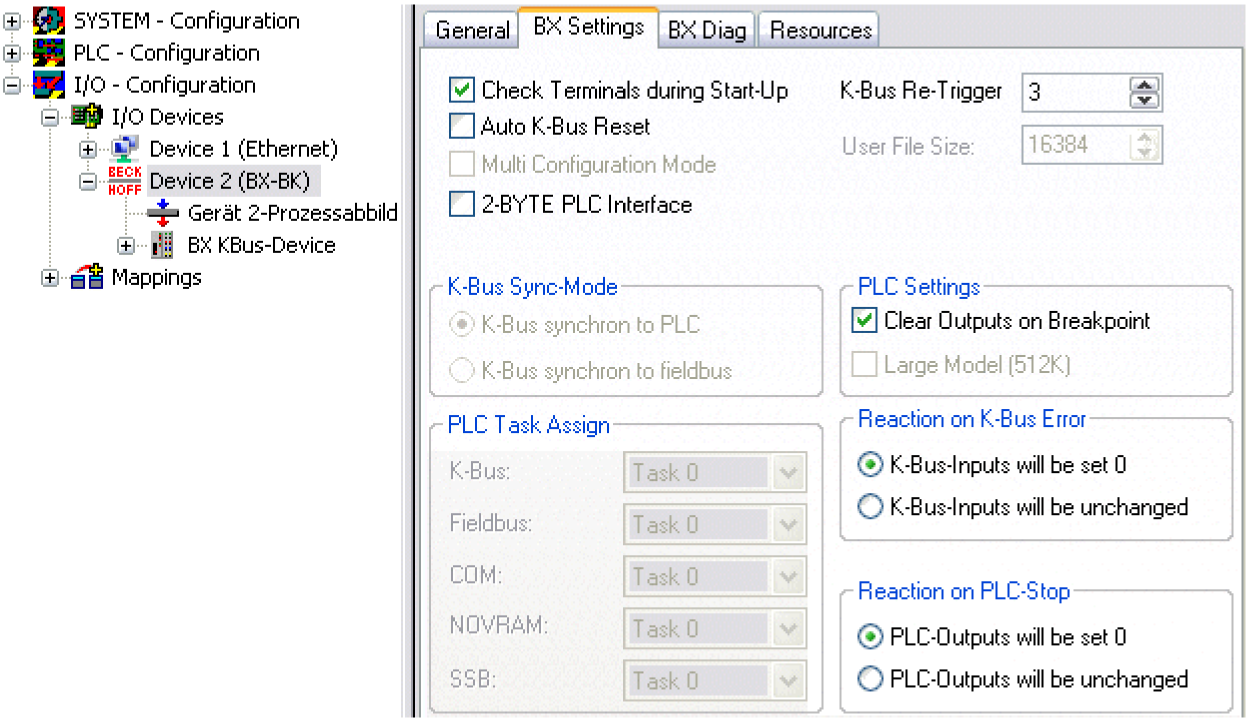
Fig. 52: Setting the poll rate

4.2.8 K-bus

Bus Terminal and end terminal required

To operate a Bus Terminal Controller of the BC or BX series, at least one Bus Terminal with process image and the end terminal must be connected to the K-bus.

BX Settings tab



- SYSTEM - Configuration
- PLC - Configuration
- I/O - Configuration
  - I/O Devices
    - Device 1 (Ethernet)
    - Device 2 (BX-BK)
      - Gerät 2-Prozessabbild
      - BX KBus-Device
  - Mappings

- ☑ Check Terminals during Start-Up
- □ Auto K-Bus Reset
- □ Multi Configuration Mode
- □ 2-BYTE PLC Interface

K-Bus Re-Trigger: 3

User File Size: 16384

K-Bus Sync-Mode
- ● K-Bus synchron to PLC
- ○ K-Bus synchron to fieldbus

PLC Settings
- ☑ Clear Outputs on Breakpoint
- □ Large Model (512K)

PLC Task Assign

K-Bus:	Task 0
Fieldbus:	Task 0
COM:	Task 0
NOVRAM:	Task 0
SSB:	Task 0

Reaction on K-Bus Error
- ● K-Bus-Inputs will be set 0
- ○ K-Bus-Inputs will be unchanged

Reaction on PLC-Stop
- ● PLC-Outputs will be set 0
- ○ PLC-Outputs will be unchanged

Fig. 53: BX Settings tab

Check Terminals during Start-up

When a boot project is created, the current Bus Terminal configuration is stored. The connected Bus Terminals are checked when the Bus Terminal Controller restarts. If this option is selected, the Bus Terminal Controller does not enter into data exchange. The PLC project will not be started.

Auto K-Bus Reset

Once a K-bus error has been rectified, the Bus Terminal Controller automatically resumes the data exchange.

⚠ CAUTION

Once a K-Bus error has been rectified, the outputs become active again immediately!

Ensure that the outputs are reactivated immediately and that analog outputs retain their programmed value, if this is not dealt with in your PLC project.

Clear Outputs on Breakpoint

If breakpoints are set in PLC Control, the K-Bus is no longer processed, i.e. the outputs are set to a safe state (zero).

K-Bus Sync Mode

Writing and reading of the Bus Terminals can take place synchronously with task 1 or the fieldbus.

K-Bus Re-Trigger

If the processor is busy dealing with the PLC project or the SSB, the K-Bus cannot be processed for a certain amount of time. This leads to triggering of the Bus Terminal watchdog and dropping of the outputs. The Bus Terminal Controller is set such that the K-bus watchdog is re-triggered 3 times after 85 ms. The K-Bus watchdog would then be activated.
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

Reaction on K-Bus Error

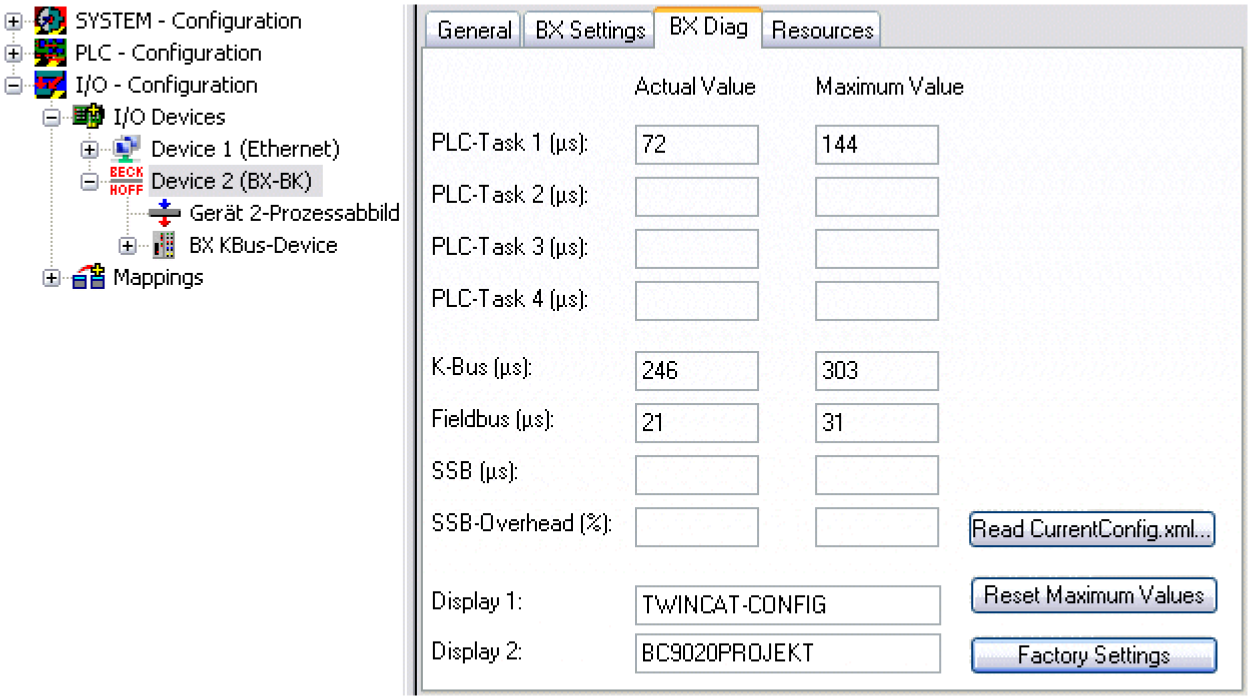
In the event of a K-Bus error, the K-Bus inputs are set to "0" or retain their last state.

Response on PLC-Stop

The user can set the behavior of the fieldbus output data in the event of the PLC project being stopped. The master will use these data as input data. In the event of a PLC stop, the data can be set to "0" or remain unchanged.

BX Diag tab

Display of the cycle time for task 1, K-bus, fieldbus processing and the SSB load.

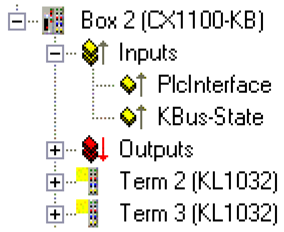


	Actual Value	Maximum Value
PLC-Task 1 (µs):	72	144
PLC-Task 2 (µs):		
PLC-Task 3 (µs):		
PLC-Task 4 (µs):		
K-Bus (µs):	246	303
Fieldbus (µs):	21	31
SSB (µs):		
SSB-Overhead (%):		
Display 1:	TWINCAT-CONFIG	
Display 2:	BC9020PROJEKT	

Fig. 54: BX Diag tab

Factory Settings: the Bus Terminal Controller is set to its delivery. These settings are reactivated via Restart System or by switching the system off and on again (display shows DEFAULT-CONFIG).
Reset Maximum Values: resets the maximum values

K-Bus variables



Box 2 (CX1100-KB)
Inputs
PlcInterface
KBus-State
Outputs
Term 2 (KL1032)
Term 3 (KL1032)

PLC interface: Not supported (only included for moving CX or BX projects)

K-bus state: see Diagnostics

4.2.9 PLC

4.2.9.1 Inserting a PLC project

For variable mapping, configuration has to be specified in the system manager. This is where the link between PLC and hardware is specified. The variables can process and link bit, byte, word or data structures. Automatic addressing via the System Manager is possible, but should be checked for offset.

Word alignment, byte orientation

With data structures, ensure that the Bus Terminal Controller saves the data in word alignment and the System Manager operates byte-oriented (see Data structures [▶ 102])

A valid project has to be compiled and saved in PLC Control. These data are saved as a *.tpy file. For inserting a PLC project, right-click on *PLC - Configuration*. Select your current PLC project.

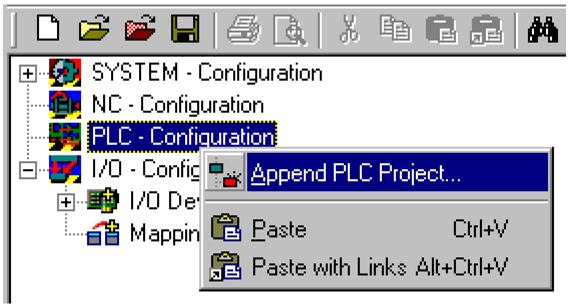
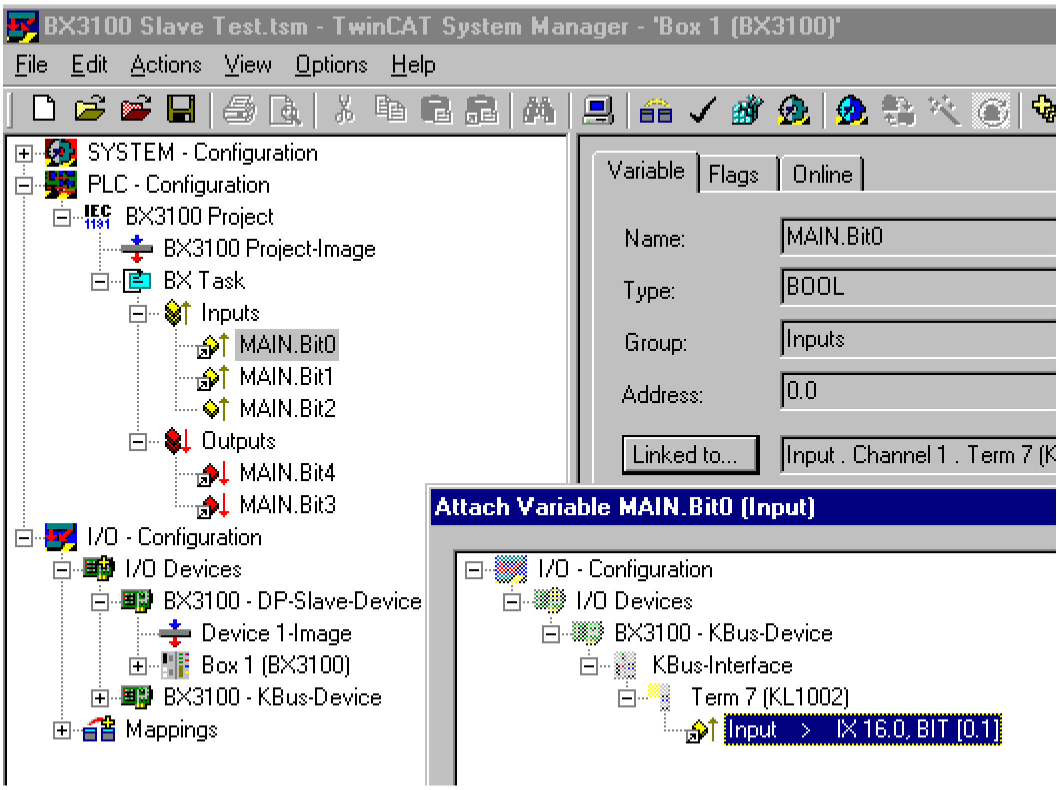


Fig. 55: Selecting the PLC project

Link the PLC variable with the hardware (e.g. digital Bus Terminal).


BX3100 Slave Test.tsm - TwinCAT System Manager - 'Box 1 (BX3100)'
File Edit Actions View Options Help
Variable
Flags
Online
Linked to...
Attach Variable MAIN.Bit0 (Input)

SYSTEM - Configuration
PLC - Configuration
BX3100 Project
BX3100 Project-Image
BX Task
Inputs
MAIN.Bit0
MAIN.Bit1
MAIN.Bit2
Outputs
MAIN.Bit4
MAIN.Bit3
I/O - Configuration
I/O Devices
BX3100 - DP-Slave-Device
Device 1-Image
Box 1 (BX3100)
BX3100 - KBus-Device
Mappings

Name: MAIN.Bit0
Type: BOOL
Group: Inputs
Address: 0.0
Linked to... Input . Channel 1 . Term 7 (K

I/O - Configuration
I/O Devices
BX3100 - KBus-Device
KBus-Interface
Term 7 (KL1002)
Input > IX 16.0, BIT [0.1]

Fig. 56: Connecting PLC variable and hardware

Once all links have been created, activate the configuration *Actions/Activate Configuration* (Ctrl+Shift+F4) and start TwinCAT *Set/Reset TwinCAT to Run Mode*. Ensure that you have selected the correct target system (bottom right in the System Manager window).

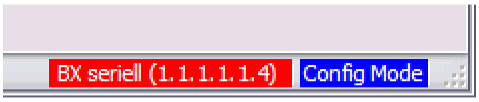
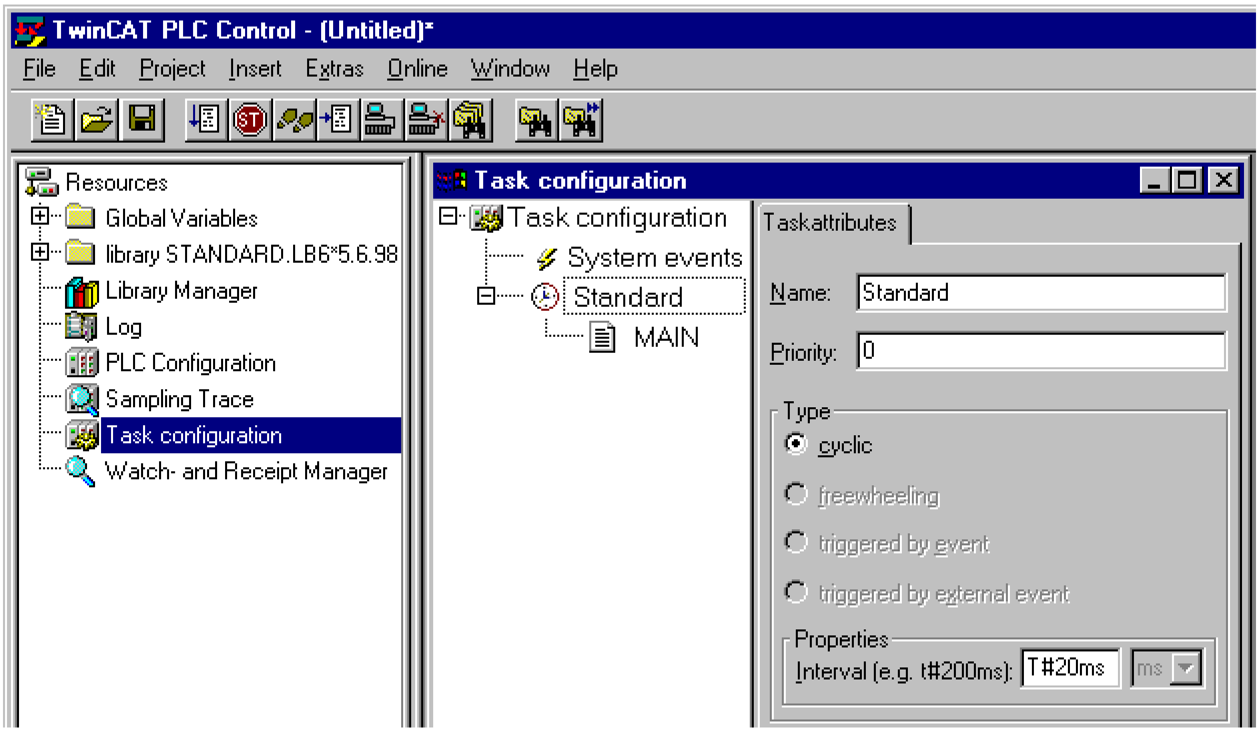

BX seriell (1.1.1.1.1.4) Config Mode

Fig. 57: Target system display

4.2.9.2 Measuring the PLC cycle time

The task time is set in PLC Control. The default setting is 20 ms.



Name: Standard

Priority: 0

Type

● cyclic

○ freewheeling

○ triggered by event

○ triggered by external event

Properties

Interval (e.g. t#200ms): T#20ms ms

Fig. 58: Setting the task time

In the default setting, the PLC program is called every 20 ms, as long as the general cycle time is less than 20 ms. To determine the load of your system, the PLC cycle time can be measured in the System Manager. In order to ensure trouble-free operation, the set task time should be 20-30 % higher than the measured total cycle time. A precise cycle time breakdown can be found under K-Bus tab [▸ 50] description. The total cycle time is displayed with the TcBase library (see TcBase.lbx or TcBaseBCxx50.lbx).

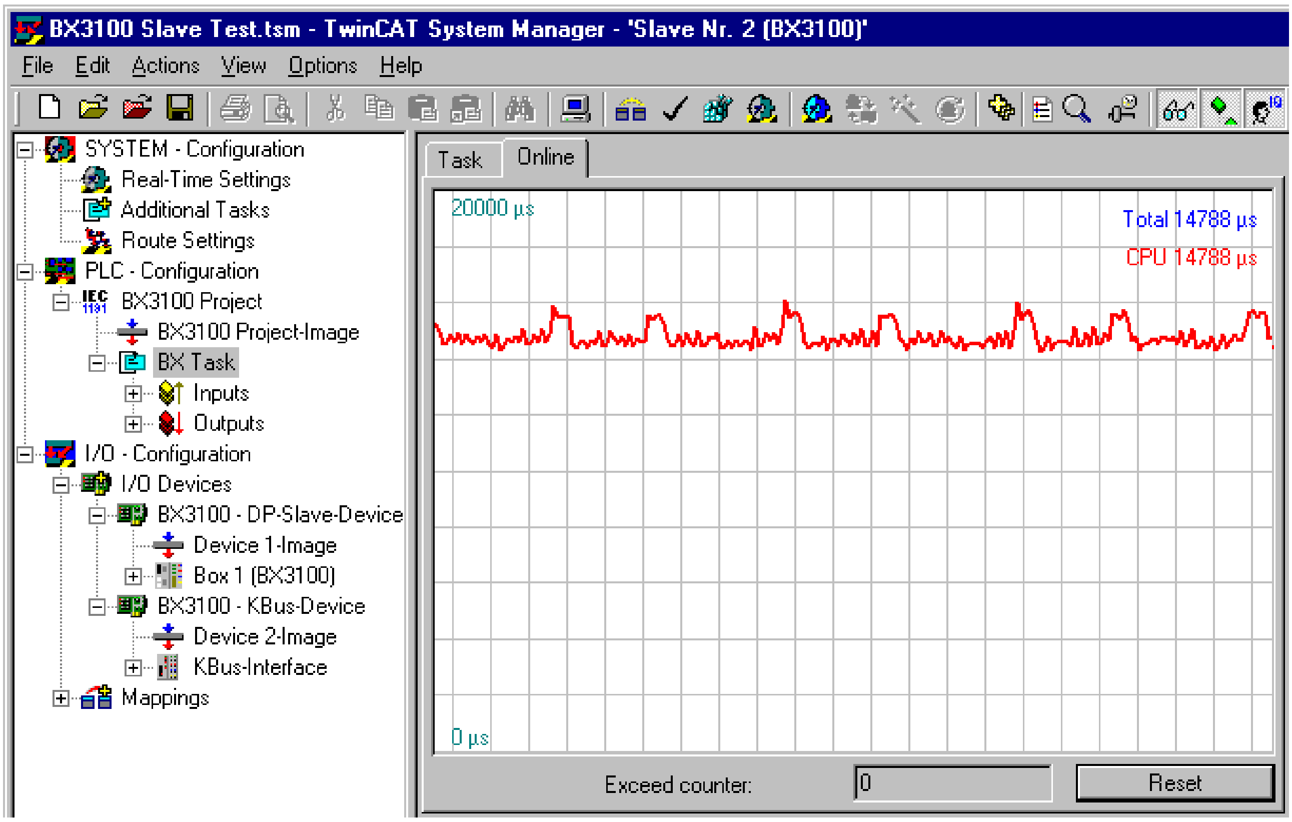


Fig. 59: Displaying the PLC cycle time

4.2.10 SSB

4.2.10.1 SSB overview

The SSB (Smart System Bus) is a sub-bus system based on CANopen. It is a CANopen master with limited functionality. CANopen slaves may be connected to this interface for reading or writing distributed I/Os. Parameterization SDOs (service data objects) can be sent to the slave via a start-up window.

Configuration

The SSB is configured via the TwinCAT System Manager (see TwinCAT config). The configuration is then loaded onto the BX controller via ADS.

Technical data

SSB	Data
Max. number of slaves	8
Max. number of PDOs	32 RxPODs / 32 TxPDOs
Baud rate	10 kbaud to 1 Mbaud
Permitted slave addresses	1 to 64

Sync telegram

The sync telegram is transferred depending on the PLC task time. If a task time of 20 ms is set, the sync telegram is also sent every 20 ms (asynchronous with the PLC run time). The sync telegram is only generated when a device requires it and is configured accordingly.
Sync telegrams are supported from firmware 1.12.

Guarding

Guarding is supported and is sent after a configurable interval.

4.2.10.2 CANopen cabling

Notes related to checking the CAN wiring can be found in the Trouble Shooting [▶ 66] section.

4.2.10.2.1 CAN topology

CAN is a 2-wire bus system, to which all participating devices are connected in parallel (i.e. using short drop lines). The bus must be terminated at each end with a 120 (or 121) Ohm terminating resistor to prevent reflections. This is also necessary even if the cable lengths are very short!

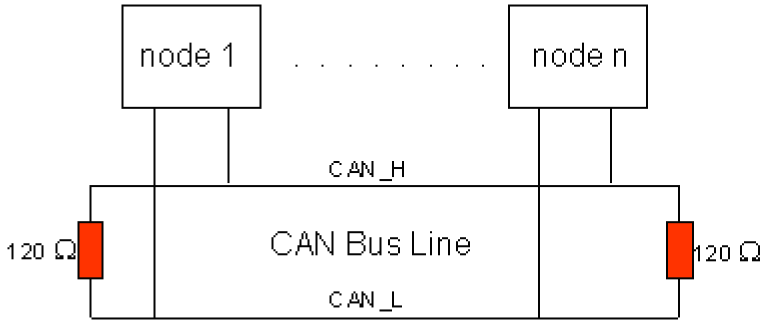


Fig. 60: Termination of the bus with a 120 Ohm termination resistor

Since the CAN signals are represented on the bus as the difference between the two levels, the CAN leads are not very sensitive to incoming interference (EMI): Both leads are affected, so the interference has very little effect on the difference.

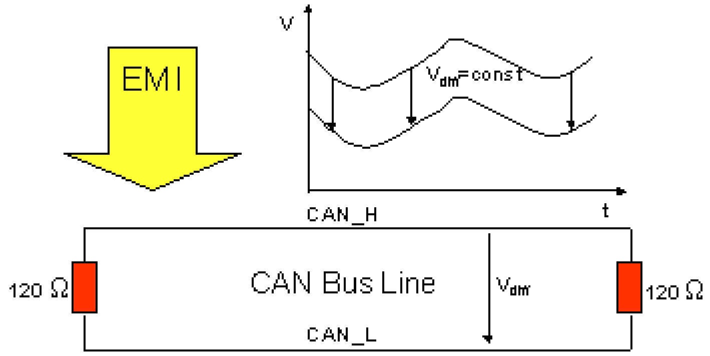


Fig. 61: Insensitivity to incoming interference

4.2.10.2.2 Bus length

The maximum length of a CAN bus is primarily limited by the signal propagation delay. The multi-master bus access procedure (arbitration) requires signals to reach all the nodes at effectively the same time (before the sampling within a bit period). Since the signal propagation delays in the CAN connecting equipment (transceivers, opto-couplers, CAN controllers) are almost constant, the line length must be chosen in accordance with the baud rate:

Baud rate	Bus length
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

*) A figure of 40 m at 1 Mbit/s is often found in the CAN literature. This does not, however, apply to networks with optically isolated CAN controllers. The worst case calculation for opto-couplers yields a figure 5 m at 1 Mbit/s - in practice, however, 20 m can be reached without difficulty.

It may be necessary to use repeaters for bus lengths greater than 1000 m.

4.2.10.2.3 Drop lines

Drop lines must always be avoided as far as possible, since they inevitably cause reflections. The reflections caused by drop lines are not however usually critical, provided they have decayed fully before the sampling time. In the case of the bit timing settings selected in the Bus Couplers it can be assumed that this is the case, provided the following drop line lengths are not exceeded:

Baud rate	Drop line length	Total length of all drop lines
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

Drop lines must not have terminating resistors.

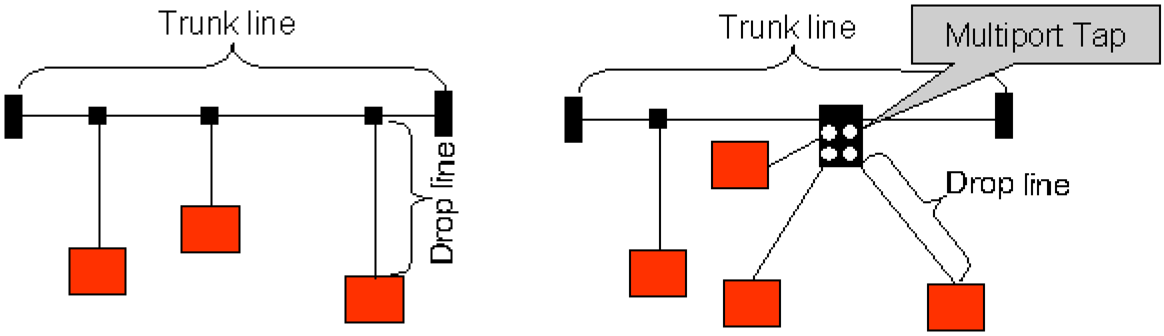


Fig. 62: Sample topology of drop lines

4.2.10.2.4 Star Hub (Multiport Tap)

Shorter drop line lengths must be maintained when passive distributors ("multiport taps"), such as the Beckhoff ZS5052-4500 Distributor Box. The following table indicates the maximum drop line lengths and the maximum length of the trunk line (without the drop lines):

Baud rate	Drop line length with multiport topology	Trunk line length (without drop lines)
1 Mbit/s	< 0,3 m	< 25 m
500 kbit/s	< 1,2 m	< 66 m
250 kbit/s	< 2,4 m	< 120 m
125 kbit/s	< 4,8 m	< 310 m

4.2.10.2.5 CAN cable

Screened twisted-pair cables (2x2) with a characteristic impedance of between 108 and 132 Ohm is recommended for the CAN wiring. If the CAN transceiver's reference potential (CAN ground) is not to be connected, the second pair of conductors can be omitted. (This is only recommended for networks of small physical size with a common power supply for all the participating devices).

ZB5100 CAN Cable

A high quality CAN cable with the following properties is included in Beckhoff's range:

- 2 x 2 x 0.25 mm² (AWG 24) twisted pairs, cable colors: red/black + white/black
- double screened
- braided screen with filler strand (can be attached directly to pin 3 of the 5-pin connection terminal)
- flexible (minimum bending radius 35 mm when bent once, 70 mm for repeated bending)
- characteristic impedance (60 kHz): 120 ohm
- conductor resistance < 80 Ohm/km
- sheath: grey PVC, outside diameter 7.3 +/- 0.4 mm
- Weight: 64 kg/km.
- printed with "Beckhoff ZB5100 CAN-BUS 2x2x0.25" and meter marking (length data every 20cm)

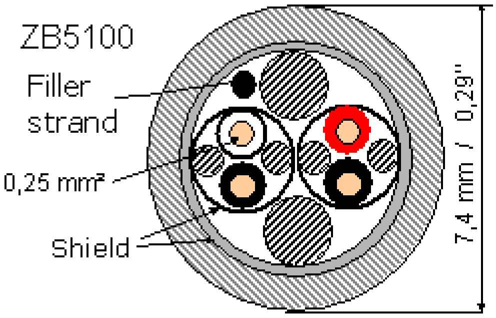


Fig. 63: Structure of CAN cable ZB5100

ZB5200 CAN/DeviceNet Cable

The ZB5200 cable material corresponds to the DeviceNet specification, and is also suitable for CANopen systems. The ready-made ZK1052-xxxx-xxxx bus cables for the Fieldbus Box modules are made from this cable material. It has the following specification:

- 2 x 2 x 0.34 mm² (AWG 22) twisted pairs
- double screened, braided screen with filler strand
- characteristic impedance (1 MHz): 126 ohm
- Conductor resistance 54 Ohm/km
- sheath: grey PVC, outside diameter 7.3 mm
- printed with "InterlinkBT DeviceNet Type 572" as well as UL and CSA ratings
- stranded wire colors correspond to the DeviceNet specification
- UL recognized AWM Type 2476 rating
- CSA AWM I/II A/B 80°C 300V FT1
- corresponds to the DeviceNet "Thin Cable" specification

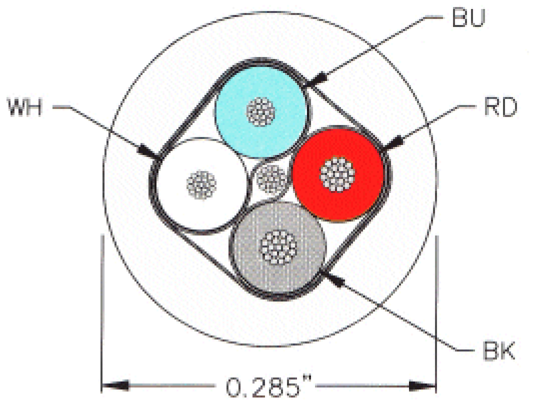


Fig. 64: Structure of CAN/DeviceNet cable ZB5200

4.2.10.2.6 Shielding

The screen is to be connected over the entire length of the bus cable, and only galvanically grounded at one point, in order to avoid ground loops.
The design of the screening, in which HF interference is diverted through R/C elements to the mounting rail assumes that the rail is appropriately earthed and free from interference. If this is not the case, it is possible that HF interference will be transmitted from the mounting rail to the screen of the bus cable. In that case the screen should not be attached to the couplers - it should nevertheless still be fully connected through.

Notes related to checking the CAN wiring can be found in the Trouble Shooting [▶ 66] section.

4.2.10.2.7 Cable colors

Suggested method of using the Beckhoff CAN cable on Bus Terminal and Fieldbus Box:

BK51x0 pin PIN BX5100 (X510)	Pin BK5151 CX8050, CX8051, CXxxxx-B510/M510	Fieldbus Box pin	Pin FC51xx	Function	ZB5100 cable color	ZB5200 cable color
1	3	3	3	CAN Ground	**black/** (red)	**black**
2	2	5	2	CAN Low	**black**	**blue**
3	5	1	5	Shield	Filler strand	Filler strand
4	7	4	7	CAN high	**white**	**white**
5	9	2	9	not used	**(red)**	**(red)**

4.2.10.2.8 BK5151, FC51xx, CX with CAN interface and EL6751: D-sub, 9 pin

The CANbus cable is connected to the FC51x1, FC51x2 CANopen cards and in the case of the EL6751 CANopen master/slave terminal via 9-pin Sub-D sockets with the following pin assignment.

Pin	Assignment
2	CAN low (CAN-)
3	CAN ground (internally connected to pin 6)
6	CAN ground (internally connected to pin 3)
7	CAN high (CAN+)

The unlisted pins are not connected.
The mounting rail contact spring and the plug shield are connected together.

