BECKHOFF New Automation Technology

Manual | EN

CX805x

Embedded-PC with CANopen

2025-02-18 | Version: 1.5

Table of contents

1 Notes on the documentation

This description is intended exclusively for trained specialists in control and automation technology who are familiar with the applicable national standards.
For installation and commissioning of the components, it is absolutely necessary to comply with the documentation and the following notes and explanations.
The qualified personnel is always obliged to use the currently valid documentation.

The responsible staff must ensure that the application or use of the products described satisfies all safety requirements, 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 notice.
No claims to modify 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 and licensed trademarks of Beckhoff Automation GmbH.
If third parties make use of designations or trademarks used in this publication for their own purposes, this could infringe upon the rights of the owners of the said designations.

Patents

The EtherCAT Technology is covered by the following patent applications and patents, without this constituting an exhaustive list:
EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
and similar applications and registrations in several other countries.

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

Copyright

1.1 Representation and structure of warnings

The following warnings are used in the documentation. Read and follow the warnings.

Warnings relating to personal injury:

⚠ DANGER
Hazard with high risk of death or serious injury.

⚠ WARNING
Hazard with medium risk of death or serious injury.

⚠ CAUTION
There is a low-risk hazard that can result in minor injury.

Warnings relating to damage to property or the environment:

NOTICE
There is a potential hazard to the environment and equipment.

Notes showing further information or tips:

This notice provides important information that will be of assistance in dealing with the product or software. There is no immediate danger to product, people or environment.

1.2 Documentation issue status

Version	Comment
0.1	• Preliminary version
1.0	• Foreword updated • Chapter *1-second UPS* added • Chapter *Operating system* added
1.1	• Foreword revised • Chapter "For your safety" added • ATEX warnings added
1.2	• Chapter "Transport and storage" added
1.3	• Chapter "FCC" added
1.4	• Warnings for hazardous areas revised • IECEx certificate added
1.5	• Information on hazardous areas adapted.

Image version CX8050

Firmware	Description
Build 2241	• First version

Image version CX8051

Firmware	Description
Build 2241	• First version

2 For your safety

Read the chapter on safety and follow the instructions in order to protect from personal injury and damage to equipment.

Limitation of liability

All the components are supplied in particular hardware and software configurations appropriate for the application. Unauthorized modifications and changes to the hardware or software configuration, which go beyond the documented options, are prohibited and nullify the liability of Beckhoff Automation GmbH & Co. KG.
In addition, the following actions are excluded from the liability of Beckhoff Automation GmbH & Co. KG:

- Failure to comply with this documentation.
- Improper use.
- Use of untrained personnel.
- Use of unauthorized replacement parts.

2.1 Intended use

The Embedded PC is designed for a working environment that meets the requirements of protection class IP20. This involves finger protection and protection against solid foreign objects up to 12.5 mm, but not protection against water. Operation of the devices in wet and dusty environments is not permitted, unless specified otherwise. The specified limits for electrical and technical data must be adhered to.

Potentially explosive atmospheres

The Embedded PC is suitable only for the following potentially explosive atmospheres:

1. For Zone 2 atmospheres in which gas is present as a combustible material. Zone 2 means that an explosive atmosphere does usually not occur during normal operation, or only for a short time.
2. For Zone 22 atmospheres in which dust is present as a combustible material. Zone 22 means that an explosive atmosphere in the form of a cloud does usually not occur during normal operation, or only for a short time.

The Embedded PC must be installed in a housing, which ensures protection class IP 54 for gas according to EN 60079-15. A housing with protection class IP 54 is required for non-conductive dust. IP 6X is required for conductive dust according to EN 60079-31.

Improper use

The Embedded PC is not suitable for operation in the following areas:

- In potentially explosive atmospheres, the Embedded PC may not be used in other zones except for 2/22 and not without a suitable housing.
- Areas with an aggressive environment, e.g. aggressive gases or chemicals.
- Living areas. In living areas, the relevant standards and guidelines for interference emissions must be adhered to, and the devices must be installed in housings or control boxes with suitable attenuation of shielding.

2.2 Staff qualification

All operations involving Beckhoff software and hardware may only be carried out by qualified personnel with knowledge of control and automation engineering. The qualified personnel must have knowledge of the administration of the Industrial PC and the associated network.

All interventions must be carried out with knowledge of control programming, and the qualified personnel must be familiar with the current standards and guidelines for the automation environment.

2.3 Safety instructions

The following safety instructions must be followed during installation and working with networks and the software.

Explosion protection

⚠ WARNING

Risk of explosion

Gases or dusts can be ignited in potentially explosive atmospheres. Read and follow the safety instructions to prevent deflagrations or explosions.

The Embedded PC must be installed in a housing, which ensures protection class IP54 for gas according to EN 60079-15. A housing with protection class IP54 is required for non-conductive dust. IP6X is required for conductive dust according to EN 60079-31.

Observe the temperature at the cable entry points into the housing. If the temperature during nominal operation is higher than 70 °C at the entry points or higher than 80 °C at the wire branching points, cables must be selected that are designed for these high temperatures and operation in potentially explosive atmospheres.

Tighten the screws of the fieldbus connectors firmly so that the connectors do not slip out due to vibrations. Only use RJ45 connectors with an intact latching lug.

Maintain the prescribed ambient temperature during operation. The permissible ambient temperature range during operation is 0 °C to +55 °C.

Take measures to prevent the rated operating voltage exceeding 119 V through short-term interference voltages.

Switch off the power supply and ensure that no explosive atmosphere occurs when:

- Bus Terminals are connected or removed,
- the Embedded PC is wired or cables are connected,
- DIP switches or ID switches are set,
- the front flap is opened,
- the MicroSD card or battery is replaced,
- the USB port behind the front flap is used.

Mounting

- Never work on live equipment. Always switch off the power supply for the device before installation, troubleshooting or maintenance. Protect the device against unintentional switching on.
- Observe the relevant accident prevention regulations for your machine (e.g. the BGV A 3, electrical systems and equipment).
- Ensure standard-compliant connection and avoid risks to personnel. Ensure that data and supply cables are laid in a standard-compliant manner and ensure correct pin assignment.
- Observe the relevant EMC guidelines for your application.
- Avoid polarity reversal of the data and supply cables, as this may cause damage to the equipment.
- The devices contain electronic components, which may be destroyed by electrostatic discharge when touched. Observe the safety precautions against electrostatic discharge according to DIN EN 61340-5-1/-3.

Working with networks

- Restrict access to all devices to an authorized circle of persons.
- Change the default passwords to reduce the risk of unauthorized access.
- Protect the devices with a firewall.

- Apply the IT security precautions according to IEC 62443, in order to limit access to and control of devices and networks.

Working with the software

- The sensitivity of a PC against malicious software increases with the number of installed and active software.
- Uninstall or disable unnecessary software.

Further information about the safe handling of networks and software can be found in the Beckhoff Information System:
http://infosys.beckhoff.com

Document name
IPC Security Guideline

2.4 Notes on information security

The products of Beckhoff Automation GmbH & Co. KG (Beckhoff), insofar as they can be accessed online, are equipped with security functions that support the secure operation of plants, systems, machines and networks. Despite the security functions, the creation, implementation and constant updating of a holistic security concept for the operation are necessary to protect the respective plant, system, machine and networks against cyber threats. The products sold by Beckhoff are only part of the overall security concept. The customer is responsible for preventing unauthorized access by third parties to its equipment, systems, machines and networks. The latter should be connected to the corporate network or the Internet only if appropriate protective measures have been set up.

In addition, the recommendations from Beckhoff regarding appropriate protective measures should be observed. Further information regarding information security and industrial security can be found in our https://www.beckhoff.com/secguide.

Beckhoff products and solutions undergo continuous further development. This also applies to security functions. In light of this continuous further development, Beckhoff expressly recommends that the products are kept up to date at all times and that updates are installed for the products once they have been made available. Using outdated or unsupported product versions can increase the risk of cyber threats.

To stay informed about information security for Beckhoff products, subscribe to the RSS feed at https://www.beckhoff.com/secinfo.

3 Transport and storage

Transport

NOTICE

Short circuit due to moisture

Moisture can form during transport in cold weather or in the event of large temperature fluctuations.

Avoid moisture formation (condensation) in the Embedded PC, and leave it to adjust to room temperature slowly. If condensation has occurred, wait at least 12 hours before switching on the Embedded PC.

Despite the robust design of the unit, the components are sensitive to strong vibrations and impacts. During transport the Embedded PC must be protected from

- mechanical stress and
- use the original packaging.

Table 1: Weight and Dimensions.

	CX80xx
Weight	180 g
Dimensions (W x H x D)	64 mm x 100 mm x 73 mm

Storage

- The battery should be removed if the Embedded PC is stored at temperatures above 60 °C. The battery should be stored separate from the Embedded PC in a dry environment at a temperature between 0 °C and 30 °C.
  The preset date and time are lost if the battery is removed.
- Store the Embedded PC in the original packaging.

4 Product overview

4.1 CX80xx - System overview

CX80xx is a device family of programmable controllers with 32-bit ARM-based CPU, which can be used for processing of PLC programs or as slave devices for higher-level fieldbus systems. Unlike with the non-programmable EtherCAT couplers of the EK series, which only act as gateway between the associated fieldbus system and the connected EtherCAT terminals, the CX80xx is programmable and able to run its own control program.

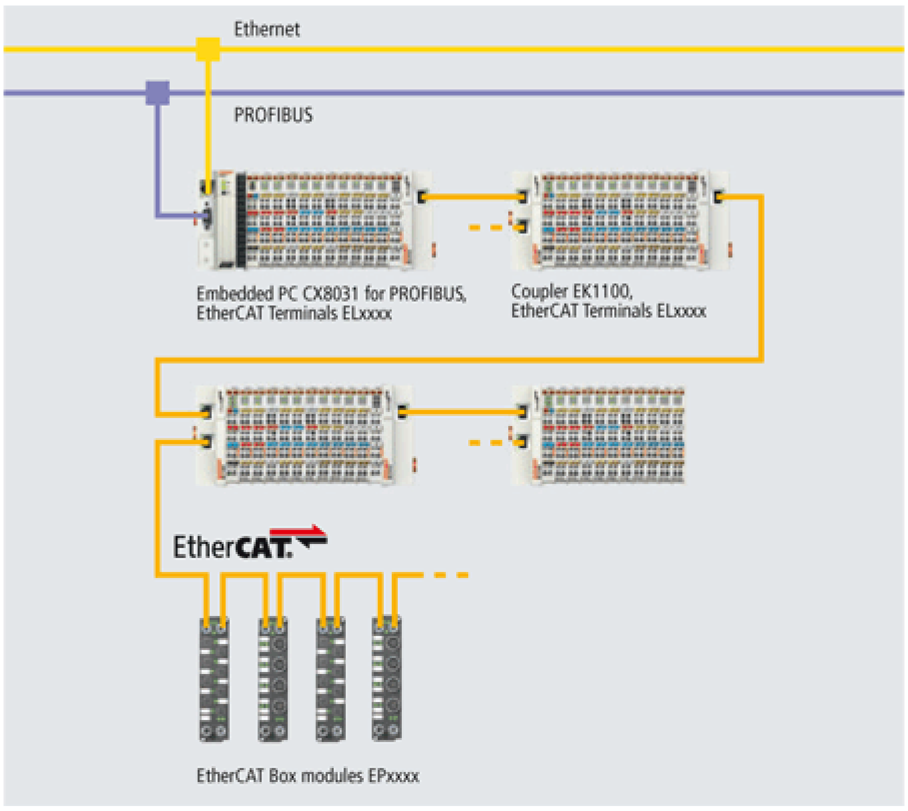
The devices from the CX80xx series represent a further development of the well-known and proven 16-bit microcontroller-based Bus Terminal Controllers from the BC and BX series including more efficient 32-bit processors. As with the BC/BX, it is also ensured in the case of the CX80xx that the control and the local program continue to be executed in the case of interruption of the higher-level fieldbus system. The CX80xx devices can therefore be used as local controllers. Alternatively, Bus Terminals (K-bus) or EtherCAT Terminals (E-bus) can be connected; the CX80xx automatically recognizes which terminal system is connected during the start-up phase. The use of EtherCAT gives rise to further options, such as the realization of different topologies, the integration of further bus systems such as CANopen, PROFIBUS and PROFINET and – with the EtherCAT Box Modules – connection to the IP67 world.

Like all CX products, the CX80xx devices are programmed and commissioned via the Ethernet interface, which can, of course, also be used for connection of the control system with a regular network. Some of the Embedded PCs have further Ethernet interfaces with switch functions, so that a linear "daisy chain" topology can be constructed inexpensively, without additional hardware. The other connections on the lower plug level are fieldbus-specific. Under the cover at the upper housing level there is an exchangeable button cell for date and time, a set of DIP switches for setting function modes, a slot for Micro-SD Flash memory cards and a type B USB connection. Thanks to their low power consumption, the devices are fanless.

Microsoft Windows CE is used as the operating system. In the absence of a monitor port, the operating system and its "virtual" display can only be accessed via the network. As for all other Beckhoff devices, the TwinCAT software is used for system configuration and the programming of the PLC functionality. The CX80xx target device features a pre-installed TwinCAT PLC runtime environment. All software required for operating the device, including the operating system, the TwinCAT files and user files and data, is stored on the MicroSD Flash card. This simplifies exchange in the case of service. Commercial card readers can be used to access the card data. The size of the MicroSD Flash card (e.g. 512 MB) can be chosen depending on the application and the quantity of data to be stored.

The CX80xx device family features an integrated, capacitive 1-second UPS, which in the event of a failure of the supply voltage provides sufficient energy for saving persistent data. Important data are thus preserved in a non-volatile manner without battery backup.

With a high-performance but nevertheless energy-saving 32-bit ARM processor, EtherCAT as I/O bus and TwinCAT PLC with extensive PLC libraries, the Embedded Controllers from the CX80xx series represent high-performance and versatile controllers with slave fieldbus connection.


Ethernet
PROFIBUS
Embedded PC CX8031 for PROFIBUS,
EtherCAT Terminals ELxxxx
Coupler EK1100,
EtherCAT Terminals ELxxxx
EtherCAT
EtherCAT Box modules EPxxxx

Fieldbus interface

The variants from the CX80xx series differ by their fieldbus interfaces. Various versions cover the most important fieldbus systems:

- CX8010: EtherCATSlave
- CX8030: PROFIBUS DP Master
  CX8031: PROFIBUS DP Slave
- CX8050: CAN Master
  CX8051: CANopen Slave
- CX8080: RS232/485
- CX8090: Ethernet (RT-Ethernet, EAP, ModbusTCP, TCP/IP, UDP/IP, Web Services)
- CX8091: BACnet IP/OPC UA
- CX8093: PROFINET RT Device (Slave)
- CX8095: Ethernet/IP Slave
- CX8097: Sercos III Slave

Programming

The CX80xx controller is programmed according to the high-performance 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.

Configuration

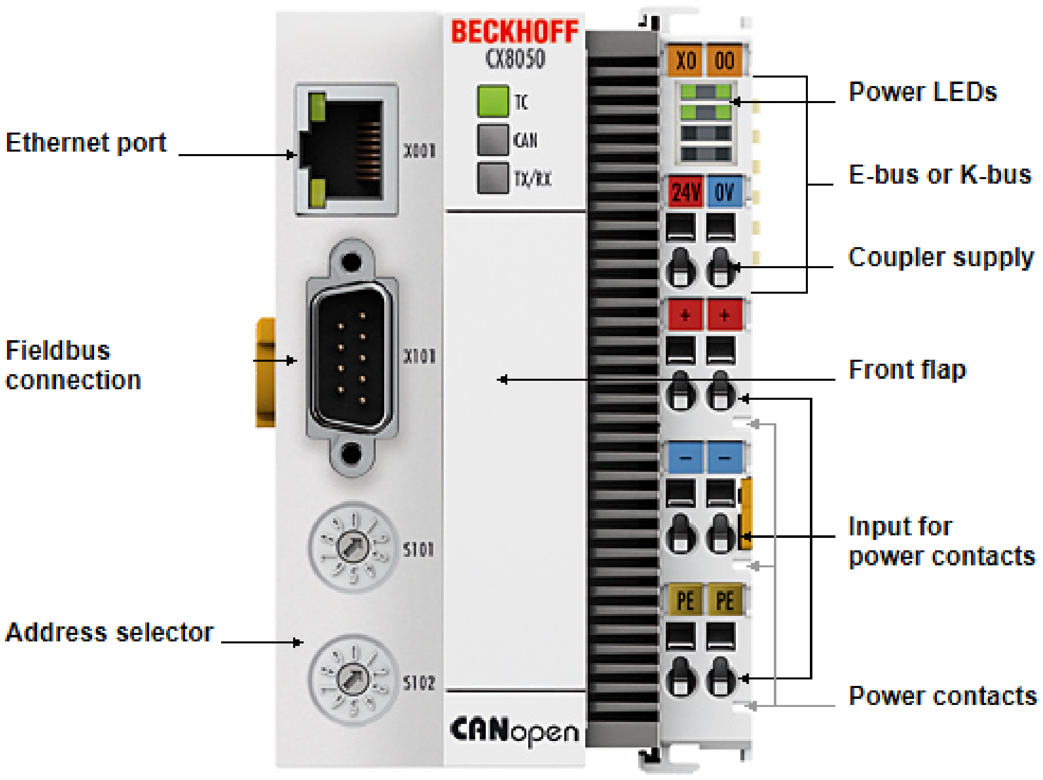
The configuration is also carried out using TwinCAT. The fieldbus interface and the real-time clock can be configured and parameterized via the System Manager. The System Manager can read all connected devices and Bus Terminals. The configuration is stored on the CX after the parameterization. The configuration thus created can be accessed again later.

4.2 CX8050, CX8051 - Introduction

In the basic version the CX80xx contains a 512 MB MicroSD Card. A fieldbus interface, an Ethernet interface and a K-bus or E-bus interface are included as standard.

