Application Note DK9222-0510-0045
Communication

Keywords
CAN
Small Controller
Networking
Master-less
Communication
Process data
Exchange
BC5150
BX5100

Three examples of communication between small local controllers on the basis of CAN

Peer-to-peer exchange of process data between small controllers can be done without requiring a higher-level master especially when the small controller's capacity is not even close to exhausted. This application example describes how small local controllers from Beckhoff can be networked on the basis of CAN and how they can exchange up to 1024 bits of process data per controller. Communication via CAN using the Beckhoff BC5150 'Compact' Bus Terminal Controller for CANopen is presented on the basis of three different examples; the principles of CAN communication are briefly outlined at the end of the text.

Small controllers are an economical solution for the automation of smaller manufacturing plants in the lower performance range of control technology. For example, the manufactured goods may be subject to regulations that require verification, the traceability of produced lots and the complete logging of the production sequence. Networking of the individual controllers is therefore indispensable.

Low-priced networking of small controllers

In the networking of small modular machines, the individual modules communicate with one another and exchange process data. Usually a higher-level controller functions as the fieldbus master. If the performance level of the small controllers is adequate for the application, then a fieldbus master in the form of a high-performance PLC is overcompensating and increases the price of the machine unnecessarily. Both the BC5150 'Compact' Bus Terminal Controller for CANopen and the BX5100 CANopen Bus Terminal Controller from Beckhoff are economical alternatives for the exchange of data via CAN without a fieldbus master. The sole prerequisite for the exchange of process data is that all connected BC5150 units enter into data exchange independently, i.e. without a master.

Application Note DK9222-0510-0045
Communication

Advantages of this solution

- very inexpensive – a cost-intensive master is not needed.
- simple cabling, CAN does not have excessive cabling requirements and is very resistant to interference.
- open and flexible programming in IEC 61131-3
- small, compact design

BC5150 | CANopen 'Compact' Bus Terminal Controller

Beckhoff Bus Terminal Controllers are small controllers that unite the functions of a Bus Coupler with a PLC. The BC5150 'Compact' Bus Terminal Controller for CANopen is a cost-optimized variant in a compact housing. The CANopen Controller offers automatic baud rate detection up to 1 Mbaud and two address selection switches for address assignment.

The Bus Terminal Controller is programmed using the TwinCAT programming system according to IEC 61131-3. The configuration/programming interface on the BC5150 is used to load the PLC program. If TwinCAT PLC software is used, the PLC program can also be loaded via the fieldbus.

Controller for local signal processing

The TwinCAT programming system for the BC5150 works independently of the manufacturer in accordance with IEC 61131-3; therefore, the PLC programs can be written in the five different programming languages (IL, FBD, LD, SFC, ST). TwinCAT offers extensive debugging functions (breakpoint, single step, monitoring etc.), which facilitate commissioning. It is also possible to perform adjustment and measurement of the cycle time.

In the default setting the inputs and outputs of the connected Bus Terminals are assigned to the controller. Each Bus Terminal can be configured in such a way that it exchanges data directly through the fieldbus with the higher-level automation device. Similarly, pre-processed data can be exchanged between the Bus Terminal Controller and the higher-level controller via the fieldbus.

BX5100 | CANopen Bus Terminal Controller

The BX5100 Bus Terminal Controller has a CANopen slave interface. It has automatic baud rate detection up to 1 Mbaud and an address selection switch for address assignment. Up to 16 TxPDOs and 16 RxPDOs can be exchanged with the controller. The controller is programmed via the COM1 or via the CANopen interface of the FC510x PC fieldbus card.

In terms of equipment and performance range, the Bus Terminal Controllers from the BX series are positioned between the Bus Terminal Controllers from the BC series and the Embedded PCs from the CX series. The main differences between BC and BX are the larger memory capacity and the extended interfaces of the BX. Additionally, two serial interfaces are integrated for programming and for the connection of further serial devices. The device itself features a backlit LCD display with two lines of

16 characters each, a 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 BX family is particularly suitable for modular machine concepts. Within a network, the Bus Terminal Controller can exchange data with other system components via the fieldbus interfaces. 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.