Note: an auxiliary voltage of up to 30 V_{DC} may be connected to pin 9. Some CAN devices use this to supply the transceiver.

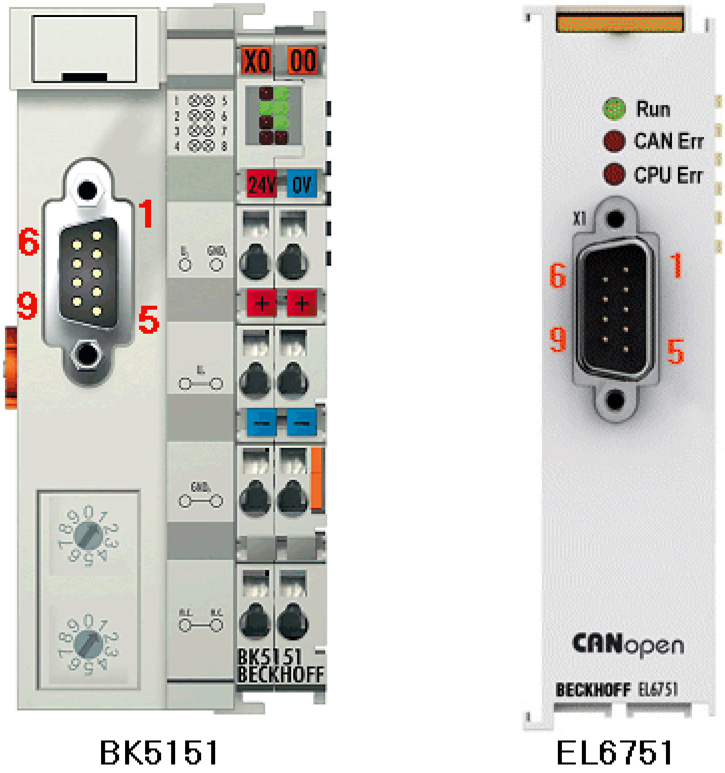


Fig. 65: BK5151, EL6751 pin assignment

FC51x2:

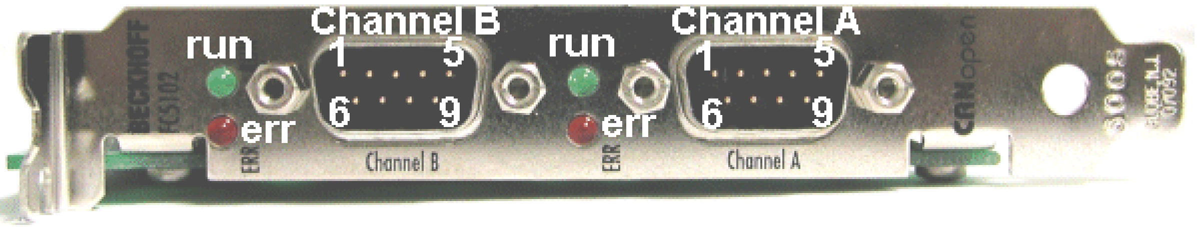


Fig. 66: FC51x2

4.2.10.2.9 BK51x0/BX5100: 5-pin open style connector

The BK51x0/BX5100 (X510) Bus Couplers have a recessed front surface on the left hand side with a five pin connector.
The supplied CANopen socket can be inserted here.

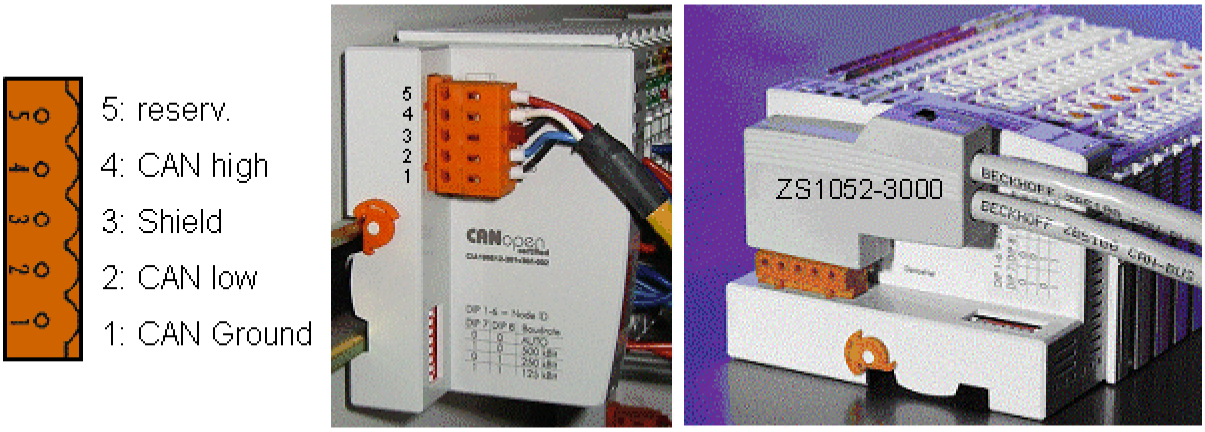


Fig. 67: BK51x0/BX5100 socket assignment

The left figure shows the socket in the BK51x0/BX5100 Bus Coupler. Pin 5 is the connection strip's top most pin. Pin 5 is not used. Pin 4 is the CAN high connection, pin 2 is the CAN low connection, and the screen is connected to pin 3 (which is connected to the mounting rail via an R/C network). CAN-GND can optionally be connected to pin 1. If all the CAN ground pins are connected, this provides a common reference potential for the CAN transceivers in the network. It is recommended that the CAN GND be connected to earth at one location, so that the common CAN reference potential is close to the supply potential. Since the CANopen BK51X0/BX5100 Bus Couplers provide full electrical isolation of the bus connection, it may in appropriate cases be possible to omit wiring up the CAN ground.

ZS1052-3000 Bus Interface Connector

The ZS1052-3000 CAN Interface Connector can be used as an alternative to the supplied connector. This makes the wiring significantly easier. There are separate terminals for incoming and outgoing leads and a large area of the screen is connected via the strain relief. The integrated terminating resistor can be switched externally. When it is switched on, the outgoing bus lead is electrically isolated - this allows rapid wiring fault location and guarantees that no more than two resistors are active in the network.

4.2.10.2.10 LC5100: Bus connection via spring-loaded terminals

In the low cost LC5100 Coupler, the CAN wires are connected directly to the contact points 1 (CAN-H, marked with C+) and 5 (CAN-L, marked with C-). The screen can optionally be connected to contact points 4 or 8, which are connected to the mounting rail via an R/C network.

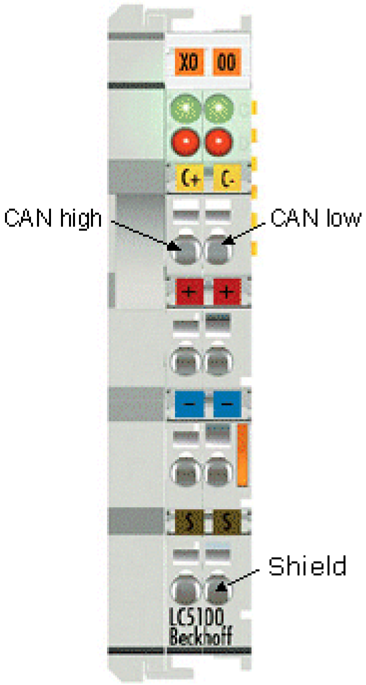


Fig. 68: LC5100

NOTE

Risk of device damage!

On account of the lack of electrical isolation, the CAN driver can be destroyed or damaged due to incorrect cabling. Always carry out the cabling in the switched-off condition.
First connect the power supply and then the CAN. Check the cabling and only then switch on the voltage.

4.2.10.2.11 Fieldbus Box: M12 CAN socket

The IPxxxx-B510, IL230x-B510 and IL230x-C510 Fieldbus Boxes are connected to the bus using 5-pin M12 plug-in connectors.

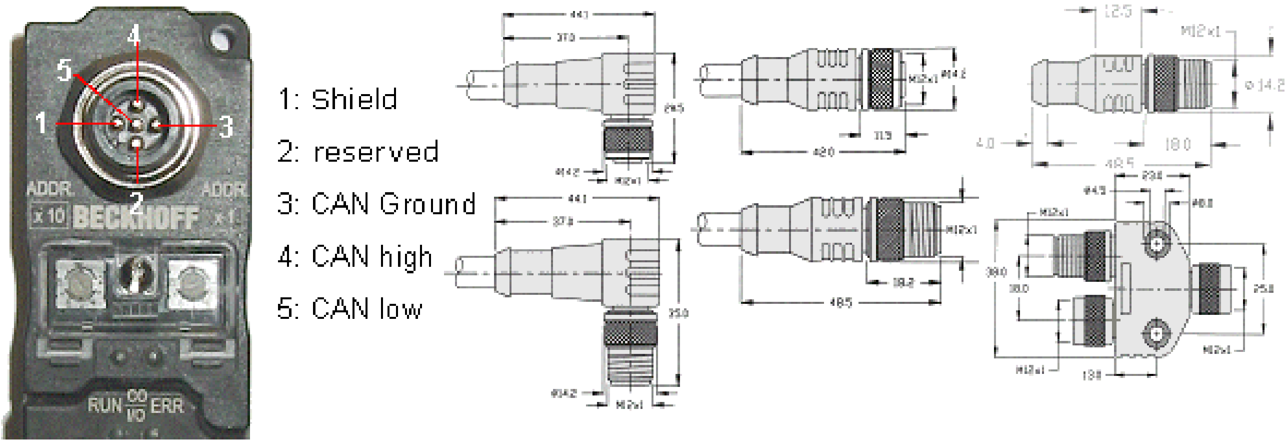


Fig. 69: Pin assignment: M12 plug, fieldbus box

Beckhoff offer plugs for field assembly, passive distributor's, terminating resistors and a wide range of pre-assembled cables for the Fieldbus Box system. Details be found in the catalogue, or under www.beckhoff.de.

4.2.10.3 SSB configuration

The SSB is configured in the system manager. Open your existing configuration, in which you have already configured the PLC project, the K-bus and the higher-level fieldbus, or open a new configuration.
Under I/O devices (left mouse button) a further device can now be appended.

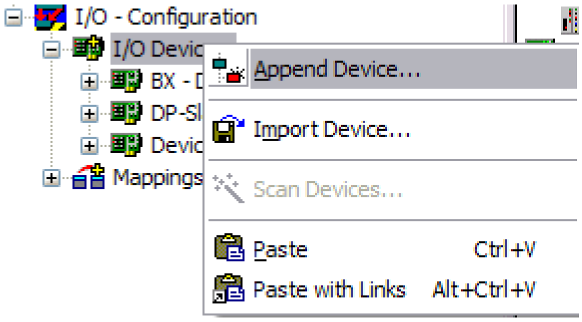
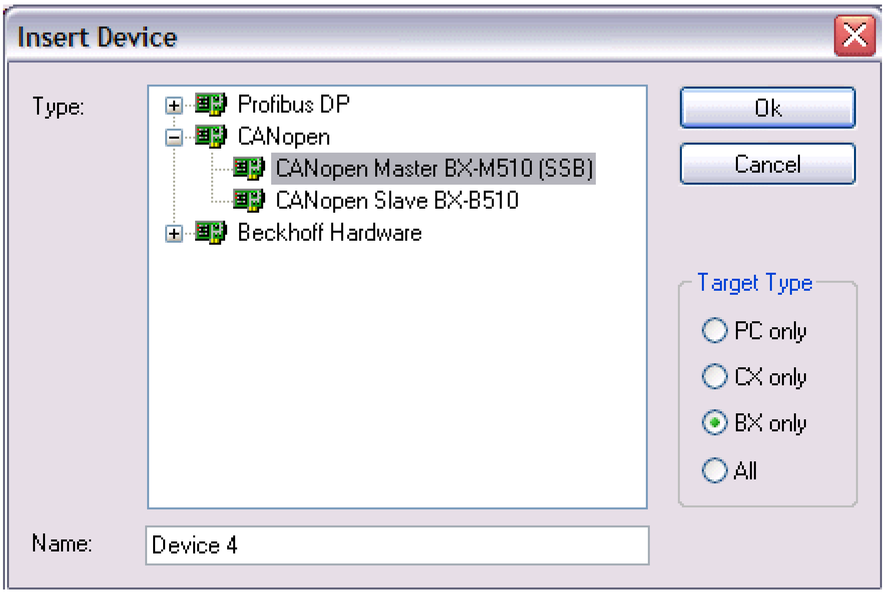


Fig. 70: Adding a further device

Select the CANopen Master SSB and confirm with OK.



Type: CANopen Master BX-M510 (SSB)

● BX only

Name: Device 4

Fig. 71: Selecting the CANopen master SSB

With the left mouse button, a CANopen node can now be selected on the SSB device.

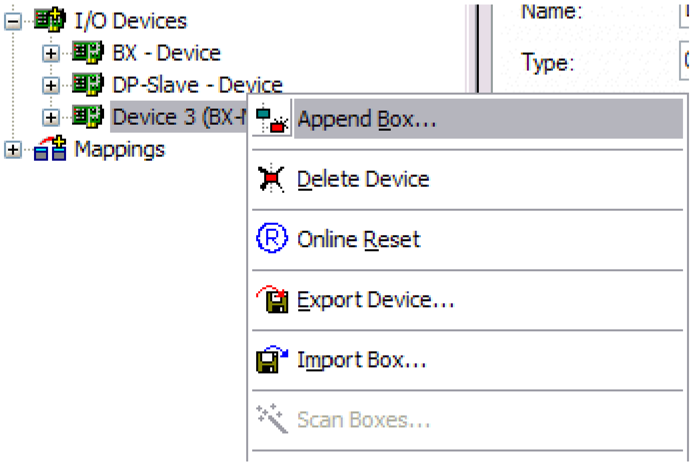


Fig. 72: Adding a CANopen device

All Beckhoff CAN nodes are available, as well as a general CANopen node for CANopen devices from other manufacturers.

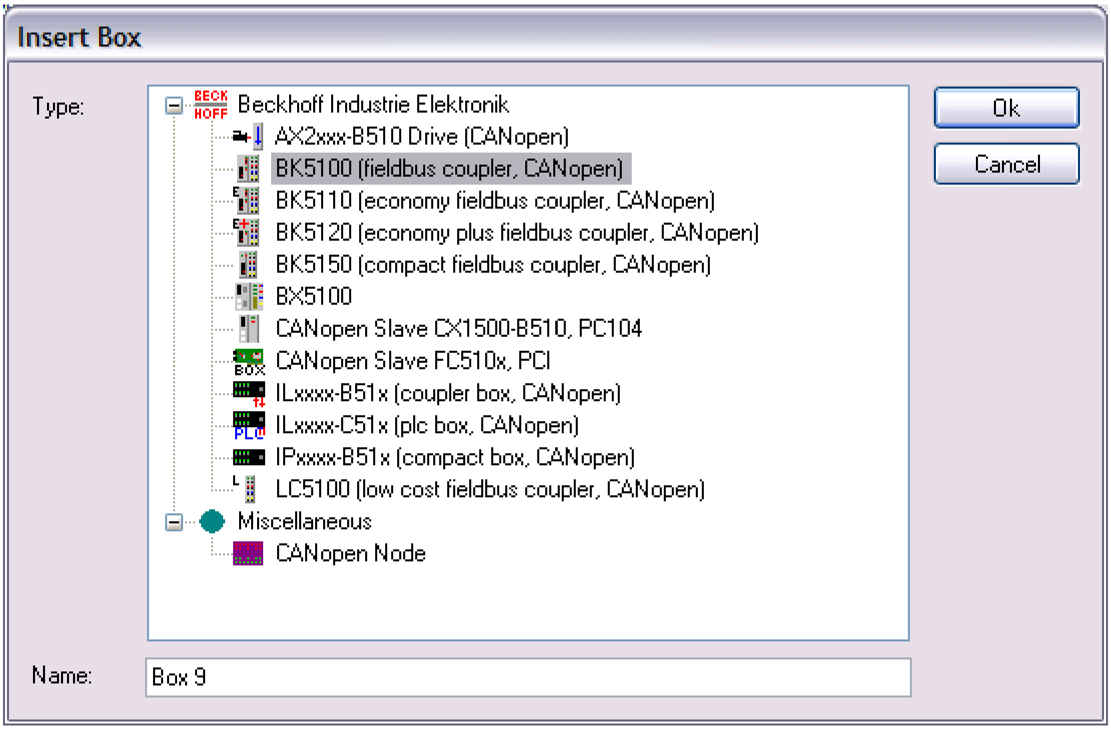

Insert Box
Type:
Beckhoff Industrie Elektronik
AX2xxx-B510 Drive (CANopen)
BK5100 (fieldbus coupler, CANopen)
BK5110 (economy fieldbus coupler, CANopen)
BK5120 (economy plus fieldbus coupler, CANopen)
BK5150 (compact fieldbus coupler, CANopen)
BX5100
CANopen Slave CX1500-B510, PC104
CANopen Slave FC510x, PCI
ILxxxx-B51x (coupler box, CANopen)
ILxxxx-C51x (plc box, CANopen)
IPxxxx-B51x (compact box, CANopen)
LC5100 (low cost fieldbus coupler, CANopen)
Miscellaneous
CANopen Node
Name:
Box 9
Ok
Cancel

Fig. 73: Selecting a CANopen node

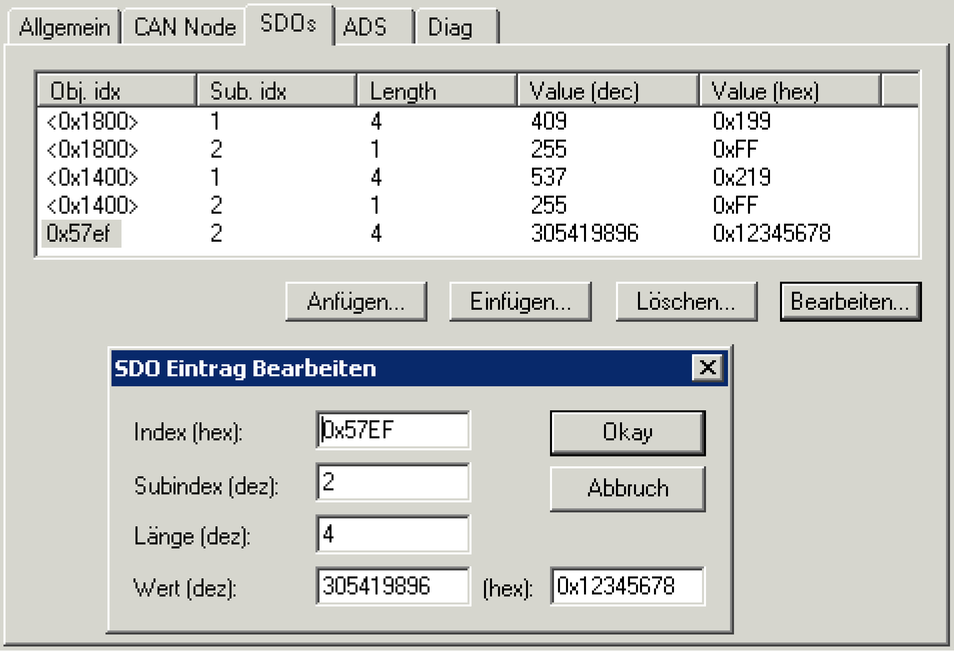
Now link the PLC variables with your CAN node. Once the configuration is complete, load it into the BX.

4.2.10.4 SSB - SDO communication

CANopen SDO communication (Service Data Object) is used to read or write any parameters in the CANopen bus node's object directory. The SSB uses the SDO communication for configuring the communication parameters during start-up.

Downloading Application-Specific Parameters when Starting Up

The appropriate parameters are to be entered here in the System Manager for the corresponding node in tab "SDO". The objects that result from the configuration under CAN node appear in square brackets. Any desired number of object directory entries can then be inserted.



Obj. idx	Sub. idx	Length	Value (dec)	Value (hex)
<0x1800>	1	4	409	0x199
<0x1800>	2	1	255	0xFF
<0x1400>	1	4	537	0x219
<0x1400>	2	1	255	0xFF
0x57ef	2	4	305419896	0x12345678

Index (hex): 0x57EF

Subindex (dez): 2

Länge (dez): 4

Wert (dez): 305419896 (hex): 0x12345678

Fig. 74: Adding/editing object directory entries

The SSB expects a positive acknowledgement of the parameter download from the respective bus device. If it was not possible to write a parameter (the bus device has aborted the SDO) the card then attempts to read the corresponding value back and to compare it with the value that was to be written. This is because it could, for instance, be a read-only value, and therefore already correctly configured within the bus device. If they agree with one another, the card moves onto the next parameter entry.

4.2.10.5 SDO communication from the PLC

ADS blocks are used for SDO communication from the PLC. These blocks can be used for sending SDO telegrams and receiving the response of the slave (ADSWRITE/ADSREAD).

Input parameters	Description
NETID	Local NetId of the BX or leave empty, e.g. with ''
Port number	0x1000$_{hex}$ + NodeId (slave number)
IDXGRP	SDO Index
IDXOFFS	SDO Subindex
LEN	Length of SDO data (1...4)

Download BX (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207257611.zip)

Setting individual or all nodes to pre-operational or operational state

The ADSWRTCTL function block can be used for setting individual CANopen nodes or all slaves to pre-operational or operational state.

Input parameters	Description
NETID	Local NetId of the BX or leave empty, e.g. with ''
Port number	0x1000$_{hex}$ + NodeId (slave number) / 153$_{dec}$ (all nodes)
ADSSTATE	ADSSTATE_RUN
DEVSTATE	1 - Pre / 0 - Operational
LEN	0
SRCADDR	0

Download BX (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207259787.zip)

Restarting the SSB interface

The ADSWRTCTL function block can be used to stop and restart the SSB. It should be stopped first before restarting it.

Input parameters	Description
NETID	Local NetId of the BX or leave empty, e.g. with ''
Port number	153_{dec}
ADSSTATE	ADSSTATE_STOP, ADSSTATE_RUN
DEVSTATE	0
LEN	0
SRCADDR	0

or

Input parameters	Description (from software version 1.16 for all BX controllers)
NETID	Local NetId of the BX or leave empty, e.g. with ''
Port number	300_{dec}
ADSSTATE	ADSSTATE_RESET
DEVSTATE	0
LEN	4
SRCADDR	ADR on a DWORD variable with the ID of the SSB device (the ID can be obtained from the System Manager file and is typically a value between 1 and 3).

4.2.10.6 CANopen Trouble Shooting

Error Frames

One sign of errors in the CAN wiring, the address assignment or the setting of the baud rate is an increased number of error frames: the diagnostic LEDs then show *Warning Limit exceeded* or *Bus-off state entered*.

Error Frames

Warning limit exceeded, passive error or bus-off state are indicated first of all at those nodes that have detected the most errors. These nodes are not necessarily the cause for the occurrence of error frames!
If, for instance, one node contributes unusually heavily to the bus traffic (e.g. because it is the only one with analog inputs, the data for which triggers event-driven PDOs at a high rate), then the probability of its telegrams being damaged increases. Its error counter will, correspondingly, be the first to reach a critical level.

Node ID / Setting the Baud Rate

Care must be taken to ensure that node addresses are not assigned twice: there may only be one sender for each CAN data telegram.

Test 1

Check node addresses. If the CAN communication functions at least some of the time, and if all the devices support the boot up message, then the address assignment can also be examined by recording the boot up messages after the devices are switched on. This will not, however, recognize node addresses that have been swapped.

Test 2

Check that the same baud rate has been set everywhere. For special devices, if the bit timing parameters are accessible, do they agree with the CANopen definitions (sampling time, SJW, oscillator).

Testing the CAN wiring

These tests should not be carried out if the network is active: No communication should take place during the tests. The following tests should be carried out in the stated sequence, because some of the tests assume that the previous test was successful. Not all the tests are generally necessary.

Network terminator and signal leads

The nodes should be switched off or the CAN cable unplugged for this test, because the results of the measurements can otherwise be distorted by the active CAN transceiver.

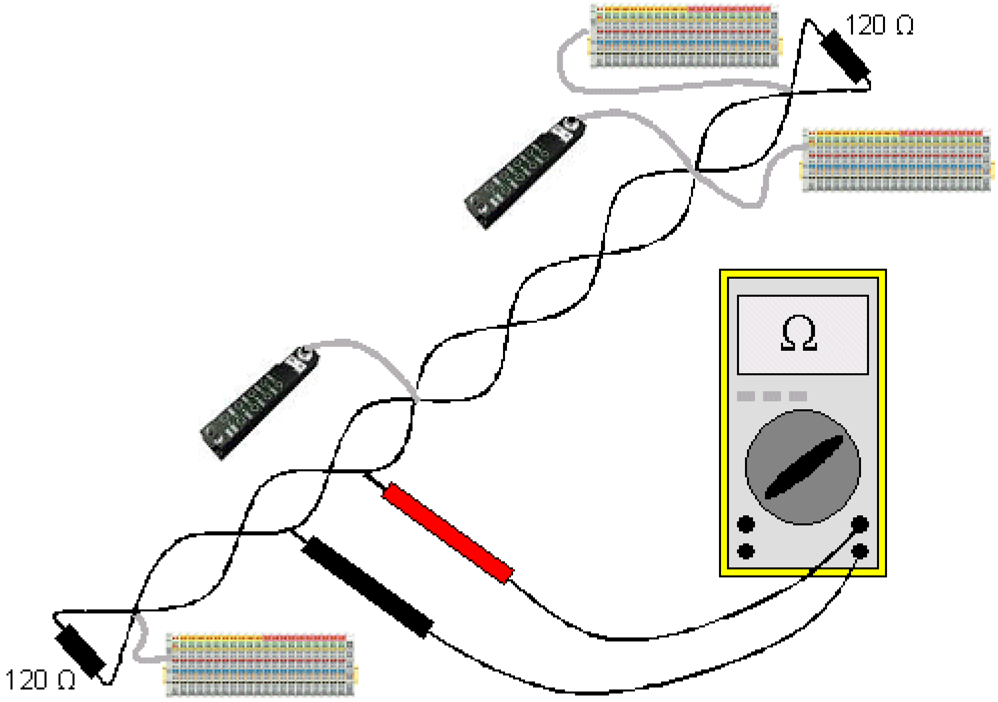


Fig. 75: Wiring diagram for test setup

Test 3

Determine the resistance between CAN high and CAN low - at each device, if necessary.

If the measured value is greater than 65 Ohms, it indicates the absence of a terminating resistor or a break in a signal lead. If the measured value is less than 50 Ohms, look for a short circuit between the CAN lines, more than the correct number of terminating resistors, or faulty transceivers.

Test 4

Check for a short circuit between the CAN ground and the signal leads, or between the screen and signal leads.

Test 5

Remove the earth connection from the CAN ground and screen. Check for a short circuit between the CAN ground and screen.

Topology

The possible cable length in CAN networks depends heavily on the selected baud rate. CAN will tolerate short drop lines - although this again depends on the baud rate. The maximum permitted drop line length should not be exceeded. The length of cable that has been installed is often underestimated - estimates can even be a factor of 10 less than the actual length. The following test is therefore recommended:

Test 6

Measure the lengths of the drop lines and the total bus lengths (do not just make rough estimates!) and compare them with the topology rules for the relevant baud rate.

Screening and earthing

The power supply and the screen should be carefully earthed at the power supply unit, once only and with low resistance. At all connecting points, branches and so forth the screen of the CAN cable (and possibly the CAN GND) must also be connected, as well as the signal leads. In the Beckhoff IP20 Bus Couplers, the screen is grounded for high frequencies via an R/C element.

Test 7

Use a DC ammeter (16 amp max.) to measure the current between the power supply ground and the shield at the end of the network most remote from the power supply unit. An equalization current should be present. If there is no current, then either the screen is not connected all the way through, or the power supply unit is not properly earthed. If the power supply unit is somewhere in the middle of the network, the measurement should be performed at both ends. When appropriate, this test can also be carried out at the ends of the drop line.

Test 8

Interrupt the screen at a number of locations and measure the connection current. If current is flowing, the screen is earthed at more than one place, creating a ground loop.

Potential differences

The screen must be connected all the way through for this test, and must not be carrying any current - this has previously been tested.

Test 9

Measure and record the voltage between the screen and the power supply ground at each node. The maximum potential difference between any two devices should be less than 5 volts.

Detect and localize faults

The "low-tech approach" usually works best: disconnect parts of the network, and observe when the fault disappears.

However, this does not work well for problems such as excessive potential differences, ground loops, EMC or signal distortion, since the reduction in the size of the network often solves the problem without the "missing" piece being the cause. The bus load also changes as the network is reduced in size, which can mean that external interference "hits" CAN telegrams less often.

Diagnosis with an oscilloscope is not usually successful: even when they are in good condition, CAN signals can look really chaotic. It may be possible to trigger on error frames using a storage oscilloscope - this type of diagnosis, however, is only possible for expert technicians.

Protocol problems

In rare cases, protocol problems (e.g. faulty or incomplete CANopen implementation, unfavorable timing at boot up, etc.) can be the cause of faults. In this case it is necessary to trace the bus traffic for evaluation by a CANopen experts - the Beckhoff support team can help here.

A free channel on a Beckhoff FC5102 CANopen PCI card is appropriate for such a trace - Beckhoff make the necessary trace software available on the internet. Alternatively, it is of course possible to use a normal commercial CAN analysis tool.

Protocol problems can be avoided if devices that have not been conformance tested are not used. The official CANopen Conformance Test (and the appropriate certificate) can be obtained from the CAN in Automation Association (http://www.can-cia.de).

4.2.10.7 Emergency telegrams and diagnostics

The status of the CAN slave is indicated by NodeState. The DiagFlag is set if an emergency telegram was received. The EmergencyCounter is incremented with each emergency telegram.

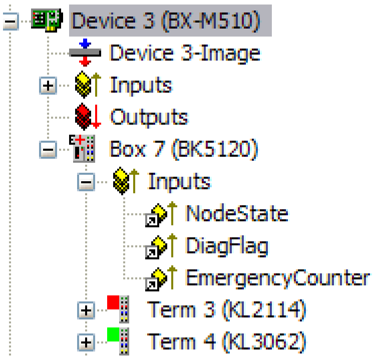


Fig. 76: NodeState, DiagFlag and EmergencyCounter

NodeState value	Description
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

Reading of emergency telegrams with AdsRead

Input parameters	Description
NETID	local NetId of BX
Port number	153
IDXGRP	16#xxxxF180 (xxxx) NodeId, the Diag flag is only reset when at least 106 bytes are read 16#xxxxF181 (xxxx) NodeId, the Diag flag is reset immediately
IDXOFFS	Byte Offset

Description of the array

Offset	Bit	Value / description
0 - 1	Bit 0	reserved
	Bit 1	Boot up message not received or incorrect
	Bit 2	Emergency-Overflow
	Bit 3 - 15	reserved
2 - 3	Bit 0 - 14	TX-PDO (i+1) received
	Bit 15	All TX PDOs 16-n received
4 - 5	Bit 0 - 4	1: Incorrect TX PDO length
		2: Synchronous TX PDO absent
		3: Node signaling PRE-OPERATIONAL
		4: Event timer timed out for TX PDO
		5: No response and guarding is activated
		6: Toggling missed several times and guarding activated
	Bit 5 - 15	Associated COB ID
6	Bit 0 - 7	1: Incorrect value during SDO upload
		2: Incorrect length during SDO upload
		3: Abort during SDO up/download
		4: Incorrect date during a boot-up message
		5: Timeout while waiting for a boot-up message
7	Bit 0 - 7	2: Incorrect SDO command specifier
		3: SDO toggle bit has not changed
		4: SDO length too great
		5: SDO-Abort
		6: SDO-Timeout
8 - 9	Bit 0 - 7	SDO up/download index
10	Bit 0 - 7	SDO up/download subindex
11	Bit 0 - 7	reserved
12	Bit 0 - 7	errorClass des Aborts
13	Bit 0 - 7	errorCode des Aborts
14 - 15	Bit 0 - 15	Abort additionalCode
16 - 19		Read value (if offset 6 = 1)
20 - 23		Expected value (if offset 6 = 1)
24 - 25		Number of consecutive emergencies
26 - n		Emergencies (8 bytes each)

Download BX (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207261963.zip)

Download sample System Manager file BX (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207264139.zip)

Reading the number of PDO telegrams with AdsRead

Input parameters	Description
NETID	local NetId of BX
Port number	153
IDXGRP	16#xxxxF930 (xxxx) NodeId
IDXOFFS	0

Download BX (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207266315.zip)

Configuration of the node ID required

The node ID must be configured before the ADS function blocks is called in the TwinCAT configuration.

Sending a CAN message

This ADSWRITE command enables any CAN message to be sent.

Input parameters	Description
NETID	local NetId of BX
Port Nummer	153
IDXGRP	16#0000F921
IDXOFFS	0
LEN	11 bytes
SRCADDR	Pointer to an 11 byte ARRAY

Structure of the 11 byte CAN data

Byte	Description	Example 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 (length)	0x08 (LEN, may be 5 in this case)
4	Data[1]	0x22 (Download Request)
5	Data[2]	0x00 (Index Low Byte)
6	Data[3]	0x21 (Index High Byte)
7	Data[4]	0x01 (Sub Index)
8	Data[5]	0x01 (Value "1")
9	Data[6]	0x00
10	Data[7]	0x00
11	Data[8]	0x00

4.2.10.8 Examples

4.2.10.8.1 BK5120 at SSB

Required material:

- TwinCAT 2.9 build 953 or higher
- BX3100 version 0.80 or higher, 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
- Cabling material and power supply
- TwinCAT System Manager file (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207268491.zip)

(The system manager file has to be transferred to the BX controller via ADS).