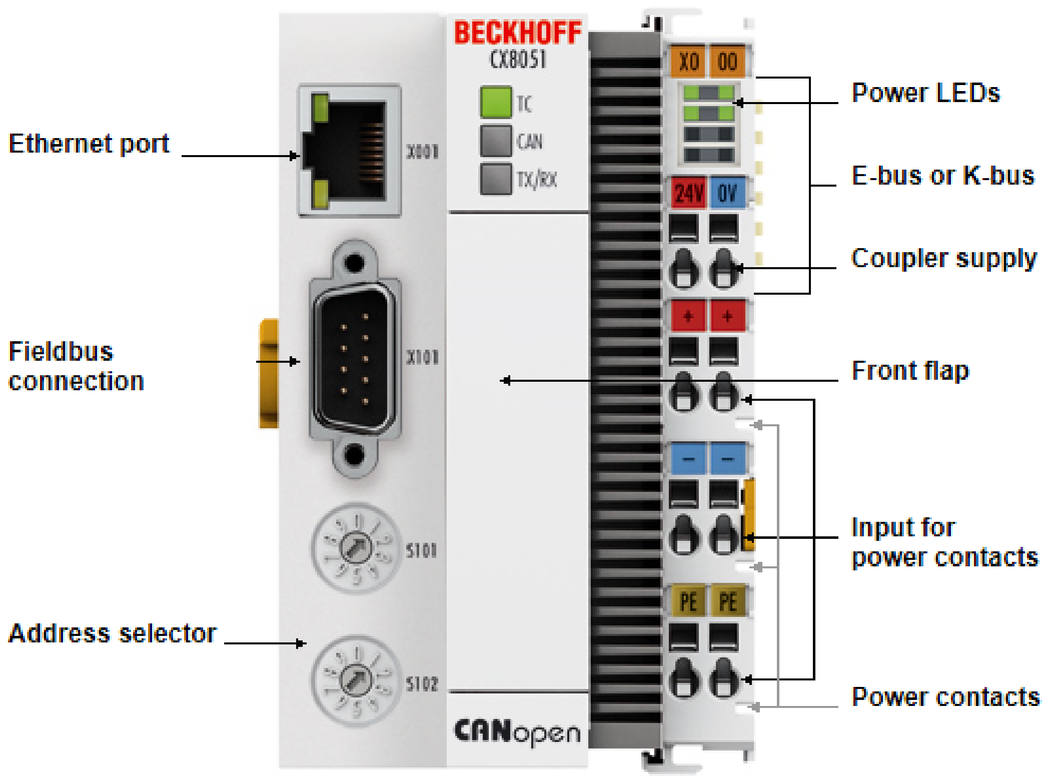
The smallest task time to be used is 1 ms (a task time of 10 ms to 50 ms is recommended for the I/O data, further tasks can also be set slower). When using short cycle times, the total system utilization rate is to be observed. If too short a cycle time is selected, the Web visualization and remote desktop may operate very slowly or cause timeouts. The user is responsible for projecting and configuring his system such that it is not overloaded.

CX8050



The CX8050 is a controller with a CANopen master interface. Apart from functioning as a CANopen master, CAN-Layer-2 communication is alternatively also possible. Alternatively K-bus or E-bus terminals can be series-connected; the CX8050 automatically detects which system is connected during the start-up phase. The controller is programmed via the Ethernet interface.

CX8051



The CX8051 is a control system with CANopen slave interface. The CANopen address is set via two rotary selection switches. The CX8051 offers automatic baud rate detection. Alternatively K-bus or E-bus terminals can be series-connected; the CX8051 automatically detects which system is connected during the start-up phase. The controller is programmed via the Ethernet interface.

4.3 Technical data

Technical data	CX8050	CX8051
Processor	Arm9™, 400 MHz	
Internal main memory	64 MB RAM (internal, not expandable)	
Operating system	Microsoft Windows CE 6.0	
Web-based Management	yes	
Flash memory	microSD card (ATP) 512 MB (optional 1, 2, 4, 8 GB)	
Interfaces	1 x USB device (behind the front flap) 1 x RJ45 Ethernet, 10/100 Mbit/s (ADS over TCP/IP) 1 x D-Sub RS485 CAN	
Protocols	CANopen master or CAN master	CANopen slave
Interface for I/O terminals	K-bus or E-bus, automatic recognition	
Process data on the K-bus	max. 2 kB input data max. 2 kB output data	
Diagnostic LED	1 x power, 1 x TC status, 2 x bus status	
Clock	internal battery-backed clock (RTC) for time and date (battery exchangeable)	
Operating system	Microsoft Windows CE 6[1)]	
Control software	TwinCAT PLC runtime (from version 2.11 R3)	
Programming	TwinCAT PLC	
Programming languages	IEC 61131-3	
Online Change	Yes	
Up/download code	yes/yes	
Power supply	24 V_{DC} (-15%/+20%)	
1-second UPS	integrated (1 MB on microSD card)	
Power supply for I/O terminals (K-bus or E-bus)	max. 2 A	
Power contact current load	max. 10 A	
Max. power consumption	3 W	
Max. power consumption E-bus/K-bus	10 W (5 V/max. 2 A)	
Dielectric strength	500 V (supply / internal electronics)	
Dimensions (W x H x D)	64 mm x 100 mm x 73 mm	
Weight	approx. 180 g	
Permissible ambient temperature during operation	0° C ... +55 °C	
Permissible ambient temperature during storage	-25° C ... +85° C see notes under: Transport and storage [▶ 11]	
Installation position	see chapter Installation positions	
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 rating	IP20	

[1)] The support period for the Microsoft Windows Embedded CE 6 operating system has already expired. Security updates can therefore no longer be provided.

4.4 Technical data - CAN

CX8050

Technical data - CANopen	**CX8050**
Fieldbus	CANopen
Data transfer rate	10, 20, 50, 100, 125, 250, 500, 800, 1.000 kBaud
Bus interface	1 x D-sub socket, 9-pin
Bus devices	max. 64
max. process image	512 Tx PDOs / 512 Rx PDOs
Autobaud	-
Galvanic isolation	Yes
Protokoll	
CANopen Slave	-
CAN (virtual slave)	-
ADS Interface	Yes (only via Ethernet)
Services	
CAN Layer 2	Yes
CAN 2.0A	Yes
CAN 2.0B	Yes, can only be used via the CAN interface
Diagnosis/Status/Alarm	
TC LED	Yes, green/red
BF LED	Yes, green/red
DIA LED	Yes, green/red
diagnostic notice	Yes

CX8051

Technical data - CANopen	**CX8051**
Fieldbus	CANopen
Data transfer rate	10, 20, 50, 100, 125, 250, 500, 800, 1.000 kBaud
Bus interface	1 x D-sub socket, 9-pin
Extendable process image	Up to 3 virtual slaves in addition
max. process image	4 slaves x (16 Tx PDOs / 16 Rx PDOs (8 byte per PDO))
Autobaud	Yes
galvanic isolation	Yes
Protokoll	
CANopen Slave	Yes
CAN (virtual slave)	4 (3 virtual CANopen nodes)
ADS Interface	Yes (only via Ethernet)
Services	
CAN Layer 2	No
CAN 2.0A	after CANopen
CAN 2.0B	No
Diagnosis/Status/Alarm	
TC LED	Yes, green/red
BF LED	Yes, green/red
DIA LED	Yes, green/red
diagnostic notice	Yes

4.5 CX80xx - MicroSD cards

⚠ CAUTION

MicroSD card as ignition source in potentially explosive atmospheres

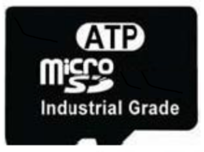
Gases or dusts can be ignited by a spark discharge when the MicroSD card is inserted or removed.

Switch off the power supply and wait until the 1-second UPS has discharged. Ensure that there is no explosive atmosphere before you insert or remove the MicroSD card.

In the basic version the CX80xx contains a MicroSD card with 512 MB. You can order it as an option with larger cards (up to 8 GB).

The cards employed are SLC memory with extended temperature range for industrial applications. Use exclusively MicroSD cards approved by Beckhoff.

Example of a MicroSD card:



Order identifier	Capacity	Description
CX1900-0123	1 GB	MicroSD card (SLC memory) with extended temperature range for industrial applications instead of the 512 MB card (ordering option)
CX1900-0125	2 GB	
CX1900-0127	4 GB	
CX1900-0129	8 GB	
Order identifier	Capacity	Description
CX1900-0122	512 MB	MicroSD card (SLC memory) with extended temperature range for industrial applications as spare part.
CX1900-0124	1 GB	
CX1900-0126	2 GB	
CX1900-0128	4 GB	
CX1900-0130	8 GB	

Further Information: https://www.beckhoff.com/CX8000

5 Mounting and wiring

5.1 Mounting

⚠ CAUTION

Application in potentially explosive atmospheres

The Embedded PC must be fitted with a suitable housing and suitable cables for use in potentially explosive atmospheres.

In potentially explosive atmospheres, the Embedded PC must always be installed in a housing with the correct protection class, and suitable cables must be used.

Install the Embedded PC in a housing or a control cabinet, if it is to be used in potentially explosive atmospheres.

Table 2: Embedded PC installation, requirements for housing in potentially explosive atmospheres.

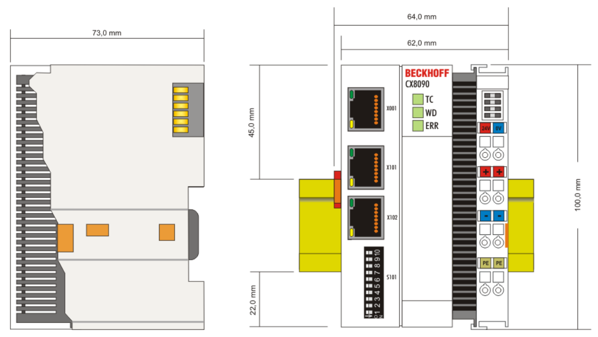
Ex area	Flammable substance	Protection class
Zone 2	Gas	IP 54, according to EN 60079-15
Zone 22	dust, non-conductive	IP 54, according to EN 60079-31
	dust, conductive	IP 6x, according to EN 60079-31

Observe the temperature at the cable entry points into the housing. If the temperature during nominal operation is higher than 70 °C at the entry points or higher than 80 °C at the wire branching points, cables that are designed for these higher temperatures and Ex operation must be used.

5.1.1 Dimensions

The following drawings show the dimensions of the CX80xx Embedded PCs.

Dimensions

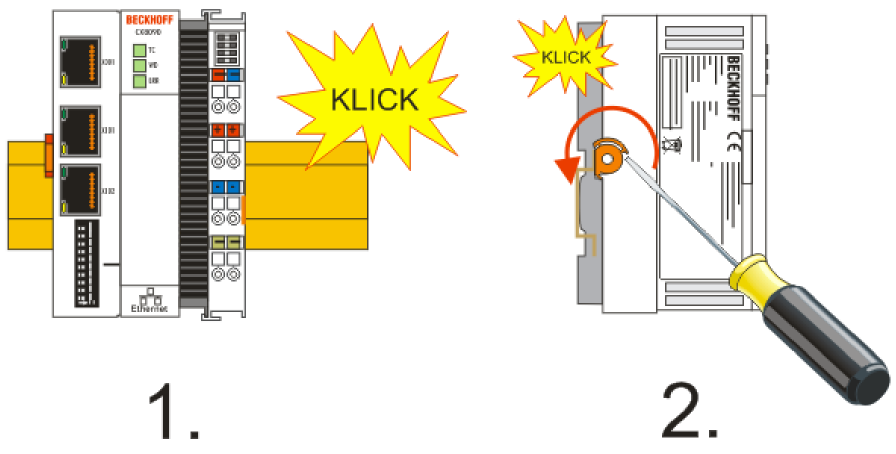


Drawings in various CAD formats can be found at: https://www.beckhoff.com

5.1.2 Installation on mounting rails

Snapping onto the mounting rail

The CX80xx can simply be snapped onto the mounting rail. To this end simply position the block on the mounting rail and push it slightly until it engages on the right-hand side. The is indicated by a distinct click. Use a screwdriver to push up the lock on the left-hand side, thereby turning it and causing it to engage audibly.



NOTICE

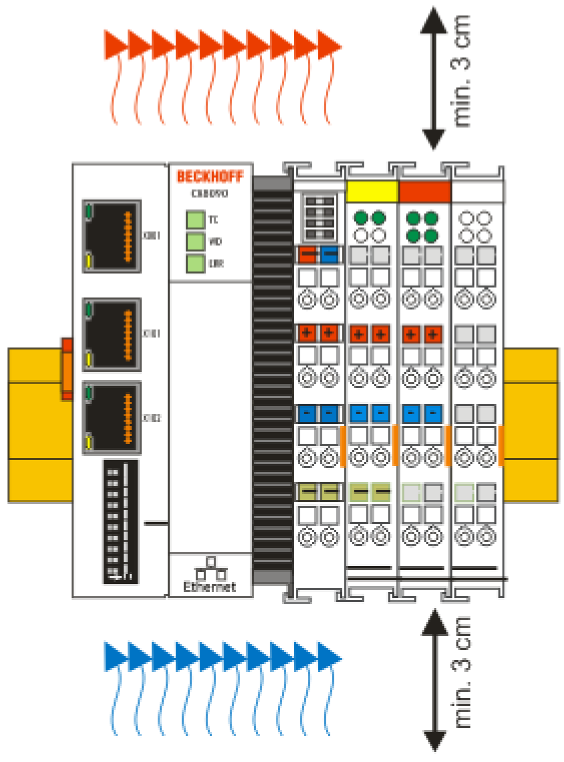
Avoid damage!

Do not force the module or apply excessive pressure!

Permissible installation positions and minimum distances

Installation positions

Installation position up to 55 °C



NOTICE

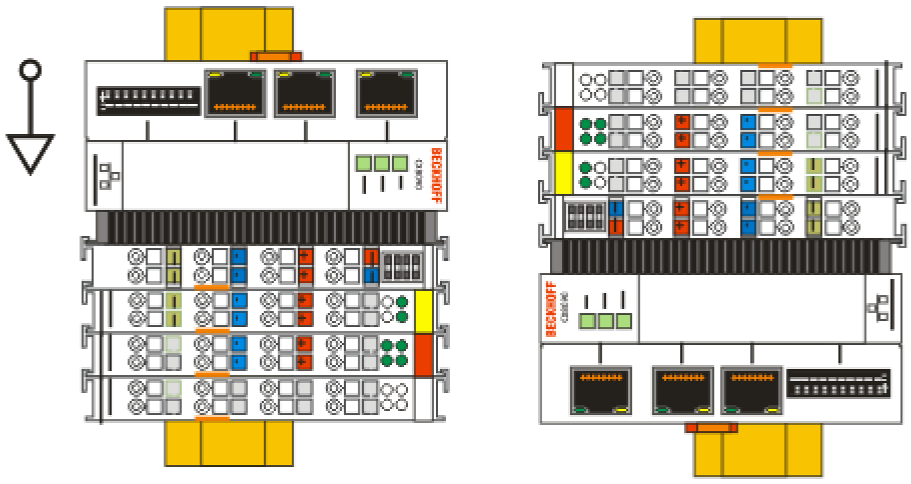
Comply with the permitted installation position and minimum distances!

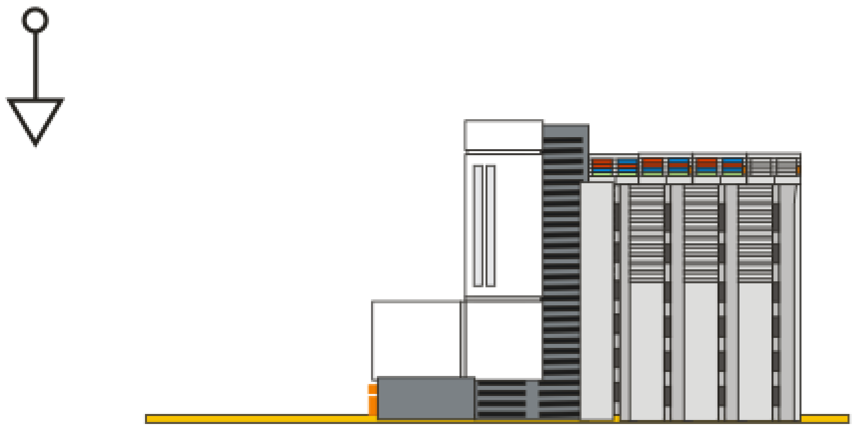
The maximum ambient temperature for CPU modules mounted on a DIN rail is 55°C. The orientation in which the device is fitted must be selected in such a way that cooling air can flow vertically through the ventilation holes. The images show the permitted and restricted installation positions. Mounting must provide a clearance of 30 mm both above and below a CX80xx device combination to ensure adequate ventilation of the base CPU module and the power supply unit.

The high performance and the compact design of the CX80xx systems may result in increased heat generation. The heat is dissipated via a passive ventilation system. This system requires the unit to be mounted correctly. Ventilation openings are located at the top and bottom of the housing. The system therefore has to be installed horizontally. This ensures optimum air flow.

Installation positions with reduced temperature range up to 45 °C

Other installation positions are permitted with a temperature range up to 45 °C.





5.2 Wiring

5.2.1 Power supply

⚠ WARNING

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

Bring the CX80xx into a safe, de-energized state before starting assembly, disassembly or wiring!

⚠ CAUTION

Connections as ignition source in potentially explosive atmospheres

Gases or dusts can be ignited by a spark discharge when the Embedded PC is wired.

Switch off the power supply and wait until the 1-second UPS has discharged. Ensure that there is no explosive atmosphere before you wire the Embedded PC and connect or disconnect Bus Terminals.

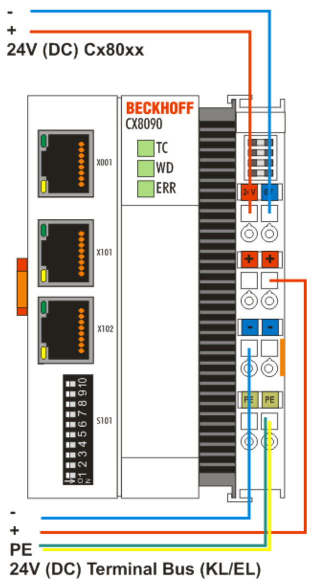
This power supply unit is equipped with an I/O interface, which permits connection of the Beckhoff Bus Terminals. The power is supplied via the upper spring-loaded terminals with the designation 24 V and 0 V.