Controller for local signal processing

Like for all other Beckhoff controllers, TwinCAT automation software is the basis for parameterization and programming. The BX devices are programmed according to the powerful IEC 61131-3 standard in the programming languages IL, FBD, LD, SFC or ST. The familiar TwinCAT tools, such as the PLC programming interface, System Manager and TwinCAT Scope, are available to the user. Data is exchanged optionally via the serial port (COM1) or via the fieldbus through Beckhoff PC FC510x Fieldbus Cards.

The configuration is also carried out using TwinCAT. The fieldbus interface, the SSB bus and the real-time clock are configured and parameterized via the System Manager. The System Manager can read all connected devices and Bus Terminals. After parameterization the configuration is saved via the serial interface to the BX, from there it can also be read out again.

Tabular overview for the comparison of the two controllers (BC5150 and BX5100): see the chapter on the **principles of CAN communication**

Application Examples of master-less process data exchange

1. Peer-to-peer communication between two BC5150 controllers
2. Peer-to-peer communication between BC5150s and third-party CAN devices
3. Communication with several BC5150 controllers

Note: when using the program examples on the BX5100, the libraries must be exchanged.

Application Note DK9222-0510-0045

Communication

1. Peer-to-peer communication between two BC5150s

This variant of master-less communication is used when just two BC5150 controllers exchange data with each other. It is particularly suitable for the replacement of extensive I/O to I/O parallel cabling: The data can be collected and sent centrally via just one CAN cable. Up to 16 PDOs (identifiers) can be configured, each of which encompass 8 bytes and is transmitted in each direction. The volume of data corresponds to 16 PDOs x 8 bytes x 8 bits = 1024 bits or four analog values per PDO = 64 analog values. Since communication is event-controlled, the PDOs are transmitted only in the case of a change in value.

Structure example 1:

1 x BC5150 (Node ID 11) with 1 x KL2134, 2 x KL1408, 2 x KL2408, 1 x KL9010

1 x BC5150 (Node ID 33) with 1 x KL2134, 2 x KL1408, 2 x KL2408, 1 x KL9010

Functional scope of example 1:

Data are to be transmitted from a KL1408 to a KL2408 on another BC and in reverse. To this end, two BC5150s are connected via CAN. Configuration takes place via the System Manager; the identifiers (COB ID) must also be set accordingly here (see fig. 1). Furthermore the number of data must be specified – a maximum of 8 bytes per PDO is possible. The PLC is put into **operational mode** (OP mode) and CAN started by the command **ADS Write** (see fig. 2).