- BX program file (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207270667.zip)

For the configuration download via ADS, either a BECKHOFF fieldbus master or a free serial port is required.

4.2.10.8.2 Communication between BX controllers (via SSB)

2 or more BX controllers can exchange data via the SSB. Use 2 telegrams for configuring this data exchange in the System Manager (CAN layer).

CAN telegram communication is specified via the COB ID. The BX type is irrelevant, since the SSB is present on each BX controller, and the behavior and configuration is identical.

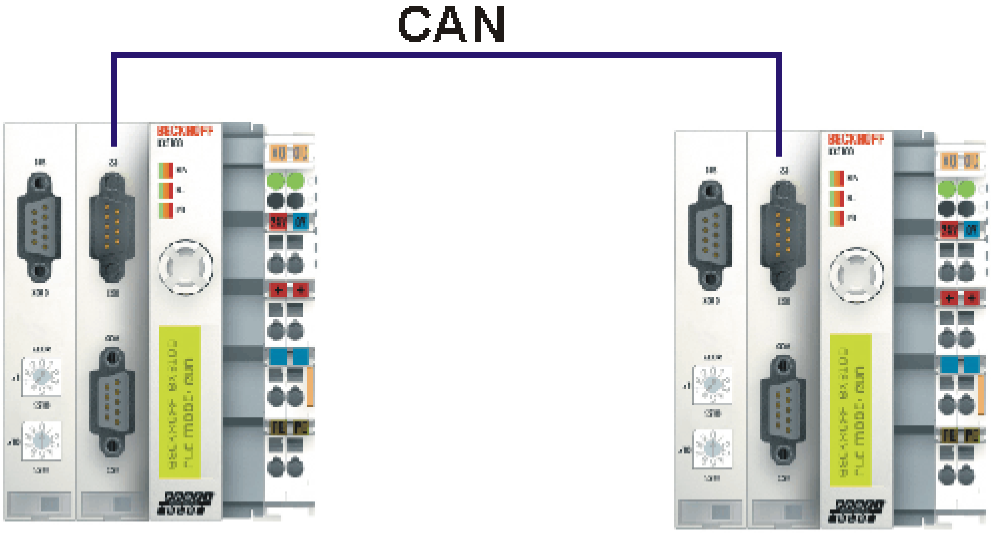


Fig. 77: Communication between BX controllers (via SSB)

Example configuration

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

Configuration and program example:

Required material

- TwinCAT 2.9 build 959 or higher
- 2 x BXxx00
- Cabling material and power supply
- TwinCAT System Manager file BX_ONE (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207272843.zip)
- Program file BX_ONE (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207275019.zip)

- TwinCAT System Manager file BX_TWO (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207277195.zip)

- Program file BX_ONE (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207279371.zip)

For the configuration download via ADS, either a BECKHOFF master (FC310x, FC510x, FC520x) or a free serial port is required.

4.2.10.8.3 AX2000 at SSB

Fig. 78: AX2000

Required material:

- TwinCAT 2.9 build 953 or higher
- BX3100 version 0.80 or higher
- 1 x KL1xx4
- 1 x KL2xx4
- 1 x KL9010
- 1 x AX2000 with the following settings: Slave address 4, baud rate 500 kbyte
- Cabling material and power supply
- Example program and configuration on the BX Controller

- TwinCAT System Manager file (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207281547.zip)

  (The System Manager file has to be transferred to the BX Controller via ADS).
- BX program file (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207283723.zip)

For the configuration download via ADS, either a BECKHOFF fieldbus master or a free serial port is required.

AX2000 description

The following sections are extracts from the AX2000 drive manual. Further information can be found on the internet at http://www.beckhoff.de.

Hardware and interfaces

Setting the Station Address

The station address (device address at the CAN bus) of the servo drive can be set in three

ways:

- Via the front panel keyboard (see AX2000 installation guide)
- Via the "Basic settings" screen of the DRIVE.EXE commissioning software
- Via the serial interface with the following ASCII command sequence:

ADDR nn > SAVE > COLDSTART (with nn = address)

The address range can be extended from 1..63 to 1..127 with the ASCII object

MDRV.

Setting the baud rate

The CAN transfer speed (baud rate) can be set in three ways:

- Via the front panel keyboard (see AX2000 installation guide)
- Via the "Basic settings" screen of the DRIVE.EXE commissioning software
- Via the serial interface with the following ASCII command sequence:
  CBAUD bb > SAVE > COLDSTART (with bb = baud rate in kB)

Possible baud rates are 10, 20, 50, 100, 125, 250, 333, 500 (default), 666, 800 or 1000 kBaud.

CANopen Interface (X6)

Interface for connection to the CAN bus (default 500 kBaud). The integrated profile is based on the DS301 CANopen communication profile and on the DSP402 drive profile. The following functions are available in combination with the position controller:

jogging with variable speed, reference motion, start travel command, start direct travel command, digital set value specification, data transfer functions and many others.
Detailed information can be found in the CANopen manual. The interface is electrically isolated via an optocoupler and has the same potential as the RS232 interface. The analog set value inputs can still be used. The two interfaces (RS232 and CAN) occupying the same connector (X6) can be split to two connectors via the optional 2 CAN extension card.

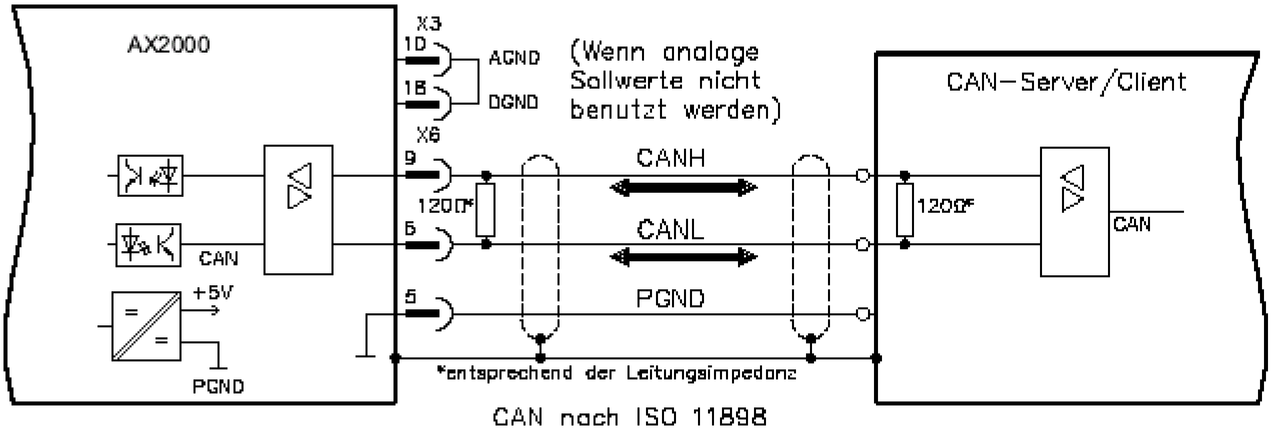


Fig. 79: CANopen Interface (X6)

4.2.10.8.4 Cimrex panel at the SSB of the BX controller

The CAN interface of the BX controller can also be used for connecting an operating panel. In this example, a panel from the company Beijers is connected. Further information on the panel can be found under http://www.beijerelectronics.de.

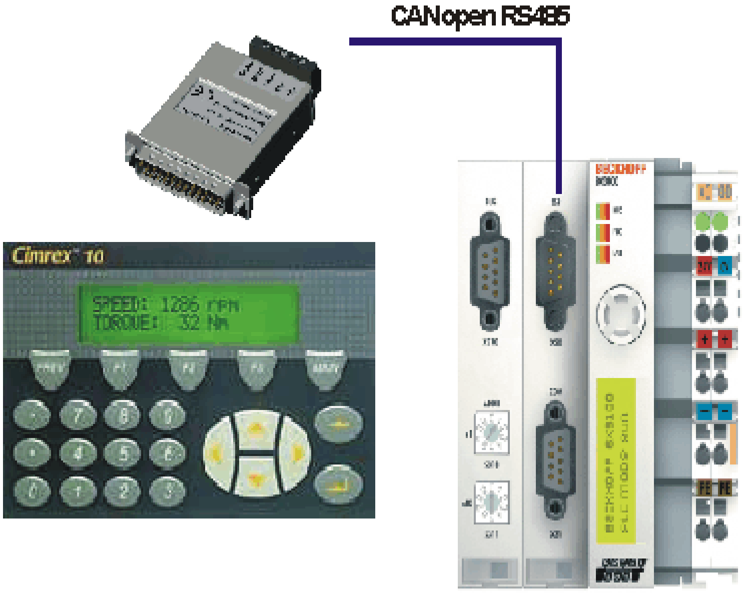


Fig. 80: Cimrex panel at the SSB of the BX controller

Necessary components

- 1 x BX3100
- Some Bus Terminals for the K-bus (here 3 x KL2114, can be adjusted in the System Manager file)
- 1 x Cimrex 41
- 1 x CAB 15 CAN adapter
- BX sample program in ST: (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207285899.zip)
- BX example configuration: (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207288075.zip)

- Example for Cimrex 41: (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207290251.zip)
    - Baud rate 500 kbaud
    - CAN slave address 10

4.2.10.8.5 IclA drive at SSB

Fig. 81: IclA drive at SSB

Required material:

- TwinCAT 2.9 build 953 or higher
- BX3100 version 0.80 or higher
- 1 x KL1xx4
- 1 x KL2xx4
- 1 x KL9010
- 1 x IclA D065 with the following settings: slave address 10, baud rate 500 kbyte (Please note: These are not the default parameters of the drive)
- Cabling material and power supply

For the configuration download via ADS, either a BECKHOFF fieldbus master or a free serial port is required.

Reconfiguration example for TwinCAT with FC510x CANopen master card

An example for converting a drive is listed below.

- TwinCAT System Manager file (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207292427.zip)
- TwinCAT PLC file (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207294603.zip)

Example program and configuration on the BX Controller

- TwinCAT System Manager file (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207296779.zip)

  (The System Manager file has to be transferred to the BX Controller via ADS).
- BX program file (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207298955.zip)

IclA D065 description

The following sections are extracts from the IclA drive manual. They were provided by the company SIG Positec Automation GmbH for the purpose of describing the basic parameters. Further information can be found on the internet at http://www.sig-positec.de.

Hardware and interfaces

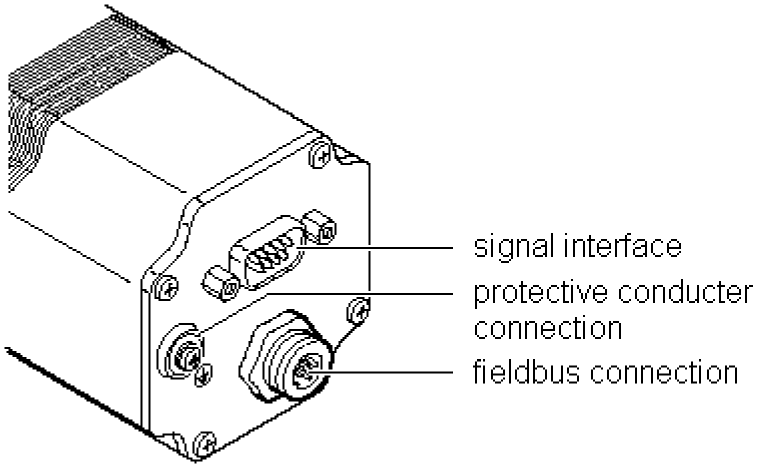


Fig. 82: IclA drive connections

- Signal interface for
  - Supply voltage
  - Control signals for manual mode
  - Connection for emergency stop signal
- Protective conductor connection for earthing via PE bus bar
- Fieldbus connection for connecting the fieldbus cable.

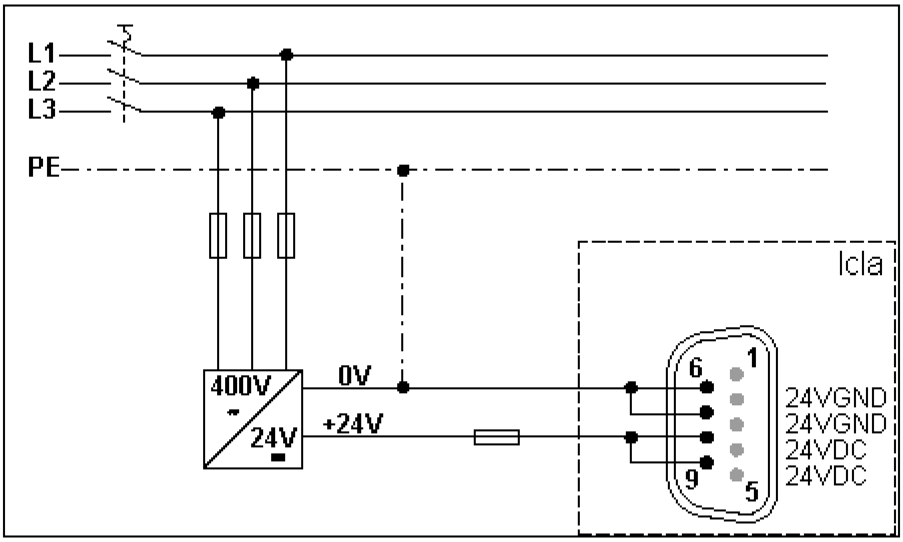


Fig. 83: Signal interface

If the emergency stop function is not required, connect pin 2 with pin 8 or 9 (24 V_{DC}).

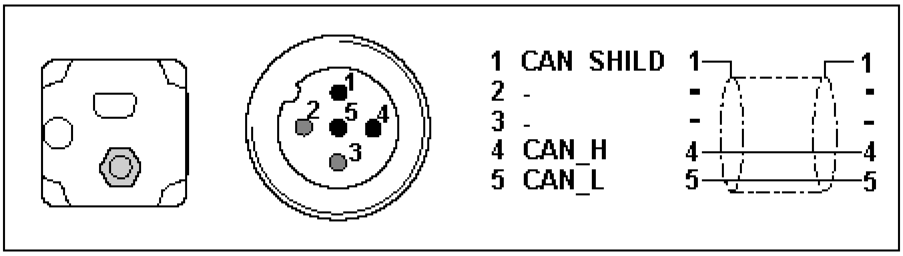


Fig. 84: Fieldbus connection

Control word 0x6040

The object represents the control word for the device. The control word is used for several control tasks:

- Changeover between different operating states. The possible states and transitions can be found under the index keyword "machine state". Bits 0 to 3 and bit 7 are relevant for a change of state.
- Starting and stopping mode-specific functions, e.g. starting a travel command via bit 4. Bits 4 to 6 are used for mode-specific settings. Further details can be found under the keywords "Operating mode, starting", "Operating mode, monitoring" and in the description of the respective operating modes in sections "Manual mode" and "Positioning mode".
- Stopping of the positioning drive from an active travel operation. Bit 8 "Stop" is used for stopping. Further details can be found under the keywords "Operating mode, starting" and "Operating mode, monitoring".

Object description	Value description
Index	6040h
Object name	Control word
Data type	Integned16
Subindex	00h, control word
Access	read-write
PDO-Mapping	R_PDO

Bit	Name	Meaning
11..15	Manufacturer specific	not used
9, 10	-	reserved
8	Stop	Stop motor
7	Reset fault	Reset fault
4..6	-	Operating mode dependent,
3	Enable operation	Execute operating mode
2	Quick stop (low active)	Breaking with quick stop ramp
1	Disable voltage (low active)	Switch off voltage
0	Switch on	Switch into ready-to-run state

Status word 0x6041

The object describes the current operating state of the device. The status word can be used for the following monitoring functions:

- Checking the operating state of the positioning controller. Bits 0 to 3, 5 and 6 are relevant.
- Bit 4 indicates whether the output stage is ready for processing a transport instruction.
- Bits 7 to 15 are used for monitoring the travel operation and for status monitoring of device-specific states.

Further details for monitoring travel operation can be found under the keywords "Operating mode, starting", "Operating mode, monitoring" and in the description of the respective operating modes in sections "Manual mode" and "Positioning mode". The bits for device status monitoring are described in section "Diagnostics and trouble shooting".
The control word is represented in the first two bytes of the R_PDOs.

Object description	Value description
Index	6041h
Object name	Status word
Data type	Unsigned16
Subindex	00h, status word
Access	read-only
PDO Mapping	T_PDO

Bit	Name	Meaning
15	Out of security area	Out of security area 0->1: Limit switch position S0 or S1 exceeded
14	Out of drive area	Out of drive area 0->1: Limit switch position D0 or D1 exceeded
12..13	-	Operating mode-dependent meaning
11	Internal limit active	Out of working area
10	Target reached	Target reached 1->0: New target position transferred 0->1: Requested target position reached or motor standstill after stop request
9	Remote	0: manual mode 1: no manual mode
8	Right out of drive area	Only valid if bit 11 = 1 - 0: Limit switch position W1 exceeded - 1: Limit switch position W0 exceeded
7	Warning	Warning
6	Switch on disabled	not ready for operation
5	Quick Stop	Quick stop active
4	Voltage disabled	Voltage off
3	Fault	Fault occurred
2	Operation enabled	Operating mode enabled
1	Switched on	Ready for operation
0	Ready to switch on	Ready to switch on

Reference ranges

A valid referencing is defined via three limit switch zones, which have to be within the possible traversing range of the drive. The limit switches protect the drive and the system from damage.

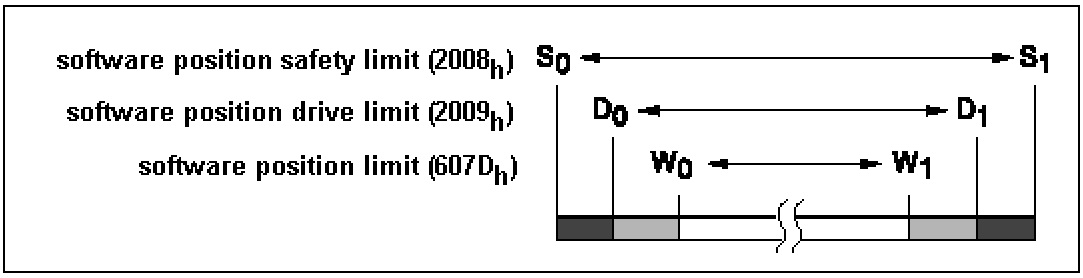
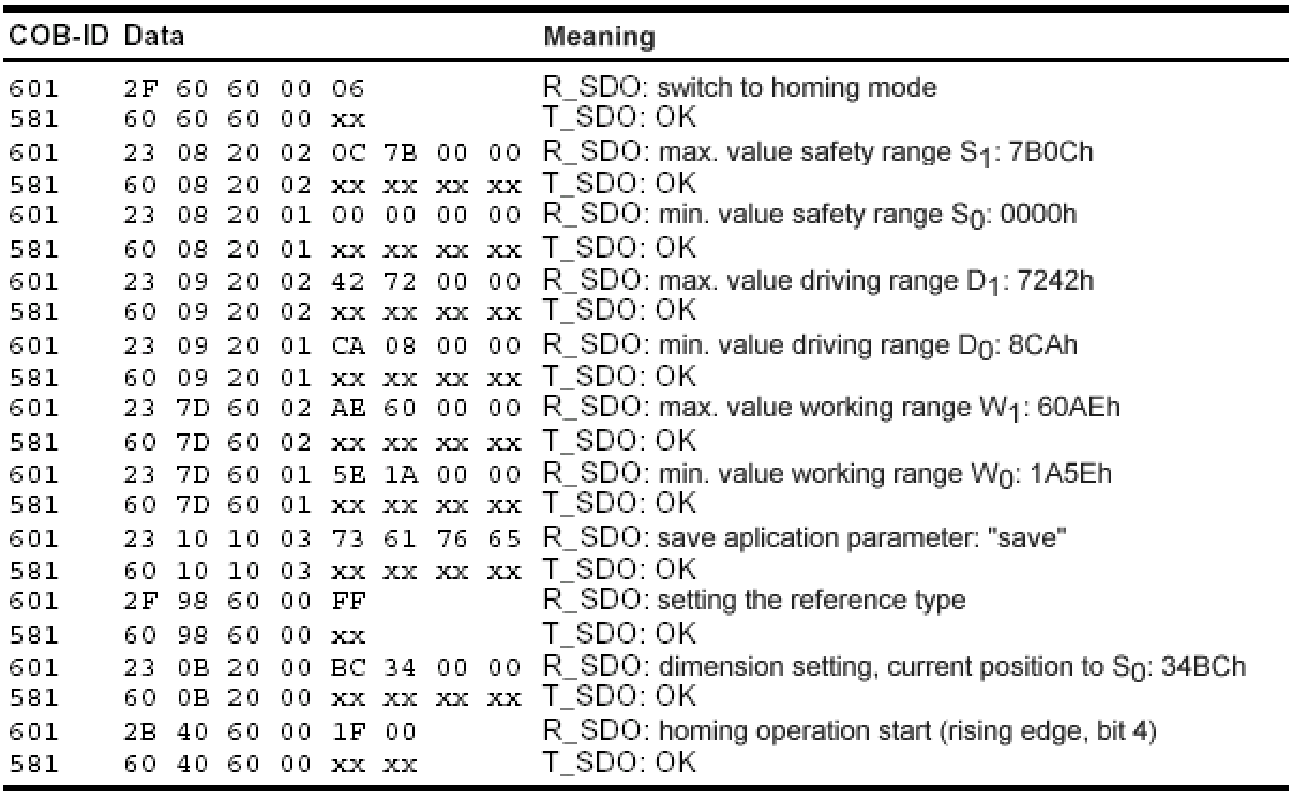

software position safety limit (2008h)
software position drive limit (2009h)
software position limit (607Dh)
S0
S1
D0
D1
W0
W1

Fig. 85: Reference ranges

- Working area W0 - W1 for positioning mode.
- Drive area D0 - D1. From ranges D0 - W0 and D1 - W1, the drive can
- only be moved backwards towards the operating range.
- Security area S0 - S1. From areas S0 - D0 and S1 - D1, the drive can only be moved backwards manually.
- CANopen objects - three CANopen objects are used for setting up the limit switches. They contain the position values for the upper and lower range limits.
- Working area limits in software position limit (607D$_{hex}$)
- Drive area limits in software position drive limit (2009$_{hex}$)
- Security area limits in software position safety limit (2008$_{hex}$)

Referencing example

The following listing demonstrates the input of the referencing values. The node address of the positioning drive is set to 01_{hex}.



COB-ID	Data	Meaning
601	2F 60 60 00 06	R_SDO: switch to homing mode
581	60 60 60 00 xx	T_SDO: OK
601	23 08 20 02 0C 7B 00 00	R_SDO: max. value safety range S_1: 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. value safety range S_0: 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. value driving range D_1: 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. value driving range D_0: 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. value working range W_1: 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. value working range W_0: 1A5Eh
581	60 7D 60 01 xx xx xx xx	T_SDO: OK
601	23 10 10 03 73 61 76 65	R_SDO: save aplication parameter: "save"
581	60 10 10 03 xx xx xx xx	T_SDO: OK
601	2F 98 60 00 FF	R_SDO: setting the reference type
581	60 98 60 00 xx	T_SDO: OK
601	23 0B 20 00 BC 34 00 00	R_SDO: dimension setting, current position to S_0: 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 (rising edge, bit 4)
581	60 40 60 00 xx xx	T_SDO: OK

Fig. 86: Listing of the referencing values

4.2.10.8.6 Lenze frequency converter at SSB

Fig. 87: Frequency converter from Lenze

Required material

- TwinCAT 2.9 build 953 or higher
- BXxx00
- 1 x KL1xx4
- 1 x KL2xx4
- 1 x KL9010
- 1 x Lenze 8200 vector + motor
- 1 x Lenze CANopen Interface 2175
- Cabling material and power supply

For the configuration download via ADS you need a BECKHOFF fieldbus master card or a free serial port.

Lenze description

The following sections are extracts from the Lenze 2175 manual. They were provided by the company Lenze Drive Systems GmbH for the purpose of describing the basic parameters. Further information can be found on the internet at http://www.Lenze.com.

Initial commissioning

Set the power supply for the bus module to internal power supply.

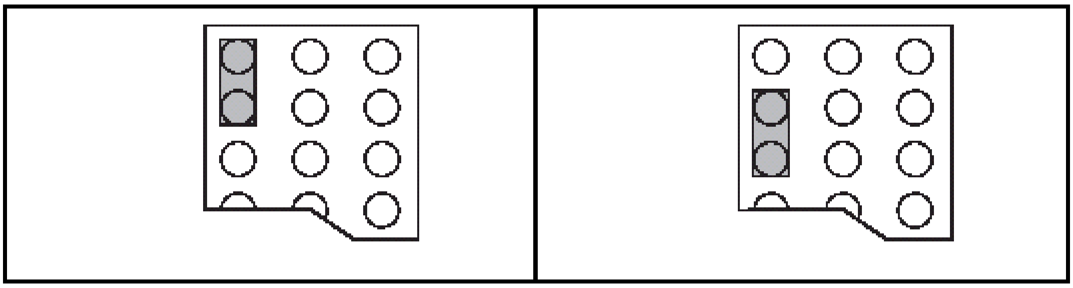


Fig. 88: External power supply - internal power supply
(State at Delivery)

For CANopen communication set DIP switch 10 to "ON".

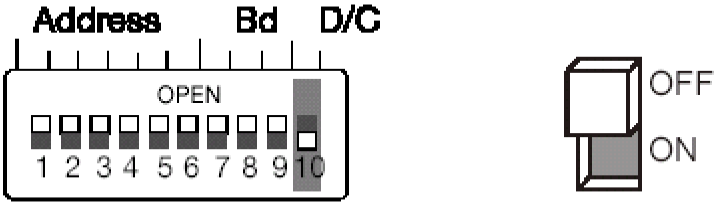


Fig. 89: DIP switch

Baud rate DIP switches 7-9

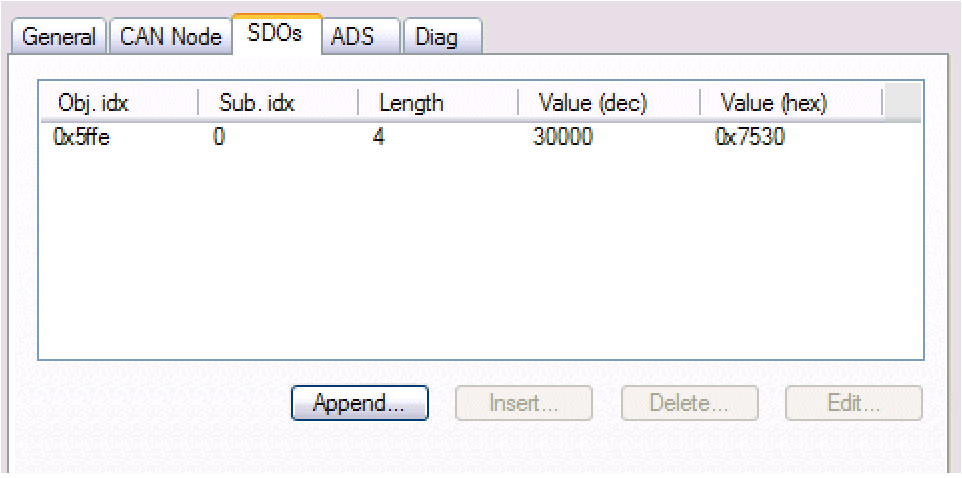
Transfer rate [kbps]	S7	S8	S9
10	ON	ON	OFF
20	ON	OFF	ON
50	OFF	ON	ON
125	OFF	ON	OFF
250	OFF	OFF	ON
500 (default)	OFF	OFF	OFF
1000	ON	OFF	OFF

Priority of the DIP switches

DIP switch 6 has the smallest weighting.
Example: Address 3 switches 5 and 6 "ON".

Enabling the communication module

Switch to operating mode 3 for enabling the communication module. This can be achieved via the SSB using the following entry:

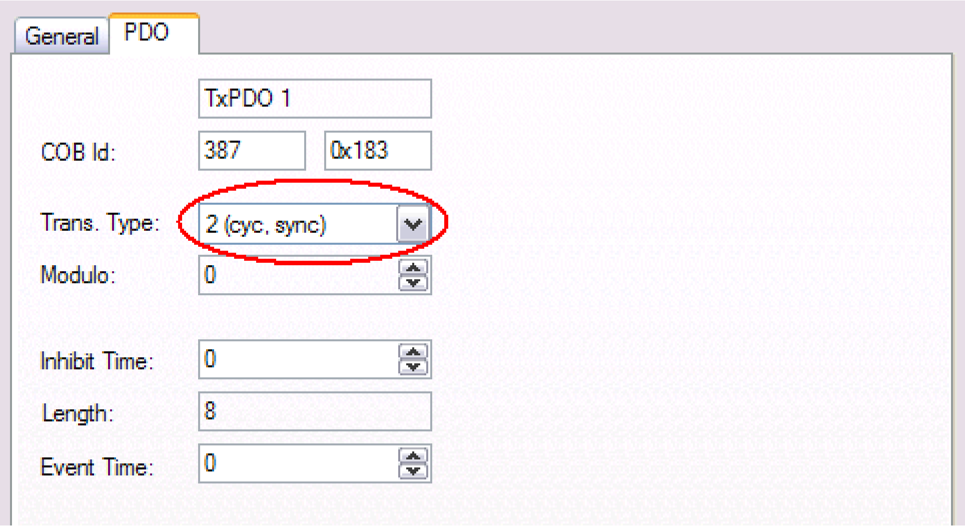


Obj. idx	Sub. idx	Length	Value (dec)	Value (hex)
0x5ffe	0	4	30000	0x7530

Fig. 90: Enabling the communication module

Sync telegram

In the default setting, the Lenze drive will send its output PDOs only once it has received a sync telegram from the CAN master. If you set the trans. type to 2, for example, the Lenze drive will send an output PDO after every second sync telegram it receives.



TxPDO 1

COB Id: 387 0x183

Trans. Type: 2 (cyc, sync)

Modulo: 0

Inhibit Time: 0

Length: 8

Event Time: 0

Sample project

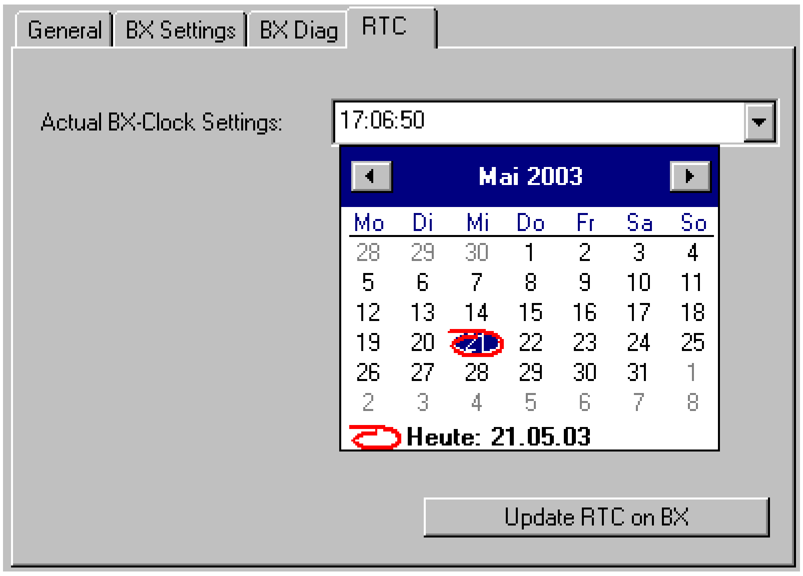
- TwinCAT-System-Manager-File: (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207301131.zip)
- TwinCAT-PLC-File: (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207303307.zip)

4.2.11 Real-Time Clock (RTC)

A real-time-clock (RTC) with backup battery is implemented on the BX controller. The clock has a battery.

Setting the real-time clock

The simplest way of setting the clock is via the System Manager. When the ADS communication is operating normally, the current time is displayed on the BX controller. To adjust the time, simply edit the time, and adjust the day, month and year with the drop down key. For setting the year, click on the year display and specify the required year. Repeat the procedure for the month. Once all parameters have been set, click on *Update RTC on BX*.



Actual BX-Clock Settings: 17:06:50

Mo	Di	Mi	Do	Fr	Sa	So
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

Fig. 91: Setting the real-time clock (RTC)

Service life of the battery

The service life of the battery may vary depending on utilization.

Reading the RTC on the BX controller (see example [▶ 121] Programming\Library)

The RTC can be read via a function block. Required libraries:

- TcSystemBX.lbx
- TCBaseBX.lbx

Writing the RTC to the BX controller

The RTC can be set via a function block. Required libraries:

- TcSystemBX.lbx
- TCBaseBX.lbx

Reading the RTC via ADS

Description	Meaning	Value
NETID	Target device	see System Manager "ADS"
Port	ADS port number	150_{dec}
IDXGRP	IDX group	$0x0000\_F100_{hex}$
IDXOFFS	IDX Offset	$0x0000\_0000_{hex}$
Length	Length of the data	16 byte
Variable type	Type of variable	TIMESTRUCT

Writing of the RTC via ADS

Description	Meaning	Value
NETID	Target device	see System Manager "ADS"
Port	ADS port number	150_{dec}
IDXGRP	IDX group	$0x0000\_F100_{hex}$
IDXOFFS	IDX Offset	$0x0000\_0000_{hex}$
Length	Length of the data	16 byte
Variable type	Type of variable	TIMESTRUCT

Setting via the navigation switch

See menu. [▸ 87]

Technical data for RTC

Accuracy: approx. 1 second/day
Duration for which the time is stored: approx. 3 months with fully charged battery
Service life of the battery: approx. 10 years for 10 cycles per day (1000 cycles for complete charge/discharge cycles)

4.2.12 COM port

The BX Controller has two serial interfaces. For PIN assignment please refer to Hardware description [▸ 24].