The supply voltage supplies the CX system and, via the terminal bus, the Bus Terminals with a voltage of 24 V_{DC} (15 %/+20 %). The dielectric strength of the power supply is 500 V. Since the Terminal Bus (K- and E-bus) only transfers data, a separate power supply is required for the Bus Terminals. This is provided by means of the power contacts, which are not connected to the power supply. Only 24 V DC may be connected to the power contacts; the maximum current load of the power contacts is 10 A.

⚠ CAUTION

Power contact PE

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



Requirements for the power supply (24 V)

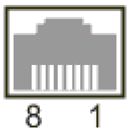
In order to guarantee the operation of the CPU (CX80xx module) and the terminal strand in all cases, the power supply must supply 2.0 A at 24 V.

LED

If the power supply unit is connected correctly and the power supply is switched on, the two upper LEDs in the terminal prism are green. The left LED (Us) indicates the CPU supply. The right LED (Up) indicates the terminal supply. The other LEDs indicate the Terminal Bus status. A detailed description of the LEDs can be found in section "LED troubleshooting".

5.2.2 Ethernet

Ethernet connections



Assignment of the RJ45 interface, port 1

X001

PIN	Signal	Description
1	TD +	Transmit +
2	TD -	Transmit -
3	RD +	Receive +
4	connected	reserved
5		
6	RD -	Receive -
7	connected	reserved
8		

Assignment of the RJ45 interface, port 2

X001 / X002

PIN	Signal	Description
1	TD +	Transmit +
2	TD -	Transmit -
3	RD +	Receive +
4	connected	reserved
5		
6	RD -	Receive -
7	connected	reserved
8		

Transmission standards

10Base5

The transmission medium for 10Base5 consists of a thick coaxial cable ("yellow cable") with a max. data transfer rate of 10 Mbaud arranged in a line topology with branches (drops) each of which is connected to one network device. Because all the devices are in this case connected to a common transmission medium, it is inevitable that collisions occur often in 10Base5.

10Base2

10Base2 (Cheaper net) is a further development of 10Base5, and has the advantage that the coaxial cable is cheaper and, being more flexible, is easier to lay. It is possible for several devices to be connected to one 10Base2 cable. It is frequent for branches from a 10Base5 backbone to be implemented in 10Base2.

10BaseT

Describes a twisted pair cable for 10 Mbaud. The network here is constructed as a star. It is no longer the case that every device is attached to the same medium. This means that a broken cable no longer results in failure of the entire network. The use of switches as star couplers enables collisions to be reduced. Using full-duplex connections they can even be entirely avoided.

100BaseT

Twisted pair cable for 100 Mbaud. It is necessary to use a higher cable quality and to employ appropriate hubs or switches in order to achieve the higher data rate.

10BaseF

The 10BaseF standard describes several optical fiber versions.

Short description of the 10BaseT and 100BaseT cable types

Twisted-pair copper cable for star topologies, where the distance between two devices may not exceed 100 meters.

UTP

Unshielded twisted-pair
This type of cable belongs to category 3, and is not recommended for use in an industrial environment.

S/UTP

Screened/unshielded twisted-pair (shielded with copper braid)
Has an overall shield of copper braid to reduce influence of external interference. This cable is recommended for use with Bus Couplers.

FTP

Foiled shielded twisted-pair (shielded with aluminum foil)
This cable has an outer shield of laminated aluminum and plastic foil.

S/FTP

Screened/foiled shielded twisted-pair (shielded with copper braid and aluminum foil)
Has a laminated aluminum shield with a copper braid on top. Such cables can provide up to 70 dB reduction in interference power.

STP

Shielded twisted-pair
Describes a cable with overall shielding without further specification of the type of shielding.

S/STP

Screened/shielded twisted-pair (wires are individually shielded)
This identification refers to a cable with a shield for each of the two wires as well as an outer shield.

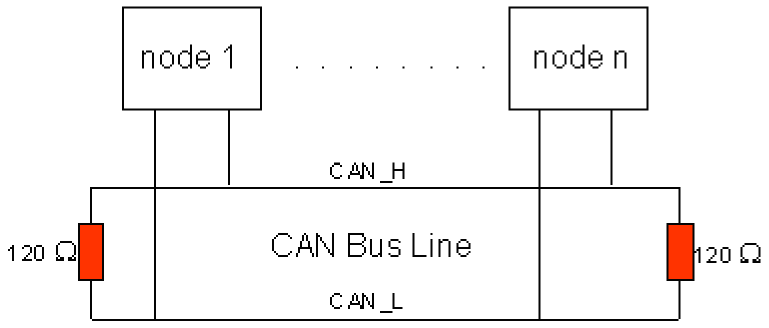
ITP

Industrial Twisted-Pair
The structure is similar to that of S/STP, but, in contrast to S/STP, it has only two pairs of conductors.

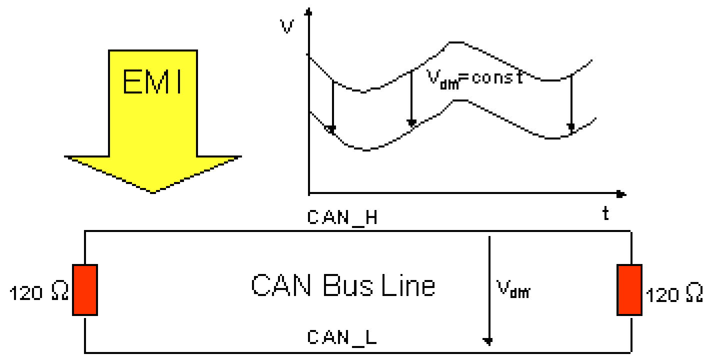
5.2.3 CANopen Cabling

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!



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.



Bus length

The maximum length of a CAN bus is primarily limited by the signal transit time. 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 transit times 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.

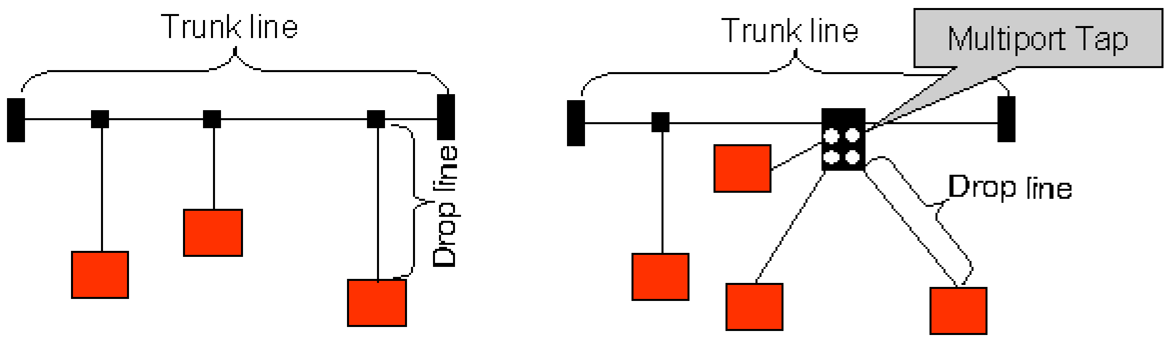
It may be necessary to use repeaters for bus lengths greater than 1000 m.

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	< 1m	< 5 m
500 kbit/s	< 5 m	< 25 m
250 kbit/s	< 10m	< 50 m
125 kbit/s	< 20m	< 100 m
50 kbit/s	< 50m	< 250 m

Drop lines must not have terminating resistors.



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

CAN cable

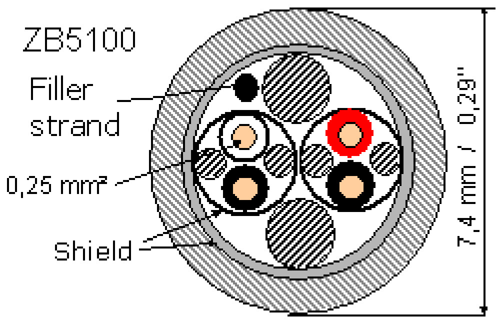
Screened twisted-pair cables (2x2) with a characteristic impedance of between 108 and 132 Ohm is recommended for the CAN wiring. If the CAN transceiver's reference potential (CAN ground) is not to be connected, the second pair of conductors can be omitted. (This is only recommended for networks of small 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, external diameter 7.3 +/- 0.4 mm

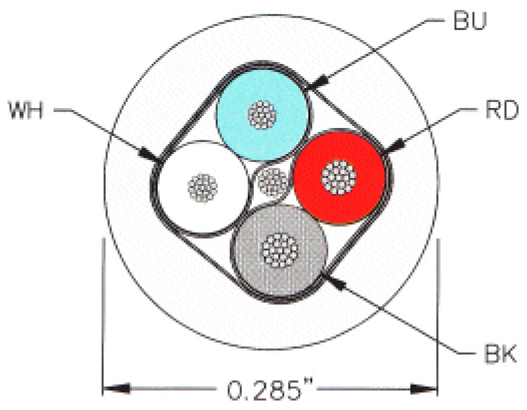
- Weight: 64 kg/km.
- printed with "BECKHOFF ZB5100 CAN-BUS 2x2x0.25" and meter marking (length data every 20 cm)



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, external diameter 7.3 mm
- printed with "InterlinkBT DeviceNet Type 572" as well as UL and CSA ratings
- stranded wire colours 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



Screening

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 section.

Cable colors

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

BK51x0 pin BC5150/ BX5100	BK5151, CX805x, CX-B510/M510	Fieldbus Box pin	FC51xx pin/ EL6751	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	Screen	Filler strand	Filler strand
4	7	4	7	CAN high	**white**	**white**
5	9	2	9	not used	**(red)**	**(red)**

BK5151, EL6751, CX805x, CX-B/M510 and FC510x: D-sub, 9 pin

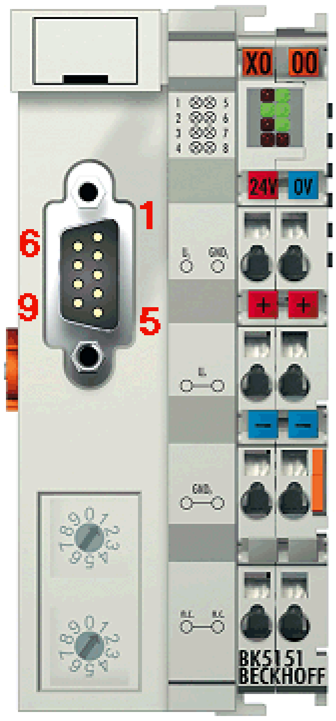
The CAN bus cable is connected to the FC51x1 and FC51x1/2 CANopen cards via 9-pin sub-D sockets, with pins assigned as follows.

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+)

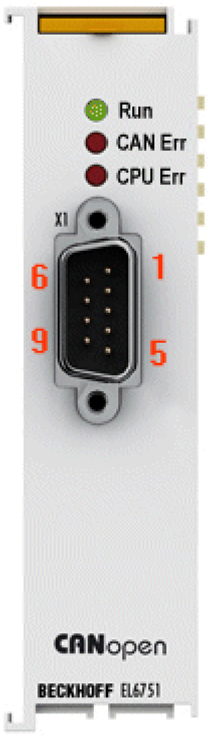
The unlisted pins are not connected.
The top-hat contact clip and the connector shield are connected..

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.

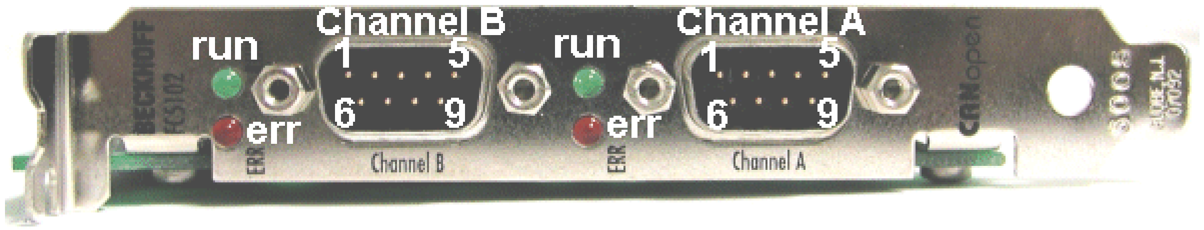
BK5151



EL6751

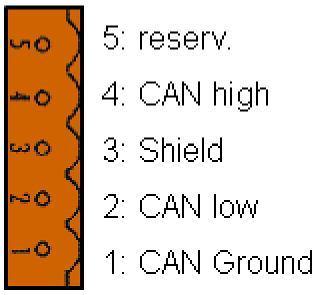


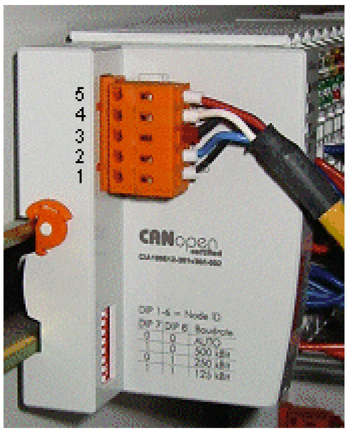
FC5102



BK51x0: 5- pin open style connector

The BK51x0 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.





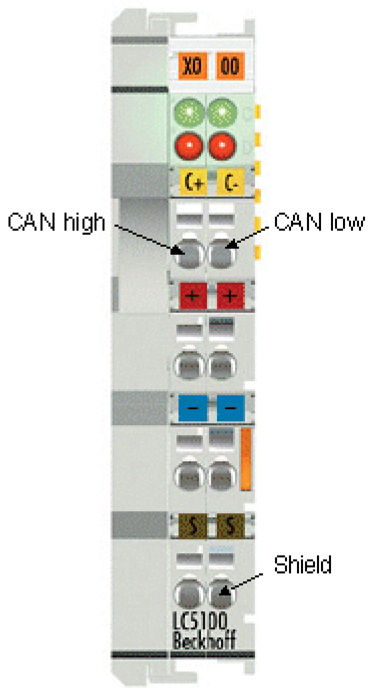
The left figure shows the socket in the BK51x0 Bus Coupler. Pin 5 is the connection strip's top most pin. Pin 5 is not used. Pin 4 is the CAN high connection, pin 2 is the CAN low connection, and the screen is connected to pin 3 (which is connected to the mounting rail via an R/C network). CAN GND can 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 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.

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.



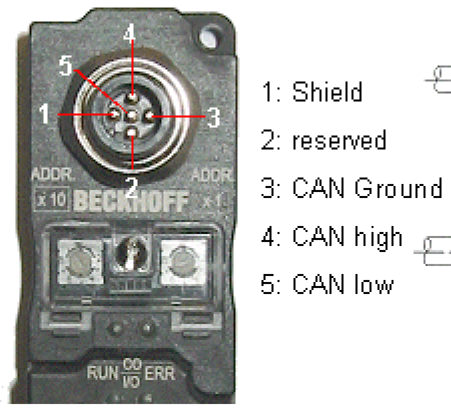
NOTICE

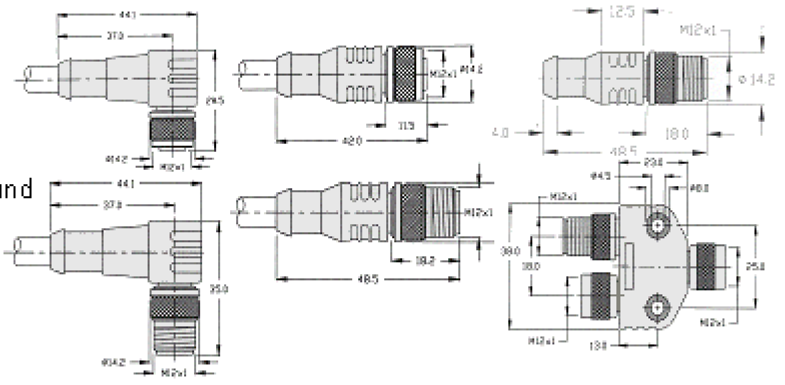
[Gefahrinformation hier einfügen!]

AchtungThe LC5100 has no galvanic isolation and an incorrect wiring can by destroyed or damaged the CAN driver.

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.





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 catalog, or under www.beckhoff.com.

5.3 Changing the battery

⚠ CAUTION

Battery as ignition source in potentially explosive atmospheres

Gases or dusts can be ignited by a spark discharge when the battery is inserted or removed.

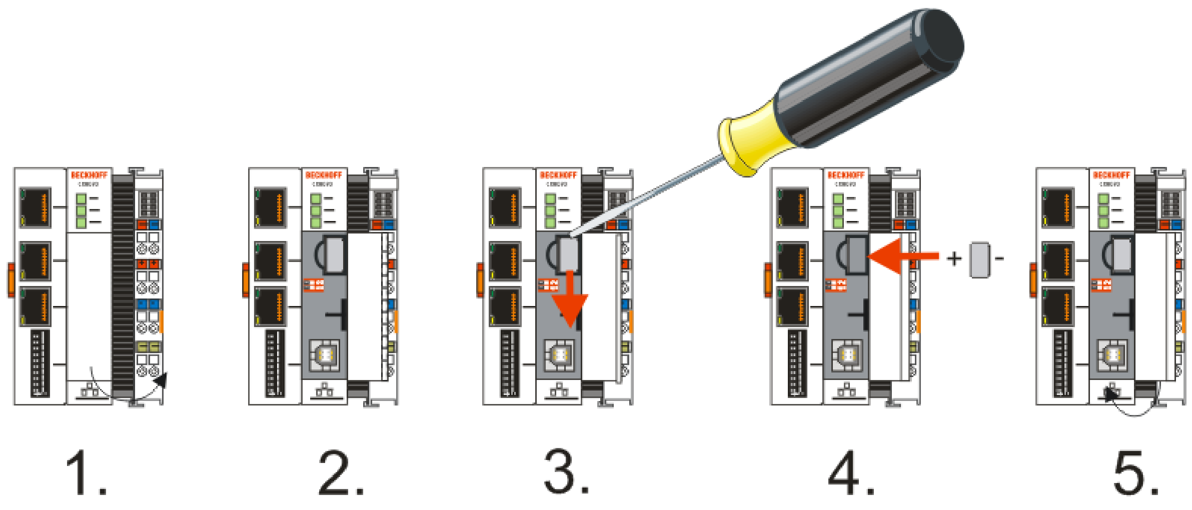
Switch off the power supply and wait until the 1-second UPS has discharged. Ensure that there is no explosive atmosphere before you insert or remove the battery.