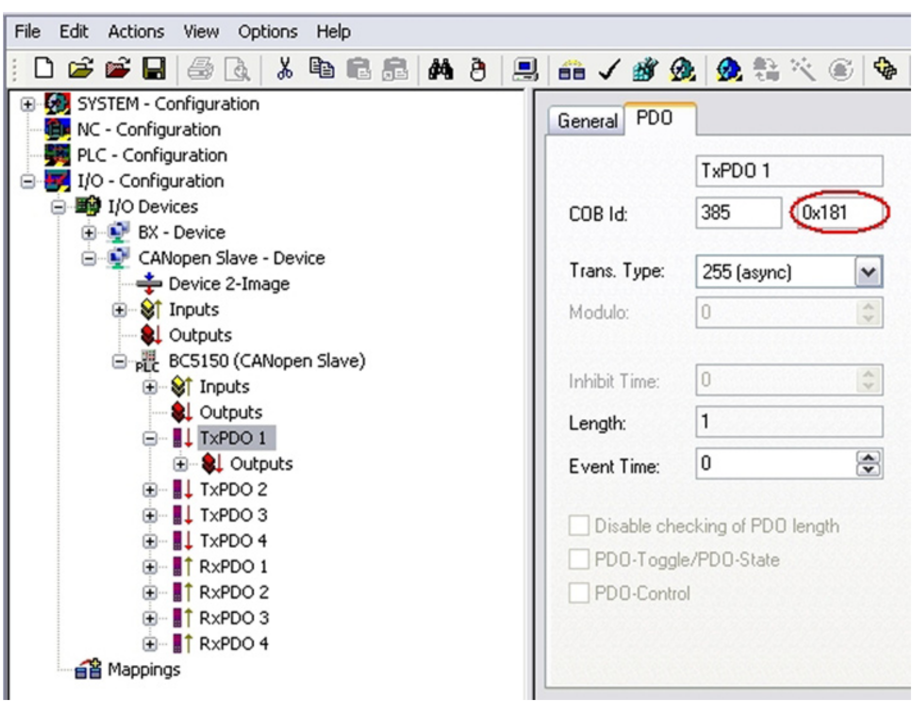

Fig. 1 View of the System Manager file: setting the identifiers (COB ID)

Application Note DK9222-0510-0045

Communication

ADS Write		Note:
NetID	Empty string	The local BC is addressed by an empty string in the parameter **NetID**.
PORT	16#64	
IndexGroup	16#0001_F880	
IndexOffset	0	
LEN	1	
Data	1	

Fig. 2 Parameters for setting the OP mode

The data of the digital terminals are mapped via the PLC, since a direct mapping of the digital signals to the PDOs is not possible. In the case of the BC5150 with the ID 11, the bits from the KL1408 are copied into the TxPDO2 in byte 7. The BC5150 with the ID 33 receives the data in RxPDO2 byte 7 and copies them to the KL2408. The BC5150 ID 33 copies the data from one of the connected KL1408s into TxPDO1 array 7. The BC5150 ID 11 receives the data in RxPDO1 byte 7 and copies them to the KL2408.

In order to check communication, PDO byte 0 will be used as a counter, which is incremented by 1 for each successful communication. The BC5150 at the opposite end copies only the counter value from the RxPDO into the TxPDO. If the counter values are identical, the counter of this controller is also incremented by 1. Successful communication is thus being monitored. CAN communication can be monitored visually by the flashing of the first LED on the KL2134: If it is not flashing, then the counter has stopped (possible causes: PLC not in RUN, CAN communication interrupted etc.).

NOTE: Since the counters increment in each 2nd or 3rd cycle and cause a CAN telegram, attention must also be paid to the bus load in the case of a low data rate and short PLC cycle time. It is sufficient for the monitoring of CAN communication to monitor only one PDO by means of a counter.