Setting options:

Description	Selection
Baud rate [▸ 113]	9600 baud 19200 baud 38400 baud (starting with auto baud rate detection) 57600 baud 115200 baud (COM 2 only)
Data bits	7 8 (Default)
Parity	NONE ODD EVEN (Default)
Stop bits	1 (default) 2

COM 1

The COM 1 interface is used for communication with the KS2000 software or with TwinCAT PLC Control (login via serial interface).

COM 2

The COM 2 interface (with RS 232 or RS 485) is used for the application of user protocols or protocol libraries (such as ModbusRTU, RK512, etc.) for the connection of other serial devices.

Library

Function blocks are available for communication with the serial interface.

- Documentation [▸ 113]
- Example [▸ 118]

- Library [▶ 113]

4.2.13 Menu

4.2.13.1 BX menu settings

To change into the menu, press the navigation switch for three seconds. The *Menu* directory appears first.

- You can change between the menu settings with the RIGHT/LEFT keys (the menu shown in row 1 is the active menu).
- Press the DOWN key for changing into a submenu.
- Press the UP key to return to the main menu.

Row 1 of the submenu shows the menu item, row 2 the current setting of this menu item.

Some settings cannot be changed *(read only)*. These items are only intended to provide checks and to give the user information. To close the menu it is necessary to be in the main menu and then to hold the navigation switch down for three seconds.

Before settings can be changed, a password has to be set. The password remains stored even during a firmware update and through a reset to the factory settings. If you forget the password, the BX controller will have to be sent in.

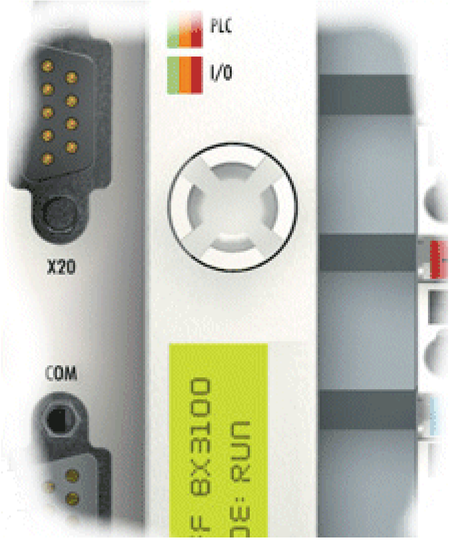


Fig. 92: Navigation switches of the BX controller

Switch assignment

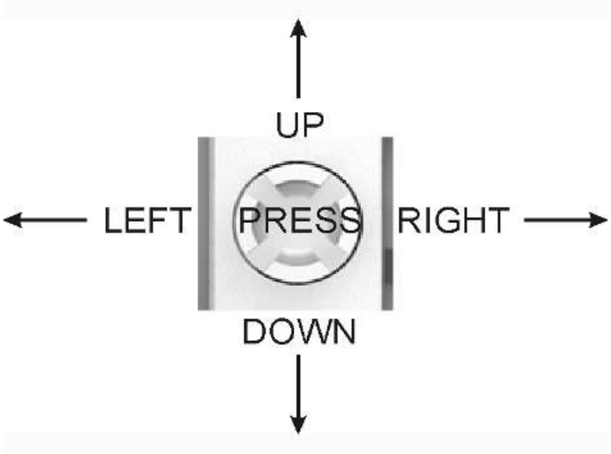


Fig. 93: Switch assignment

Main menu	Submenu 1st row	Submenu 2st row	Read/Write
MENU	Password	**** not set ???? set	See below
	Factory settings	Activate?	Pressing the key causes the factory setting to be reset and the controller to be rebooted automatically
	Reboot	Activate?	Pressing the key causes the controller to be rebooted
AMS	AMS	>AMS Net-ID<	read only
PLC	NAME	>current NAME<	read only
	Curr. Exex. Time	>current value<	[ms]
	Task time	>current value<	The cycle time can be set if the key is pressed
	Status	>Boot-Prj< >PLC Status<	Boot project exists PLC status
Config	NAME	>current NAME<	read only
	Delete config	Activate?	Pressing the right key causes the current configuration to be deleted
Real Time Clock	Date and Time	>current time<	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 *read only*	Baud rate	>current value<	9600/19200/38400/56800
COM 2 *read only*	Baud rate	>current value<	9600/19200/38400/56800/115k
SSB *read only*	Baud rate	>current value<	1MBaud, 500k, 250k, 125k, 100k, 50k
	Cycle Time	>current value< [in µs]	read only
	Utilization	>current value< [in %]	read only
K-bus *read only*	Diagnosis	>current diagnosis<	read only
	Number of Bus Terminals	>current value<	read only

Bus-specific menu items

BX3100

F-bus PROFIBUS *read only*	Address*	>current value<	1-126
	Baud rate*	>current value<	read only
	Status	>current value<	read only
	Diagnostic*	>current value<	read only

BX5100

F-bus CANopen *read only*	Address*	>current value<	1-126
	Baud rate*	>current value<	read only
	Status	>current value<	read only
	Diagnostic*	>current value<	read only

*) in preparation

BX9000

Ethernet	MAC ID	>current value<	000105-xx-xx-xx, read only
	ADDR.STATE	>current value<	read only
	ADDRESSING MODE	FIXED IP (default) DHCP BOOTP BOOTP & SAVE	read / write
	NAME	>current value<	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

The default setting is "****", i.e. no password is active. A password is required for setting parameters.

Menu navigation

Press the navigation switch for three seconds to switch to the Directory menu. Some of the menu items are described below.

MENU

MENU	Main Menu

DOWN (press briefly)

PASSWORD **????**	???? - password set **** - no password set

PRESS (press briefly)

PASSWORD ENTER? **????**	Do you want to enter the password Yes - PRESS (press for approx. 2 seconds, then enter the password) / No - UP

PRESS (press for approx. 2 seconds, then enter the password)

PASSWORD **????**	Enter password PRESS <OK>

PRESS

PASSWORD **1234**	Enter password OK <PRESS> and <PRESS> again to confirm (press for approx. 1 second until OK appears in the display)

F bus (only BX3100)

F-BUS **WAIT FOR SETPRM**	Fieldbus status (read only) WAIT FOR SETPRM - waiting for parameter data from PROFIBUS

SSB

SSB	Smart System Bus

COM2

COM2	Serial interface COM2 (read only)

DOWN

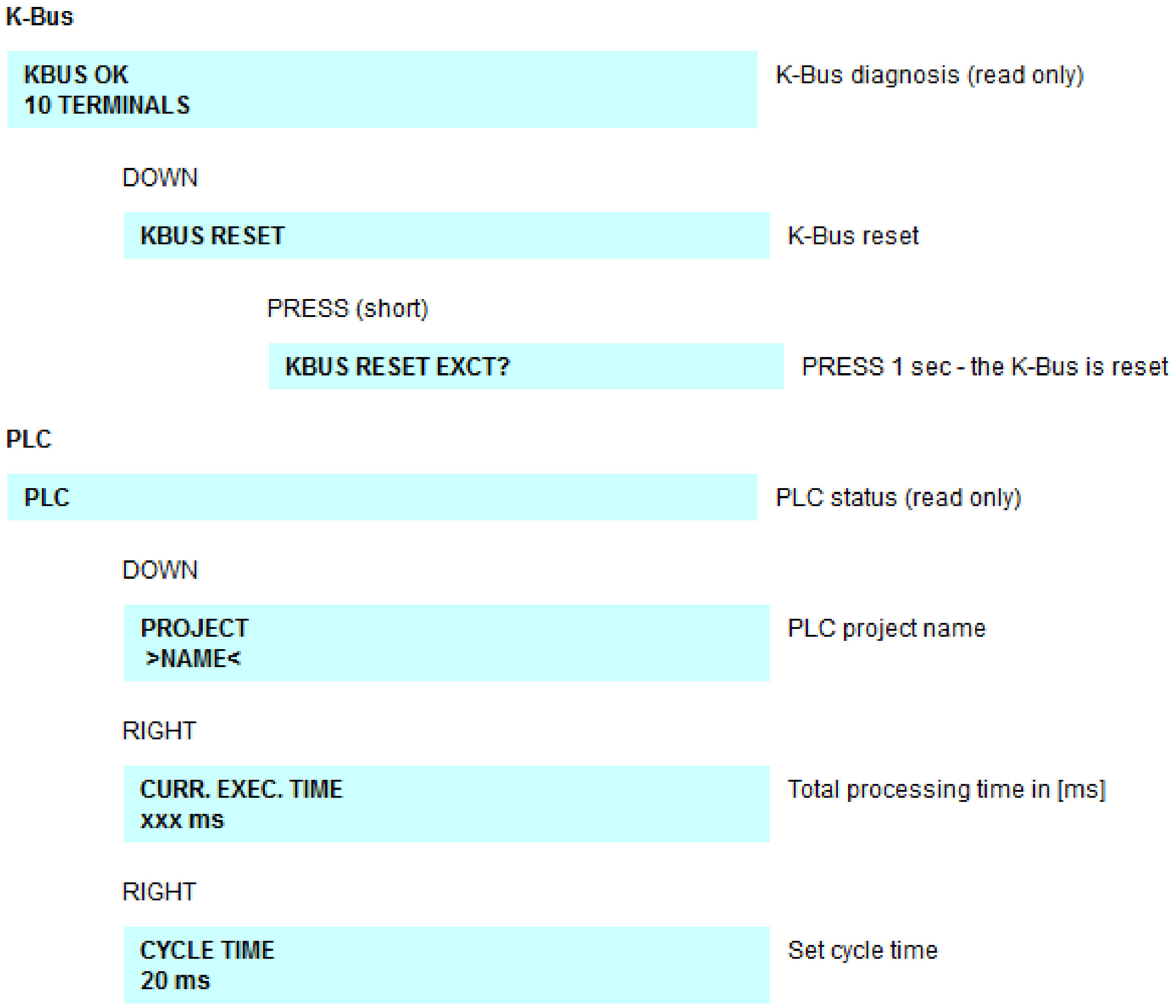
Baudrate **xxx**	Current baud rate (read only)

COM1

COM1	Serial interface COM2 (read only)

DOWN

Baudrate **xxx**	Current baud rate (read only)



K-Bus

KBUS OK
10 TERMINALS

K-Bus diagnosis (read only)

DOWN

KBUS RESET

K-Bus reset

PRESS (short)

KBUS RESET EXCT?

PRESS 1 sec - the K-Bus is reset

PLC

PLC

PLC status (read only)

DOWN

PROJECT
>NAME<

PLC project name

RIGHT

CURR. EXEC. TIME
xxx ms

Total processing time in [ms]

RIGHT

CYCLE TIME
20 ms

Set cycle time

4.2.13.2 Creating own menus

The display and the navigation switch can also be used for user-specific purposes, for example displaying diagnostic information or changing parameters. A simple example is provided that can be used and adapted to get you started.

Download (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207305483.zip):

4.2.14 Configuration software KS2000

Bus Terminal controllers of the BCxx50, BXxx20 and BXxx00 series cannot be parameterized and configured with the KS2000 configuration software. These devices must be configured with the TwinCAT System Manager.

The KS2000 configuration software offers configuration and diagnostic support for the Bus Terminals attached to the Bus Terminal Controller.
It is advisable to set the baud rate in the KS2000 configuration software and the BCxx50 BCxx20 and BXxx00 to 38400 baud (8 data bits, even, 1 stop bit).

COM1 - automatic baud rate detection

The COM 1 interface of the BXxx00 features automatic baud rate detection between 9.6 kbaud and 56.4 kbaud.

Required KS2000 version

Configuration or diagnostics of Bus Terminals at BXxx00 is supported from KS2000 version 4.3.14.

In some Bus Terminals (e.g. KL25xx, KL6811, KL6201, KL6401) the following parameters must be set in order to be able to use the configuration dialogs:

- A PLC project or boot project must be deactivated.
- The BX controller must be in its default configuration. Set the manufacturer’s setting or switch to Config Mode in the TwinCAT System Manager (blue TwinCAT icon).
- The BX controller must be in FreeRun mode. Activate it with the TwinCAT System Manager.

You can now log in with the KS2000 configuration software via ADS (port 100) or the serial cable and use the KS2000 dialogs in the Bus Terminals.

5 Programming

5.1 PLC features of the BX controllers

Description	Value
Data memory	256 kbyte
Program memory	256 kbyte minus task-configuration minus POUs during online change
Source code memory	256 kbyte
RETAIN	2 kbyte
INPUT	2 kbyte
OUTPUT	2 kbyte
FLAG	4 kbyte
Max. variable size	16 kbyte
Max. POUs	Limited by memory

5.2 TwinCAT PLC

The Beckhoff TwinCAT Software System turns any compatible PC into a real-time controller with a multi-PLC system, NC axis control, programming environment and operating station. The TwinCAT programming environment is also used for programming the BC/BX. If you have TwinCAT PLC (Windows NT4/2000/XP) installed, you can use the fieldbus connection or the serial port for downloading and debugging software.

TwinCAT I/O or TwinCAT PLC can also be used as the Ethernet Master (host), in order to exchange process data with the Bus Terminal Controller. TwinCAT provides you with the System Manager as a configuration tool, as well as the drivers and the ADS protocol.

Bus Terminal Controllers of the BCxx50, BCxx20 and BXxx00 series

These 2nd-generation Bus Terminal Controllers are configured with the TwinCAT System Manager and programmed with TwinCAT PLC Control. TwinCAT PLC must be installed for these couplers (Windows NT4, Windows 2000, Windows XP).

Programming and program transfer

- via the serial interface [▸ 142]
- via the fieldbus interface (only for Bus Terminal controllers for PROFIBUS, CANopen and Ethernet)

Online change

The Bus Terminal Controllers of the BX series and the BCxx50 support online change. This means that the PLC program is replaced with a new program without interrupting the program. The switch-over to the new program occurs after the task is completed. This means that two versions of the PLC program have to be stored. 512 kbyte are available, which therefore have to be divided by two, leaving 256 kbyte for the actual PLC program. In addition, several kbyte are required for task configuration etc. During an online change, dynamic data are stored in memory. Should a program approach the memory limit (program size greater than 240 kbyte), the online change may no longer work, even though the program may still be written to the BX after "Rebuild all".

When is online change not available?

Online change is not available under certain conditions,.

- Inserting of a new library
- Changing the task setting

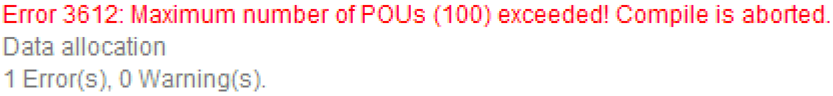
- "Rebuild all"
- Controller memory limit is almost reached (PLC program greater than 90%)

5.3 TwinCAT PLC - Error codes

Error type	Description
PLC compiler error	Maximum number of POUs (...) exceeded
PLC compiler error	Out of global data memory ...

Error POUs

For each function block one POU (process object unit) is created. 256 function blocks are available by default.



```
Error 3612: Maximum number of POUs (100) exceeded! Compile is aborted.
Data allocation
1 Error(s), 0 Warning(s).
```

Fig. 94: Maximum number of POUs exceeded

If libraries are integrated this value may be insufficient. In this case, the number of POUs should be increased.

To this end, open in PLC Control under Projects/Options...

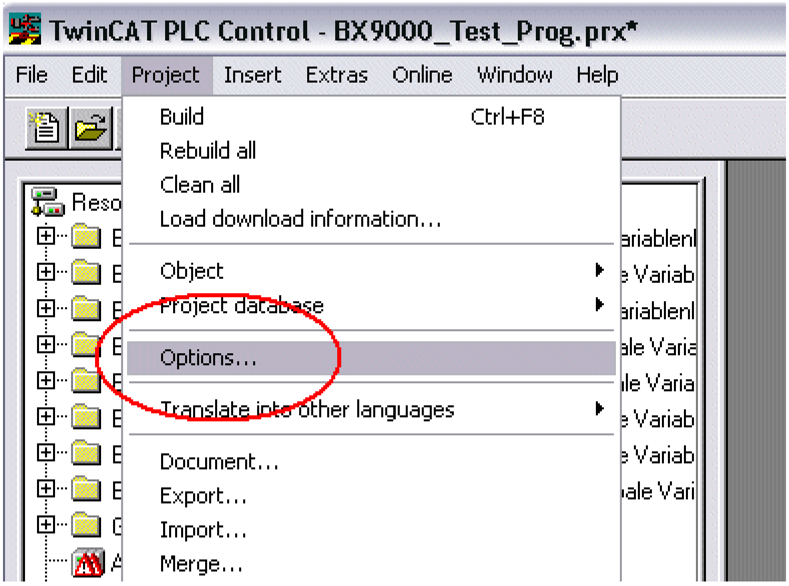
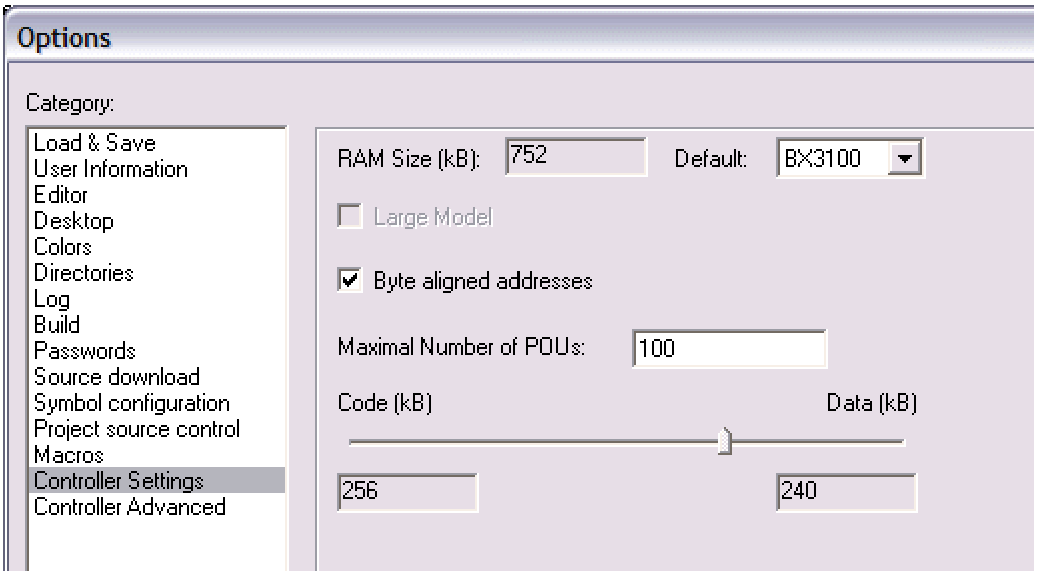


Fig. 95: Menu path Projects / Options / Controller Settings

...the controller settings.



Options

Category:

RAM Size (kB): 752 Default: BX3100

□ Large Model

☑ Byte aligned addresses

Maximal Number of POUs: 100

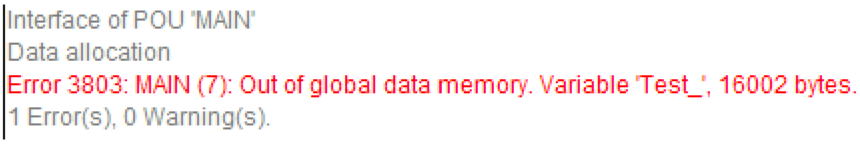
Code (kB) 256

Data (kB) 240

Fig. 96: Controller settings

Changing these settings will deactivate online changes.

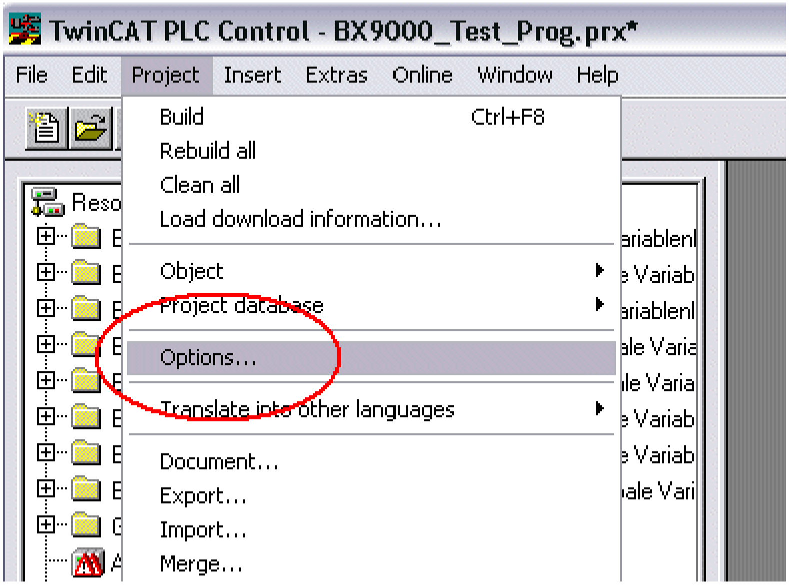
Global memory error



```
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).
```

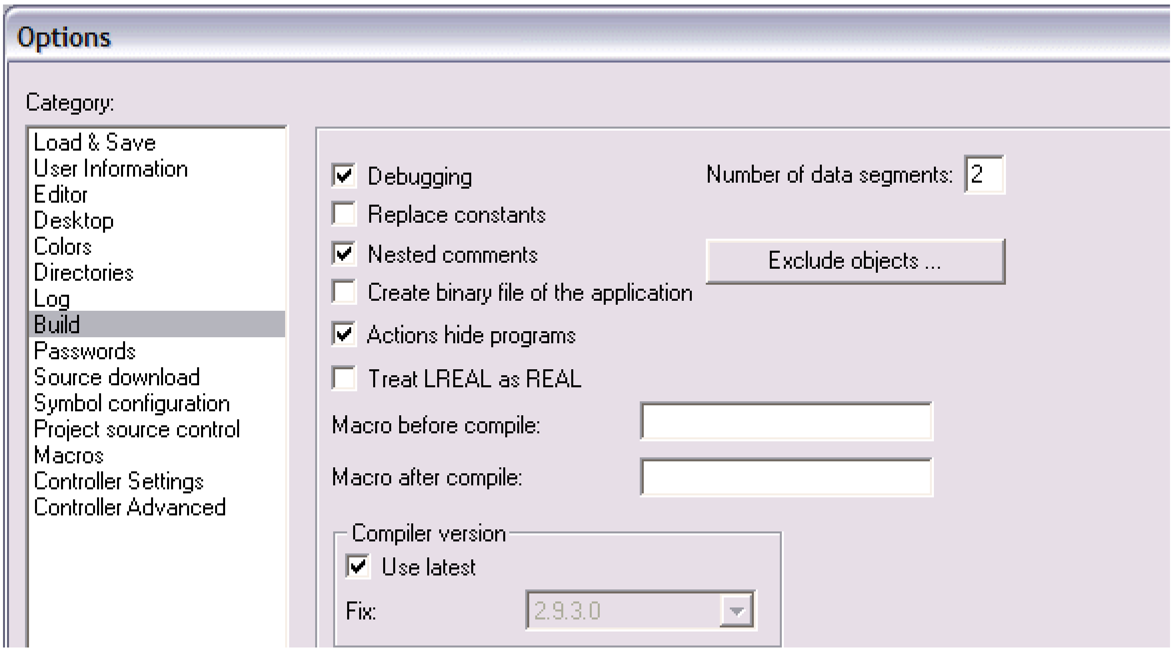
Fig. 97: Global memory insufficient

2 x 16 kbyte of data are available by default. If large data quantities are to be used, this range should be increased. A maximum of 14 data segments are possible for the BX.



TwinCAT PLC Control - BX9000_Test_Prog.prx*

Fig. 98: Menu path Projects / Options / Build



Options

Category:

☑ Debugging

Number of data segments: 2

□ Replace constants

☑ Nested comments

□ Create binary file of the application

☑ Actions hide programs

□ Treat LREAL as REAL

Macro before compile:

Macro after compile:

Compiler version

☑ Use latest

Fix: 2.9.3.0

Fig. 99: Build

5.4 Remanent data

2000 bytes of remanent data are available on the BC9191 and the BX controller. These data are declared as VAR RETAIN in PLC Control.

Example

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

Retain data are located between VAR RETAIN and END_VAR. These data are stored in a NOVRAM and are consistent across the whole 2 kbyte range. The RETAIN data are stored in the NOVRAM after each cycle. For 2 kbyte approx. 2 ms are required (for 1 kbyte approx. 1 ms). The variables can be configured locally or globally. Allocated variables (%MB, %QB, %IB) cannot be used as remanent data.

Do not use VAR_RETAIN in function blocks

VAR_RETAIN should not be used in function blocks. All FB data are copied into the retain memory. This leads to an unnecessary increase in cycle time, and the retain memory is filled with unnecessary data.

Do not use variables with address as remanent data

Variables that have been assigned an address (%MB, %QB, %IB) must not be used as remanent data.

Example for remanent data in the function block

This should be avoided, if possible, since all the data of a function block, in which even just a single remanent bit is found, are stored by default. A program sample can be found below.

Function block test (no program code required - in ST semicolon is sufficient)

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

MAIN program

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

```
fb_Test(Counter:=iCounter1);
```

5.5 Persistent data

The Bus Terminal Controller has 1000 bytes of persistent data available.
In contrast to the retain data, these are not deleted, even with a new project, a PLC reset or a new download.

In order to use the persistent data, these must first be activated once with a function block from the PLC.

Secondly, the variables should reside in the allocated flag area. Here you can choose where the persistent data reside.
4 kbytes of allocated flags are available, of which 1000 bytes can be declared as persistent data.

Example

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

The **Persistent_Data** function block can be used to specify the start address and the length (in bytes) from which the data are to be persistent.

The input variable *WriteOffset* is used to specify the byte offset of the flag area, *WriteSize* is used for the length in bytes.

The function block can be found in the TcSystemBX.lbx library. Should this not be available, it can be downloaded from this documentation (see Libraries [▶ 107]).

Example values

WriteOffset 1000
WriteSize 10

All data in the range %MB1000 - %MB1009 are then persistent. The variable type is irrelevant.

Like the retain data, the data are copied to the NOVRAM and are therefore writeable in each cycle.

Persistent data from firmware 1.17

Persistent data is supported for all BX controllers from firmware 1.17 or higher.

Parameters are valid immediately

The parameters only have to be written once, after which they are valid immediately. These data are stored permanently.
Activation of the factory setting deletes everything, including the persistent data.

Sample Program

Click on the link (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207307659.zip) to download a sample program from this documentation.

5.6 Allocated flags

4 kbyte of allocated flags are available. They can be used to assign different variable types to the same address, e.g. for converting strings to bytes. Data can also be placed here that can be read or written via ADS by the controller.

Allocated variables are not remanent data

For the Bus Terminal Controllers of the BX series and the BCxx50 the allocated variables are **not** saved as remanent data.

Reading/writing of allocated flags via ADS

The flags may also be read via the controller and ADS. In PROFIBUS, the DPV-1 services are used for this purpose, in CANopen SDO communication is used.
The AmsNetID can be obtained from the System Manager, or it can be displayed via the Bus Terminal Controller menu.
The PLC port number is 800.