NOTICE

An incorrectly inserted battery may explode!

Use exclusively the specified battery type. Make absolutely sure that positive and negative terminals of the battery are inserted correctly. (Plus pole on the left). Never open the battery or throw it into a fire. The battery cannot be recharged.

The battery of the CX80xx is required for the real-time clock (RTC) of the CX80xx. It ensures that the RTC continues to run in the power-off state so that the set time is available again on restarting.



- Step 1: Open the flap
- Step 2/3: Take a small flat-blade screwdriver, insert it above the battery and prise the battery carefully out of the device
- Step 4: Insert the new battery. The plus pole must be on the left
- Step 5: Close the flap again

Battery type	Technical data
Duracell 303/357 SR44	1.5 V / 165 mAh

Battery maintenance

The battery must be replaced every 5 years.

6 Parameterization and commissioning

6.1 DIP switch

⚠ CAUTION

DIP switches as ignition source in potentially explosive atmospheres

Gases or dusts can be ignited by a spark discharge when DIP switches are used.

Switch off the power supply and wait until the 1-second UPS has discharged. Ensure that there is no explosive atmosphere before you use DIP switches.

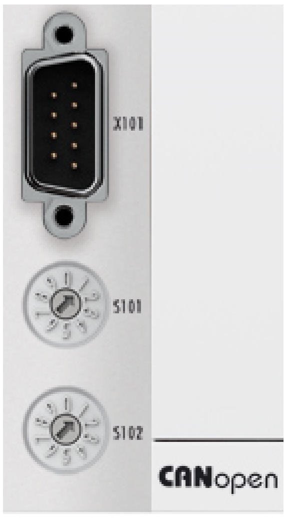
CX8050 DIP switch

The address selection switch of the CX8050 has no purpose, although it can be read by the PLC (see programming).

CX8051 DIP switch

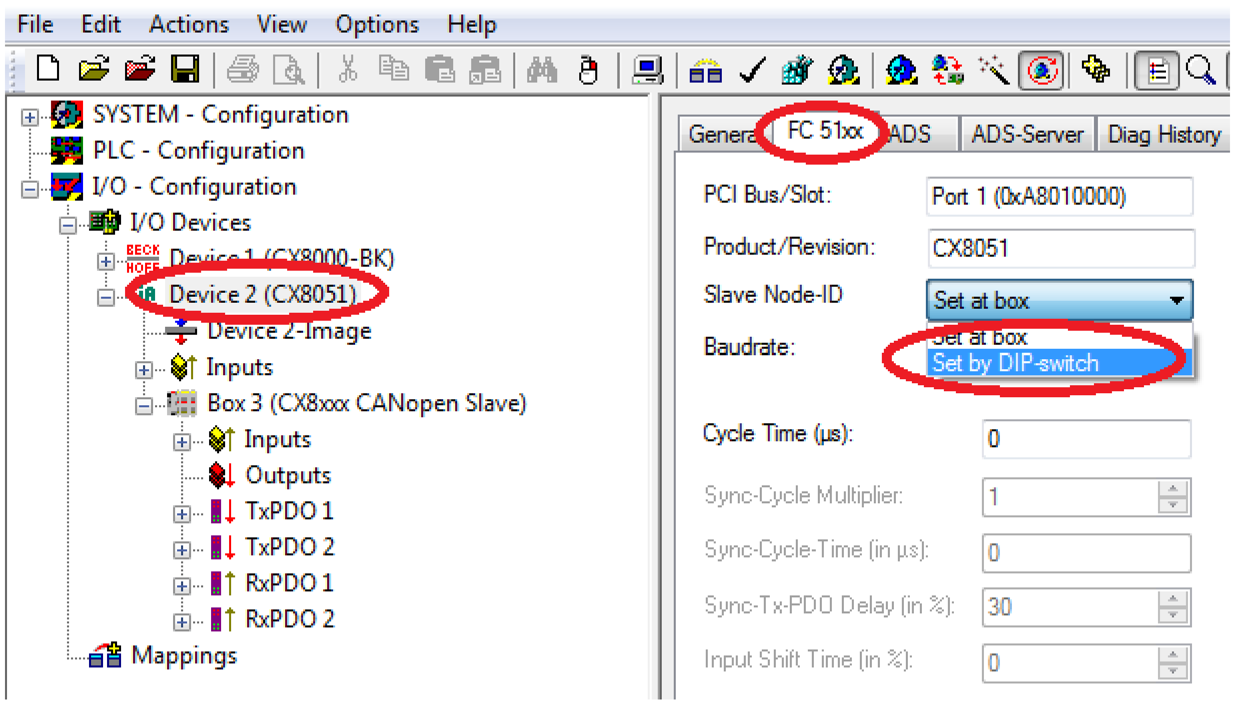
2x 10-pole address switch S101/S102

The address selection switch can be used for CAN address, although it can also be read by the PLC (see programming).



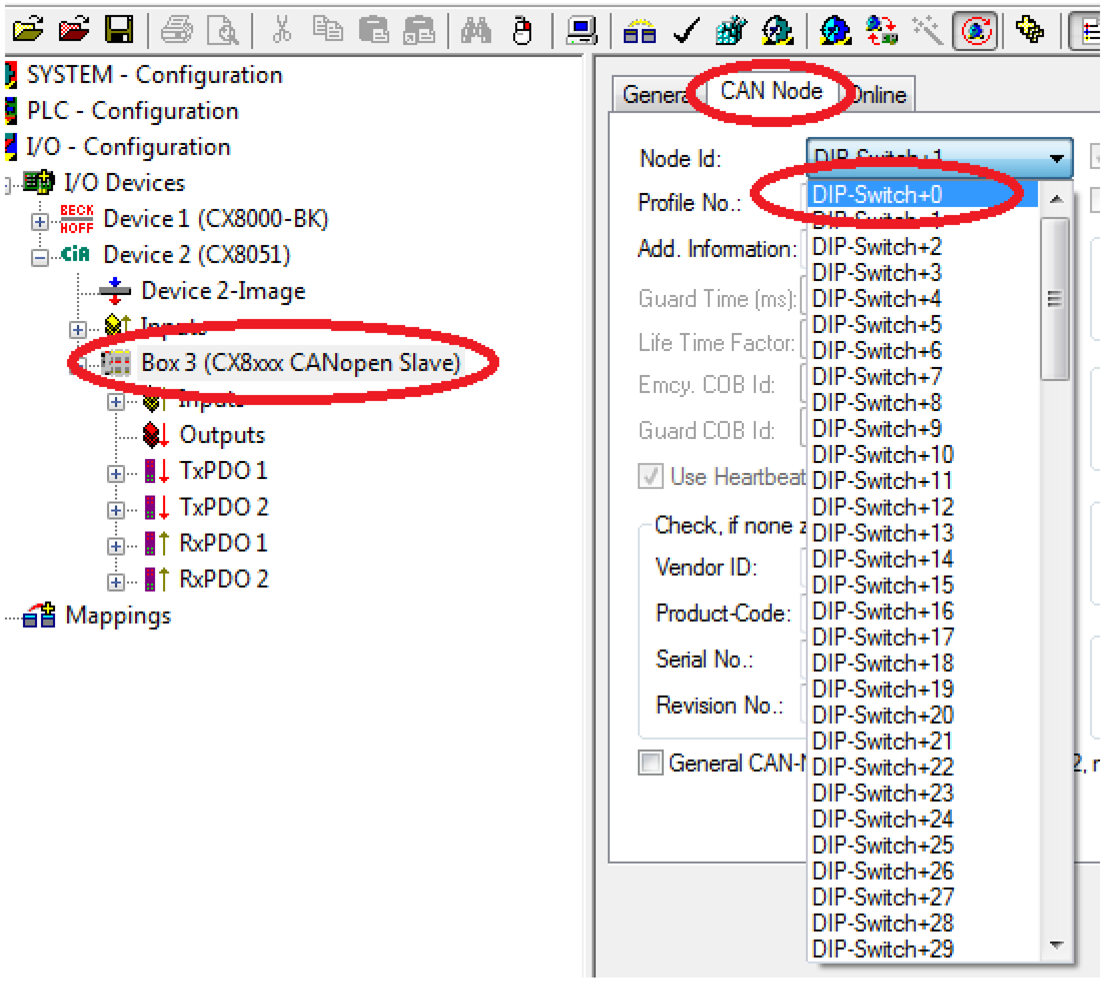
S101 for the address x 1, S102 for the address x10, example S101=2, S102=3 CAN node address 32

So that the address is also used via the address selector, this must be activated in the System Manager.



PCI Bus/Slot: Port 1 (0xA8010000)
Product/Revision: CX8051
Slave Node-ID: Set at box
Baudrate:
Cycle Time (µs): 0
Sync-Cycle Multiplier: 1
Sync-Cycle-Time (in µs): 0
Sync-Tx-PDO Delay (in %): 30
Input Shift Time (in %): 0

The station address can now be set for each slave with a DIP switch + number. For the virtual slaves you can use +1, +2, +3 for example.



2-pole DIP switch (under the flap between the battery and the SD card slot)

Requirements

DIP switch (red)	Meaning
1 off and 2 off	normal mode, TwinCAT is started
1 on and 2 off	The CX mode starts in Config Mode; the flash memory or, in the case of the CX80xx the SD card, is reachable via the USB interface (for example for an image update).
1 off and 2 on	Restore the registry
1 on and 2 on	No function so far

6.2 Setting the IP adress

6.2.1 IP address

The CX8010, CX803x, CX805x and CX8080 have an Ethernet interface, X001.

X001

IP addressing via the operating system; default is DHCP (represented in the operating system as FEC1)

EtherCAT interface

The EtherCAT interface is a further Ethernet interface that is not visible in the operating system for the IP addressing.

6.2.2 Setting the address via DHCP server

Port 1 (X001) is set to DHCP by default.

If DHCP is switched on, the CX is automatically assigned an IP address by the DHCP server. The DHCP server must know the MAC ID of the Bus Terminal Controller for this. The IP address should be assigned statically by the DHCP server. A local IP address is used if no DHCP server is reachable.

The DNS name is formed from the type and the last 3 byte of the MAC ID. The MAC ID is given on the production label of the Bus Terminal Controller.

Example: CX80xx

- MAC ID: 00-01-05-01-02-03
- DNS name: CX-010203

6.2.3 Subnet mask

The subnet mask is subject to the control of the network administrator, and specifies the structure of the subnet.

Small networks without a router do not require a subnet mask. The same is true if you do not use registered IP numbers. A subnet mask can be used to subdivide the network with the aid of the mask instead of using a large number of network numbers.

The subnet mask is a 32-bit number:

- Ones in the mask indicate the subnet part of an address space.
- Zeros indicate that part of the address space which is available for the host IDs.

Description	Binary representation	Decimal representation
IP address	10101100.00010000.00010001.11001000	172.16.17.200
Subnet mask	11111111.11111111.00010100.00000000	255.255.20.0
Network ID	10101100.00010000.00010000.00000000	172.16.16.0
Host ID	00000000.00000000.00000001.11001000	0.0.1.200

Standard subnet mask

Address class	Standard subnet mask (decimal)	Standard subnet mask (hex)
A	255.0.0.0	FF.00.00.00
B	255.255.0.0	FF.FF.00.00
C	255.255.255.0	FF.FF.FF.00

Assignment of subnets, host numbers and IP addresses

Neither subnet 0 nor the subnet consisting only of ones may be used. Host number 0, and the host number consisting only of ones, must not be used. Under BootP or DHCP the subnet mask is transmitted also by the server.

6.3 Configuration

6.3.1 CX80xx - Operating system

The CX80xx comes with a Microsoft CE operating system, version 6.0. This operating system is adapted and optimized for the CX80xx. Not all CE6.0 components are available.

Safety

From image version 3.54b security was tightened. This applies to CERHOST and TELNET. Both services are now switched off in delivery state. To reactivate these services, you need a Micro SD card reader.

CERHOST

CERHOST is deactivated by current images on first start-up via the registry file *CeRemoteDisplay_Disable.reg*, which is located in the folder *RegFiles*.

To reactivate CERHOST, delete the file *CeRemoteDisplay_Disable.reg* from the folder *RegFiles* and also the folder *Documents and Settings*.
Then reinsert the Micro SD card in the CX and reboot. The CX creates a new *Document and Settings* directory and then reboots automatically.
The CX is then accessible again via CERHOST.

TELNET

TELNET is deactivated by current images on first start-up via the registry file *Telnet_Disable.reg*, which is located in the folder *RegFiles*.

To reactivate TELNET, delete the file *Telnet_Disable.reg* from the folder *RegFiles* and also the folder *Documents and Settings*.
Then reinsert the Micro SD card in the CX and reboot. The CX creates a new *Document and Settings* directory and then reboots automatically.
The CX is then accessible again via TELNET.

IMAGE

If you do not know what image is loaded on the CX80xx, you can determine it quite easily.

- Via the web diagnostics page of the CX. Here you can find the build number under the *TwinCAT* device.
  Opening the web diagnostics page:
  - IP address</config
  or
  - CX name/config
  Example:
  - *172.16.17.201/config*
  or
  - *CX-01551E/config*

- Via a Micro SD card reader.
  The Micro SD card contains a file with the name of the image.
  Example CX8000_CE600_LF_v354b_TC211R3_B2248.
  TC211R3_2248 indicates the TwinCAT build; in the example the build is 2248.

Prerequisites

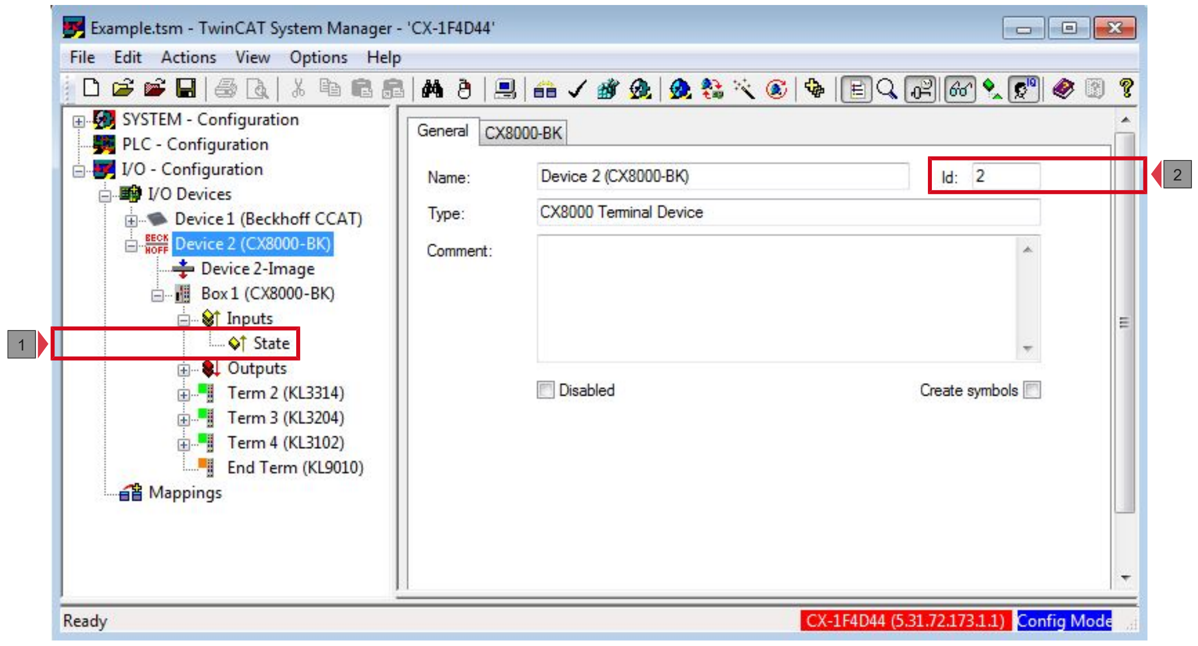
Feature / platform	CX80x0 LF version 3.xx
ATL	Xtd
MFC	X
XML DOM	X
COM	X
COM Storage	-
Winsock	X
TCP/IP	X
TCP/IPv6	-
Firewall	X
Network Utilities (IpConfig, Ping, Route)	X
UPnP	
Control Point	-
Device Host	X
SOAP	
Client	-
Server	-
DCOM	**-**
Object Exchange Protocol OBEX	-
Message Queuing MSMQ	-
Server	
File Server (SMB/CIFS)	X
File Server	X
Print-Server (SMB/CIFS)	-
RAS Server / PPTP Server	-
Simple Network Management Protocol (SNMP)	X
Telnet Server	X
HTTP / ASP / FTP / SNTP -Server	X
Web Server (HTTPD) / Active Server Pages (ASP) Support / JScript 5.6 / VBScript 5.6	X
Simple Network Time Protocol (SNTP)	X
HTML / DHTML, TLS, ISAPI extensions	X
Internet Explorer 6.0	-
Java Applets	-
NET Compact Framework	v3.5
RDP Client (Remote Desktop protocol)	-
CAB File Installer/Uninstaller	X
TwinCAT (Level PLC)	X
USB support	X
Printer, storage on Compact Flash, for example	-
HID (Human interface devices)	-
Touch	-

6.3.2 Power supply terminal

K-bus interface

It is possible to operate K-bus terminals on the CX80xx.

The CX80xx recognizes these terminals automatically on scanning, reads out the terminal types and automatically places them in the System Manager.



Name: Device 2 (CX8000-BK)
Id: 2
Type: CX8000 Terminal Device
Comment:

Fig. 1: K-Bus Interface

K-bus state

The K-bus status is saved in the state byte (see fig. K-bus interface "1"). If the value is 0 the K-bus is operating synchronously and without errors. If the value should be <>0 this can be an error, but it may also be *just* a notice that, for example, the K-bus requires longer than the employed task and is thus no longer synchronous to the task. The task time should be faster than 100 ms. We recommend a task time of less than 50 ms. The K-bus update time typically lies between one and five ms.