Application Note DK9222-0510-0045

Communication

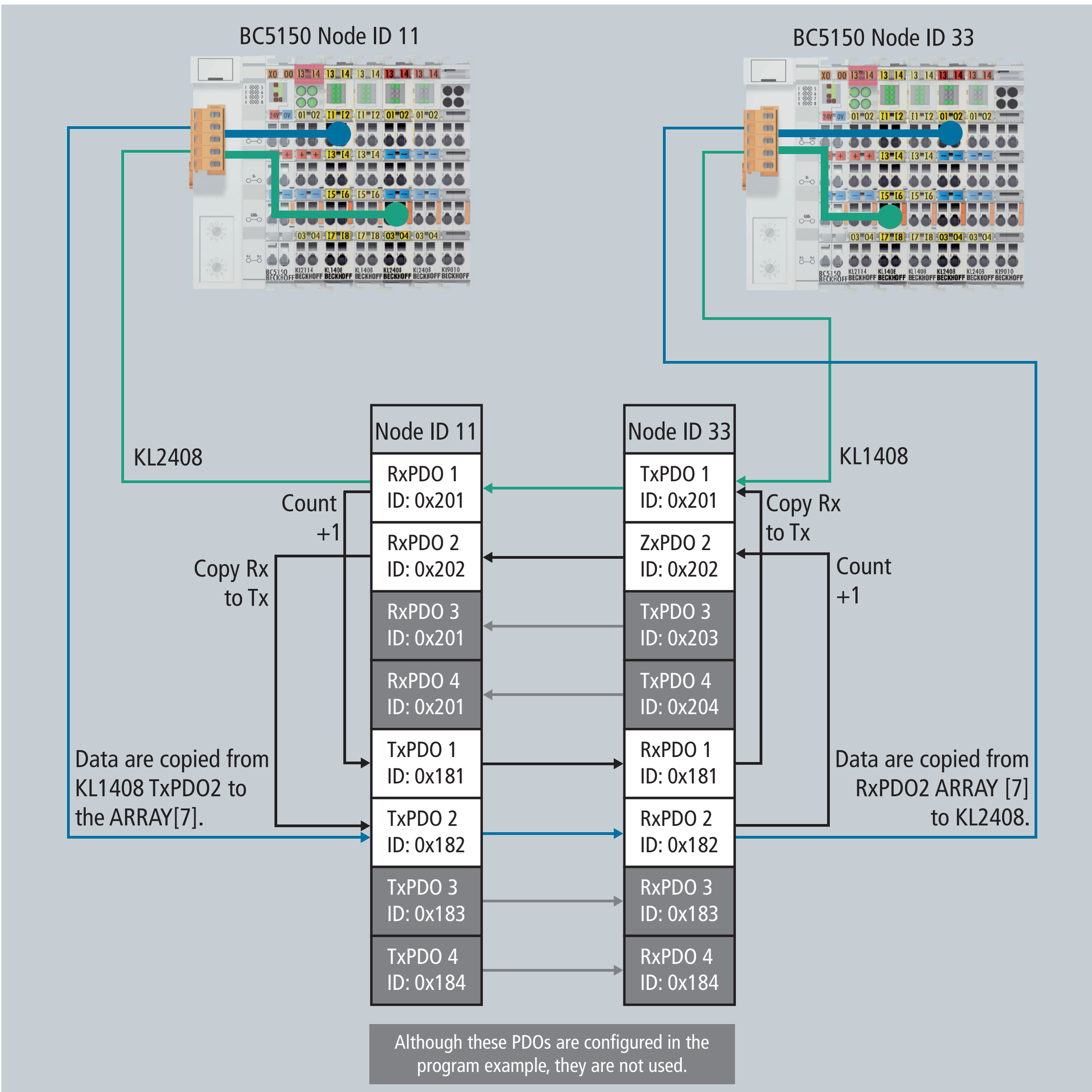

Fig. 3 Data exchange between KL2408 and KL1408 via CAN with integrated checking of communication by means of a counter block

Application Note DK9222-0510-0045
Communication

2. Peer-to-peer communication between BC5150s and third-party CAN devices

If a CAN (not CANopen) device is already present in an existing application and offers the corresponding interface, then the BC5150 can exchange process data with the connected device via this CAN interface.

Structure example 2:

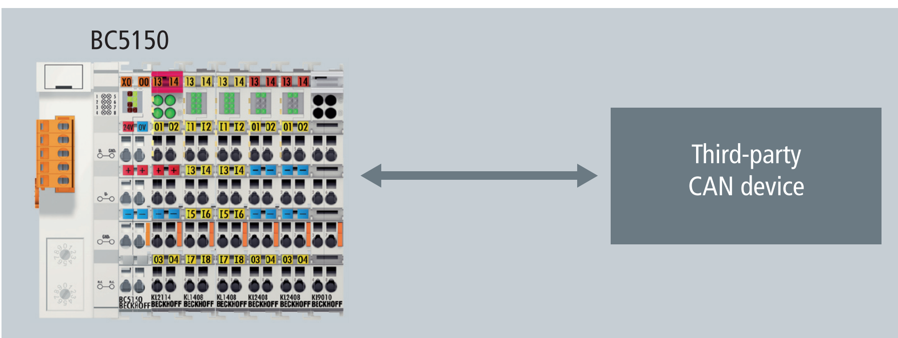

Fig. 4 Peer-to-peer communication between a BC5150 and a third-party CAN device

Conditions for example 2:

– the device communicates via CAN which is not the same as CANopen!
– the device is already integrated and the individual identifiers are known.
– see conditions for example 3

Application Note DK9222-0510-0045

Communication

3. Communication with several BC5150 controllers

Small controllers are an economical solution for the automation of smaller manufacturing plants in the lower performance range of control technology. The ideal here is a machine structure with three modules in one-dimensional expansion. Each module contains an independent BC5150; these communicate with one another peer-to-peer.

Up to a maximum of 32 devices may be connected to the CAN physics. Since 16 PDOs can be parameterized for each BC5150, 16 further BC5150s can be connected to a BC5150 if only one PDO is used per device. Data that are to be distributed equally to all devices can be sent via a broadcast. To do this, the same identifier must be configured in the RxPDOs in all connected BC5150s. Via this connection, for example, a clock or other data can be made available with a BC5150 (see fig. 5 TxPDO_3 black line).

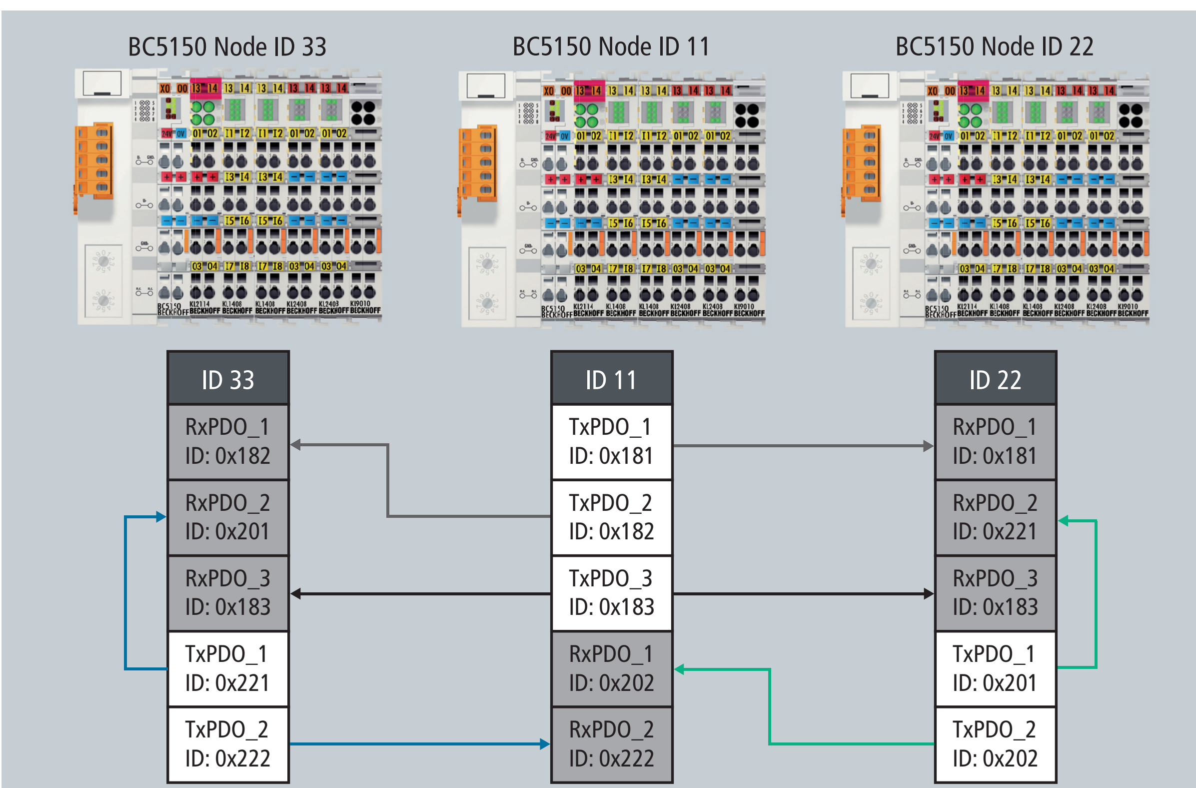

Fig. 5 Assignment of the PDOs in the case of three BC5150s

The important thing with this structure is careful planning and equally careful programming so that the identifier is not doubly assigned.

Application Note DK9222-0510-0045

Communication

Conditions:

- data are sent only in the case of a change.
- a maximum of 16 different identifiers with 8 bytes each are available (16 for sending data and 16 for receiving).
- monitoring of the nodes (devices still there?) can only take place by sending changed data. If the data content of the identifiers does not change, then no distinction can take place.

Sequence:

- the baud rate is permanently set in the System Manager and can be from 10 kbaud to 1 Mbaud.
- the BC sends a 0x700 + Node ID and all configured TxPDOs once when starting.
- after start-up, the data are only sent in the case of a change.
- the user must ensure that the load on the bus is limited. This can be done either by means of a high cycle time in the PLC or, in the case of a low cycle time, by means of appropriate timer blocks. In the case of low cycle times, the effect of the timer blocks is that the data are not sent in each cycle, but at an interval adapted to the CAN communication (a multiple of the cycle time).