Index group	Meaning	Index offset (value range)
0x4020	Flag (only BXxxx0)	0..4096

Example

BX program

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

TwinCAT PC/CX master program

```
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 Local process image in delivery state (default config)

The process image of the Bus Terminal Controller consists of input, output and flag area. In addition, there are unallocated data without fixed address. They are created without specifying an address. For these variable types the memory allocation is as follows:

- BCxx50 48 kbyte,
- BC9x20, BC9191 128 kbyte and
- BXxx00 256 kbyte.

The maximum size of a variable or structure (array) is 16 kbyte. For the allocated data 2048 bytes of input data and 2048 bytes of output data are available. The Bus Terminal Controller has 4 kbyte of memory allocated for the flag area.

In the delivery state (default configuration) of the BX/BCxx50, fixed addresses are allocated for all connected Bus Terminals. The data for Ethernet communication start from address offset 1000_{dec}. The length of the Ethernet data depends on how much data has been configured; on the BX9000 it has a maximum length of 1000 bytes.

Inputs	Outputs
Bus Terminal %IB0 ...	Bus Terminal %QB0 ...
Ethernet DATA (PLC variables) %IB1000 ...(Modbus TCP/ADS-TCP/ADS-UDP)	Ethernet DATA (PLC variables) %QB1000 ... (Modbus TCP/ADS-TCP/ADS-UDP)
... %IB2047 maximum	... %QB2047 maximum

Addressing of the connected Bus Terminals

The default setting is for all the connected Bus Terminals to be assigned to the local process image. Mapping within the Bus Terminal Controller is carried out according to the following rule:
First all the complex Bus Terminals, in the sequence they are physically inserted, followed by the digital Bus Terminals which are filled to a byte. The default mapping of the complex Bus Terminals is:

- complete evaluation
- Intel format
- Word alignment

Example structure

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

Table 1: Process image

Bus Terminal	Position	Input image	Output image	Size
KL1501	5	%IB0...%IB5	%QB0...%QB5	6 bytes
KL3002	6	%IB6...%IB13	%QB6...%QB13	8 bytes
KL4002	7	%IB14...%IB21	%QB14...%QB21	8 bytes
KL6001	8	%IB22...%IB29	%QB22...%QB29	6 bytes
KL1012	1	%IX30.0...%IX30.1	-	2-bit
KL1104	2	%IX30.1...%IX30.5	-	4-bit
KL2012	3	-	%QX30.0...%IX30.1	2-bit
KL2034	4	-	%QX30.2...%IX30.5	4-bit
KL9010	9	-	-	-

5.8 Mapping the Bus Terminals

The precise assignment of the byte-oriented Bus Terminals may be found in the configuration guide for the particular bus terminal. This documentation can be found on the Internet at http://www.beckhoff.de.

Byte oriented Bus Terminals	Bit oriented Bus Terminals
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 Creating a boot project

The following memory resources are available for generating the boot project

- approx. 250 kbyte flash on the Bus Terminal controllers of the BX series;
- approx. 48 kbyte flash on the Bus Terminal controllers of the BCxx50 series.

PLC Control

After logging into TwinCAT PLC Control, a boot project can be created.

- Opening a PLC project
- Selecting the target system (or selection the serial interface)
- Logging into the BX/BCxx50
- Creating a boot project (Online\Create boot project)

The PLC LED lights up green once a valid boot project is available on the BX/BCxx50.

In the Bus Terminal controllers of the BX series, the PLC LED flashes orange while boot project is created. The PLC LED lights up orange if no boot project is available on the BX.

Deleting a boot project

The boot project can be deleted from the Bus Terminal Controller. The following steps must be followed:

- Opening the project
- Logging into the Bus Terminal Controller
- Deleting the boot project (Online\Delete boot project)

The PLC LED lights up orange when the boot project is deleted.

Using the current project as boot project

After an online change the old project is still shown as boot project. To use the current project (after the online change) as the boot project, the boot project has to be recreated.

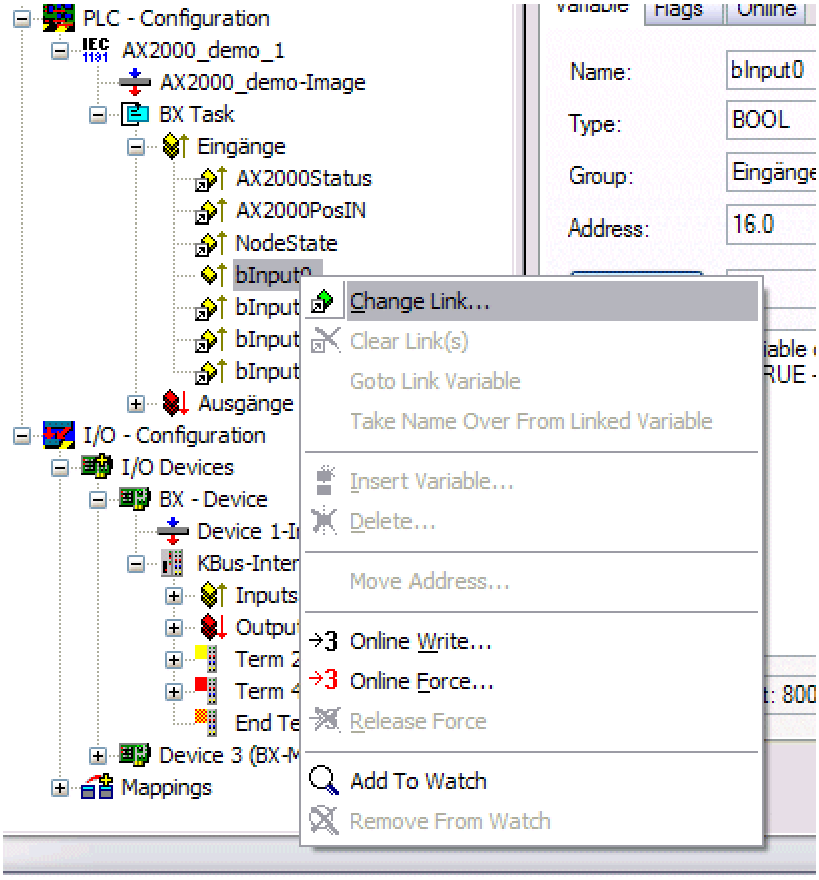
Bypassing the start of the boot project*

With the Bus Terminal controllers of the BX series, starting of the boot project during booting can be prevented by pressing the Navi button. This does not delete the boot project. The project is reloaded when the Bus Terminal Controller is rebooted.

* from version 0.85

5.10 Local process image in the TwinCAT configuration

The TwinCAT configuration (TwinCAT CONFIG) enables free mapping between fieldbus, K-bus and PLC variables. Variables can be linked independent of their address via the System Manager.



Name: bInput0
Type: BOOL
Group: Eingänge
Address: 16.0

Fig. 100: Changing variable links

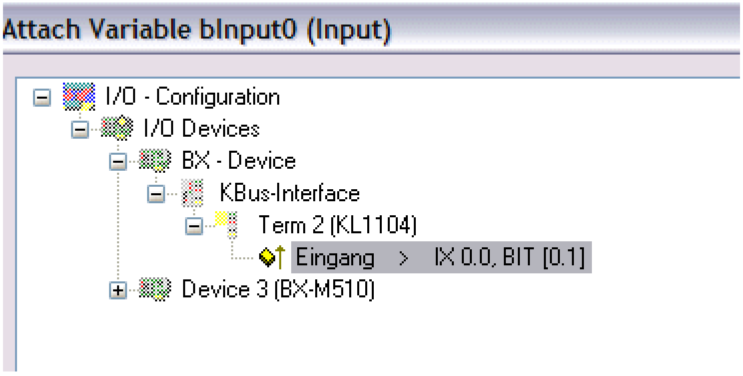

Attach Variable bInput0 (Input)

Fig. 101: Linking a variable with an input

In the default configuration all Bus Terminals are assigned fixed addresses. If a Bus Terminal is inserted, the whole address range may be shifted. The TwinCAT configuration enables allocated variables to be linked to a Bus Terminal, as required. This is parameterized in the System Manager, and the configuration is then downloaded to the Bus Terminal Controller (see TwinCAT configuration [▸ 32]). It is also possible to upload an existing TwinCAT configuration.

5.11 Communication between TwinCAT and BX/BCxx50

For transferring data from TwinCAT to the Bus Terminal Controller, it makes sense to organize the data in a structure. Please note the following to account for the differences in data management on the two systems.

- If two different data types are sent in sequence (e.g. byte and INT), the following variable is set to the next even address offset
- Boolean variables should never be allocated individually within a structure, since they would invariably occupy 1 byte. Boolean expressions should always be masked in a byte or word.

Example 1: A structure on the BX/BCxx50 and on the PC

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

Due to the fact that another variable type (INT) follows the first byte, in the BX/BCxx50 it was assigned the next free even address. In order to achieve the same data structure on both systems, a dummy byte has to be inserted in the PC project (see example 2).

Example 2: A structure on the BX/BCxx50 and on the PC with the same memory allocation

Variable	BX/BCxx50 memory	PC memory (TwinCAT)
Byte	%..B0	%..B0
Byte (dummy)	%..B1 (not necessarily required, since the system deals with this itself if the variable does not exist)	%..B1
INT (1)	%..B2	%..B2
INT (2)	%..B4	%..B4

Data structure

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

Creating a variable structure

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

Small programming example

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

Do not use real values in a mixed data structure

A mixed data structure should not contain real values. If this is nevertheless the case, the high and low words must be swapped in the BX/BCxx50 or in the TwinCAT master project. It is better to use an array of Real values or to transfer the Real values individually.

Larger fieldbus data blocks

You can transfer larger fieldbus data blocks, in order to have a reserve for your structure. Disadvantage: These reserves are then transferred with each fieldbus telegram, resulting in overload of the fieldbus communication.

5.12 Up- and downloading of programs

The Bus Terminal Controller has a memory for the source code. It can be used for storing the program, the task configuration, and the libraries. Should the memory be insufficient, the source code may be stored without task configuration and libraries. This takes up significant less memory space!

General settings

The timing of the source code download to the target system can be specified via Edit/Options. Open the options menu.

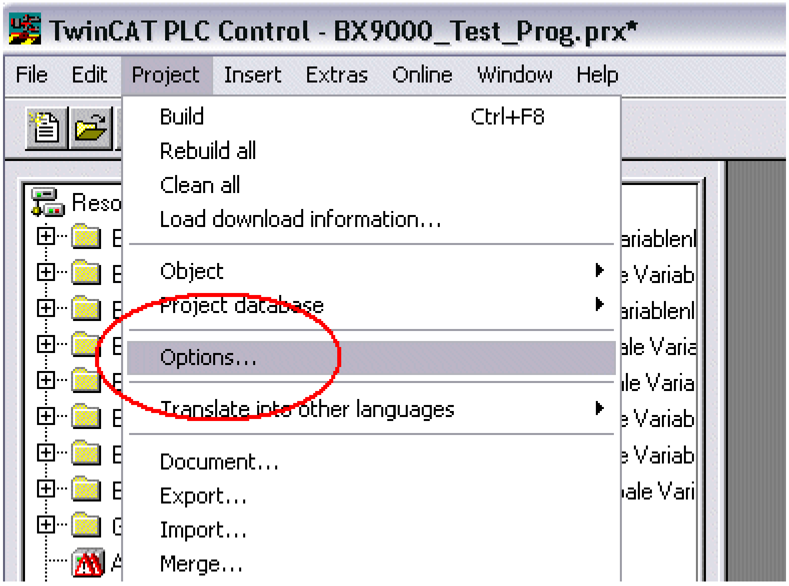
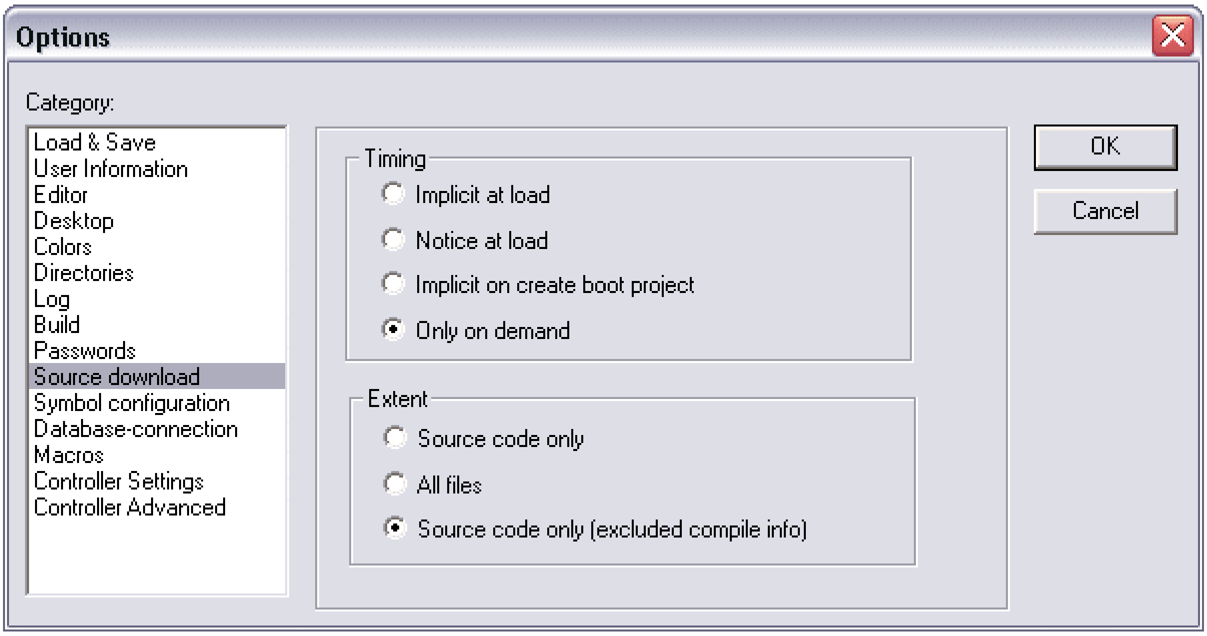

TwinCAT PLC Control - BX9000_Test_Prog.prx*
File Edit Project Insert Extras Online Window Help
Build Ctrl+F8
Rebuild all
Clean all
Load download information...
Object
Project database
Options...
Translate into other languages
Document...
Export...
Import...
Merge...
Reso

Fig. 102: Opening the options menu

Select Source Download.



Timing
○ Implicit at load
○ Notice at load
○ Implicit on create boot project
● Only on demand

Extent
○ Source code only
○ All files
● Source code only (excluded compile info)

Fig. 103: Selecting Source Download

Here you can set which parts of the source code are to be downloaded to the Bus Terminal Controller, and when.

Source code only: the prx file with information on the online change is transferred. Login via online change is possible (the PLC does not stop).
All files: as *Source code only*, plus all required libraries.
Source code only (compile info excluded): only the prx file is transferred. Login is only possible when the PLC stops.

Which option you can use depends on the size of your projects.

Downloading a program

The source code can be transferred to the target system on request. This requires the user to be logged in with his program. Under Online/Source code download the program code can now be transferred to the Bus Terminal Controller.

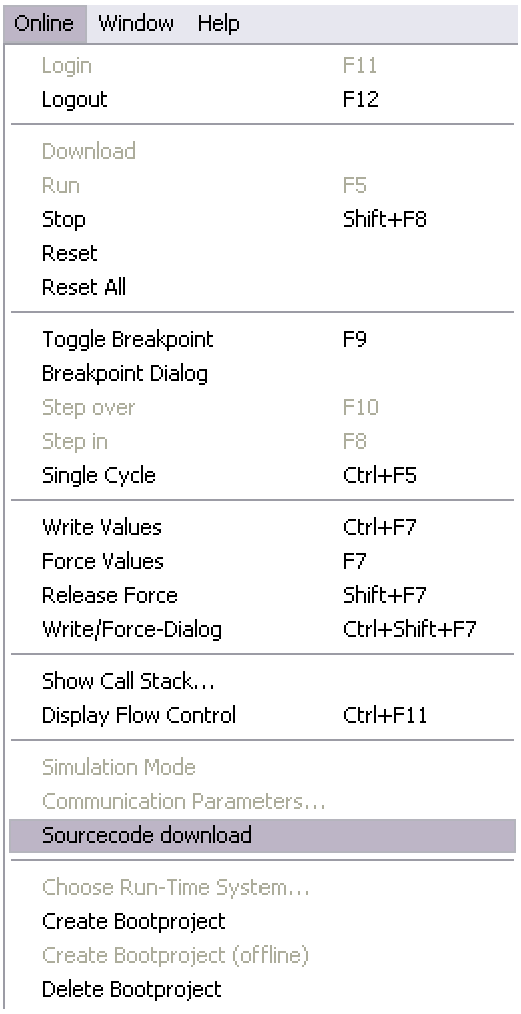
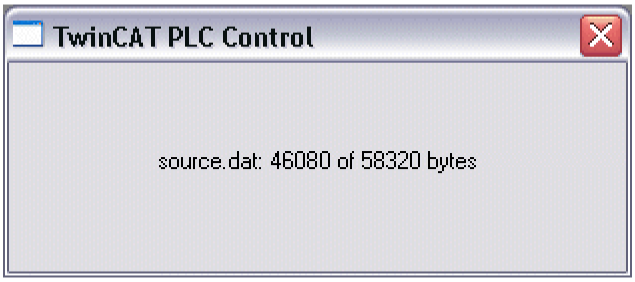


Fig. 104: Downloading the program code

After a short delay, a window will open that indicates the download progress.

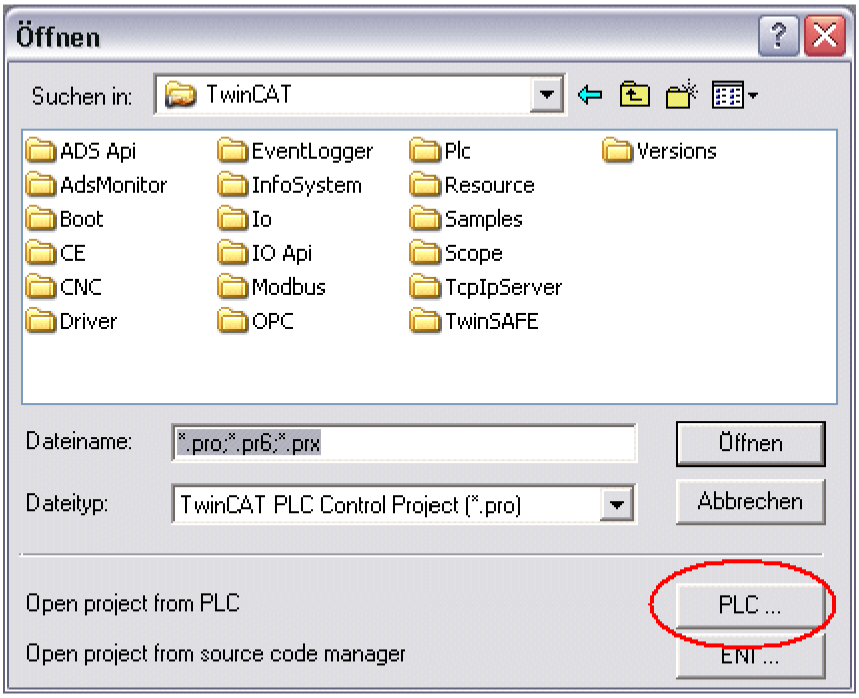


source.dat: 46080 of 58320 bytes

Fig. 105: Download progress

Uploading a program

For uploading the program code again, open a new file in PLC Control. Then click on the PLC button.



Öffnen

Suchen in: TwinCAT

ADS Api, AdsMonitor, Boot, CE, CNC, Driver, EventLogger, InfoSystem, Io, IO Api, Modbus, OPC, Plc, Resource, Samples, Scope, TcpIpServer, TwinSAFE, Versions

Dateiname: *.pro;*.pr6;*.prx

Dateityp: TwinCAT PLC Control Project (*.pro)

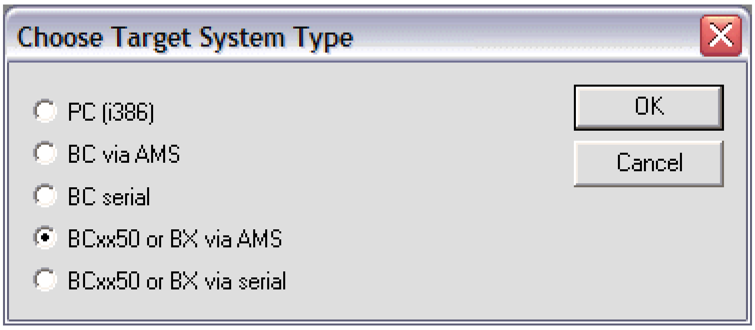
Open project from PLC

Open project from source code manager

Fig. 106: Uploading a program

Select the data transfer route:

- *BCxx50 or BX via AMS*, if you are connected to the Bus Terminal Controller via the fieldbus, or
- *BCxx50 or BX via serial*, if you are connected to the Bus Terminal Controller via the serial interface.

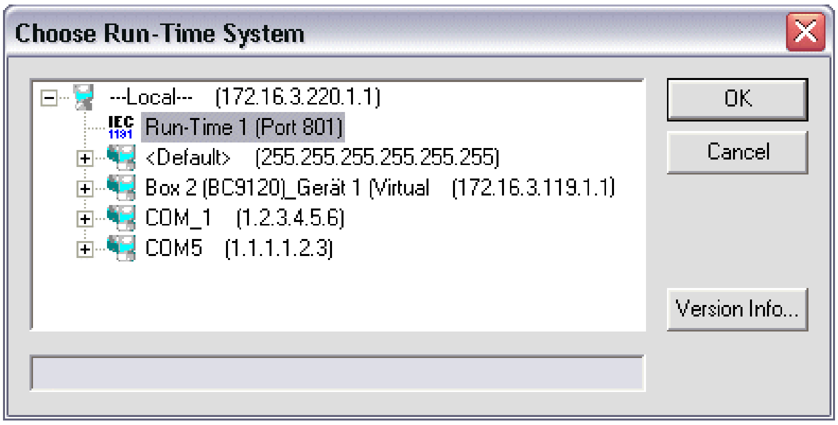


Choose Target System Type

○ PC (i386)
○ BC via AMS
○ BC serial
● BCxx50 or BX via AMS
○ BCxx50 or BX via serial

Fig. 107: Selecting the data transfer route

Then select the device and confirm with OK.



Choose Run-Time System

```
⊟ --Local--   (172.16.3.220.1.1)
     Run-Time 1 (Port 801)
  ⊞ <Default>   (255.255.255.255.255.255)
  ⊞ Box 2 (BC9120)_Gerät 1 (Virtual   (172.16.3.119.1.1)
  ⊞ COM_1   (1.2.3.4.5.6)
  ⊞ COM5   (1.1.1.1.2.3)
```

Fig. 108: Selecting the device

The source code will now be uploaded.

Password

You can protect your project with a password (in PLC Control Project/Options/Passwords).

5.13 Libraries

5.13.1 Libraries overview

Various libraries are available for the Bus Terminal Controllers (Bus Coupler with PLC functionality: BXxxxx) (see Beckhoff Information System).

Download

To download the libraries click on the link. Please copy the libraries to directory TwinCAT\PLC\LIB.

- Standard (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207309835.zip)

  💾
- TcSystemBX (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207312011.zip)

  💾 (the TcSystemBX requires the TcBaseBX library)
- TcBaseBX (Download)

  💾 (TcDisplayBX, TcNaciSwitchBX and TcDebugBX are now included here)
- TcComPortBX (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207314187.zip)

  💾
- ChrAscBX.lbx (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207316363.zip)

  💾

Use the library that matches the firmware

The latest firmware requires the latest library. If you update your BX Controller, please also change the libraries.
Copy these libraries to the LIB directory, remove these libraries from your project and add them again.

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
ADSRDWRT	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_DispWrite [▶ 120]	31.03.03	0.28	0.01	1.00	0.01	1.00

Diagnosis	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
FirmwareVersionString	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
F_ReadDebugTimer [▶ 119]	08.08.03	0.59	0.14	1.00	0.02	1.00
F_StartDebugTimer [▶ 119]	08.08.03	0.59	0.14	1.00	0.02	1.00

NaviSwitch	Version	Firmware				
		BX3100	BX5100	BX5200	BX8000	BX9000
All function blocks [▶ 119]	10.10.03	0.64	0.14	1.00	0.02	1.00

TcComPortBX

Com FBs	Version	Firmware				
		BX3100	BX5100	BX5200	BX8000	BX9000
All function blocks [▶ 125]	20.08.03	0.60	0.02	1.00	0.01	1.00

5.13.2 TcBaseBX

5.13.2.1 System task information

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

System flags are implicitly declared variables. Using the Input Assistant, a variable SystemTaskInfoArr can be found under system variables. This variable is a field with four structures of type SYTEMTASKINFOTYPE [▶ 110]. The structure definition can be found in the system library. The index in this field is the task ID.

Development environment	Target platform	PLC libraries to be linked
TwinCAT v2.9.0	BX Controller	TcBaseBX.lbx

5.13.2.2 System task information 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
```

Legend

active: This variable indicates whether the task is active.
taskName: the task name.
firstCycle: During the first PLC task cycle, this variable has the value: TRUE.
cycleTimeExceeded: this variable indicates whether the set task cycle time was exceeded.
cycleTime: set task cycle time in multiples of 100 ns.
lastExecTime: cycle time required for the last cycle in multiples of 100 ns.
priority: set task priority.
cycleCount: cycle counter.

Development environment	Target platform	PLC libraries to be linked
TwinCAT v2.9.0	BX Controller	TcBaseBX.lbx

5.13.2.3 System info

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

System flags are implicitly declared variables. Using the Input Assistant, a variable Systeminfo can be found under system variables. The type SYSTEMINFOTYPE [▶ 110]is declared in the system library. For accessing the variable, the system library has to be integrated in the project.

Development environment	Target platform	PLC libraries to be linked
TwinCAT v2.9.0	BX Controller	TcBaseBX.lbx

5.13.2.4 System information type

```
TYPE SYSTEMINFOTYPE
STRUCT
    runTimeNo            :       BYTE;
    projectName          :       STRING(32);
    numberOfTasks       :       BYTE;
    onlineChangeCount   :       UINT;
    bootDataFlags       :       BYTE;
    systemStateFlags    :       WORD;
END_STRUCT
END_TYPE
```

Legend

runTimeNo: specifies the number of the runtime system (1).
projectName: project name as STRING.
numberOfTasks: number of tasks contained in the runtime system (max. 2).
onlineChangeCount: number of online changes since the last complete download.
bootDataFlags: Reserved
systemStateFlags: Reserved.

Development environment	Target platform	PLC libraries to be linked
TwinCAT v2.9.0	BX Controller	TcBaseBX.lbx

5.13.2.5 ADS

5.13.2.5.1 Local ADS Port Numbers - Overview

Port number	Description
100 [▶ 111]$_{dec}$	Reading and writing of registers and tables from the coupler and the complex Bus Terminals
150 [▶ 85]$_{dec}$	Reading and writing of RTC (real-time clock)
153 [▶ 69]$_{dec}$	SSB - reading of the emergency message
800 [▶ 111]$_{dec}$	Local process image of the PLC, see also port 801
801 [▶ 111]$_{dec}$	Local process image of the PLC, see also port 800
0x1000 + node ID [▶ 65]	SSB - SDO communication with CANopen node (slave number)

5.13.2.5.2 ADS services

Local Process Image PLC Task 1 Port 800/801

Data can be read from and written to the local process image. If it is necessary for outputs to be written, it is important to ensure that they are not used by the local PLC, because the local controller will overwrite these values. The data are not associated with a watchdog, and therefore must not be used for outputs that would have to be switched off in the event of a fault.

Index group	Meaning	Index offset (value range)
0xF020	Inputs	0...2047
0xF021	Bit inputs	0...16376
0xF030	Outputs	0...2047
0xF031	Bit outputs	0...16376
0x4020	Flags	0...4095
0x4021	Flag bit	0...32760

ADS services

AdsServerAdsState

Data type (read only)	Meaning
String	Start - the local PLC is running Start - the local PLC is stopped

AdsServerDeviceState

Data type (read only)	Meaning
INT	0 - Start - the local PLC is running 1 - Stop - the local PLC is stopped

AdsServerType

Data type (read only)	Meaning
String	BX PLC Server

ADSWriteControl

Data type (write only)	Meaning
NetID	Net ID of the Ethernet Controller*
Port	800
ADSSTATE	5 - RUN / 6 - STOP
DEVSTATE	0
LEN	0
SRCADDR	0
WRITE	rising edge starts the function block
TMOUT	example: T#1000 ms

* BC9050, BC9020, BC9120, BX9000

Register access port 100

On the Bus Terminal Controllers of the BX series, and on the BCxx50/xx20, the ADS port number for register communication is fixed at 100.

Index group	Index offset (value range)		Meaning
	Hi-Word	Low Word	
0 [READ ONLY]	0...127	0...255	Registers in the Bus Coupler Hi-Word, table number of the Bus Coupler Lo-Word, register number of the table
1...255	0...3	1...255	Register of the Bus Terminals Hi-Word, channel number Lo-Word, register number of the Bus Terminal

Minimum timeout

For reading the registers, ensure that the timeout for the ADS function block is set to more than one second.

Setting the password

When writing to the registers, the password has to be set (see the documentation for the particular Bus Terminal).

5.13.2.5.3 Deactivating the LED for cycle time exceeding

The BX controller monitors the set task cycle time. If it is exceeded, the cycleTimeExceeded [▸ 110] bit and the red *PLC* LED are set. In some applications the value may be exceeded for short periods, which is tolerable. This may be the case when many data are received in serial communication, for example. In order to avoid flickering of the red *PLC* LED, it can be switched off via ADSWRITE.

Structure of the ADSWRITE command

This ADSWRITE command enables the red *PLC* LED of the BX controller to be deactivated.

Input parameters	Description
NETID	local NetId of BX
Port number	800
IDXGRP	16#0000_4080
IDXOFFS	0
LEN	1 bytes
SRCADDR	Pointer on 1 bytes 0: red LED ON 1: red LED OFF

5.13.2.6 COM Port

5.13.2.6.1 COM port - overview

The library includes function blocks that enable data exchange between the **BXxxxx** Bus Controller and a remote partner. The maximum COM buffer is 512 bytes for both directions.

Function blocks

Name	Description
FB_ComPortOpen [▸ 115]	Opens a serial connection to a partner.
FB_ComPortClose [▸ 115]	Closes a serial connection to a partner.

Functions

Name	Description
F_ComPortRead [▸ 114]	Reading data from the COM buffer
F_ComPortWrite [▸ 114]	Writing data into the COM buffer

Supported baud rates

Baud rate [baud]	COM 1	COM 2
300	NO	YES
600	NO	YES
1200	NO	YES
2400	NO	YES
4800	NO	YES
9600	YES	YES
19200	YES	YES
38400	YES	YES
57600	YES	YES
115200	NO	YES

Further helpful function blocks can be found in TcComPortBX.lbx [▶ 124].

- Function block for using ComLib, ModbusRTU etc. for the BX COM ports
- Function block for communication with the BK8x00 Bus Couplers
- Function block for emulation of a BK8x00 slave

5.13.2.6.2 COM port functions

COM Port Read

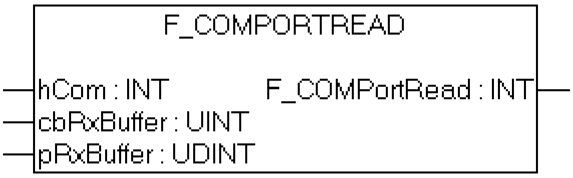


Fig. 109: Function block F_COMPORTREAD

FUNCTION F_COMPORTREAD

VAR_INPUT

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

Legend

hCom: is connected with the iHandle of FB_COMPORTOPEN

cbRxBuffer: maximum length of data that can be read.

pRxBuffer: pointer to data to be written with the COM buffer content

Return value	Meaning
> 0	Number of bytes that is to be copied from the COM buffer into the PLC.
0x8000	Memory overflow

COM Port Write

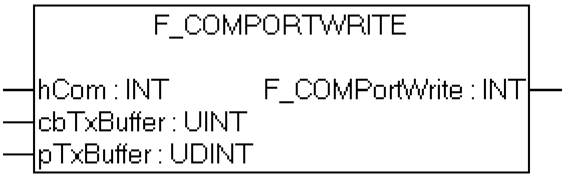


Fig. 110: Function block F_COMPORTWRITE

FUNCTION F_COMPORTWRITE

VAR_INPUT

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

Legend

hCom: is connected with the iHandle of FB_COMPORTOPEN

cbTxBuffer: Number of data bytes that were copied into the COM buffer.

pTxBuffer: Pointer to the data from which the COM buffer is to be filled.

Return value	Meaning
> 0	Number of bytes that is to be copied into the COM buffer from the PLC
0x8000	Memory overflow

5.13.2.6.3 COM port function block

COM Port Open

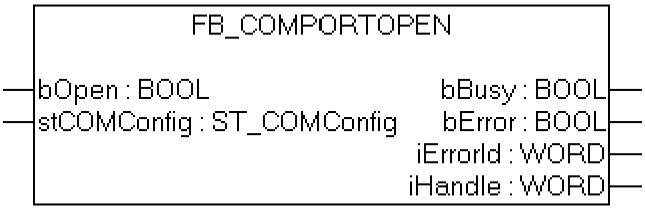


Fig. 111: Function block FB_COMPORTOPEN

FUNCTION_BLOCK FB_COMPORTOPEN

VAR_INPUT

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

Legend

bOpen: rising edge starts the function block
stComConfig [▶ 116]: COM interface data structure

VAR_OUTPUT

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

Legend

bBusy: The function block is active as long it is TRUE.
bErr: Error bit.
iErrId: Error number.
iHandle: pointer transfer.

Return parameter iErrId	Meaning
0	No error
-1, 0xFFFF	Incorrect COM port
-2, 0xFFFE	Incorrect or unsupported baud rate. Check the parameter stComConfig.BaudRate.
-3, 0xFFFD	Incorrect data format. Check the parameter stComConfig.
-4, 0xFFFC	Incorrect initialization of the COM interface
-5, 0xFFFB	Unsupported instance
-6, 0xFFFA	Incorrect size of the RX buffer
-7, 0xFFF9	Incorrect size of the TX buffer
-8, 0xFFF8	COM port is blocked

COM Port Close

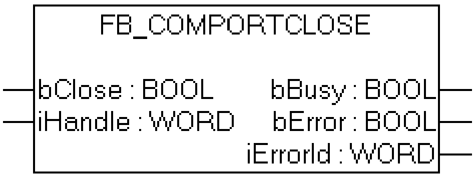


Fig. 112: Function block FB_COMPORTCLOSE

FUNCTION_BLOCK FB_COMPORTCLOSE

VAR_INPUT

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

Legend

bClose: rising edge starts the function block
iHandle: pointer transfer of FB_COMPORTOPEN.

VAR_OUTPUT

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

Legend

bBusy: The function block is active as long it is TRUE.
bErr: Error bit.
iErrId: Error number.

Return parameter iErrId	Meaning
0	No error
> 0	error number (#not documented#)

5.13.2.6.4 ComConfig data structure

The settings for the serial interfaces of the BX are transferred with the following data structure.

```
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
```

Legend

cbRxBufferLen: Has no purpose (was retained for compatibility reasons)
cbTxBufferLen: Has no purpose (was retained for compatibility reasons)
dwMode: data mode COM 1 only "0" - COM 2 RS232 "0" and RS485 "1"
BaudRate: Baud rate
eCommPort: Com Port COM1/COM2
eDataBits: number of data bits SEVEN_DATABITS/EIGHT_DATABITS
eParity: EVEN/ODD/NONE
eStoppBits: Number of stop bits ONE_STOPPBIT/TWO_STOPPBITS

5.13.2.6.5 Example

ST sample program

Download (https://infosys.beckhoff.com/content/1033/bx5200/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 debugging function

These functions can be used for measuring command execution times in a PLC project. The unit is a tick. One tick corresponds to 5.12 µs.

Start Debug Timer function

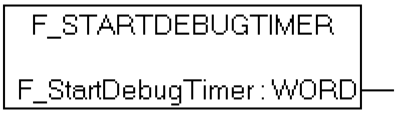


Fig. 113: Function block F_STARTDEBUGTIMER

Calling this function starts the timer. The return value is "0".

Read Debug Timer function

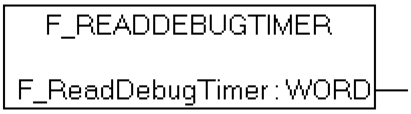


Fig. 114: Function block F_READDEBUGTIMER

This function reads the timer value. The return value has to be multiplied with 5.12 µs.

Example

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

Program

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

5.13.2.8 Navigation switch

5.13.2.8.1 FUN GetNavSwitch

This function block enables reading of the navigation switch.

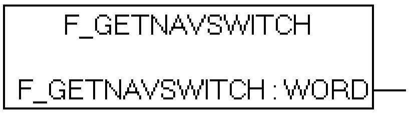


Fig. 115: Function block F_GETNAVSWITCH

VAR_Output

```
F_GETNAVSWITCH    :WORD;
```

Legend

F_GETNAVSWITCH: Switch data

WORD description

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

If bit 15 is set, you are in the BX controller menu. This bit is set for as long as the user remains in the BX3100 menu. On exit of the menu, the navigation switch is immediately released for the PLC, i.e. pressing of the Press button is still visible in the program. Please take account of this in your application. For example, the switch should only be evaluated after a short delay by starting a timer with falling edge from bit 15.

Download sample ST program https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207322891.zip

5.13.2.9 Display

5.13.2.9.1 FB DISPWRITE

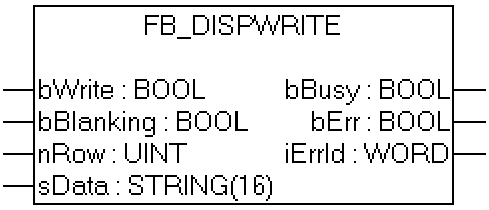


Fig. 116: Function block FB_DISPWRITE

VAR_INPUT

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

Legend

bWrite: rising edge starts the function block

bBlanking: FALSE backlight on, TRUE backlight off, default is on (supported in all BX controllers from FW 1.15).

nRow: row in display 1 or 2.

sData: displayed character string

VAR_OUTPUT

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

Legend

bBusy: The function block is active as long it is TRUE.

bErr: Error bit.

iErrId: Error number.

Return parameter	Meaning
0	No error
> 0	Error number

ST sample program

Download https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207325067.zip

```
PROGRAM MAIN
VAR
    fb_DispWrite1: FB_DispWrite;

i:                 INT;
    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 of ASCII table

Example for the "&" sign (see row 1 column 7): $00100110_{bin} = 38_{dec} = 26_{hex}$. In the PLC values this corresponds to '$26' (string.)

https://infosys.beckhoff.com/content/1033/bx5200/Resources/pdf/3207327243.pdf

5.13.3 TcSystemBX

5.13.3.1 Real-time clock - example

The BX Controller features a real-time clock. The current time can be read via a function block. The following example will illustrate this.

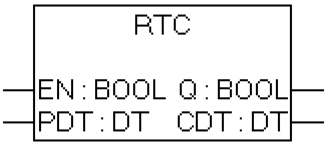


Fig. 117: Function block RTC

FUNCTION_BLOCK RTC

VAR_INPUT

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

Legend

EN: Rising edge sets the time to the value available at the PDT input.

PDT: Date and time to be set.

VAR_OUTPUT

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

Legend

CDT: Current time.

Required libraries:

- TcSystemBX.lb6
- TcBaseBX.lb6

Download sample ST program (https://infosys.beckhoff.com/content/1033/bx5200/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
```

Do not call the RTC function block in every PLC cycle

Calling the RTC block increases the cycle time by approx. 5 ms, due to the data conversion into a TIME AND DATE variable. The function block should therefore not be called during each PLC cycle. Alternatively, you can read the time via an ADS function block. The ADS function block returns the date and the time as WORD variables.
Example 19:30 hrs - hour: 19 / minute: 30

5.13.3.2 Loading and storing of recipes

The function block fb_ReadWriteFile enables data (up to 16,000 bytes) to be stored permanently in the flash memory of the BX controller. A new program or a project reset does not affect the content of this memory. This function block is not suitable for sustained and continuous use. A maximum of 10000 write cycles are permitted. There is no limit on read operations.