Bit 0 = K-Bus Err
Bit 1 = Terminal State Err
Bit 2 = Process Data Length Err
Bit 8 = No valid Inputs
Bit 9 = K-Bus Input Update busy
Bit 10 = K-Bus Output Update busy
Bit 11 = Watchdog Err
Bit 15 = Acyc. Function atcive (e.g. K-Bus Reset)

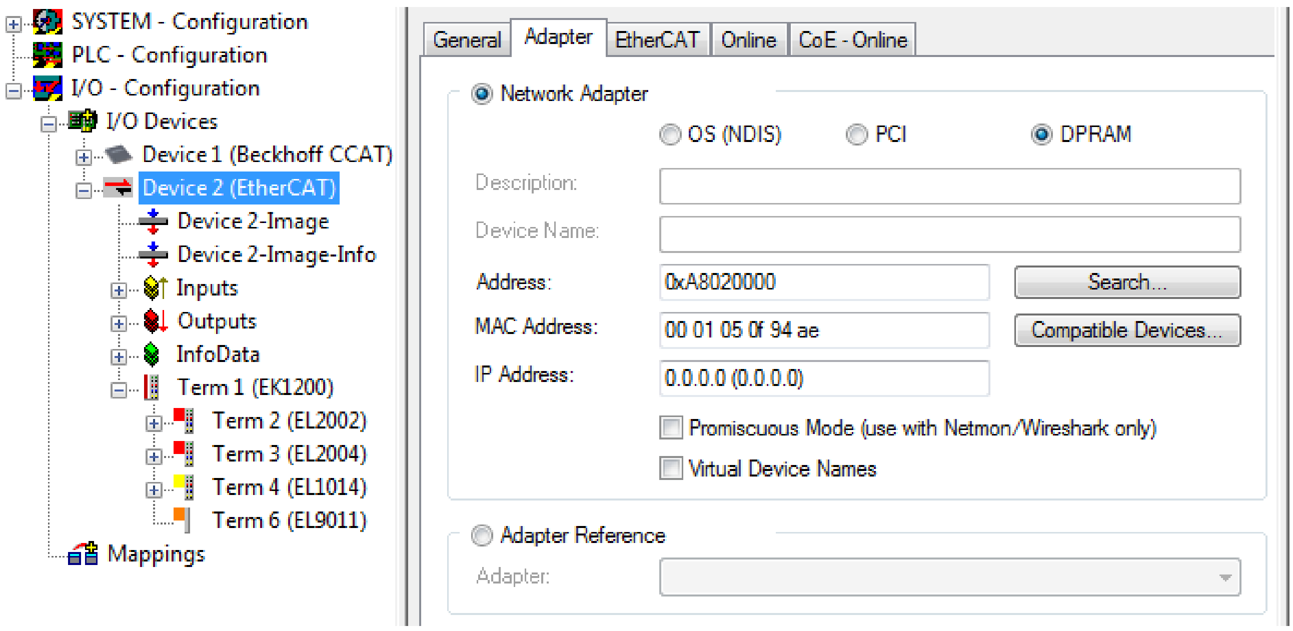
If there is a K-bus error, this can be reset via the IOF_DeviceReset function block (in the TcIoFunctions.lib).

The NetID is that of the CX80xx and can thus be entered as an empty string, the Device ID (see fig. K-bus Interface "2") is to be taken from the System Manager.

E-bus interface

The operation of E-bus terminals and EtherCAT devices is possible on the CX80xx.

The CX80xx recognizes these terminals automatically on scanning, reads out the terminal types and automatically places them in the System Manager.


SYSTEM - Configuration
PLC - Configuration
I/O - Configuration
I/O Devices
Device 1 (Beckhoff CCAT)
Device 2 (EtherCAT)
Device 2-Image
Device 2-Image-Info
Inputs
Outputs
InfoData
Term 1 (EK1200)
Term 2 (EL2002)
Term 3 (EL2004)
Term 4 (EL1014)
Term 6 (EL9011)
Mappings

General | Adapter | EtherCAT | Online | CoE - Online

[✓ Network Adapter]

OS (NDIS) PCI [✓ DPRAM]

Description:

Device Name:

Address: 0xA8020000

MAC Address: 00 01 05 0f 94 ae

IP Address: 0.0.0.0 (0.0.0.0)

DC Distributed Clocks

The CX80xx series is not suitable for the use of EtherCAT slaves that use or need distributed clocks functionality.

6.3.3 CAN

CX8051

CANopen-Interface

The CANopen communication takes place via D-Sub port X101.

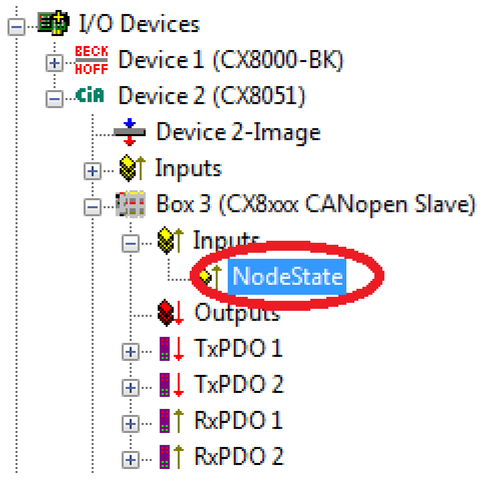
Debug via the Ethernet interface only

The CX8051 does not support the ADS via CANopen. The program download and debugging can take place exclusively via the Ethernet interface.

CANopen address

The CANopen address can be assigned via the rotary selector, or permanently in the System Manager. If the address is assigned permanently, the address switch is ignored.

CANopen NodeState



The NodeState can be used to display the state of the CANopen communication to find out whether the slave is engaged in data exchange (NodeState=0) or whether there is an error or problem.

0 = No error
128 = Node is Operational but not all RxPDOs were received
129 = Node is Pre-Operational
130 = Node is Stopped

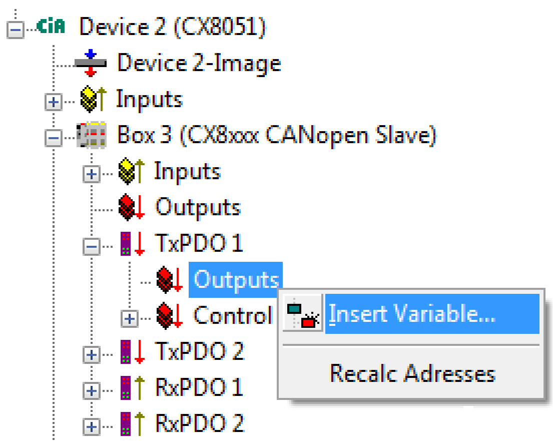
CANopen process data

The CX8051 can exchange up to 16 PDOs (each with 8 bytes of process data) with the CANopen master in input and output direction via CANopen.

By default 2 PDOs are created in Tx and Rx direction. The PDOs can be filled with user data. The limit of 8 bytes per PDO must not be exceeded. Data are sent automatically when there is a change, unless the master is configured differently. At the planning stage please ensure that the data in a PDO "only" change at a moderate rate (e.g. not with ms frequency). Failure to adhere to this can lead to CAN overload. Particularly for low baud rates, the CAN can reach its limit quite quickly.

Creating data in the PDO

For each PDO you can create up 8 bytes of data. Variables of different type may be used, as long as the limit of 8 bytes is adhered to. For TxPDOs there is an additional control word, which can be used in cases where data are to be sent not only in the event of changes, but also when the data in the PDO have not changed. To change the control word it can simply be incremented, for example. If incrementation and data modification happen at the same time (i.e. in the same cycle), only one telegram is sent.
The RxPDOs had an additional status word, which is incremented on arrival of the PDO. This is useful in cases where data in the PDO are unchanged, since attention is drawn to the fact that new telegram with old data has arrived. This can be used for monitoring or to check whether a device still sends data on a regular basis.



Further PDOs can be added by clicking on the "CX8xxx CANopenSlave box". Please note that the COB ID is always zero from the 5th PDO, in which case it has to be entered manually.

CAN load

The CAN load should be taken into account during network planning and configuration: 500 kbit at 8 bytes of user data per frame results in a maximum number of 3707 frames per second. For reasons of network stability it is never advisable to run a CAN at 100% load. An upper bus load limit of 60% is recommended, which corresponds to 2221 frames per second. Example: A CX8051 with 8 Rx PDOs and 10 ms task time, resulting in 100 cycles per second for 8 PDOs. If PDO sending is event-driven, they are sent when the process data change. On the slave side this may be more frequent than every 10 ms. If the values only change once per 10 ms cycle, this results in 800 frames per second on the slave side and perhaps another 800 frames per second on the master side, plus heartbeat, sync telegrams and SDO communication. The example indicates that the upper limit of 2221 frames can be reached or indeed exceeded quite quickly in cases where rapid changes in input data lead to sending of PDOs with high frequency. This may be the case for analog inputs, for example, since their values usually change continuously. It is therefore advisable to control the send behavior by setting suitable parameter (inhibit time, filters) or to switch to cyclic sending.

Baud rate	1 bytes data	2 bytes data	4 bytes data	8 bytes data
1 Mbaud	15384	13333	10526	7407
500 kbaud	7692	6666	5263	3703
100 kbaud	1538	1333	1052	740

Table showing the number of theoretical CAN frames at 100% load for different CAN data sizes.

Virtual CANopen device interface

The virtual slave interface enables the creation of up to three virtual slaves on the same hardware interface. This enables the user to exchange more data with a CANopen master.

A maximum of 16 PDOs data can be configured for each slave, i.e. in total you can exchange 4 x 16 PDOs data in each direction.

Append a maximum of four CX8051 devices to your CAN device (fig. 1.0). Each of these devices is given a CAN address via the System Manager which can also be linked with the address selector (see Address switch). Add the process data PDOs under the box. For the CANopen master configure each of the four slaves like an independent device.

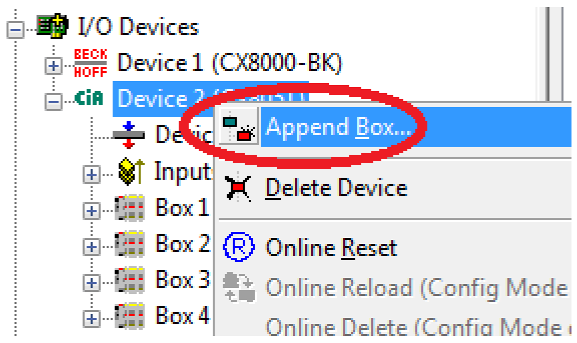


Fig. 2: Creation of the 4 CANopen slave devices

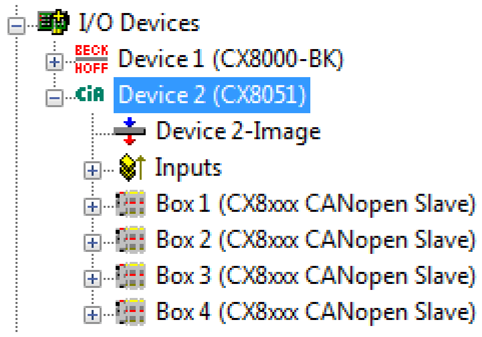

I/O Devices
Device 1 (CX8000-BK)
Device 2 (CX8051)
Device 2-Image
Inputs
Box 1 (CX8xxx CANopen Slave)
Box 2 (CX8xxx CANopen Slave)
Box 3 (CX8xxx CANopen Slave)
Box 4 (CX8xxx CANopen Slave)

Fig. 3: Appending the CAN modules

CX8050

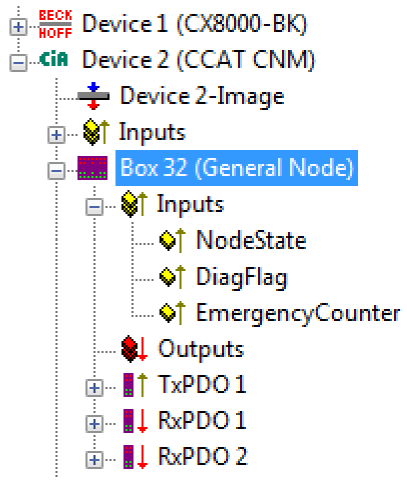
CANopen interface / CAN interface

The CANopen communication takes place via D-Sub port X101. The CX8050 enables a CANopen master or "simple" CAN communication to be used.

CANopen address

The rotary selector (S101/102) of the CAN master has no purpose. The address selector can be read via the PLC (see address) and then be used for the applications.

CANopen NodeState



The NodeState (red) can be used to display the state of the CANopen communication to find out whether the slave is engaged in data exchange (NodeState=0) or whether there is an error or problem.

0 = No error
1 = Node deactivated
2 = Node not found
4 = SDO syntax error at StartUp
5 = SDO data mismatch at StartUp
8 = Node StartUp in progress
11 = FC510x 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

The DiagFlag indicates whether an emergency telegram was received from the slave. The telegram can then be read in the System Manager or the PLC via ADS (see ADS interface). Please consult the slave manufacturer regarding interpretation of the data. The flag is reset once the Diag buffer was read.

The EmergencyCounter is incremented after each emergency telegram.

CAN Layer 2 communication

If you have selected this checkbox, the entire CANopen network management for this device is deactivated. It is not started, monitored etc. The PDO inputs are detected as pure CAN (layer 2) telegrams and enable the controller to operate in event driven mode.

CAN interface

Any CAN data can be sent via the CAN interface. There is a choice between 11-bit identifier (CAN 2.0A) or 29-bit identifier (CAN 2.0B).

Message structure with 29-bit support

- Length (0..8)
- CobId
    - Bit 0-28: 11 Bit identifier / 29 Bit identifier
    - Bit 30: RTR
    - Bit 31: 0: normal Message (11 Bit Identifier), 1: extended Message (29 Bit-Identifier)
- Data[8]

Sending data: In NoOfTxMessages enter the number of data to be sent from the Tx buffer. If the buffer has capacity for 10 entries, the maximum number of telegrams that can be send consecutively is 10. "Length" defines the number of PDO data bytes (maximum 8 bytes). Enter the data, then enter the CAN message ID in "codId". Now increment the TxCounter value.

Sample code: Sending messages from the PLC

```
if Outputs.TxCounter = Inputs.TxCounter then
     for i=0 to NumberOfMessagesToSend do
          Outputs.TxMessage[i] = MessageToSend[i];
     End_for
     Outputs.NoOfTxMessages = NumberOfMessagesToSend;
     Outputs.TxCounter := Outputs.TxCounter + 1;
end_if
```

Sample code: Receiving messages from the PLC

```
if Outputs.RxCounter <> Inputs.RxCounter then
     for i := 0 to (Inputs.NoOfRxMessages-1) do
          MessageReceived[i] := Inputs.RxMessage [i];
     End_for
     Outputs.RxCounter := Outputs.RxCounter+1;
end_if
```

Also see about this

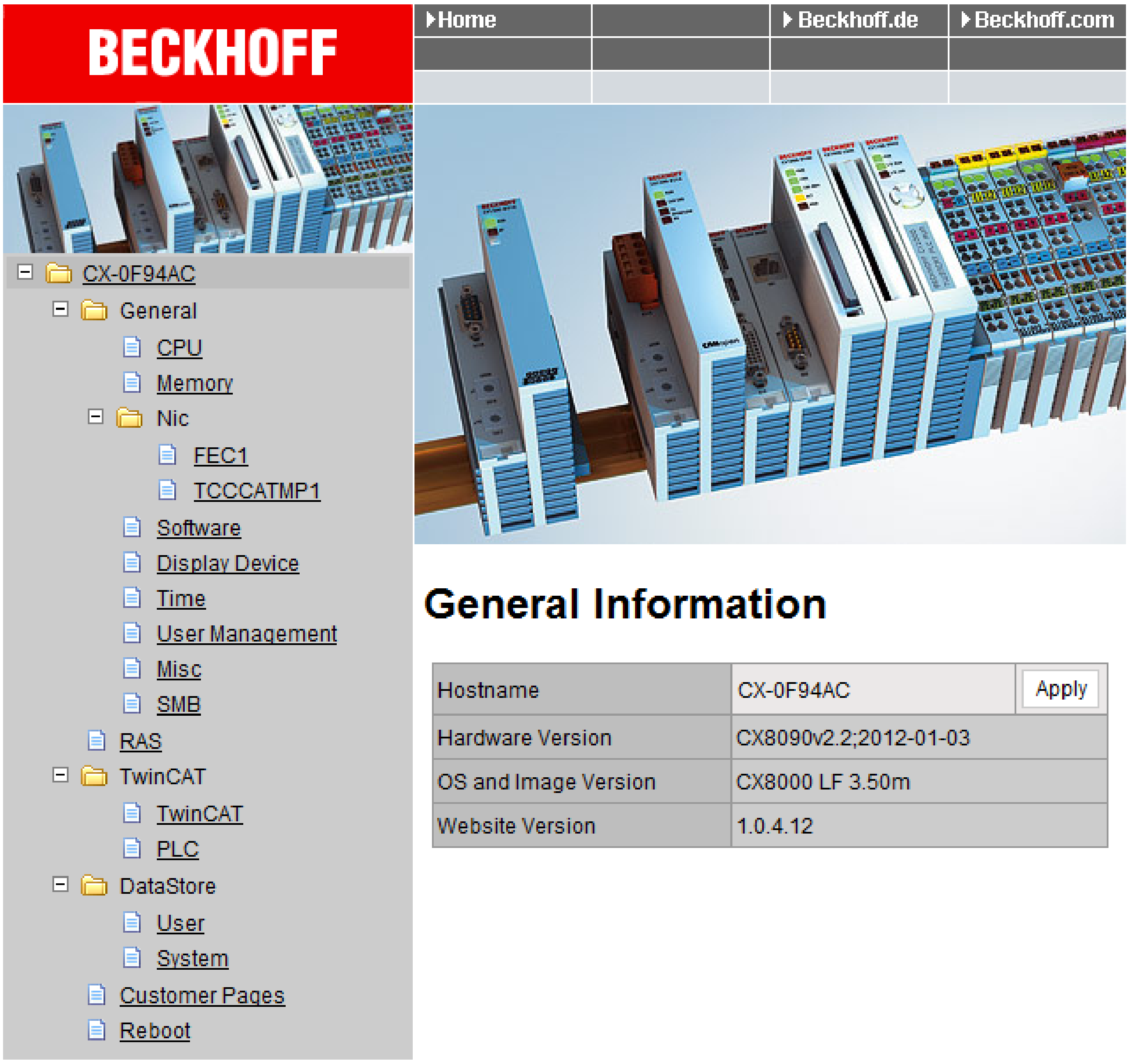
- DIP switch [▸ 32]
- Emergency telegrams and diagnostics [▸ 55]

6.3.4 Web Services

Upnp webpages

There is a Upnp webpage on the CX80xx for diagnostics.