Principles of CAN communication

CAN is a very widespread bus system that owes its beginnings to prevalence in the automotive industry. CAN is based on RS485 cabling, and therefore only two conductors are necessary for communication. CAN is a multi-master system that permits each node to send data with equal rights and access to the bus. Collisions are excluded by a special procedure, in order to prevent frame destruction when two devices transmit at the same time.

The transmission of a CAN frame is specified by the address: all connected devices ‘listen’ on the bus and only forward the data to the PLC if they are addressed. The devices are not addressed in a CAN frame; instead, the telegram itself is indexed. This indexing is known as an ‘identifier’. If, for example, a CAN device sends data with the identifier 0x18, then all devices recognize the identifier. If the identical identifier is projected in the device, then the data are passed on to the PLC. A data packet is also known as a Process Data Object (PDO), whereby TxPDO stands for transmitting (Tx – Transmit) and RxPDO for receiving (Rx – Receive). A frame with the identifier 0x18 is sent via the TxPDO_1 and is only read out in those devices in which an RxPDO_1 is configured with the identifier 0x18.

The maximum size of a CAN telegram (PDO) is 8 bytes. Using the BC5150 ‘Compact’ Bus Terminal Controller, 16 identifiers (16 x Tx and 16 x Rx) can be configured. The BX5100 Bus Terminal Controller contains double the number of Tx and Rx identifiers (32). The data volume is 128 bytes in the case of the BC5150 and 256 bytes in the case of the BX5100. The most important differences between the two small controllers are listed in table 1 below.

Application Note DK9222-0510-0045

Communication

Differences between BC5150 and BX5100		
	BC5150	**BX5100**
Number of CAN identifiers	16	32
PLC code	48 kByte	256 kByte
CPU performance for 1000 IL lines	3 ms	1 ms
Interfaces/hardware	- CAN - programming interface for KS2000-Z2(-USB) cable	- CAN - 1 x RS232 - 1 x RS232 oder RS485 - additional CAN interface - RTC (Real-time clock) - Display with 2 x 16 characters

Tab. 1 Comparison of BC5150 and BX5100

– CANopen „Compact“ Bus Terminal Controller **www.beckhoff.com/BC5150**

– CANopen Bus Terminal Controller **www.beckhoff.com/BX5100**

– I/Os for all common fieldbus systems **www.beckhoff.com/FieldbusComponents**