Application: Saving of recipes or settings that only change rarely or not at all, for example controller parameters.

Note the following during writing of data

- The voltage must not be interrupted during writing. It is therefore advisable to initiate writing via an operating panel or the navigation keys or simply via a digital input, in order to ensure that the BX Controller is not switched off during writing. Automatic writing is not recommended, since uninterrupted voltage supply during writing cannot be guaranteed.
- Writing of data takes approx. two seconds, irrespective of the number of data that are written.
- The data are lost if the BX controller is switched off during the write operation.
- Only one instance of this function block is permitted.

Function block fb_ReadWriteFile

Function block for reading and writing of recipes

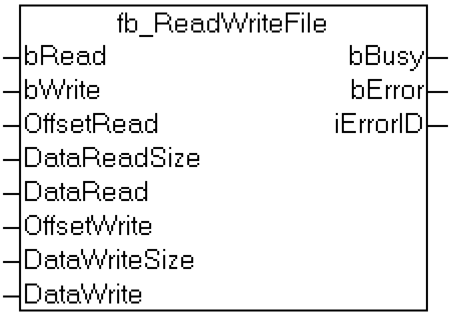


Fig. 118: Function block fb_ReadWriteFile

VAR_INPUT

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

Legend

bRead: A rising edge triggers reading of the function block (bWrite must be FALSE)
bWrite: A rising edge triggers writing of the function block (bRead must be FALSE)
OffsetRead: Offset in the memory 16,000 bytes max.
DataReadSize: Size of data to be read in bytes (16,000 bytes max.)
DataRead: The pointer should be pointed to the data via ADR
OffsetWrite: Offset in the memory 16,000 bytes max.
DataWriteSize: Size of data to be written in bytes (16,000 bytes max.)
DataWrite: The pointer should be pointed to the data via ADR

VAR_OUTPUT

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

Legend

bBusy: Indicates that the function block is still active
bError: Function block error
bErrorId: Error number

Return parameter iErrorId	Meaning
0	No error
1_{dec}	READ: Data offset and data length more than 16,000 bytes
2_{dec}	WRITE: Data offset and data length more than 16,000 bytes
0x31440708	CRC error in the data memory
0x31470708	Writing of data is not yet complete

Required libraries:

- TcSystemBX.lb6
- TcBaseBX.lb6

Download sample ST program (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207331595.zip)

5.13.4 TcComPortBX

5.13.4.1 TcComPortBX overview

Required libraries:

- TcBaseBX
- TcSystemBX

Overview

Name	Description
fb_BX_BK8x00 Master [▶ 125]	BK8x00 COM port function block, communication with Bus Coupler BK8x00 or BC8x00
fb_BX_BK8x00 Slave [▶ 125]	BK8x00 COM port function block, communication with PC. A BK8x00 is simulated.

Name	Description
FB_BX_COM_5 [▶ 128]	Function block for emulating a KL60x1(if COMlib.lb6 or ModbusRTU.lb6 is used).
FB_BX_COM_64 [▶ 129]	Function block for emulating a PC interface (if COMlib.lib or ModbusRTU.lib is used).
FB_BX_COM_64ex [▶ 129]	Function block for emulating a PC interface (if COMlib.lib or ModbusRTU.lib is used). Here the COM interface can be closed during operation.

5.13.4.2 BK8x00 - FB COM-Port

This function block can be used to connect (via the serial interface of the BXxxxx) the BK8000 serial Bus Coupler with RS485 and BK8100 with RS232 connection. The maximum baud rate is 38400 baud.

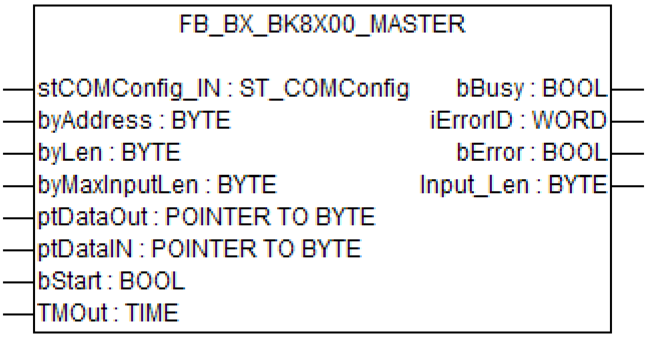


Fig. 119: Function block FB_BX_BK8X00_master

VAR_INPUT

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

Legend

stComConfig: Structure for selecting the COM parameters
byAddress: BX8x00 address 1-98 (0 and 99 are reserved)
byLen: data length in [BYTES] (only even numbers are permitted, i.e. 0, 2, 4, ...)
byMaxInputLen: is connected with SIZEOF and the variable that is linked with ptDataIN
ptDataOut: is connected with ADR and data out
ptDataIn: is connected with ADR and data in
bStart: rising edge starts the function block
TMOut: delay until process is aborted

VAR_OUTPUT

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

Legend

bBusy: The function block is active as long it is TRUE.
bError: error bit
iErrorID: Error number
Input_Len: number of data that were received

Return parameter iErrorId	Meaning
0	No error
100_{dec}	Error during opening of the COM port
101_{dec}	Error during sending of data
102_{dec}	Watchdog error, no response from the slave within the WD time
105_{dec}	The input buffer is too small
200_{dec}	CRC error
$0x80xx_{hex}$	Bus Coupler error xx status byte of the Bus Coupler (see BX8x00 documentation)

Hardware

RS 232 communication PIN assignment

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

RS485 communication PIN assignment

FB settings: When using the RS485 connection it is important that the stCOMConfig variable is set to 1 and that the COM2 interface is selected.

BX COM2 RS485	BK8000
1	3
6	8

ST example program

Download (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207333771.zip)

Required material:

- BX3100 + Bus Terminal
- BK8100, KL1xx8, KL2xx8, KL9010
- Serial cable, PIN assignment: see sample program

5.13.4.3 BK8x00 - FB Slave COM-Port

With this function block, the PC (TwinCAT or KS8000) can be connected with the BXxxxx via the serial interface. The PC acts as the serial master, and the BXxxxx emulates a BK8x00 with the aid of the function block.

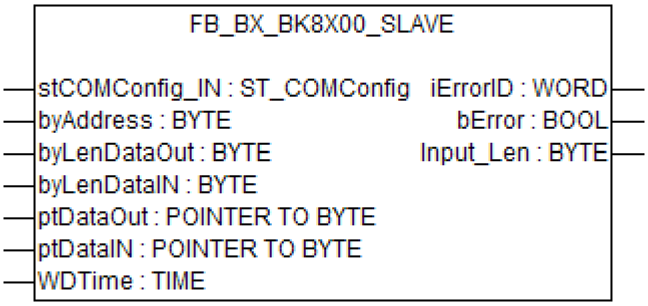


Fig. 120: Function block FB_BX_BK8X00_SLAVE

VAR_INPUT

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

Legend

stComConfig: Structure for selecting the COM parameters
byAddress: BX8x00 address 1-98 (0 and 99 are reserved)
byLenDataOut: data length in [BYTES] (only even numbers are permitted, i.e. 0, 2, 4, ...)
byLenDataIn: data length in [BYTES] (only even numbers are permitted, i.e. 0, 2, 4, ...)
ptDataOut: is connected with ADR and data out
ptDataIn: is connected with ADR and data in
WDTime: error message if no new data are received within the watchdog time (0 ms disable WD)

VAR_OUTPUT

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

Legend

bError: error bit
iErrorId: Error number
Input_Len: number of data that were received

Return parameter iErrorId	Meaning
0	No error
1	Watchdog error, if greater than 0 ms (WD disable if 0 ms)
100_{dec}	Error during opening of the COM port
101_{dec}	Error during sending of data
103_{dec}	Internal receive buffer overflow
104_{dec}	Data exceed the PLC buffer capacity (more than 500 bytes)
105_{dec}	Data cannot be copied into the PLC buffer
200_{dec}	CRC error

Hardware

RS232 communication PIN assignment

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

RS485 communication PIN assignment

FB settings: When using the RS485 connection it is important that the stCOMConfig variable is set to 1 and that the COM2 interface is selected.

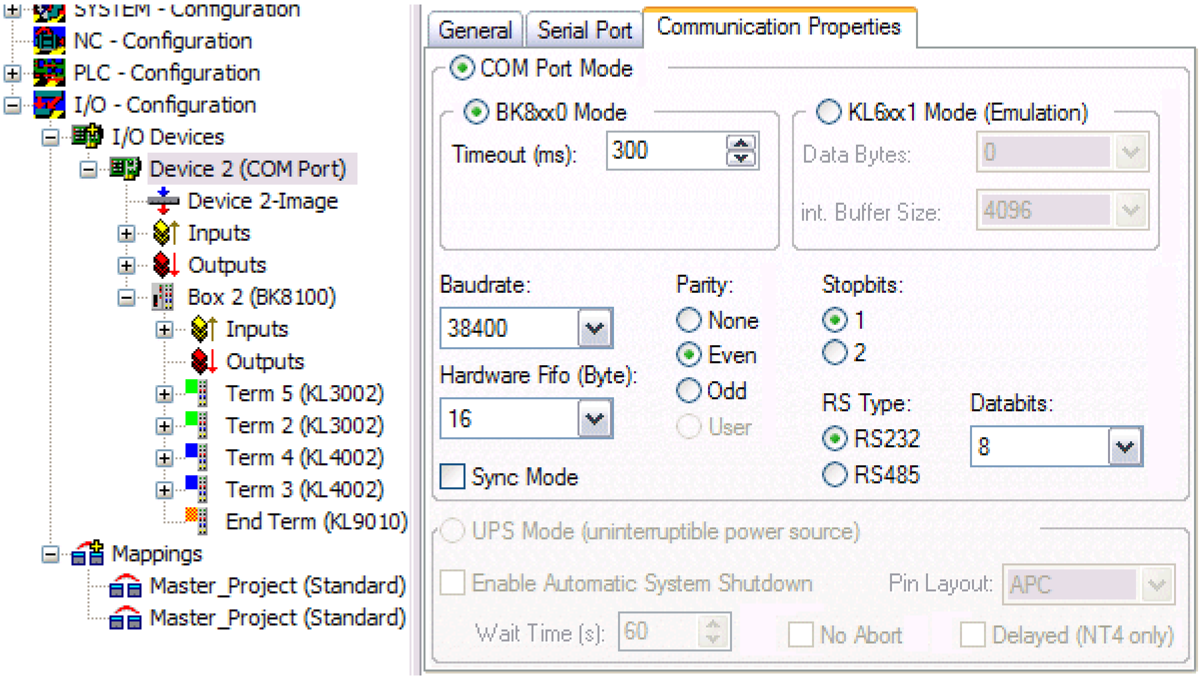
BX COM 2 RS485	PC COM port (e.g. RS485 card W&T #13601, 2-wire, without echo, automatic)
1	1 - 2 bridges
6	6 - 7 bridges

ST sample program for BXxxxx

Download (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207335947.zip)

System Manager file for TwinCAT as master. As shown in the figure, a Bus Coupler with Bus Terminals is configured. In this case, the type and number of Bus Terminals determines the data length. In principle, the type of Bus Terminal is irrelevant. Sample:

- 2 x KL3002 results in 4 words of input
- 2 x KL4002 results in 4 words of output



- SYSTEM - Configuration
- NC - Configuration
- PLC - Configuration
- I/O - Configuration
  - I/O Devices
    - Device 2 (COM Port)
      - Device 2-Image
      - Inputs
      - Outputs
      - Box 2 (BK8100)
        - Inputs
        - Outputs
        - Term 5 (KL3002)
        - Term 2 (KL3002)
        - Term 4 (KL4002)
        - Term 3 (KL4002)
        - End Term (KL9010)
  - Mappings
    - Master_Project (Standard)
    - Master_Project (Standard)

General | Serial Port | Communication Properties

● COM Port Mode

● BK8xx0 Mode
Timeout (ms): 300

○ KL6xx1 Mode (Emulation)
Data Bytes: 0
int. Buffer Size: 4096

Baudrate: 38400
Hardware Fifo (Byte): 16
□ Sync Mode

Parity: ○ None ● Even ○ Odd ○ User

Stopbits: ● 1 ○ 2

RS Type: ● RS232 ○ RS485

Databits: 8

○ UPS Mode (uninterruptible power source)
□ Enable Automatic System Shutdown
Pin Layout: APC
Wait Time (s): 60
□ No Abort
□ Delayed (NT4 only)

Fig. 121: Communication features

Required material:

- BX3100 + Bus Terminal
- PC with RS232 interface and TwinCAT from version 2.9, serial cable, PIN assignment: see above

5.13.4.4 FB_BX_COM_5

This function block connects ModbusRTU.lb6, ModbusRTU.lib or ComLib.lb6 with the serial interface of the BX Controller. It emulates a KL60x1 - data output is not via a Bus Terminal, but via one of the two serial interfaces of the BX.

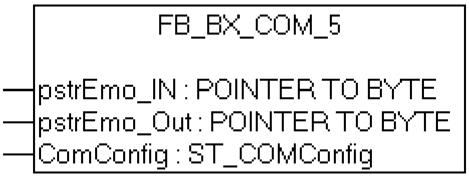


Fig. 122: Function block FB_BX_COM_5

VAR_INPUT

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

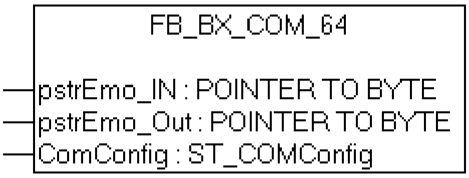
Legend

pstrEmo_IN: Pointer to KL6inData5B
pstrEmo_OUT: Pointer to KL6outData5B
ComConfig [▶ 116]: Parameterization of the COM interface

Download sample program in ST for linking COMLib and BX: (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207338123.zip)

5.13.4.5 FB_BX_COM_64

This function block connects ModbusRTU.lib or ComLib.lib with the serial interface of the BX Controller. It emulates a PC interface. Data output is not via a PC interface, but via one of the two serial interfaces of the BX (COM1 or COM2).



```
FB_BX_COM_64
pstrEmo_IN : POINTER TO BYTE
pstrEmo_Out : POINTER TO BYTE
ComConfig : ST_COMConfig
```

Fig. 123: Function block FB_BX_COM_64

VAR_INPUT

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

Legend

pstrEmo_IN: Pointer to ModbusPCComInData
pstrEmo_OUT: Pointer to ModbusPCComInData
ComConfig [▶ 116]: Parameterization of the COM interface

Download sample program in ST for linking ModbusRTU and BX: (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207340299.zip)

Download sample program in ST for linking ModbusRTU version 2 and BX: (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207342475.zip)

The sample requires the ModbusRTU library!

5.13.4.6 FB_BX_COM_64ex

This function block connects ModbusRTU.lib or ComLib.lib with the serial interface of the BX Controller. A PC interface with 64 byte of user data is emulated. Data output is not via a PC interface, but via one of the two serial interfaces of the BX (COM1 or COM2).

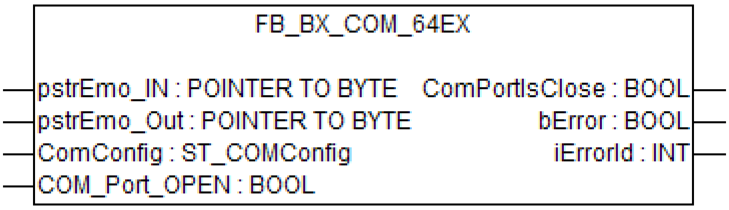


Fig. 124: Function block FB_BX_COM_64EX

VAR_INPUT

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

VAR_OUTPUT

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

Legend

pstrEmo_IN:
Pointer to ModbusPCComInData
pstrEmo_OUT:
Pointer to ModbusPCComOutData
ComConfig [▶ 116]**:**
Parameterization of the COM interface
COM_Port_Open:
If this bit is set, the interface is opened. If this bit is reset, the interface is closed.
ComPortIsClose:
If the interface is closed, this bit is set.
bError:
There is an error.
iErrorId:
Error code (see FB_COMPortOpen) [▶ 115]

Download sample program in ST for linking ModbusRTU and BX: (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207344651.zip)

The sample requires the ModbusRTU library!

5.13.4.7 Further samples

5.13.4.7.1 BX COM port as ModbusRTU master

The serial interface of the BX can also be used as Modbus master.

Necessary components

1 x BX3100
Bus Terminals for the K-Bus (any, since they are not used for the example)
1 x BK7300
2 x KL2xx4
2 x KL1xx4
1 x KL9010

RS 485 cable*

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

*) active termination resistor required for short cable lengths (< 5 m) and low baud rates (<19200 baud)

Download ST sample program for linking the ModbusRTU master and BX: (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207346827.zip **)**

Download ST sample program for linking the ModbusRTU master version 2 and BX: (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207349003.zip **)**

The example requires the ModbusRTU, TcComPortBC, TcBaseBX libraries.
Baud rate 9600, n, 8.1 default BK7300,
BK7300 address 11

Reaction times

The reaction times depend on the set task time, the number of slaves, the length of the Modbus telegrams and the response time of the slaves.
Beckhoff BK7300 Modbus slaves were used for determining the following table. Since this is not transferable to all slaves, the table should only be used for guidance.

Baud rate 38400 baud (one read reg. and one write reg. telegram per slave)

Number of slaves	Task time on the BX	Time for one cycle
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 words inputs and 2 words outputs
*) 20 words inputs and 20 words outputs

5.13.4.7.2 BX COM port - ComLibV2

Examples for ComLibV2 sending of strings via the internal COM interface of the BX controller. For receiving a bridge can be established from PIN 7 and 8 to X01 (COM2).

Required material

Hardware:

- BX Controller

Software:

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

Download BX sample program: (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207351179.zip)

5.13.4.7.3 BX COM port - Cimrex panel

The serial interface of the BX controller can also be used as Modbus slave. In this example, a panel from the company Beijers is connected. Further information on the panel can be found under http://www.beijerelectronics.de.

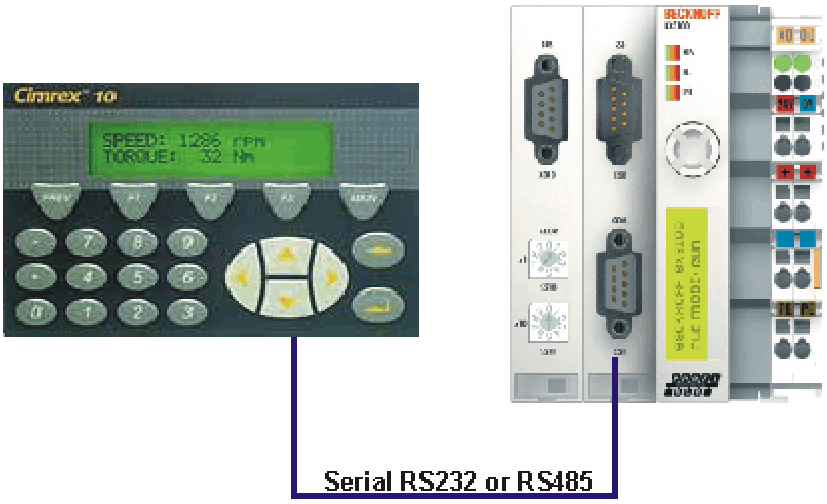


Fig. 125: Cimrex panel at the COM port of the BX controller

Necessary components

- 1 x BX3100
- 1 x Cimrex 12
- any Bus Terminals (any, since no Bus Terminals are used in the example)

RS232 cable

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

RS485* cable

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

*) active termination resistor is not required for short cable lengths (≤ 5 m) and low baud rates (≤19200 baud)

Download sample program in ST for the BX: (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207340299.zip)

Download sample with Cimrex panel: (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207353355.zip)

The example requires the ModbusRTU, TcComPortBC and TcBaseBX libraries.
- Baud rate 9600,n,8,1 D
- Cimrex 12

5.13.4.7.4 BX COM port - RK512 protocol

The RK512 protocol can exchange data with a distant station via the COM1 or COM2 interface of the BX controller. Documentation for the RK512 function block can be found in the Beckhoff Information System. The serial PC interface is simulated via the 64 byte emulation of the BX controller.

Required material

Hardware:

- PC with RS232 interface and TwinCAT PLC from 2.9
- BX Controller
- Serial cable for the BX - PC connection

Software:

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

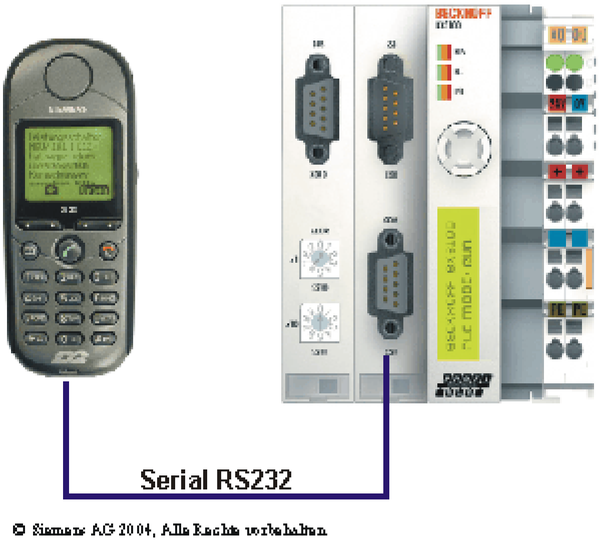
Download sample program BX3100: (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207355531.zip)

Download PC sample program: (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207357707.zip)

Download sample System Manager file PC: (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207359883.zip)

5.13.4.7.5 BX COM port - text message via mobile phone

The serial interface can also be used for sending a text message from the BX controller. The following example uses the SMS library with a Siemens S35 mobile phone.



© Siemens AG 2004, Alle Rechte vorbehalten

Fig. 126: Mobile phone at the COM port of the BX controller

Download: (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207362059.zip)

Pin assignment (Siemens cable 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 Overview

The Bus Terminal Controllers of the BX series support the TwinSAFE Bus Terminals when the following conditions are met:

- At the Bus Terminal Controller only one logic terminal is permitted. It must be connected to the K-bus interface, not the SSB.
- At this logic terminal a maximum of seven connections are permitted.
- TwinSAFE-input and output terminals can be connected to the K-bus or the SSB, for example via BK5120 or BK515x.
- If the online change feature is to be used, the connection timeout must be set to 500 ms or greater.
- An ADS connection must exist for downloading the TwinSAFE projects. The connection can be serial or via the fieldbus.
- The firmware version of the Bus Terminal Controller must be 1.17 or higher.

TwinSAFE library

The TwinSAFE library includes function blocks for executing services/functions in connection with the TwinSAFE terminals KL1904, KL2904 and KL6904.

Name	Description
F_GetVersionTcTwinSAFE [▶ 136]	Library version number
FB_TwinSAFE_KLx904_input [▶ 136]	Evaluation of TwinSAFE data sent from a KL1904 or KL2904 to a KL6904
FB_TwinSAFE_KLx904_output [▶ 139]	Evaluation of TwinSAFE data sent from a KL6904 to a KL1904 or KL2904

Download of the TwinSAFE library: (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3207364235.zip)

5.13.5.2 FUNCTION F_GetVersionTcTwinSAFE

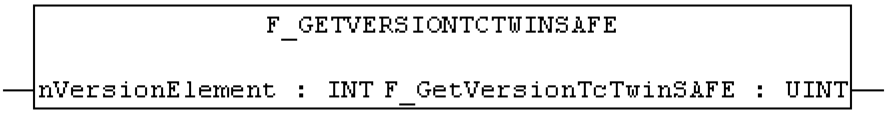


Fig. 127: Function block F_GETVERSIONTCTWINSAFE

This function can be used to read PLC library version information.

FUNCTION F_GetVersionTcTwinSAFE : UINT

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

nVersionElement: Version element to be read. Possible parameters:

- 1 : major number;
- 2 : minor number;
- 3 : revision number;

Development environment	Target platform	IO Hardware	PLC libraries to include
TwinCAT v2.10.0 Build > 914	PC (i386)	-	TcTwinSAFE.Lib (Standard.lib, TcBase.Lib and TcSystem.Lib are integrated automatically)
TwinCAT v2.10.0	BX series	-	TcTwinSAFE.LBX (Standard.LBX; TcBaseBX.LBX; TcSystemBX.LBX are integrated automatically)

5.13.5.3 FUNCTION_BLOCK FB_TwinSAFE_KLx904_input

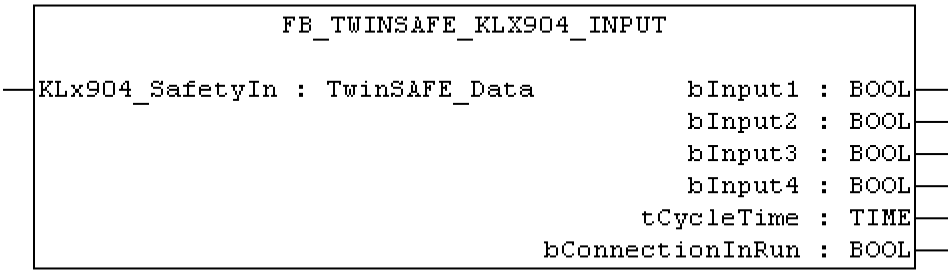


Fig. 128: Function block FB_TWINSAFE_KLX904_INPUT

The function block *FB_TwinSAFE_KLx904_input* can be used for evaluation of TwinSAFE data sent from a KL1904 or KL2904 to a KL6904. The input parameter is doubly linked to the SafetyIn data of a KL1904 or KL2904 in the System Manager.

VAR_INPUT

```
VAR_INPUT
    KLx904_SafetyIn AT%I* : TwinSAFE_Data; (* Additional link to "SafetyIn" *)
END_VAR
```

KLx904_SafetyIn: TwinSAFE telegram sent from a KL1904 or KL2904 to a KL6904. This parameter is doubly linked to SafetyIn in the System Manager (input data of the KLx904).

VAR_OUTPUT

```
VAR_OUTPUT
    bInput1               : BOOL;
    bInput2               : BOOL;
    bInput3               : BOOL;
    bInput4               : BOOL;
    tCycleTime            : TIME;
    bConnectionInRun      : BOOL;
END_VAR
```

bInput1: Returns input 1 of a KL1904. If this function block is used for connection to a KL2904, the value is always 0.

bInput2: Returns input 2 of a KL1904. If this function block is used for connection to a KL2904, the value is always 0.

bInput3: Returns input 3 of a KL1904. If this function block is used for connection to a KL2904, the value is always 0.

bInput4: Returns input 4 of a KL1904. If this function block is used for connection to a KL2904, the value is always 0.

tCycleTime: Returns the cycle time in ms for exchanging the TwinSAFE telegram between the devices.

bConnectionInRun: Returns TRUE if there is no error in the connection between the KLx904 and the KL6904.

Example of a call in the FBD:

```
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
```

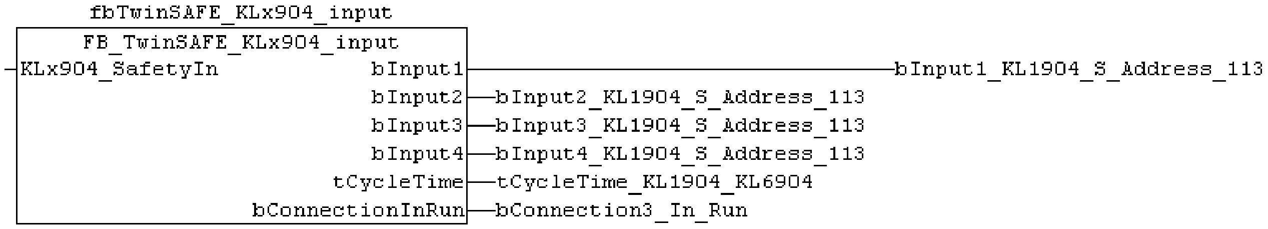


Fig. 129: Function block FB_TWINSAFE_KLX904_input

In the example the values of the KL1904 input data are written to the connected variables. If the output bConnectionInRun is FALSE, all outputs are set to FALSE.

To link the input data, select the parameter KLx904_SafetyIn and select "Modify link..." from the context menu.

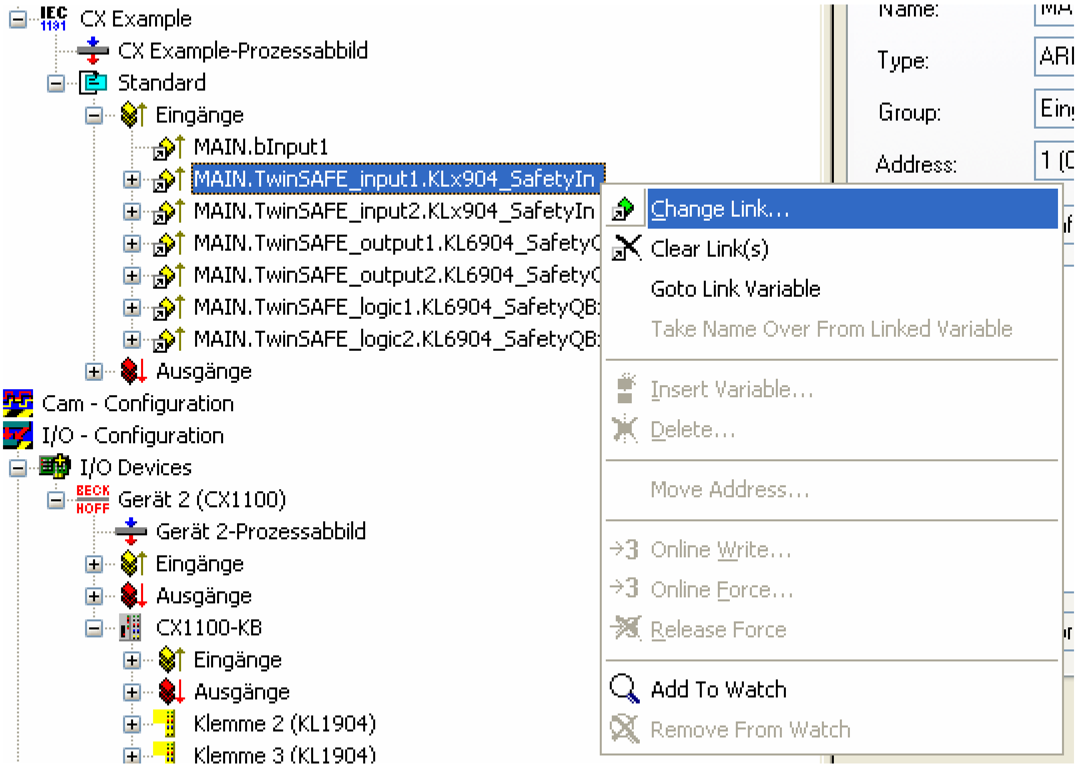
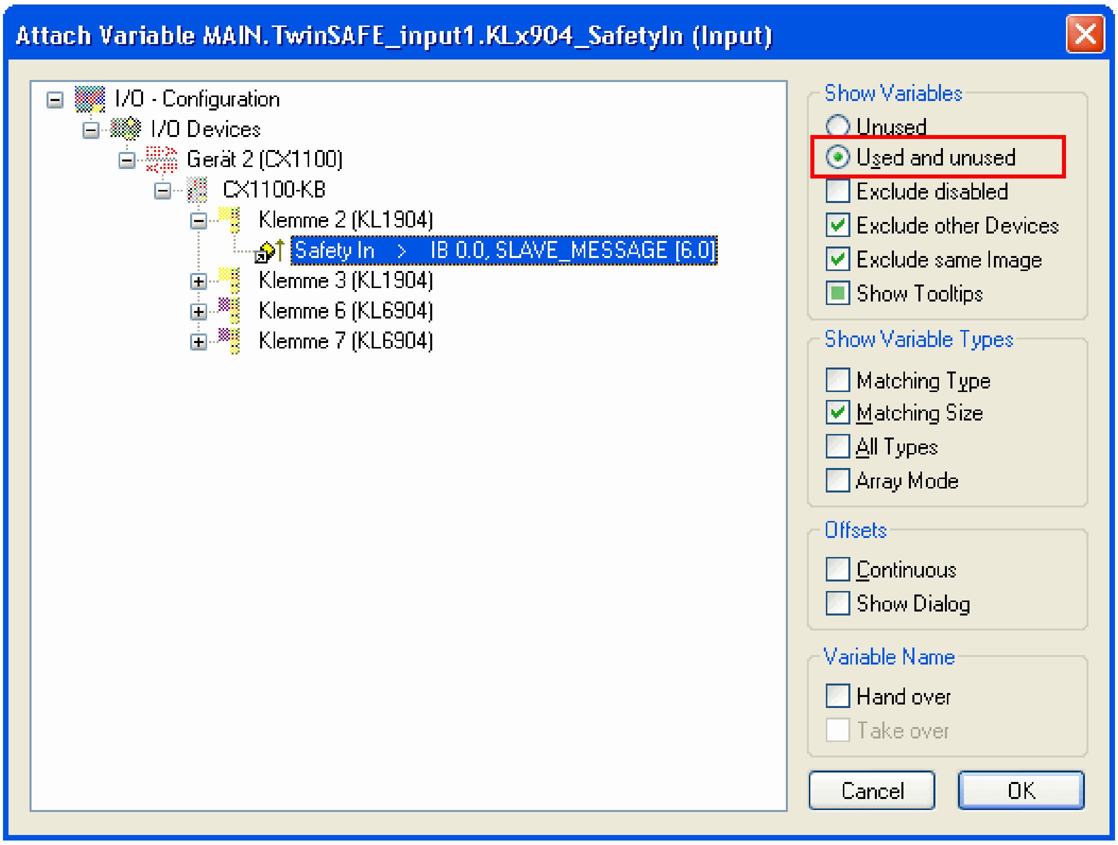


Fig. 130: Linking the input data

and select the corresponding SafetyIn variable in the dialog that follows.