User name: guest
Password: 1



General Information

Hostname	CX-0F94AC
Hardware Version	CX8090v2.2;2012-01-03
OS and Image Version	CX8000 LF 3.50m
Website Version	1.0.4.12

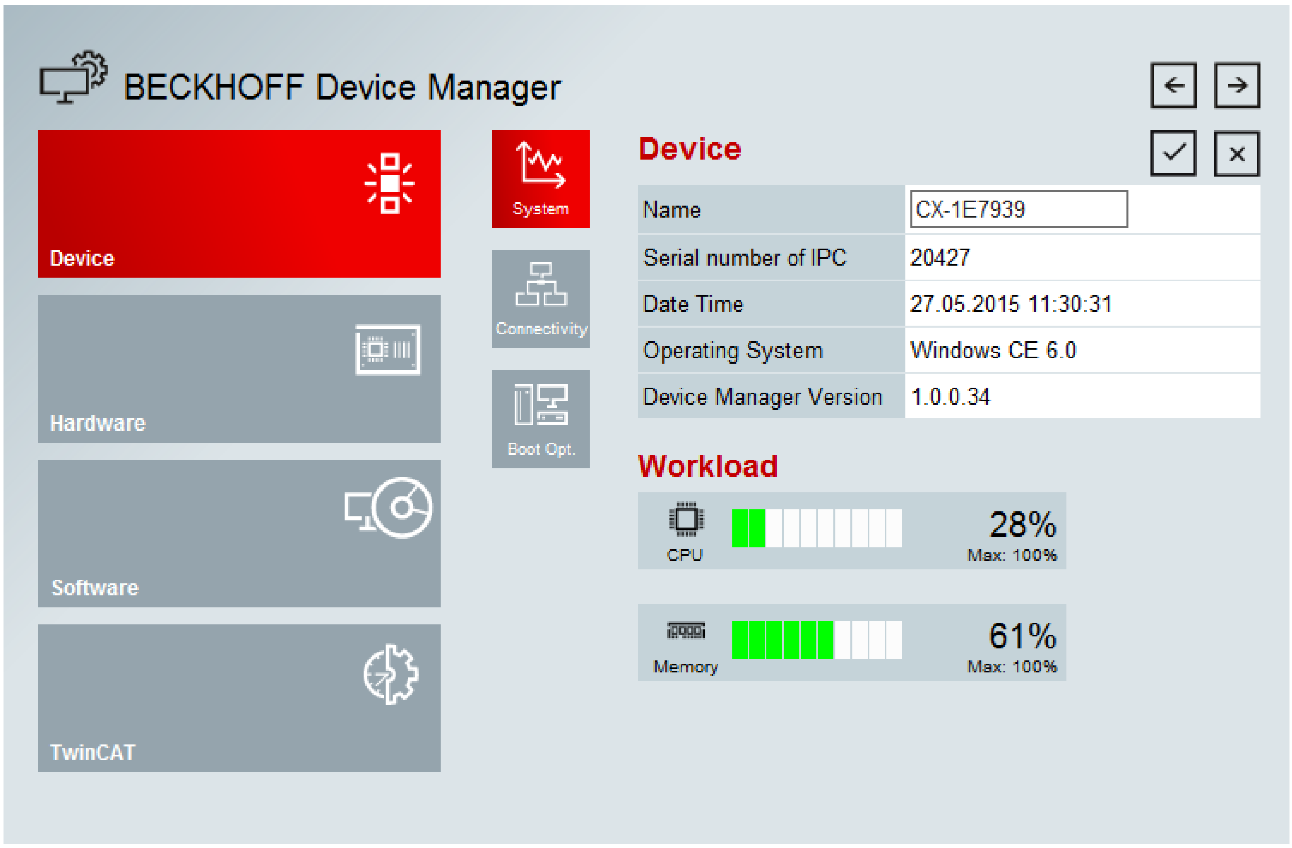
Enter the IP address or the device name.

Example

http://**cx-0f94ac**/config

http://**172.16.17.55**/config

The diagnostic page was revised starting from image v354c.



Device

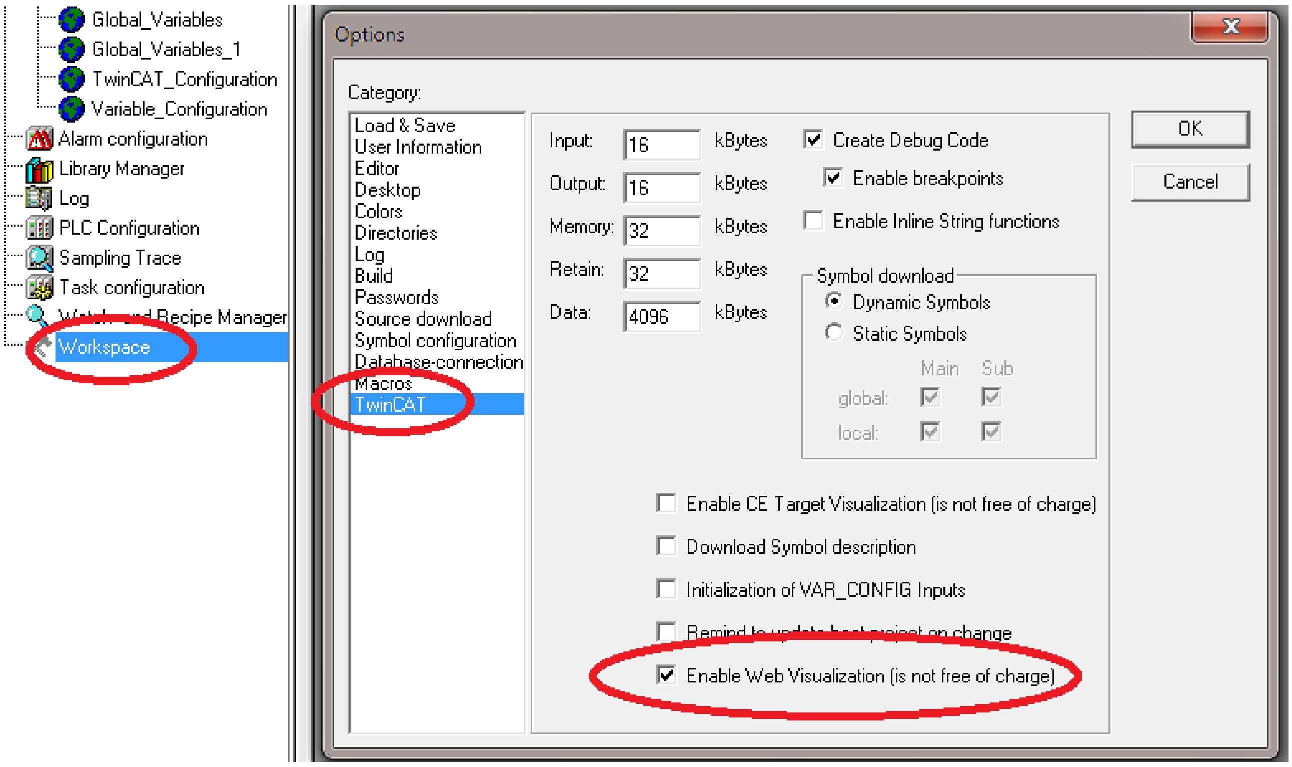
Name	CX-1E7939
Serial number of IPC	20427
Date Time	27.05.2015 11:30:31
Operating System	Windows CE 6.0
Device Manager Version	1.0.0.34

CPU 28% Max: 100%

Memory 61% Max: 100%

Web visualization

There is a web visualization on the CX80xx. This can be prepared and activated with the help of the PLC Control in TwinCAT.



Input: 16 kBytes
Output: 16 kBytes
Memory: 32 kBytes
Retain: 32 kBytes
Data: 4096 kBytes

[✓] Create Debug Code
[✓] Enable breakpoints
[ ] Enable Inline String functions

Symbol download
(●) Dynamic Symbols
( ) Static Symbols

	Main	Sub
global:	✓	✓
local:	✓	✓

[ ] Enable CE Target Visualization (is not free of charge)
[ ] Download Symbol description
[ ] Initialization of VAR_CONFIG Inputs
[ ] Remind to update boot project on change
[✓] Enable Web Visualization (is not free of charge)

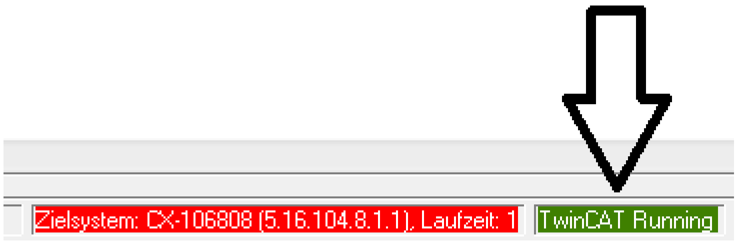
The call is made via the IP address or the device name in a web browser.
Further information can be taken from the documentation on the web visualization (see TwinCAT Supplements PLC HMI Web).

Example

http://**cx-0f94ac**/TcWebVisu/

http://**172.16.17.44**/TcWebVisu/

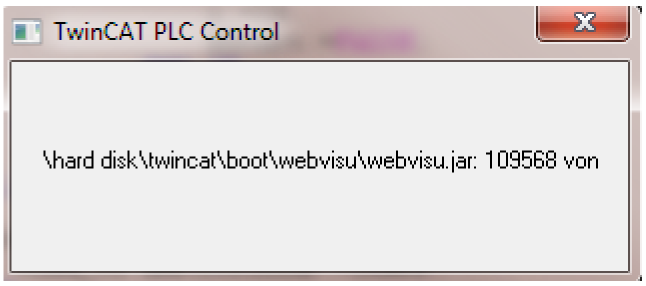
Ascertain before logging in (i.e. in the logged out condition) whether a ADS connection is established to the CX – "TwinCAT Running" in the bottom right-hand corner must be green. If that is not the case, please go onto Online/Selection of the target system again and call the CX once again.



The following path must be specified for downloading the web data for the web user interface:

\hard disk\twincat\boot\webvisu\

If that is not the case, the PLC Control will copy the data into the wrong folder and the webpage will be displayed incorrectly or not at all.


TwinCAT PLC Control

\hard disk\twincat\boot\webvisu\webvisu.jar: 109568 von

Remote Display

This page describes the steps for remotely controlling a CE device with CE operating system from a further PC by **'Remote Display'**.

Software required on the PC:

- Windows NT, Windows 2000, Windows XP or Windows 7
- Microsoft Remote Display (CERHOST, available license-free from Microsoft)

Establishing the connection

The "Remote Display" tool is started on the PC. The address of the CE device can now be entered under the menu option "File - > Connect"; this can be both the TCP-IP address or, if available, also the name of the CE device.

If the CE device is provided with password protection, then the password must also be entered accordingly. No password is set in the delivery condition.

After entering the target address, the user interface of the CE device is available for remote control on the PC.

Download : https://infosys.beckhoff.com/content/1033/cx805x_hw/Resources/1608562059.zip

6.3.5 Real Time Clock (RTC)

The RTC is read out via the FB_LocalSystemTime function blocks and can be set with the NT_SetLocalTime block (see TcUtilities.lib).

The RTC is supplied by the battery and can thus continue to run in the power-off state.

6.3.6 1-second UPS (Uninterruptible Power Supply)

Technical concept

The 1-second UPS is an UltraCap capacitor, which, in the event of a voltage outage, continues to supply power to the processor for approx. 4 to 5 seconds, so that persistent data can be saved. Data saving generally takes less than 4 to 5 seconds. However, due to ageing of the components used, one should assume that the UPS can provide power for a maximum of 1 second. You can assume that data saving continues to work smoothly, even after many years. If you save data yourself, we recommend that this should take place within 1 second. Should it take longer, we would advise against it.

The 1-second UPS supplies neither the K-bus nor the E-bus with power. Please note that the data of these devices may already be invalid when the 1-second UPS is activated. Also, the fieldbus system (or Ethernet) may not work or not work properly once the 1-second UPS was activated.

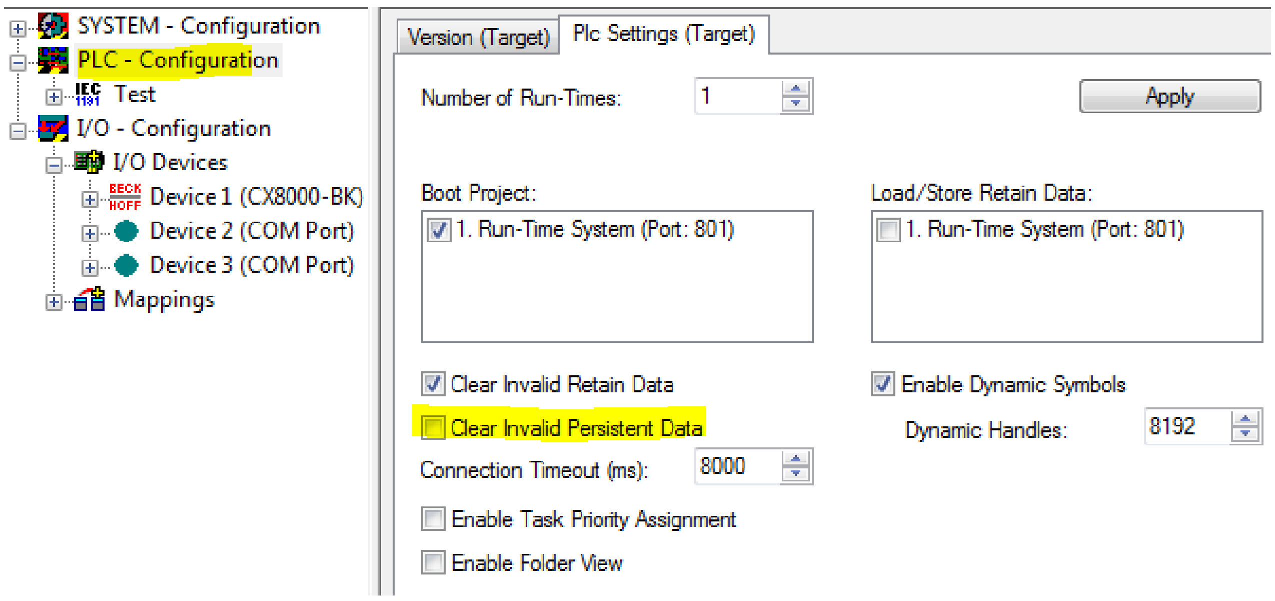
Saving of the persistent data only takes place in conjunction with the function block FB_S_UPS_CX80xx. This block must be called cyclically. We strongly recommend using the default values for the block.

Saving and loading persistent data

The persistent data are stored on the SD card as a wdp file. When the PLC starts up, the wdp file is loaded from the SD card, saved there as a wd~-file (backup), and then deleted. Another current wpd file is not written until the system is shut down or the 1-second UPS is activated. If no wdp file is present when the CX starts up, the persistent data are invalid and are deleted (default setting).
The reason is that the 1-second UPS was activated before the TwinCAT PLC was started during startup of the CX. In this case no persistent data were saved, since the system was unable to ensure sufficient buffer time for saving the data.

Loading a backup of the persistent data

To load the persistent data from the backup (wp~-file), it has to be enabled in the System Manager.



Number of Run-Times: 1

Boot Project: [✓] 1. Run-Time System (Port: 801)

Load/Store Retain Data: [ ] 1. Run-Time System (Port: 801)

[✓] Clear Invalid Retain Data

[ ] Clear Invalid Persistent Data

Connection Timeout (ms): 8000

[ ] Enable Task Priority Assignment

[ ] Enable Folder View

[✓] Enable Dynamic Symbols

Dynamic Handles: 8192

Or via the following registry entry:

[HKEY_LOCAL_MACHINE\SOFTWARE\Beckhoff\TwinCAT\Plc]"ClearInvalidPersistentData"= 0

The default factory setting is "1".

Checking whether current persistent data (from wdp file) or saved persistent data from the backup (wd~-file) were loaded

In this example, the CX8090 indicates via the ERR LED whether the persistent data were loaded. The LED cannot be used for other CX8xxx models.

```
IF systeminfo.bootDataFlags.4 AND NOT
systeminfo.bootDataFlags.5 THEN


F_CX8090_LED_ERR(eLED_GREEN_ON);     (* persistent
data is OK *)

ELSIF systeminfo.bootDataFlags.4 AND systeminfo.bootDataFlags.5
THEN


F_CX8090_LED_ERR(eLED_RED_ON);
(* load backup persistent data *)

ELSE

    F_CX8090_LED_ERR(eLED_RED_FLASHING_200ms); (* no
persistent data *)

END_IF
```

Purpose of the 1-second UPS

The 1-second UPS should only be used for managing the persistent data. Other applications are not supported and are not covered by our complaints procedure. Retain data cannot be used for the 1-second UPS!

6.3.7 CPU load

In the delivery condition the CPU load display is deactivated on all CX80xx devices (it is displayed with a constant 10%). The CPU load display is deactivated because it accounts for a considerable portion of the CPU load itself. The CPU load can be activated for brief diagnostic help; however, we recommend that you deactivate it again after the diagnostics.

HKEY_LOCAL_MACHINE/SOFTWARE/BECKHOFF/TWINCAT/RTime/EnableRTimeMeasurement   0 deactivated, 1 activated

A TwinCAT restart of the CX80xx is necessary after making the setting.

CPU load

The CPU load is calculated internally with 10 ms. The CPU load display may fluctuate very strongly if one or more tasks exceeding 10 ms are used.