Attach Variable MAIN.TwinSAFE_input1.KLx904_SafetyIn (Input)

- I/O - Configuration
  - I/O Devices
    - Gerät 2 (CX1100)
      - CX1100-KB
        - Klemme 2 (KL1904)
          - Safety In > IB 0.0, SLAVE_MESSAGE [6.0]
        - Klemme 3 (KL1904)
        - Klemme 6 (KL6904)
        - Klemme 7 (KL6904)

Show Variables
○ Unused
● Used and unused
□ Exclude disabled
☑ Exclude other Devices
☑ Exclude same Image
▣ Show Tooltips

Show Variable Types
□ Matching Type
☑ Matching Size
□ All Types
□ Array Mode

Offsets
□ Continuous
□ Show Dialog

Variable Name
□ Hand over
□ Take over

Fig. 131: Selecting the SafetyIn variable

Development environment	Target platform	IO Hardware	PLC libraries to include
TwinCAT v2.10.0 Build > 914	PC (i386)	KLx904	TcTwinSAFE.Lib (Standard.lib, TcBase.Lib and TcSystem.Lib are integrated automatically)
TwinCAT v2.10.0 Build > 914	BX series	KLx904	TcTwinSAFE.LBX (Standard.LBX, TcBaseBX.LBX and TcSystemBX.LBX are integrated automatically)

5.13.5.4 FUNCTION_BLOCK FB_TwinSAFE_KLx904_output

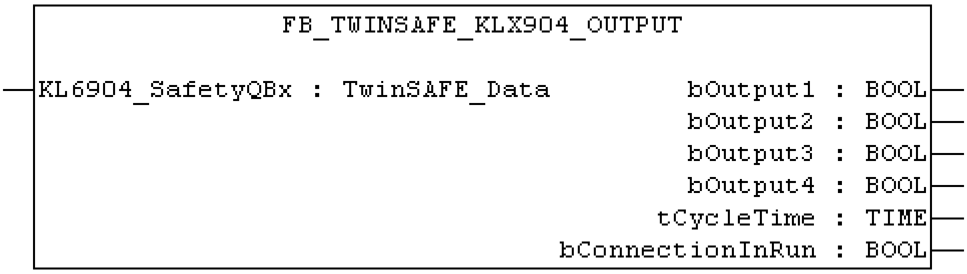


Fig. 132: Function block FB_TWINSAFE_KLX904_output

The function block *FB_TwinSAFE_KLx904_output* can be used for evaluation of TwinSAFE data sent from a KL6904 to a KL1904 or KL2904. The input parameter is doubly linked to the SafetyQBx data of a KL6904 in the System Manager.

VAR_INPUT

```
VAR_INPUT
    KL6904_SafetyQBx AT%I* : TwinSAFE_Data; (* Additional link to "SafetyQBx" *)
END_VAR
```

KL6904_SafetyQBx: TwinSAFE telegram sent from a KL6904 to a KL1904 or KL2904. This parameter is doubly linked to SafetyQBx in the System Manager (input data of the KL6904); x represents for numerals between 1 and 15, according to the TwinSAFE connection used.

VAR_OUTPUT

```
VAR_OUTPUT
    bOutput1            : BOOL;
    bOutput2            : BOOL;
    bOutput3            : BOOL;
    bOutput4            : BOOL;
    tCycleTime          : TIME;
    bConnectionInRun    : BOOL;
END_VAR
```

bOutput1: Returns output 1 of a KL2904. If the function block is used for a connection to the KL1904, this value is always 0.

bOutput2: Returns output 2 of a KL2904. If the function block is used for a connection to the KL1904, this value is always 0.

bOutput3: Returns output 3 of a KL2904. If the function block is used for a connection to the KL1904, this value is always 0.

bOutput4: Returns output 4 of a KL2904. If the function block is used for a connection to the KL1904, this value is always 0.

tCycleTime: Returns the cycle time in ms for exchanging the TwinSAFE telegram between the devices.

bConnectionInRun: Returns TRUE if there is no error in the connection between the KL6904 and the KLx904.

Example of a call in the FBD

```
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
```

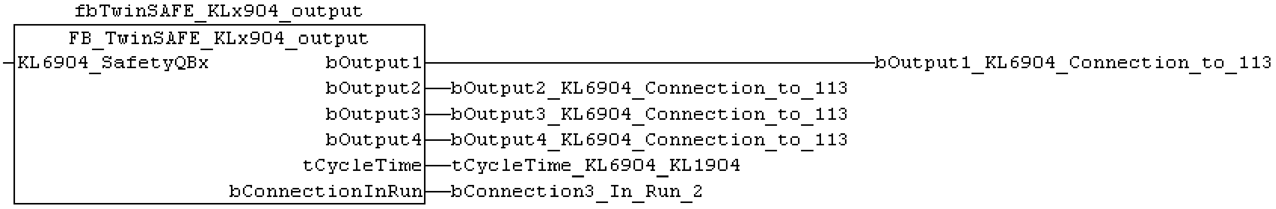

fbTwinSAFE_KLx904_output
FB_TwinSAFE_KLx904_output
KL6904_SafetyQBx
bOutput1
bOutput2
bOutput3
bOutput4
tCycleTime
bConnectionInRun
bOutput1_KL6904_Connection_to_113
bOutput2_KL6904_Connection_to_113
bOutput3_KL6904_Connection_to_113
bOutput4_KL6904_Connection_to_113
tCycleTime_KL6904_KL1904
bConnection3_In_Run_2

Fig. 133: Call of function block FB_TWINSAFE_KLX904_OUTPUT

In the example the values of TwinSAFE terminals KL6904 and KL1904 are evaluated. Since no output signals are used in this connection, the outputs are always FALSE. Only tCycleTime and bConnectionInRun can be evaluated.

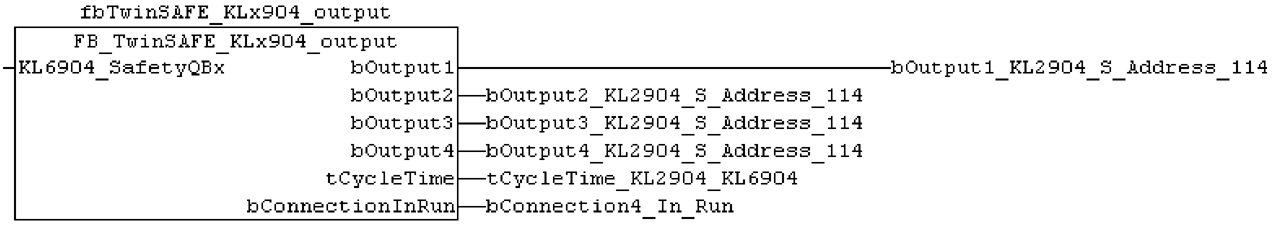

fbTwinSAFE_KLx904_output
FB_TwinSAFE_KLx904_output
KL6904_SafetyQBx
bOutput1
bOutput2
bOutput3
bOutput4
tCycleTime
bConnectionInRun
bOutput1_KL2904_S_Address_114
bOutput2_KL2904_S_Address_114
bOutput3_KL2904_S_Address_114
bOutput4_KL2904_S_Address_114
tCycleTime_KL2904_KL6904
bConnection4_In_Run

Fig. 134: Call of function block FB_TWINSAFE_KLX904_OUTPUT

In the example the values of TwinSAFE terminals KL6904 and KL1904 are evaluated. In this connection the output signals are written to the KL2904 and copied from the function block to the connected variables. If the output bConnectionInRun is FALSE, all outputs are set to FALSE.

To link the input data, select the parameter KL6904_SafetyQBx and select "Modify link..." from the context menu.

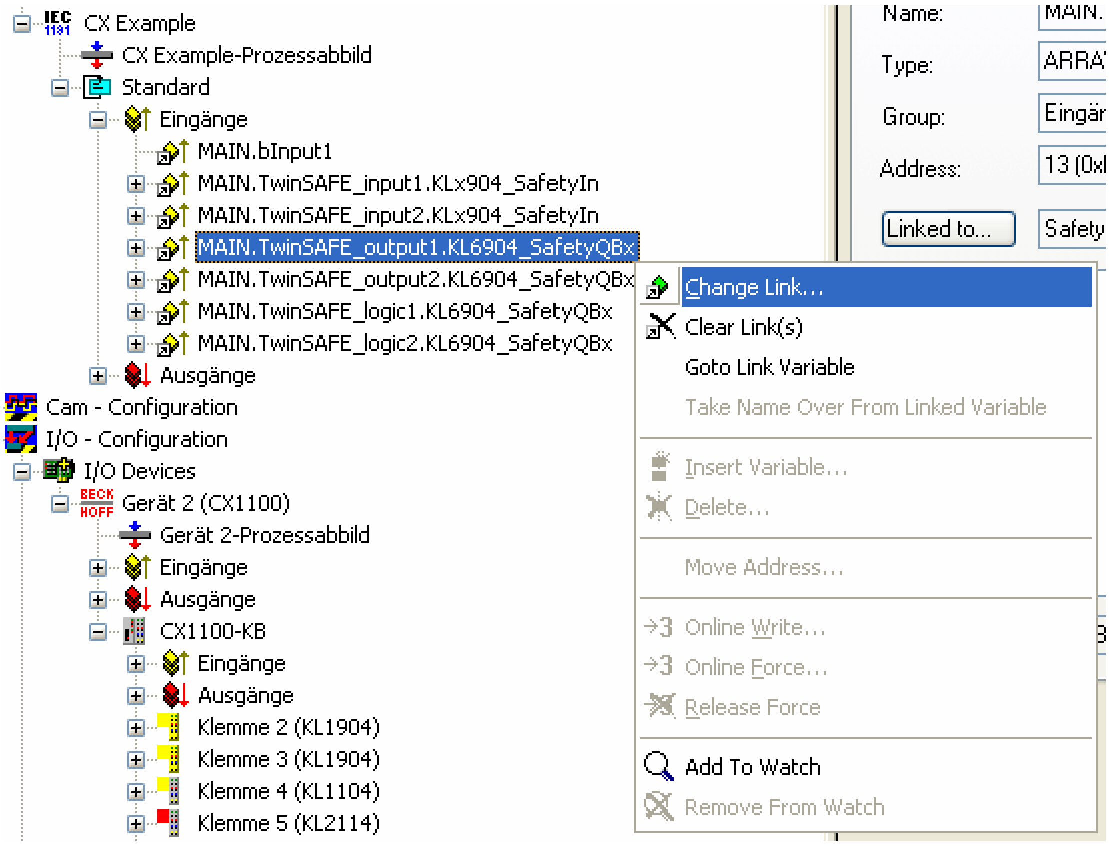
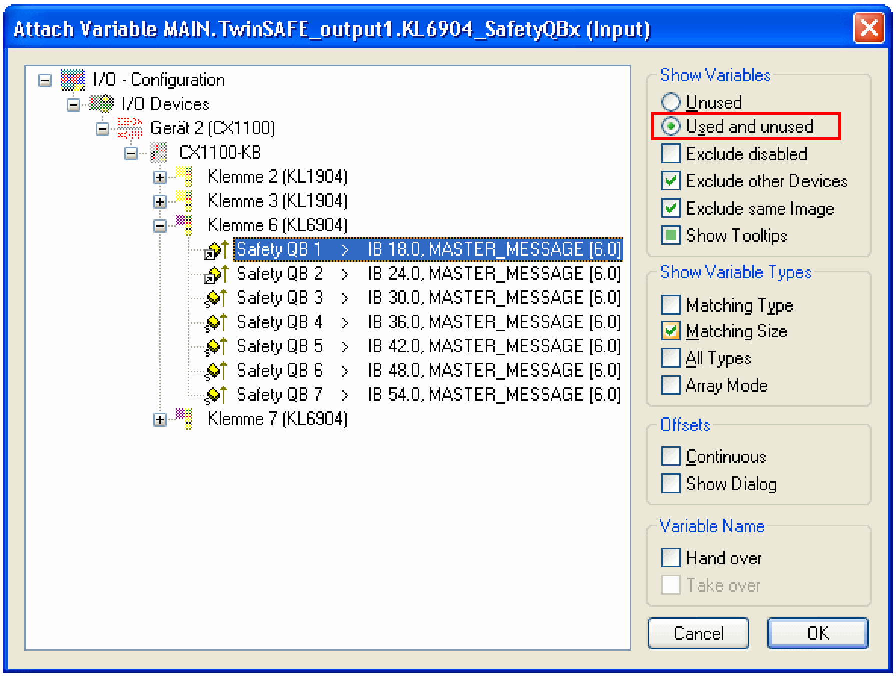


Fig. 135: Linking the input data

and select the corresponding SafetyQBx variable in the dialog that follows.


Attach Variable MAIN.TwinSAFE_output1.KL6904_SafetyQBx (Input)

I/O - Configuration
I/O Devices
Gerät 2 (CX1100)
CX1100-KB
Klemme 2 (KL1904)
Klemme 3 (KL1904)
Klemme 6 (KL6904)
Safety QB 1 > IB 18.0, MASTER_MESSAGE [6.0]
Safety QB 2 > IB 24.0, MASTER_MESSAGE [6.0]
Safety QB 3 > IB 30.0, MASTER_MESSAGE [6.0]
Safety QB 4 > IB 36.0, MASTER_MESSAGE [6.0]
Safety QB 5 > IB 42.0, MASTER_MESSAGE [6.0]
Safety QB 6 > IB 48.0, MASTER_MESSAGE [6.0]
Safety QB 7 > IB 54.0, MASTER_MESSAGE [6.0]
Klemme 7 (KL6904)
Cancel
OK

Show Variables
○ Unused
● Used and unused
□ Exclude disabled
☑ Exclude other Devices
☑ Exclude same Image
▣ Show Tooltips

Show Variable Types
□ Matching Type
☑ Matching Size
□ All Types
□ Array Mode

Offsets
□ Continuous
□ Show Dialog

Variable Name
□ Hand over
□ Take over

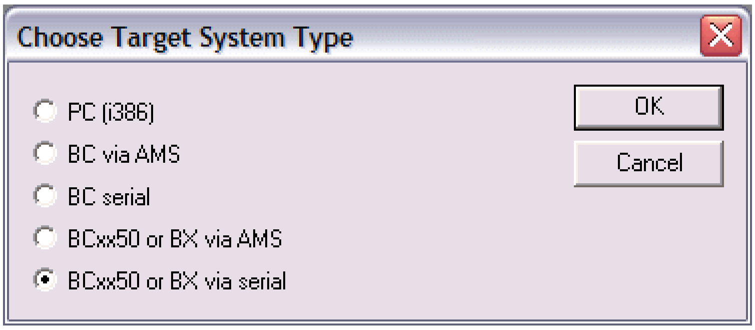
Fig. 136: Selecting the corresponding SafetyQBx variable

Development environment	Target platform	IO Hardware	PLC libraries to include
TwinCAT v2.10.0 Build > 914	PC (i386)	KLx904	TcTwinSAFE.Lib (Standard.lib, TcBase.Lib and TcSystem.Lib are integrated automatically)
TwinCAT v2.10.0 Build > 914	BX series	KLx904	TcTwinSAFE.LBX (Standard.LBX, TcBaseBX.LBX and TcSystemBX.LBX are integrated automatically)

5.14 Program transfer via the serial interface

Every Bus Terminal Controller can be programmed via the PC's RS232 interface.

Select the serial interface in TwinCAT PLC Control.



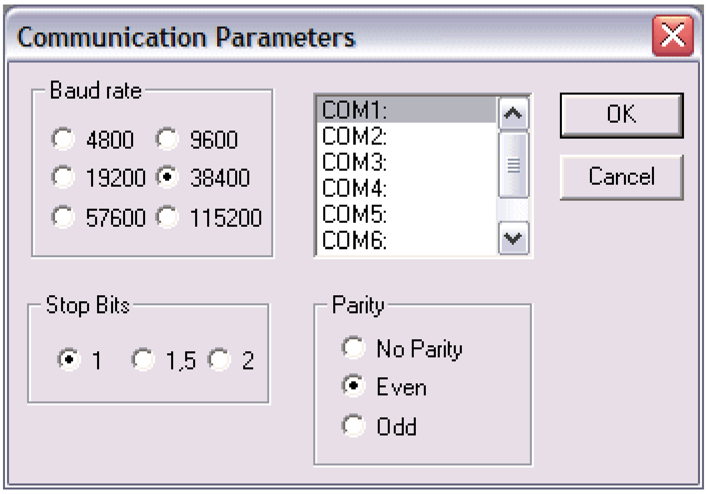
○ PC (i386)
○ BC via AMS
○ BC serial
○ BCxx50 or BX via AMS
● BCxx50 or BX via serial

Fig. 137: *Selecting the data transfer route - serial interface*

The settings for the serial interface, port number, baud rate etc. are found under Online/Communication parameters in PLC Control.

The Bus Terminal Controller requires the following setting:

- Baud Rate: 9600/19200/38400/57600 baud (automatic baud rate detection)
- Stop bits: 1
- Parity: Straight line



Baud rate
○ 4800 ○ 9600
○ 19200 ● 38400
○ 57600 ○ 115200

COM1:
COM2:
COM3:
COM4:
COM5:
COM6:

Stop Bits
● 1 ○ 1,5 ○ 2

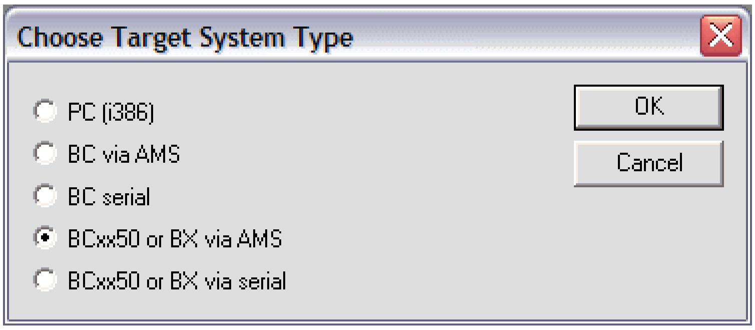
Parity
○ No Parity
● Even
○ Odd

Fig. 138: Parameterization of the serial interface

Program transfer via the serial interface and ADS

The Bus Terminal Controller can be programmed via the PC's RS232 interface. Before you can work with the Bus Terminal Controller, TwinCAT must be notified of it (see serial ADS [▶ 41]).

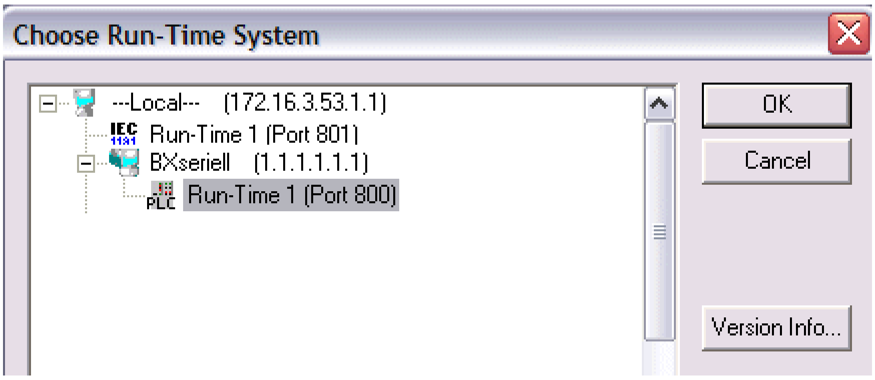
Select the ADS connection in TwinCAT PLC Control.


Choose Target System Type

○ PC (i386)
○ BC via AMS
○ BC serial
● BCxx50 or BX via AMS
○ BCxx50 or BX via serial

Fig. 139: *Selecting the data transfer route - AMS*

PLC Control can be accessed via *Online/Communication Parameters...*.


Choose Run-Time System

```
---Local---   (172.16.3.53.1.1)
    Run-Time 1 (Port 801)
    BXseriell   (1.1.1.1.1.1)
        Run-Time 1 (Port 800)
```

Fig. 140: Selecting the device

6 DeviceNet communication

6.1 DeviceNet Introduction

Fig. 141: DeviceNet

DeviceNet is an open system based on CAN. CAN was developed some years ago by R. Bosch for data transmission in motor vehicles. Millions of CAN chips are now in use. A disadvantage for application in automation is that CAN does not contain definitions for the application layer. CAN only defines the physical and data link layer.

DeviceNet specifies a uniform application layer and this makes it possible to use the CAN protocol for industrial applications. ODVA (the Open DeviceNet Vendor Association) is an independent association which supports manufacturers and users of the DeviceNet system. ODVA ensures that all devices which conform to the specification can operate together in one system, regardless of their manufacturer.

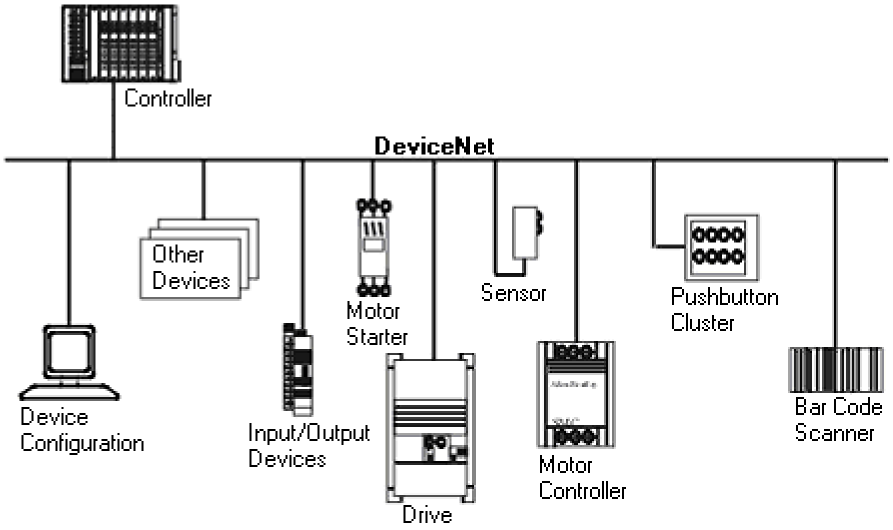


Fig. 142: Example of DeviceNet in use

DeviceNet is a sensor/actuator bus system. It is internationally standardised (EN50325) and is based on CAN (Controller Area Network). DeviceNet supports a number of communication types for the input and output data:

- Polling: The master module ("scanner") sends the output data cyclically to the assigned devices and receives the input data in an answer telegram.
- Change-of-State: Telegrams are sent as soon as their contents have changed.
- Cyclic: The modules send the data automatically after a cycle time has elapsed.
- Strobed: The scanner requests the input data using a broadcast telegram to all the devices.

The DeviceNet devices support all I/O communication types.

The DeviceNet devices are parameterized via acyclical services (explicit messaging).

The effective utilization of the bus bandwidth allows DeviceNet, particularly in Change-of-State mode, to achieve short system reaction times in spite of the relatively low data rates. The BECKHOFF DeviceNet devices have a powerful implementation of the protocol. Through active participation in the ODVA's technical committees, BECKHOFF are contributing to the further development of this bus system, and has in this way itself gathered profound DeviceNet expertise.

Configuration

The node address is set in the range from 0 to 63 using two decimally coded rotary switches. The data transfer rate set at the DeviceNet scanner is automatically recognized by the DeviceNet Box (auto baud rate). "Electronic Data Sheets" (EDS files) for DeviceNet configuration tools are available for download from the Beckhoff internet site (http://www.beckhoff.de), and on the BECKHOFF product CDs. Special I/O parameters that are not covered by the DeviceNet standard can be set via the KS2000 software (serial connection) or via acyclical explicit messages.

Diagnostics

The extensive diagnostic functions of the BECKHOFF DeviceNet devices allow rapid fault localisation. The diagnostic messages are transmitted over the bus and collated by the master. The status of the network connection, the device status, the status of the inputs and outputs and of the power supply are displayed by LEDs.

Data transfer rates

Three data transfer rates from 125 kbaud to 500 kbaud are available for different bus lengths. The effective utilization of the bus bandwidth allows DeviceNet to achieve short system reaction times at relatively low data rates.

Topology

DeviceNet is based on a linear topology. The number of devices participating in each network is logically limited by DeviceNet to 64, but physically the present generation of drivers allows up to 64 nodes in one network segment. The maximum possible size of the network for any particular data rate is limited by the signal propagation delay required on the bus medium. For 500 kbaud, for instance, the network may extend 100 m, whereas at 125 kbaud the network may reach up to 500 m. At low data rates the size of the network can be increased by repeaters, which also allow the construction of tree structures.

Bus access procedures

CAN utilizes the Carrier Sense Multiple Access (CSMA) procedure, i.e. all participating devices have the same right of access to the bus and may access it as soon as it is free (multi-master bus access). The exchange of messages is thus not device-oriented but message-oriented. This means that every message is unambiguously marked with a prioritized identifier. In order to avoid collisions on the bus when messages are sent by different devices, a bit-wise bus arbitration is carried out at the start of the data transmission. The bus arbitration assigns bus bandwidth to the messages in the sequence of their priority. At the end of the arbitration phase only one bus device occupies the bus, collisions are avoided and the bandwidth is optimally exploited.

Configuration and parameterization

The TwinCAT System Manager allows all the DeviceNet parameters to be set conveniently. An "eds" file (electronic data sheet) is available on the BECKHOFF website (http://www.beckhoff.de) for the parameterization of BECKHOFF DeviceNet devices using configuration tools from other manufacturers.

6.2 Protocol Description

6.2.1 Network Management

In preparation

6.3 Object directory

In preparation

6.4 ADS-Communication

6.4.1 ADS services

Local Process Image PLC Task 1 Port 800/801

Data can be read from and written to the local process image. If it is necessary for outputs to be written, it is important to ensure that they are not used by the local PLC, because the local controller will overwrite these values. The data are not associated with a watchdog, and therefore must not be used for outputs that would have to be switched off in the event of a fault.

Index group	Meaning	Index offset (value range)
0xF020	Inputs	0...2047
0xF021	Bit inputs	0...16376
0xF030	Outputs	0...2047
0xF031	Bit outputs	0...16376
0x4020	Flags	0...4095
0x4021	Flag bit	0...32760

ADS services

AdsServerAdsState

Data type (read only)	Meaning
String	Start - the local PLC is running Start - the local PLC is stopped

AdsServerDeviceState

Data type (read only)	Meaning
INT	0 - Start - the local PLC is running 1 - Stop - the local PLC is stopped

AdsServerType

Data type (read only)	Meaning
String	BX PLC Server

ADSWriteControl

Data type (write only)	Meaning
NetID	Net ID of the Ethernet Controller*
Port	800
ADSSTATE	5 - RUN / 6 - STOP
DEVSTATE	0
LEN	0
SRCADDR	0
WRITE	rising edge starts the function block
TMOUT	example: T#1000 ms

* BC9050, BC9020, BC9120, BX9000

Register access port 100

On the Bus Terminal Controllers of the BX series, and on the BCxx50/xx20, the ADS port number for register communication is fixed at 100.

Index group	Index offset (value range)		Meaning
	Hi-Word	Low Word	
0 [READ ONLY]	0...127	0...255	Registers in the Bus Coupler Hi-Word, table number of the Bus Coupler Lo-Word, register number of the table
1...255	0...3	1...255	Register of the Bus Terminals Hi-Word, channel number Lo-Word, register number of the Bus Terminal

Minimum timeout

For reading the registers, ensure that the timeout for the ADS function block is set to more than one second.

Setting the password

When writing to the registers, the password has to be set (see the documentation for the particular Bus Terminal).

7 Error handling and diagnosis

7.1 Diagnostics

DeviceNet state

In many cases it is important to know whether the communication with the higher-level master is still OK. To this end, link the NodeState variable with your PLC program. A TwinCAT configuration is required for this purpose.

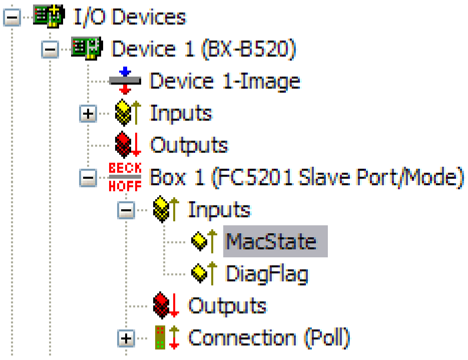


Fig. 143: DeviceNet diagnostics byte in the System Manager

Error number	Description	Remedy
0	no error	none
1	Node is deactivated by the scanner, node is no longer in the masters scan-list. For Slave Devices it means that there is no Data-Exchange between the Master and the Slave. The master has released the DeviceNet slave	Check the configuration
2	No data exchange between node and DeviceNet Master, node access timeout	Check the connection
18	Node is configured, ready for data exchange	none
42	Electronic Key Error: Vendor Id	Check identity of DeviceNet slave
43	Electronic Key Error: Device Type	Check identity of DeviceNet slave
44	Electronic Key Error: Product Code	Check identity of DeviceNet slave
45	Electronic Key Error: Revision	Check identity of DeviceNet slave
46	Error writing start-up attributes	Check identity of DeviceNet slave
47	Wrongproduced IO-Data length	Check identity of DeviceNet slave
48	Wrong consumed IO-Data length	Check identity of DeviceNet slave
49	Idle Mode (for Slave Devices): no valid IO-Data is exchanged via DeviceNet	Check whether the master is in idle mode

Reading fieldbus state by ADS

You can read the fieldbus state via ADSREAD in the default configuration or in the TwinCAT configuration.

Parameter ADSREAD function block	Description
NetID	local – empty string
Port	1
IndexGroup	16#0006
IndexOffset	16#000C_AE00
LEN	1

State of the K-bus

An internal bus or Bus Terminal error is indicated in the *K-bus state* variable. A more precise fault description can be obtained via a function block (in preparation). To this end, link the *K-bus state* variable with your PLC program.

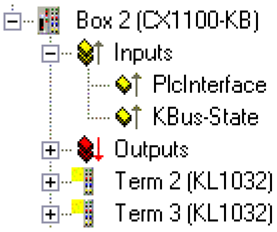


Fig. 144: State of the K-bus

Error bit	Description	Error type
0	No error	No error
Bit 0	K-bus error	Error
Bit 2	K-bus is re-triggered	Note

Reading K-Bus state by ADS

You can read the fieldbus state via ADSREAD in the default configuration or in the TwinCAT configuration.

Parameter ADSREAD function block	Description
NetID	local – empty string
Port	1
IndexGroup	16#0006
IndexOffset	16#000C_9000
LEN	1

7.2 Diagnostic LEDs

A BX Controller has two groups of LEDs for status display:

- On the top right of the BX controller there are two green power LEDs. These are provided for diagnostics of the power supply of the BX Controller and of the power contacts.
- The DIAG LEDs are located above the navigation switch on the left. They indicate the state of the fieldbus, the PLC and the K-bus.

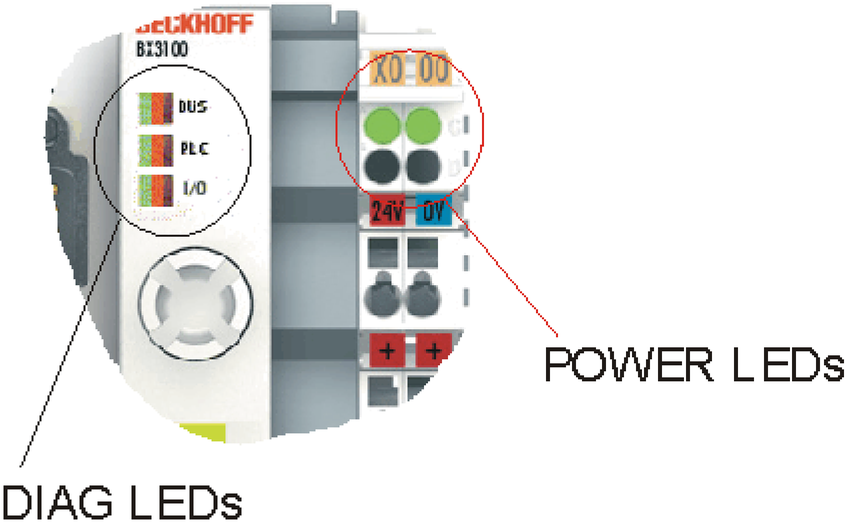


Fig. 145: Diagnostic LEDs for the fieldbus, the PLC, the K-bus and the power supply units

POWER LEDs for diagnostics of power supply

Power LEDs	Meaning
Left LED off	BX Controller has no power supply
Right LED off	Power contacts has no power supply

DIAG LEDs

The DIAG LEDs are sub-divided as follows:
- Bus: fieldbus diagnostics
- PLC: PLC diagnostics
- I/O: K-bus diagnostics

The LEDs can be off, green, orange or red.

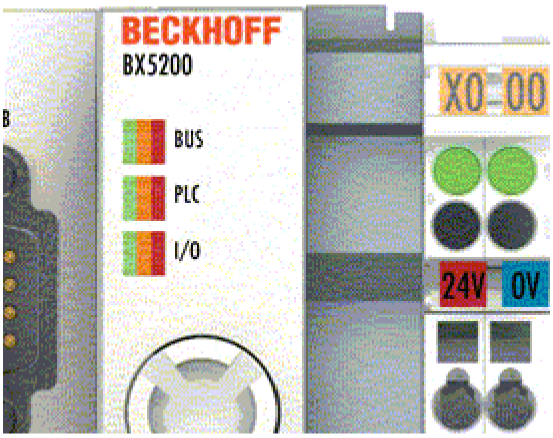


Fig. 146: Diagnostic LEDs for the fieldbus, the PLC and the K-bus

After switching on, the BX Controller immediately checks the connected configuration. The *I/O* LED goes out if the start-up was successful. A red *I/O* LED indicates a Bus Terminal error. The error type is shown in the display. This permits rapid rectification of the error.

LEDs for fieldbus diagnostics

LED Bus	Meaning
off	no fieldbus connected
red illuminated	no fieldbus connected, DeviceNet 24 V missing, DeviceNet communication error - wrong baud rate, incorrect termination
flashes red	communication error, IO timeout - communication to DeviceNet master interrupted
flashes orange	Baud rate search active, no baud rate found
flashing green	communication with master not yet established
green illuminated	DeviceNet communication active

LED for PLC diagnostics

LED PLC	Meaning
off	PLC in stop or no program available
flashes red	The set task time is sometimes exceeded (see also Chapter Deactivating the LED for cycle time exceeding [▶ 113]).
red illuminated	The set task time is always exceeded.
orange illuminated	PLC runs without boot project (only lights up during the cycle), when the boot project is generated, the LED flashes orange
green illuminated	Boot project - PLC is running (only lights up during cycle)

LED for K-bus diagnostics

LED I/O	Meaning
off	No data are exchanged via the K-bus
red illuminated	error flashing - error type - display
orange illuminated	Register or KS2000 online access
green illuminated	K-bus OK and running

Error code for K-bus diagnostics

Error code	Error code argument	Description	Remedy
0	-	EMC problems	• Check power supply for undervoltage or overvoltage peaks • Implement EMC measures • If a K-bus error is present, it can be localized by a restart of the coupler (by switching it off and then on again)
1	0	EEPROM checksum error	Enter factory settings with the KS2000 configuration software
	1	Code buffer overflow	Insert fewer Bus Terminals. The programmed configuration has too many entries in the table
	2	Unknown data type	Software update required for the BX Controller
2	-	reserved	-
3	0	K-bus command error	• No Bus Terminal inserted • One of the Bus Terminals is defective; halve the number of Bus Terminals attached and check whether the error is still present with the remaining Bus Terminals. Repeat until the defective Bus Terminal is located.
4	0	K-bus data error, break behind the BX Controller	Check whether the n+1 Bus Terminal is correctly connected; replace if necessary.
	n	Break behind Bus Terminal n	Check whether the KL9010 Bus End Terminal is connected
5	n	K-bus error in register communication with Bus Terminal n	Exchange the nth Bus Terminal
6	0	Error at initialization	Replace BX Controller
	1	Internal data error	Perform a hardware reset on the BX Controller (switch off and on again)
	2	DIP switch changed after a software reset	Perform a hardware reset on the BX Controller (switch off and on again)
7	0	Note: cycle time was exceeded	Warning: the set cycle time was exceeded. This indication (flashing LEDs) can only be cleared by booting the BX Controller again. Remedy: increase the cycle time
9	0	Checksum error in Flash program	Transmit program to the BX again
	1	Incorrect or faulty library implemented	Remove the faulty library
10	n	Bus Terminal n is not consistent with the configuration that existed when the boot project was created	Check the nth Bus Terminal. The boot project must be deleted if the insertion of an nth Bus Terminal is intentional.
14	n	nth Bus Terminal has the wrong format	Start the BX Controller again, and if the error occurs again then exchange the Bus Terminal.
15	n	Number of Bus Terminals is no longer correct	Start the BX Controller again. If the error occurs again, restore the manufacturers setting using the KS2000 configuration software
16	n	Length of the K-bus data is no longer correct	Start the BX Controller again. If the error occurs again, restore the manufacturers setting using the KS2000 configuration software

7.3 Diagnostics display

During start-up, the display shows the current firmware version for approx. three seconds.

If an error occurs during start-up, this will be indicated via a flash sequence of the associated LED.

Configuration errors are shown in the display via TC-Config and an error number. In this case, please use the System Manager to check your hardware configuration or contact support.

Display	Meaning
TC-Config 0xE02E	A complex Bus Terminal is assigned a bit address. Check the TwinCAT configuration.
TC-Config 0xF0nn	Bus Terminal no. nn does not correspond to the configuration. Compare the bus structure of Bus Terminal no. nn with the configuration.
TC-Config 0xC0nn	Bus Terminal no. nn does not correspond to the configuration. Compare the bus structure of Bus Terminal no. nn with the configuration.

Firmware errors are shown in the display via FW-Error and an error number. Please contact support.

Display	Meaning
FW-Error 0xnnnn	Please contact support

8 Appendix

8.1 Firmware Update

Firmware update program

The firmware update program is required for loading a new firmware to the Bus Coupler. The program is transferred via the serial interface.

Note for BX3100:
Updates are not available with BX3100 firmware 0.64 (or lower). If these devices need updating, send the BX3100 to the manufacturer with a corresponding note.

Beckhoff Automation GmbH & Co. KG
Service Department
Stahlstr. 31
33415 Verl, Germany

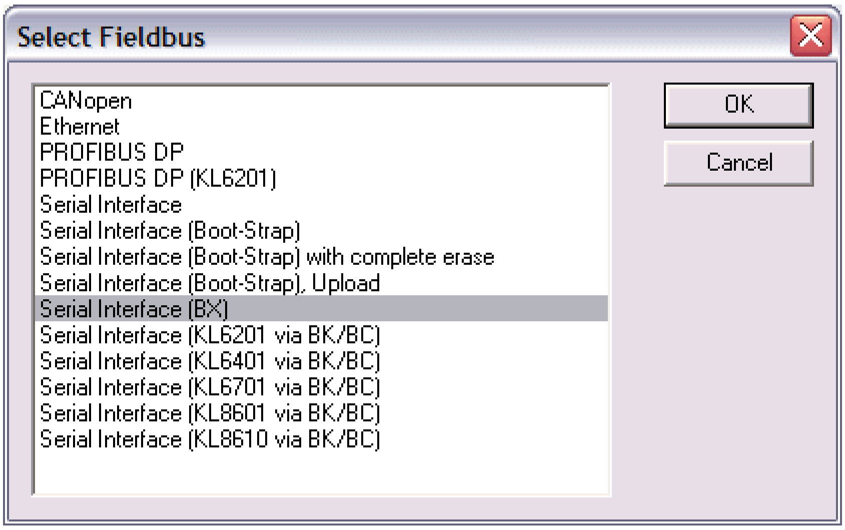
Firmware update program 241 (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3238791819.zip) 71 kbytes (for Windows NT4.0 SP6, 2000, XP).

The program *FirmwareUpdate.exe* and the file *TcRouterHelper.dll* have to be in the same directory. Open the program by double-clicking on *FirmwareUpdate.exe*.

Update for Bus Terminal Controllers

BX series

Select the appropriate device of - in this example "Serial interface (BX)".



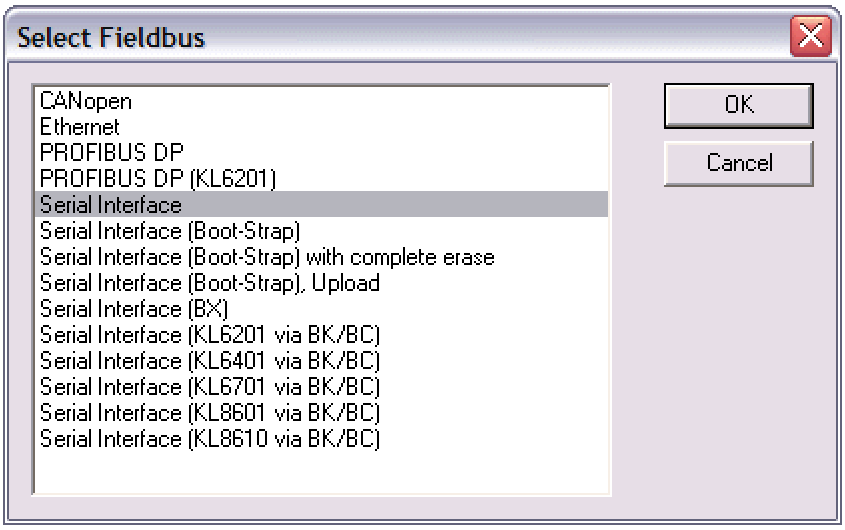
Select Fieldbus

CANopen
Ethernet
PROFIBUS DP
PROFIBUS DP (KL6201)
Serial Interface
Serial Interface (Boot-Strap)
Serial Interface (Boot-Strap) with complete erase
Serial Interface (Boot-Strap), Upload
Serial Interface (BX)
Serial Interface (KL6201 via BK/BC)
Serial Interface (KL6401 via BK/BC)
Serial Interface (KL6701 via BK/BC)
Serial Interface (KL8601 via BK/BC)
Serial Interface (KL8610 via BK/BC)

Fig. 147: Selecting a BX series Bus Terminal Controller

BCxx50 series

Select the corresponding device, in this case "Serial Interface".



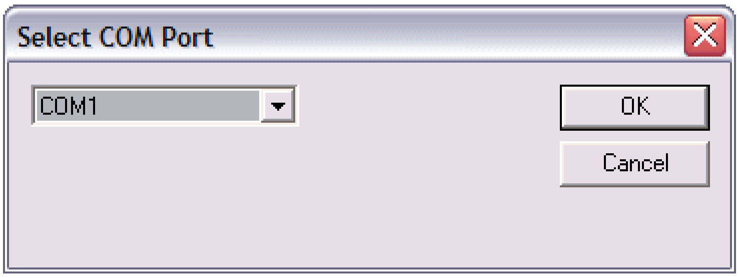
Select Fieldbus

CANopen
Ethernet
PROFIBUS DP
PROFIBUS DP (KL6201)
Serial Interface
Serial Interface (Boot-Strap)
Serial Interface (Boot-Strap) with complete erase
Serial Interface (Boot-Strap), Upload
Serial Interface (BX)
Serial Interface (KL6201 via BK/BC)
Serial Interface (KL6401 via BK/BC)
Serial Interface (KL6701 via BK/BC)
Serial Interface (KL8601 via BK/BC)
Serial Interface (KL8610 via BK/BC)

Fig. 148: Selecting a BC series Bus Terminal Controller

BX and BCxx50 series

Then select the COM port.

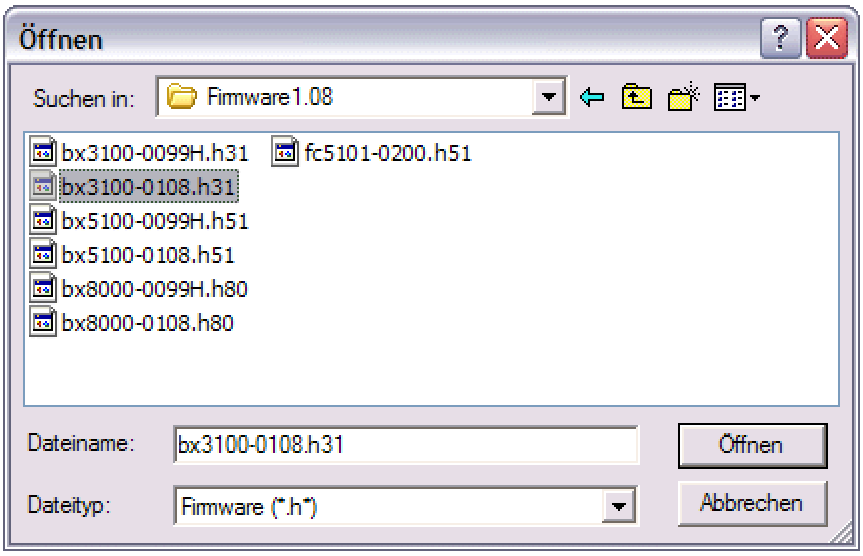


Select COM Port

COM1

Fig. 149: Select the COM port

Open the file you wish to download.



Öffnen

Suchen in: Firmware1.08

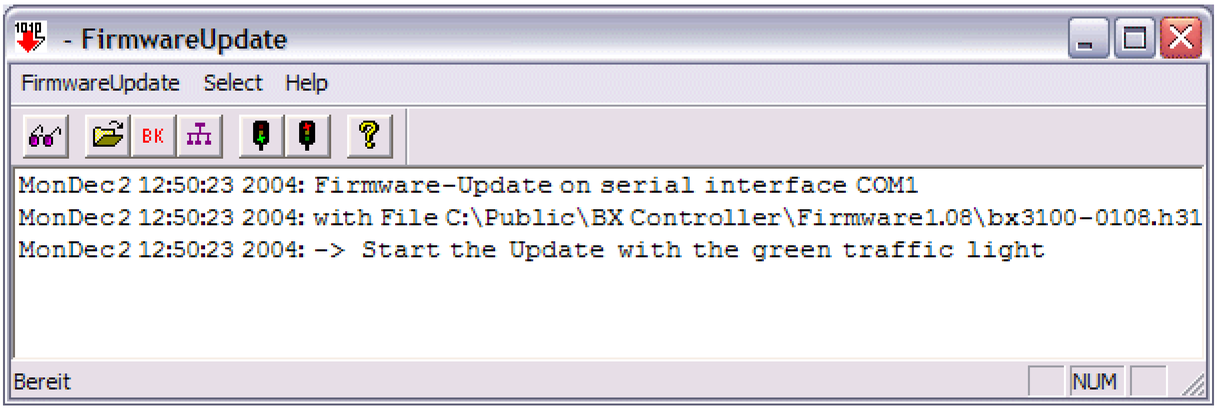
bx3100-0099H.h31
bx3100-0108.h31
bx5100-0099H.h51
bx5100-0108.h51
bx8000-0099H.h80
bx8000-0108.h80
fc5101-0200.h51

Dateiname: bx3100-0108.h31

Dateityp: Firmware (*.h*)

Fig. 150: Open the firmware file

Start the download via the green 'traffic light'. The download begins after about a minute, and is then also shown on the BX's display. After successful download (approx. 2 to 3 minutes) the Bus Terminal Controller reboots automatically.



```
MonDec2 12:50:23 2004: Firmware-Update on serial interface COM1
MonDec2 12:50:23 2004: with File C:\Public\BX Controller\Firmware1.08\bx3100-0108.h31
MonDec2 12:50:23 2004: -> Start the Update with the green traffic light
```

Fig. 151: Status messages relating to the firmware update

8.2 CFC-Client*

With the CFC client, the BX device offers the option of copying the complete memory content of a BX Controller. The data from the memory are saved in a BIN file and can be loaded into an identical controller.

The CFC client runs on Windows 9x, NT, 2000 and XP and is completely independent of TwinCAT.

CFC client (https://infosys.beckhoff.com/content/1033/bx5200/Resources/zip/3238793995.zip) (Note: The DLLs have to be registered manually via regsvr32)

When the CFC client is started, the COM parameters have to be set first.

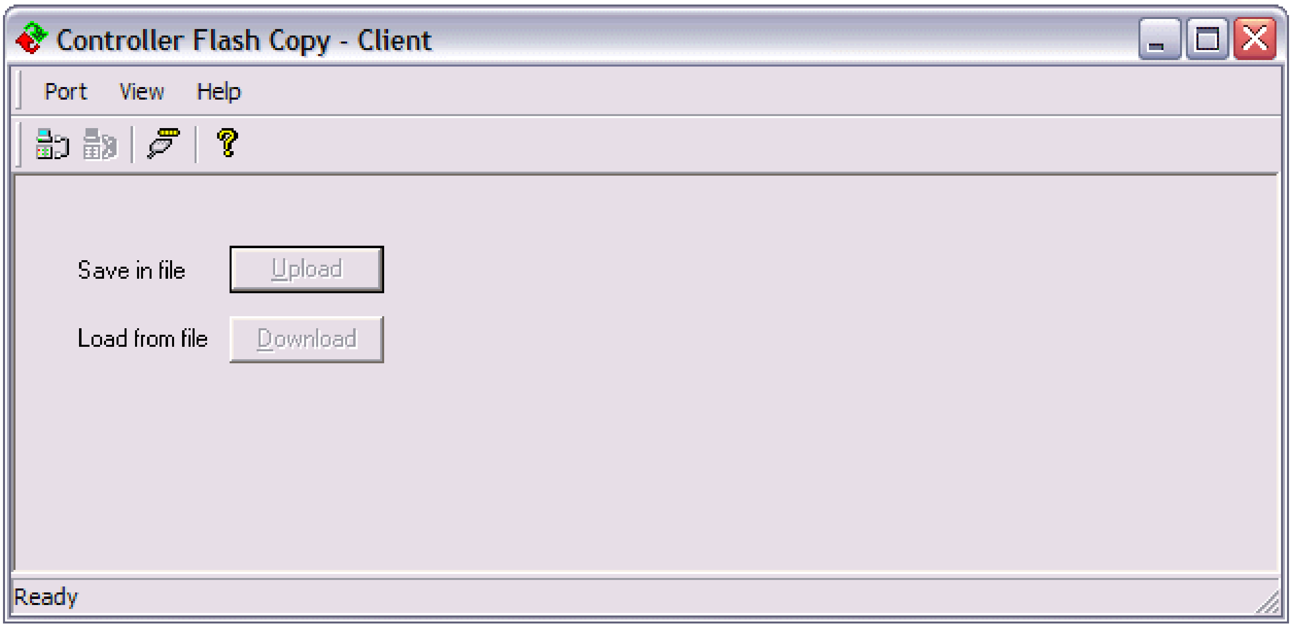


Fig. 152: CFC client

Port/Port Setup...

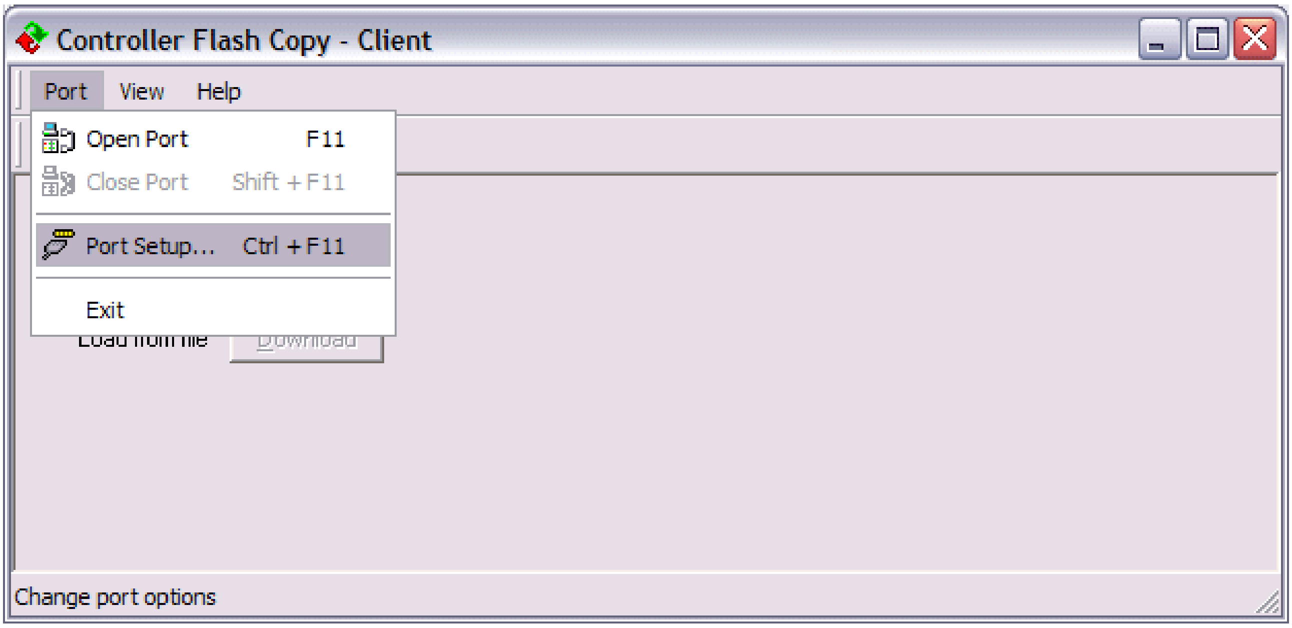
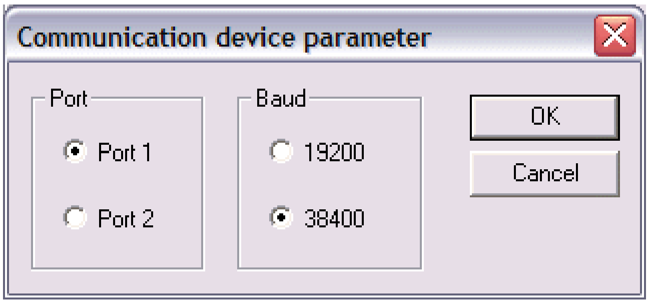


Fig. 153: Call the port setup

Select the required setting.


Communication device parameter

Port
- ● Port 1
- ○ Port 2

Baud
- ○ 19200
- ● 38400

Fig. 154: Set the COM parameters

Before opening the COM port, a physical connection to the BX Controller has to be established. Then open the COM port.

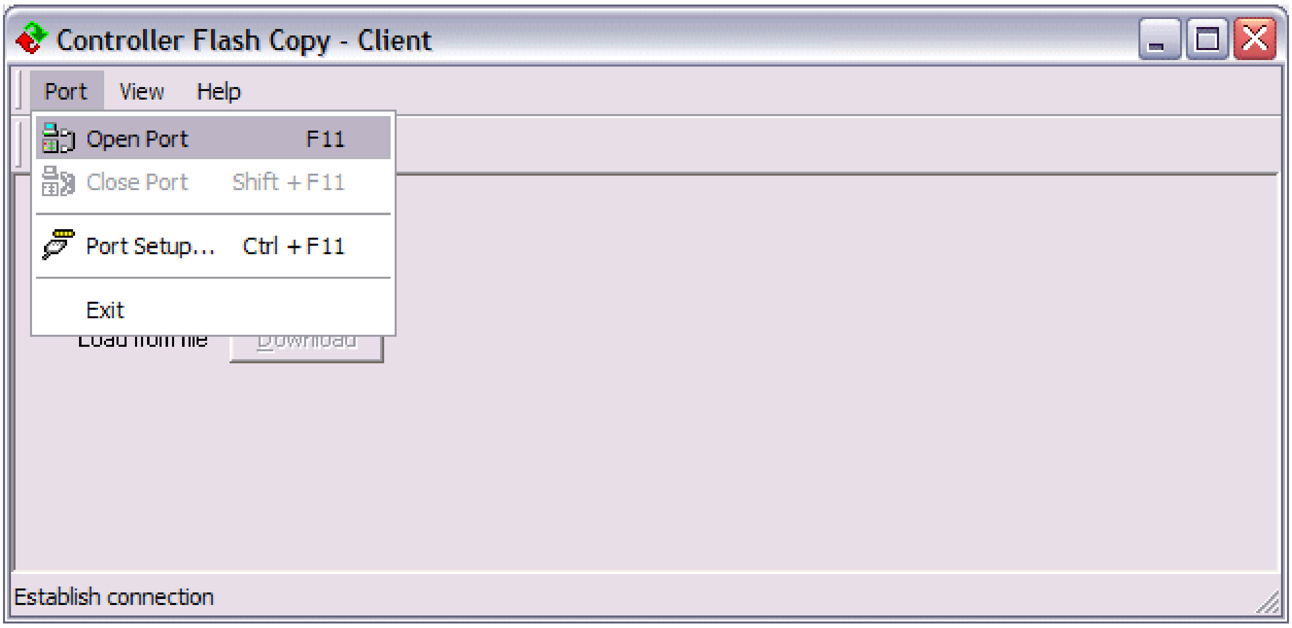
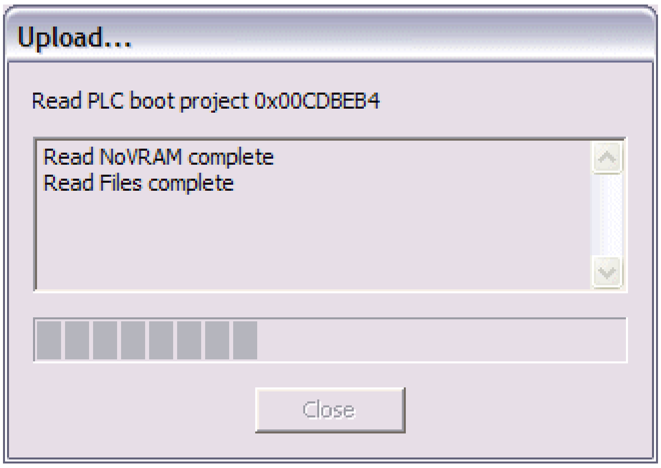


Fig. 155: Open the COM port

Use upload to create a copy of the BX memory and save it as a BIN file.

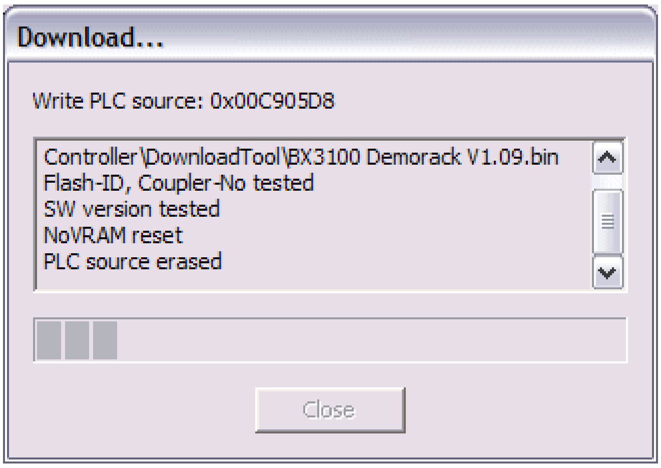


Read PLC boot project 0x00CDBEB4

```
Read NoVRAM complete
Read Files complete
```

Fig. 156: Upload the BX memory content to a BIN file

Use download to load the BIN file onto the BX controller. The controller starts automatically after a successful download.



Write PLC source: 0x00C905D8

```
Controller\DownloadTool\BX3100 Demorack V1.09.bin
Flash-ID, Coupler-No tested
SW version tested
NoVRAM reset
PLC source erased
```

Fig. 157: Downloading a BIN file to the BX controller

* The CFC server is available on the BX Controller range from firmware version 1.09.

8.3 Sample programs - overview

Denomination	Description
Display [▶ 120]	Example for controlling the display
Navigation switch [▶ 119]	Reading of the navigation switch from the PLC
Menu [▶ 91]	Example for own menu with navigation switch and display
RTC [▶ 85]	Example for reading the real-time clock (RTC) via function blocks
COM port - BK/BC8x00 master interface [▶ 125]	COM1 or COM2 interface as master with the BK8x00 protocol
COM port - BK8x00 slave interface [▶ 126]	COM1 or COM2 interface as slave with the BK8x00 protocol
COM port - Cimrex 12 [▶ 132]	Example for controlling a Cimrex 12 display via ModbusRTU
COM port - ModbusRTU slave [▶ 129]	Link of ModbusRTU Lib with the COM 1 or COM 2 interface of the BX
COM port - ModbusRTU master [▶ 130]	Link of ModbusRTU Lib with the COM 1 or COM 2 interface of the BX
COM port - RK512 protocol [▶ 134]	RK512 protocol via COM 1 or COM 2
COM port - text message via COM port [▶ 135]	Connecting a Siemens S35 mobile phone to the COM interface for sending text messages
COM port - COMlibV2 [▶ 131]	Sending and receiving strings with COMlibV2
SSB - Display [▶ 75]	Cimrex panel at SSB
SSB - AX2000 [▶ 73]	AX2000 at SSB
SSB - BK51x0 [▶ 71]	BK5120 at SSB
SSB - BX / BX communication [▶ 72]	Communication between BXs (via SSB)
SSB - IclA Drive [▶ 76]	IclA drive at SSB
SSB - Lenze Drive [▶ 82]	Lenze frequency converter at SSB

8.4 General operating conditions

The following conditions must be met in order to ensure flawless operation of the fieldbus components.

Environmental conditions

Operation

The components may not be used without additional protection in the following locations:

- in difficult environments, such as where there are corrosive vapors or gases, or high dust levels
- in the presence of high levels of ionizing radiation

Condition	Permissible range
Permissible ambient temperature during operation	See technical data
Permissible ambient temperature during operation	-25°C ... +85 °C
Installation position	variable
Vibration resistance	conforms to EN 60068-2-6
Shock resistance	conforms to EN 60068-2-27, EN 60068-2-29
EMC immunity	conforms to EN 61000-6-2
Emission	conforms to EN 61000-6-4

Transport and storage

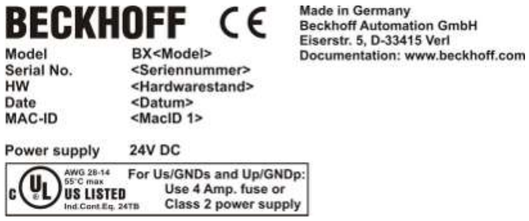
Condition	Permissible range
Permissible ambient temperature during storage	-25°C... +85 °C
Relative humidity	95 %, no condensation
Free fall	up to 1 m in the original packaging

Protection classes and types

Condition	Permissible range
Protection class in accordance with IEC 536 (VDE 0106, Part 1)	A protective conductor connection to the profile rail is necessary!
Protection class conforms to IEC 529	IP20 (protection against contact with a standard test finger)
Protection against foreign objects	Less than 12 mm in diameter
Protection against water	no protection

Component identification

Every supplied component includes an adhesive label providing information about the product's approvals.



BECKHOFF CE

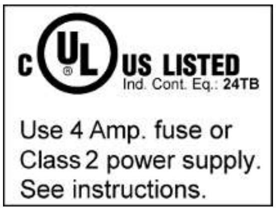
Made in Germany
Beckhoff Automation GmbH
Eiserstr. 5, D-33415 Verl
Documentation: www.beckhoff.com

Model	BX<Model>
Serial No.	<Seriennummer>
HW	<Hardwarestand>
Date	<Datum>
MAC-ID	<MacID 1>
Power supply	24V DC

c UL US LISTED
AWG 28-14
55°C max
Ind.Cont.Eq. 24TB
For Us/GNDs and Up/GNDp:
Use 4 Amp. fuse or
Class 2 power supply

Fig. 158: Name plate of a BX controller

The following information is printed on the label:

Printed item	Meaning for this label
Precise product identification	BX model
Supply voltage Us	24 V_{DC} (Use a 4 A fuse or a Class 2 power supply to meet UL requirements!)
Manufacturer	Beckhoff Automation GmbH
CE mark	Conformity mark
UL mark  c UL US LISTED Ind. Cont. Eq.: 24TB Use 4 Amp. fuse or Class 2 power supply. See instructions.	Mark for UL approval. UL stands for the Underwriters Laboratories Inc., the leading certification organization for North America, based in the USA. C = Canada, US = USA, UL file number: E172151
Production identification	Serial No.: Serial number HW: hardware version Date: Date of manufacture optional for BX9000 MAC-ID only

8.5 Approvals

Conformity mark

CE

Protection class

IP20 conforms to EN60529

8.6 Test standards for device testing

EMC

EMC immunity

EN 61000-6-2

Electromagnetic emission

EN 61000-6-4

Vibration / shock resistance

Vibration resistance

EN 60068-2-6

Shock resistance

EN 60068-2-27

8.7 Bibliography

German books

General fieldbus technology

- Gerhard Gruhler (publisher): **Fieldbus and Device Communication Systems**, Practical Know-how with Comparative Resources. Franzis Verlag 2001. 244 pages. ISBN 3-7723-5745-8

8.8 List of Abbreviations

CAN

Controller Area Network. Serial bus system standardized in ISO 11898 that is used as the basic technology for CANopen

CiA

CAN in Automation e.V.. An international association of manufacturers and users based in Erlangen, Germany.

COB

Communication Object. A CAN telegram with up to 8 data bytes.

COB-ID

Communication Object Identifier. Telegram address (not to be confused with the node address). CANopen uses the 11-bit identifier according to CAN 2.0A.

NMT

Network Management. One of the service primitives of the CANopen specification. Network management is used to initialize the network and to monitor nodes.

PDO

Process Data Object. A CAN telegram for the transfer of process data (e.g. I/O data).

RxPDO

Receive PDO. PDOs are always identified from the point of view of the device under consideration. Thus a TxPDO with input data from an I/O module becomes an RxPDO from the controller's point of view.

SDO

Service Data Object. A CAN telegram with a protocol for communication with data in the object directory (typically parameter data).

TxPDO

Transmit PDO (named from the point of view of the CAN node).

8.9 Support and Service

Beckhoff and their partners around the world offer comprehensive support and service, making available fast and competent assistance with all questions related to Beckhoff products and system solutions.

Beckhoff's branch offices and representatives

Please contact your Beckhoff branch office or representative for local support and service on Beckhoff products!

The addresses of Beckhoff's branch offices and representatives round the world can be found on her internet pages: https://www.beckhoff.com

You will also find further documentation for Beckhoff components there.

Beckhoff Support

Support offers you comprehensive technical assistance, helping you not only with the application of individual Beckhoff products, but also with other, wide-ranging services:

- support
- design, programming and commissioning of complex automation systems
- and extensive training program for Beckhoff system components

Hotline:	+49 5246 963 157
Fax:	+49 5246 963 9157
e-mail:	support@beckhoff.com

Beckhoff Service

The Beckhoff Service Center supports you in all matters of after-sales service:

- on-site service
- repair service
- spare parts service
- hotline service

Hotline:	+49 5246 963 460
Fax:	+49 5246 963 479
e-mail:	service@beckhoff.com

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

Phone:	+49 5246 963 0
Fax:	+49 5246 963 198
e-mail:	info@beckhoff.com
web:	https://www.beckhoff.com

Table of figures

More Information:
www.beckhoff.com/BX5200

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
Phone: +49 5246 9630
info@beckhoff.com
www.beckhoff.com