7 Programming

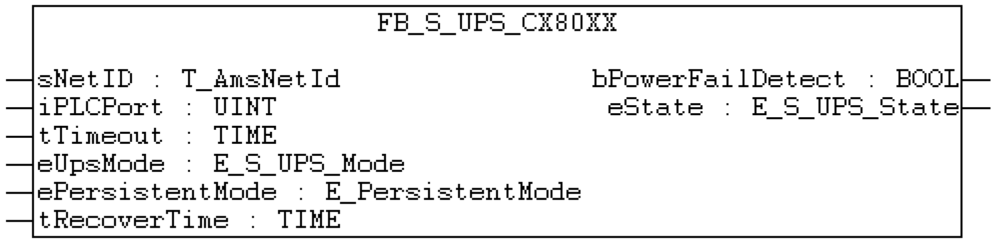
7.1 Library for CX80xx

Download : https://infosys.beckhoff.com/content/1033/cx805x_hw/Resources/1608565003.zip

7.2 Seconds UPS

7.2.1 Function blocks

FUNCTION_BLOCK FB_S_UPS_CX80xx



The FB_S_UPS function block can be used on the CX80xx with the seconds UPS in order to activate the seconds UPS from the PLC. This allows the persistent data to be saved and a quick shutdown to be performed in the event of a power failure. If possible the default values of the INPUTs of the FB_S_UPS should be retained.

NOTICE

Loss of data

The seconds UPS can be used only for a few seconds in the event of a power failure in order, for example, to save persistent data. The data must be saved in the fast persistent mode "SPDM_2PASS", even though this can lead to real-time violations. Sufficient router memory must be configured for the storage of the persistent data!

The second UPS does not have sufficient capacity for bridging power failures. Saving can take place only on Micro SD cards.

A QuickShutdown is performed automatically in the eSUPS_WrPersistData_Shutdown mode (standard setting) after the storage of the persistent data.

In the eSUPS_WrPersistData_NoShutdown mode only the persistent data are saved, no QuickShutdown is performed.

In the eSUPS_ImmediateShutdown mode a QuickShutdown is executed immediately without saving data.

In the eSUPS_CheckPowerStatus mode only a check is performed as to whether a power failure has occurred. If this is the case, the module only switches back to the PowerOK state after the expiry of tRecoverTime (10s).

Independent of the mode and thus independent of the saving or the shutting down of the controller, the UPS switches the main board off after the capacitors have discharged, even if the voltage has returned in the meantime.

NOTICE

Caution when using files:

If other applications or the PLC keep other files open or write to them, this can lead to faulty files if the UPS switches off the controller.

VAR_INPUT

```
VAR_INPUT
    sNetID      : T_AmsNetId := '';            (* '' = local netid *)
    iPLCPort    : UINT := AMSPORT_R0_PLC_RTS1;     (* PLC Runtime System for writing persistent data
*)
    iUPSPort    : UINT := 16#4A8;          (* Port for reading Power State of UPS, dafault 16#4A8 *)
    tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;     (* ADS Timeout *)
    eUpsMode    : E_S_UPS_Mode := eSUPS_WrPersistData_Shutdown; (* UPS mode (w/
wo writing persistent data, w/wo shutdown) *)
    ePersistentMode : E_PersistentMode := SPDM_2PASS; (* mode for writing persistent data *)
    tRecoverTime    : TIME := T#10s;           (* ON time to recover from short power failure in mode
 eSUPS_WrPersistData_NoShutdown/eSUPS_CheckPowerStatus *)
END_VAR
```

E_S_UPS_Mode

sNetID : AmsNetID of the controller.

iPLCPort : Port number of the PLC runtime system (AMSPORT_R0_PLC_RTS1 = 801, AMSPORT_R0_PLC_RTS2 = 811, AMSPORT_R0_PLC_RTS3 = 821, AMSPORT_R0_PLC_RTS4 = 831).

iUPSPort : Port number via which the UPS status is read (standard value is 16#4A8).

tTimeout : Timeout for the execution of the QuickShutdown.

eUpsMode : The eUpsMode defines whether persistent data are to be written and whether a QuickShutdown is to be performed.
Standard value is eSUPS_WrPersistData_Shutdown, i.e. with writing of the persistent data and then QuickShutdown. See E_S_UPS_Mode.

ePersistentMode : Mode for the writing of the persistent data. Standard value is SPDM_2PASS.
SPDM_2PASS, all persistent data are saved at once, which can lead to the cycle time being exceeded.
SPDM_VAR_BOOST, here, each persistent variable is written separately; if there is a large amount of persistent data this can accordingly take many cycles. This is not recommended as some data may be lost if the time of the seconds UPS is not sufficient.

tRecoverTime : Time after which the UPS reverts to the PowerOK status in the case of UPS modes without shutdown.
The tRecoverTime must be somewhat longer than the maximum holding time of the UPS, since the UPS switches off even when the voltage returns.

VAR_OUTPUT

```
VAR_OUTPUT
    bPowerFailDetect    : BOOL;        (* TRUE while powerfailure is detected *)
    eState      : E_S_UPS_State;    (* current ups state *)
END_VAR
```

E_S_UPS_State

bPowerFailDetect : True during the power failure; False if the supply voltage is present.

eState : Internal state of the function block, for values see E_S_UPS_State.

VAR_GLOBAL

```
VAR_GLOBAL
    eGlobalSUpsState : E_S_UPS_State;     (* current ups state *)
END_VAR
```

E_S_UPS_State

eGlobalUpsState : Internal state of the function block as a global copy of the VAR_OUTPUT **eState**; for values see E_S_UPS_State.

Prerequisites

Development environment	Target platform	Hardware	PLC libraries to be linked
TwinCAT v2.11.0 build 2220 or higher (R3)	ARM	Seconds UPS	TcSystemCX80xx.lib

7.2.2 Data types

TYPE E_S_UPS_Mode

```
eSUPS_WrPersistData_Shutdown: Schreiben der Persistenten Daten und dann QuickShutdown
eSUPS_WrPersistData_NoShutdown: Nur Schreiben der Persistenten Daten (kein QuickShutdown)
eSUPS_ImmediateShutdown: Nur QuickShutdown (kein Schreiben der Persistenten Daten)
eSUPS_CheckPowerStatus: Nur Status ermitteln (weder Schreiben der Persistenten Daten noch QuickShutd
own)
```

Prerequisites

Development environment	Target platform	Hardware	PLC libraries to be linked
TwinCAT v2.11.0 build 2220 or higher (R3)	ARM	Seconds UPS	TcSystemCX80xx.lib

TYPE E_S_UPS_State

```
eSUPS_PowerOK:
    in allen Modi: Versorgungsspannung ist OK

eSUPS_PowerFailure:
    in allen Modi: Versorgungsspannung fehlerhaft (steht nur einen Zyklus an)

eSUPS_WritePersistentData:
    im Modus eSUPS_WrPersistData_Shutdown: Schreiben der Persistenten Daten ist aktiv
 im Modus eSUPS_WrPersistData_NoShutdown: Schreiben der Persistenten Daten ist aktiv

eSUPS_QuickShutdown:
    im Modus eSUPS_WrPersistData_Shutdown: QuickShutdown ist aktiv
 im Modus eSUPS_ImmediateShutdown: QuickShutdown ist aktiv

eSUPS_WaitForRecover:
    im Modus eSUPS_WrPersistData_NoShutdown: Warten auf Wiederkehr der Spannung
 im Modus eSUPS_CheckPowerStatus: Warten auf Wiederkehr der Spannung

eSUPS_WaitForPowerOFF:
    im Modus eSUPS_WrPersistData_Shutdown: Warten auf das Abschalten durch die USV
 im Modus eSUPS_ImmediateShutdown: Warten auf das Abschalten durch die USV
```

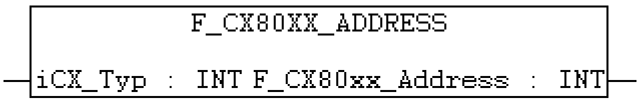
Prerequisites

Development environment	Target platform	Hardware	PLC libraries to be linked
TwinCAT v2.11.0 build 2220 or higher (R3)	ARM	Seconds UPS	TcSystemCX80xx.lib

7.3 Diagnostics

7.3.1 FUNCTION F_CX80xx_ADDRESS

With this function the address selection switch or the DIP switch of the CX80xx device can be read out. Here, for example, you can activate different parts of the program depending on the address by reading the switch position.



VAR_INPUT

```
VAR_INPUT
    iCX_Typ      : INT;
END_VAR
```

iCX_Typ : The CX type used is entered here - just the number without the designation CX: for example, CX8031 is then entered as 8031.

VAR_OUTPUT

```
F_CX80xx_ADDRESS     : INT;
```

F_CX80xx_ADDRESS : -1, non-implemented CX, address of the switch

Prerequisites

Development environment	Target platform	Hardware	PLC libraries to be linked
TwinCAT v2.11.0 build 2220 or higher (R3)	ARM	CX80xx	TcSystemCX80xx.lib

7.4 CAN

7.4.1 Reading the CAN baud rate

The baud rate can be displayed and evaluated via InfoData[1]. This can be helpful for slaves with AutoBaud, to ascertain whether the right baud rate was found if there is an issue with the communication, for example.

NodeState value	Description
0x01040400	1 MBaud
0x01040600	800 kBaud
0x01040C00	500 kBaud
0x010A0C00	250 kBaud
0x01160C00	125 kBaud
0x011C0C00	100 kBaud
0x013A0C00	50 kBaud
0x01940C00	20 kBaud
0x01941A10	10 kBaud

7.4.2 Sending any CAN message

Sending any CAN message

The ADSWRITE command can be used to send any CAN message.

Input parameters	Description
NETID	NetId of the CAN interface
Port number	200
IDXGRP	16#0000F921
IDXOFFS	0
LEN	11 bytes
SRCADDR	Pointer to an 11 byte ARRAY

Table 3: 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

7.4.3 CX8050 Master

7.4.3.1 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	ADS NetID of the CAN interface
Port number	0x1000$_{hex}$ + NodeId (slave number)
IDXGRP	SDO Index
IDXOFFS	SDO Subindex
LEN	Length of SDO data (1...4)

Setting individual CANopen nodes to pre-operational or operational state

The ADSWRTCTL block can be used to set individual CANopen nodes to pre-operational or operational state.

Input parameters	Description
NETID	ADS NetID of the CAN interface
Port number	$0x1000_{hex}$ + NodeId (slave number)
ADSSTATE	ADSSTATE_RUN
DEVSTATE	0 - Pre / 1 - Operational
LEN	0
SRCADDR	0

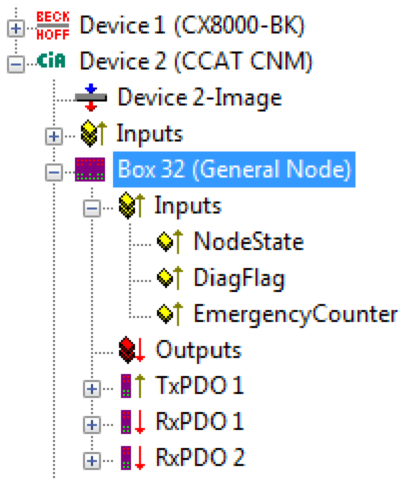
Restarting the CAN interface

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

Input parameters	Description
NETID	ADS NetID of the CAN interface
Port number	200_{dec}
ADSSTATE	ADSSTATE_STOP, ADSSTATE_RUN
DEVSTATE	0
LEN	0
SRCADDR	0

7.4.3.2 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.



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	Bus-OFF
12	Pre-Operational
13	Servere 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 200

Reading of emergency telegrams with AdsRead

Input parameters	Description
NETID	NetId of the CAN interface
Port number	200
IDXGRP	16#xxxxF180 (xxxx) Node-Id, the Diag flag is only reset when at least 106 bytes are read 16#xxxxF181 (xxxx) Node-Id, the Diag flag is reset immediately
IDXOFFS	Byte Offset

Table 4: Description of the array

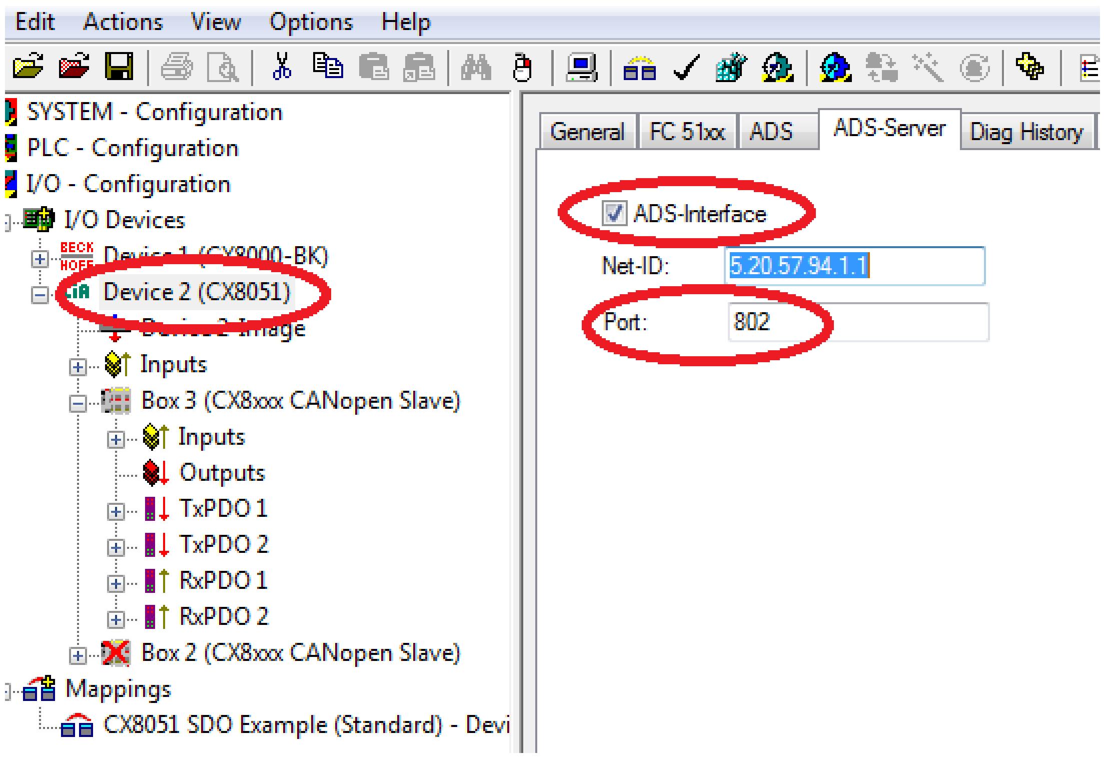
Offset	Bit	Value / description
0 - 1	Bit 0	reserved
	Bit 1	Boot up message not received or incorrect
	Bit 2	Emergency-Overflow
	Bit 3 - 15	reserved
2 - 3	Bit 0 - 14	TX-PDO (i+1) received
	Bit 15	All TX PDOs 16-n received
4 - 5	Bit 0 - 4	1: Incorrect TX PDO length
		2: Synchronous TX PDO absent
		3: Node signalling 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 sub-index
11	Bit 0 - 7	reserved
12	Bit 0 - 7	Abort errorClass
13	Bit 0 - 7	Abort errorCode
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)

7.4.4 CX8051 Slave

7.4.4.1 Receiving SDO data in the PLC

SDO data that are unknown to the CANopen part of the software and cannot be processed automatically are transferred to the PLC, where they are evaluated and answered via ADS notification.

To this end the ADS port must be enabled in the System Manager under CAN device (CX8051).



[✓] ADS-Interface
Net-ID: 5.20.57.94.1.1
Port: 802

SDO Read request

Data to be read must be received with ADSREADIND and answered with ADSREADRES.

Input parameter ADSREADIND	Description
NETID	NetID of the CAN interface
Port number	0x1000$_{hex}$ + node number
IDXGRP	16#8000_0000 + SDO Index (IDXGRP.31 = ADS-Notification)
IDXOFFS	SDO sub index
LEN	not required for reading

You now have to respond to the ADS indication with an ADS read response.

Input parameter ADSREADRES	Description
NETID	NetID of the CAN interface
Port number	0x1000$_{hex}$ + node number
INVOKEID	INVOKEID of the ADSREADIND block
RESULT	error <> 0, error-free = 0
LEN	Length of the data

SDO Write request

Data to be written must be received with ADSWRITEIND and answered with ADSWRITERES.

Output parameter ADSWRITEIND	Description
NETID	NetID of the CAN interface
Port number	0x1000$_{hex}$ + node number
IDXGRP	16#8000_0000 + SDO Index (IDXGRP.31 = ADS Notification)
IDXOFFS	SDO Subindex
LEN	Number of received data in bytes

You now have to respond to the ADS indication with an ADS write response.

Input parameter ADSWRITERES	Description
NETID	NetID of the CAN interface
Port number	0x1000$_{hex}$ + node number
INVOKEID	INVOKEID of the ADSWRITEIND block
RESULT	error <> 0, error-free = 0

7.4.4.2 Switching slave node to PreOp from the PLC

The ADSWRTCTL block can be used to set individual CANopen nodes to pre-operational or operational state. A fixed baud rate is required for this purpose.

Input parameters	Description
NETID	NetId of the CAN interface
Port number	0x1000$_{hex}$ + NodeId (slave number)
ADSSTATE	ADSSTATE_RUN
DEVSTATE	0 - Pre / 1 - Operational
LEN	0
SRCADDR	0

8 Ethernet X001 Interface

8.1 System introduction

8.1.1 Ethernet

Ethernet was originally developed by DEC, Intel and XEROX (as the "DIX" standard) for passing data between office devices. The term nowadays generally refers to the *IEEE 802.3 CSMA/CD* specification, published in 1985. Because of the high acceptance around the world this technology is available everywhere and is very economical. This means that it is easy to make connections to existing networks.

There are now a number of quite different transmission media: coaxial cable (10Base5), optical fiber (10BaseF) or twisted pairs (10BaseT) with screen (STP) or without screen (UTP). A variety of topologies such as ring, line or star can be constructed with Ethernet.

Ethernet transmits Ethernet packets from a sender to one or more receivers. This transmission takes place without acknowledgement, and without the repetition of lost packets. To achieve reliable data communication, there are protocols, such as TCP/IP, that can run on top of Ethernet.

MAC-ID

The sender and receiver of Ethernet packets are addressed by means of the MAC-ID. The MAC-ID is a 6 byte identification code unique to every Ethernet device in the world. The MAC-ID consists of two parts. The first part (i.e. the first 3 bytes) is a manufacturer identifier. The identifier for Beckhoff is 00 01 05. The next 3 bytes are assigned by the manufacturer and implement a unique serial number. The MAC-ID can, for example, be used for the BootP protocol in order to set the TCP/IP number. This involves sending a telegram containing the information such as the name or the TCP/IP number to the corresponding node. You can read the MAC-ID with the KS2000 configuration software.

The Internet Protocol (IP)

The internet protocol (IP) forms the basis of this data communication. IP transports data packets from one device to another; the devices can be in the same network, or in different networks. IP here looks after the address management (finding and assigning MAC-IDs), segmentation and routing. Like the Ethernet protocol, IP does not guarantee that the data is transported - data packets can be lost, or their sequence can be changed.

TCP/IP was developed to provide standardized, reliable data exchange between any number of different networks. TCP/IP is thus substantially independent of the hardware or software being used. Although the term is often used as if it were a single concept, a number of protocols are layered together: e.g. IP, TCP, UDP, ARP and ICMP.

Transmission Control Protocol (TCP)

The Transmission Control Protocol (TCP) which runs on top of IP is a connection-oriented transport protocol. It includes error detection and error handling mechanisms. Lost telegrams are repeated.

User Datagram Protocol (UDP)

UDP is connectionless transport protocol. It provides no control mechanism when exchanging data between sender and receiver. This results in a higher processing speed than, for example, TCP. Checking whether or not the telegram has arrived must be carried out by the higher-level protocol.

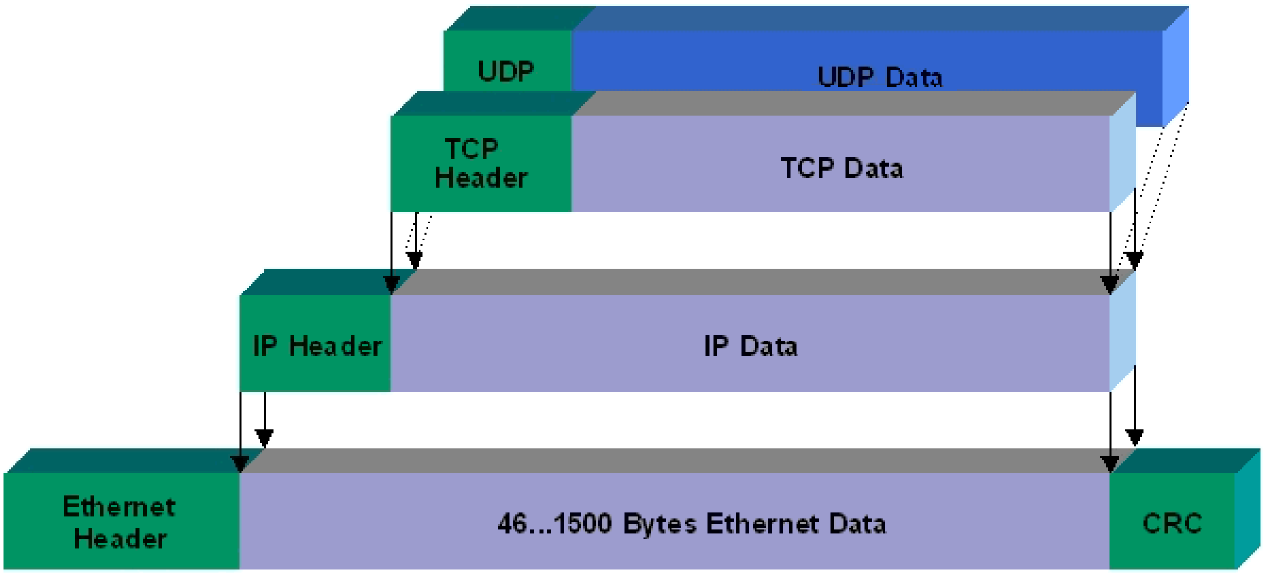


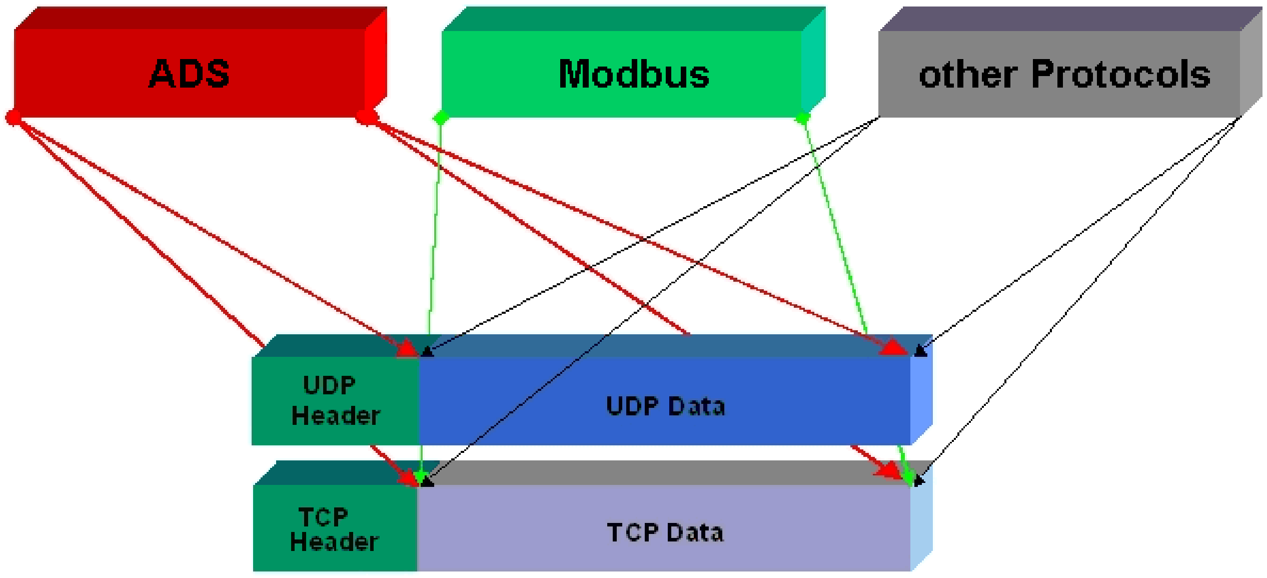
Fig. 4: Ethernet protocol

Protocols running on top of TCP/IP and UDP/IP

The following protocols can run on top of TCP/IP or UDP:

- ADS
- ModbusTCP

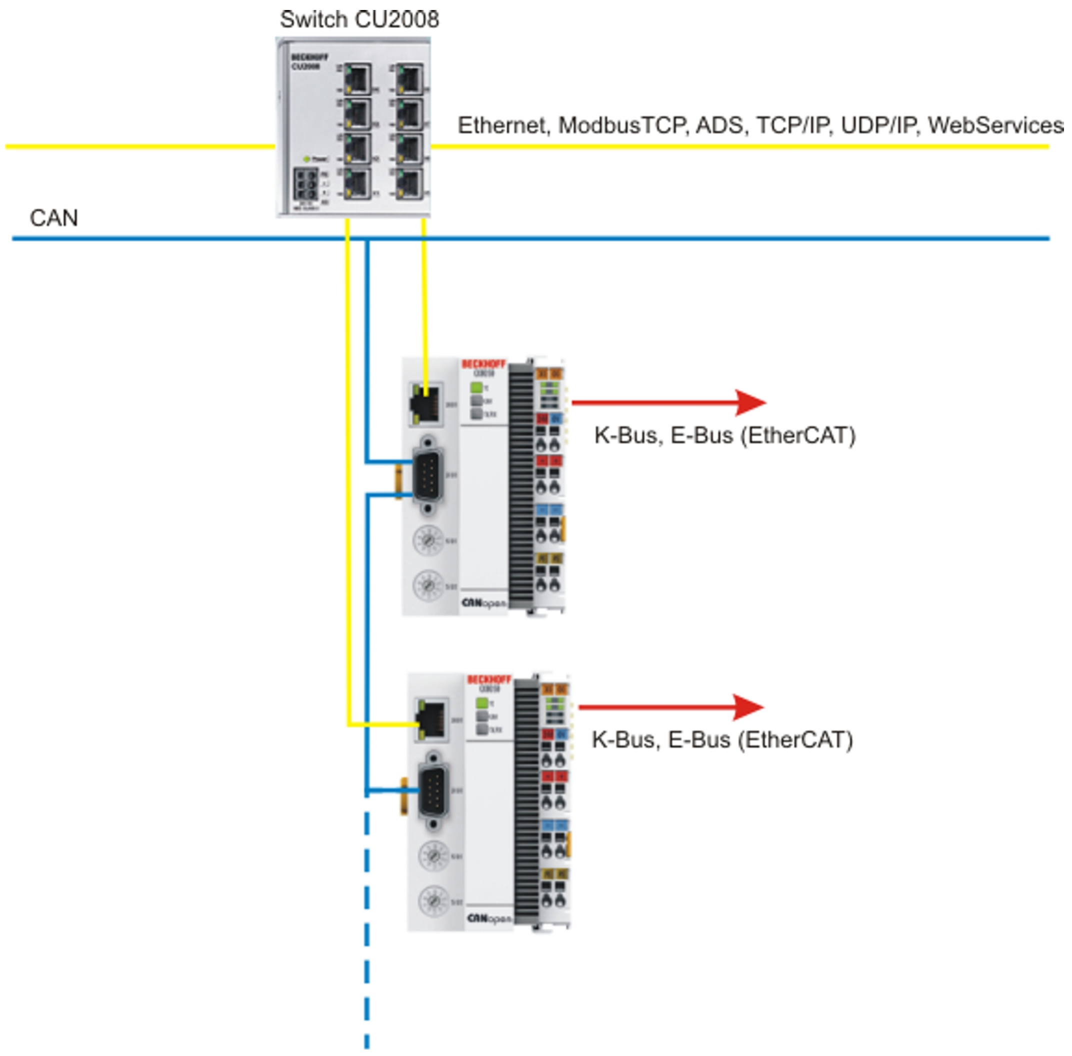
Both of these protocols are implemented in parallel on the Bus Coupler, so that no configuration is needed to activate the protocols.



ADS can be used on top of either TCP or UDP, but ModbusTCP is always based on TCP/IP.

8.1.2 Topology example

CX805x



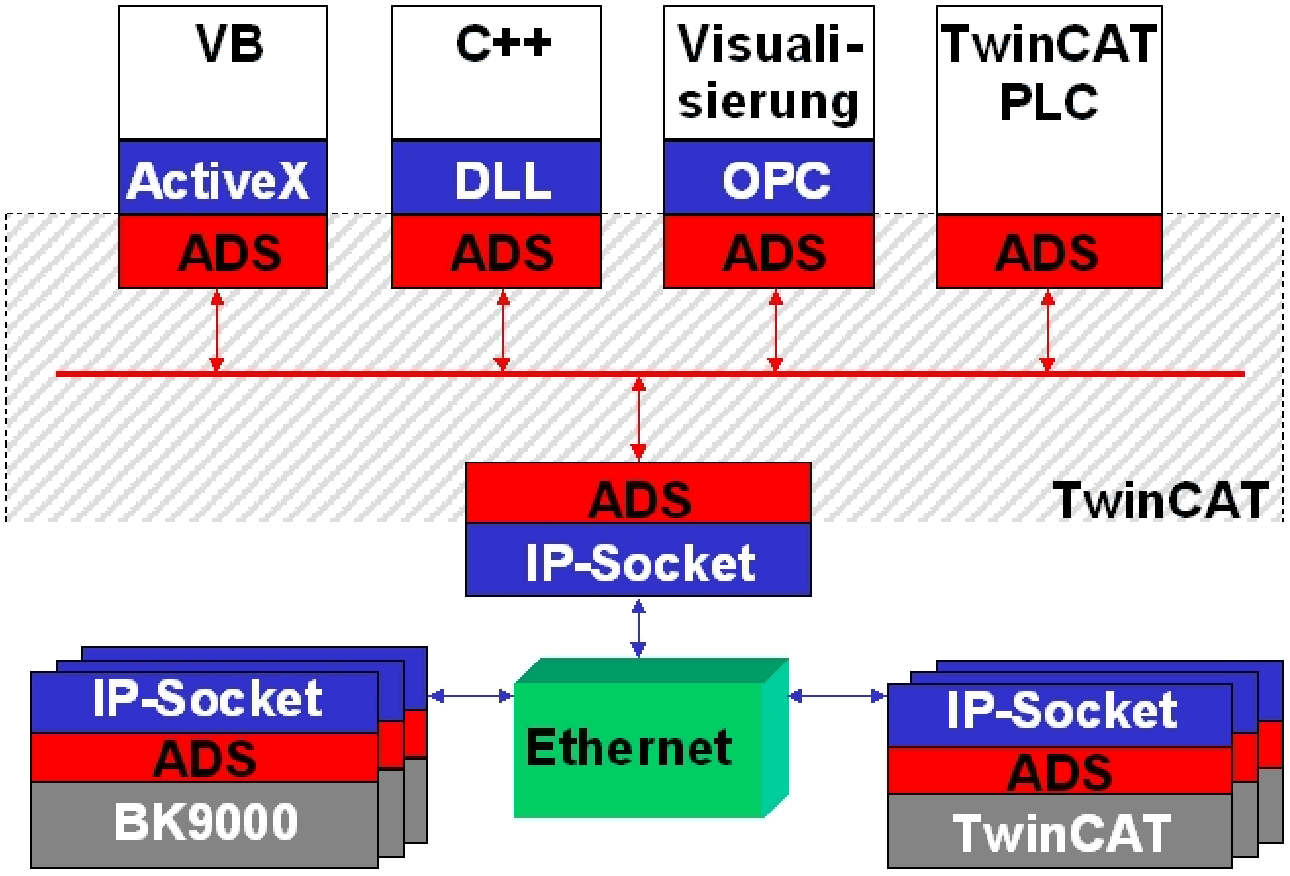
Observe system load

Observe the system load of your CX805x when using further Ethernet protocols such as ModbusTCP/UDP or Web Services. A high load can slow down Ethernet communication significantly.

8.2 ADS-Communication

Communication

The ADS protocol (ADS: Automation Device Specification) is a transport layer within the TwinCAT system. It was developed for data exchange between the different software modules, for instance the communication between the NC and the PLC. This protocol enables communication with other tools from any point within the TwinCAT. If communication with other PCs or devices is required, the ADS protocol can use TCP/IP as a basis. Within a networked system it is thus possible to reach all data from any point.



The ADS protocol runs on top of the TCP/IP or UDP/IP protocols. It allows the user within the Beckhoff system to use almost any connecting route to communicate with all the connected devices and to parameterize them. Outside the Beckhoff system a variety of methods are available to exchange data with other software tools.

Software interfaces

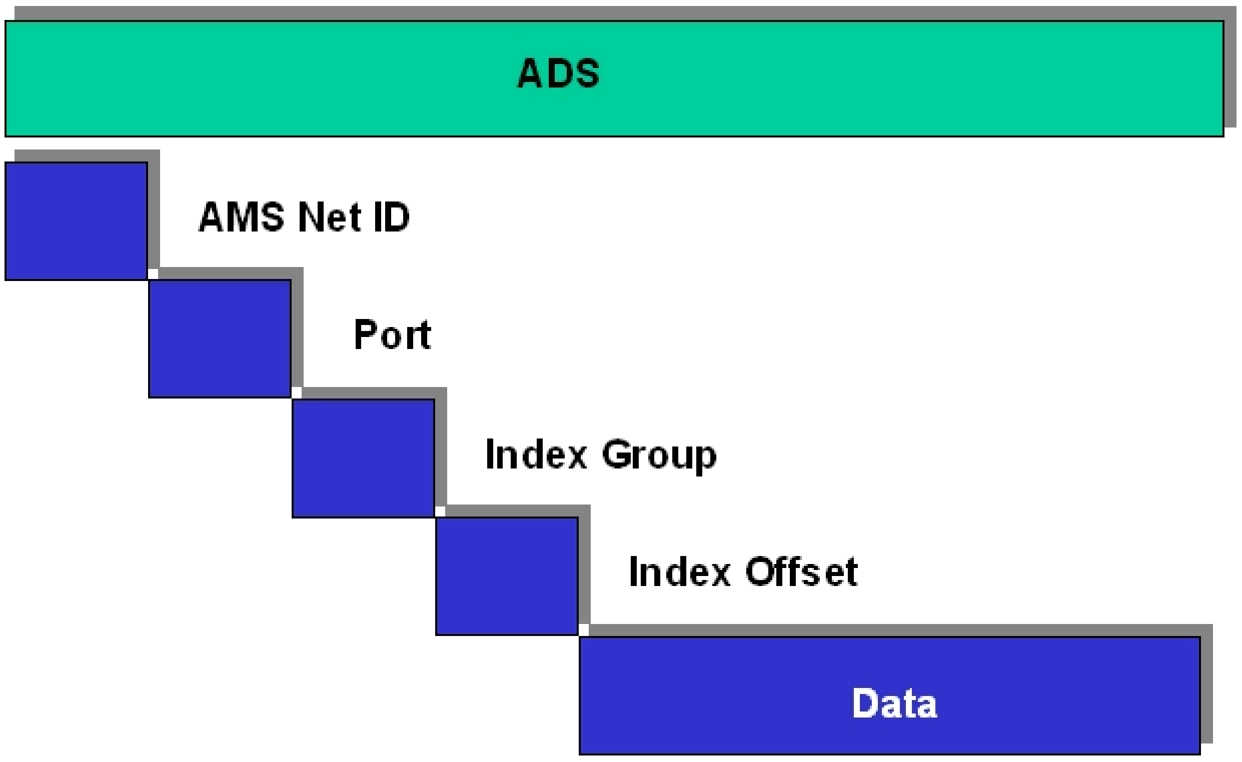
ADS-OCX
The ADS-OCX is an Active-X component. It offers a standard interface to, for instance, Visual Basic, Delphi, etc.

ADS-DLL
You can link the ADS-DLL (DLL: Dynamic Link Library) into your C program.

OPC
The OPC interface is a standardized interface for communication used in automation engineering. Beckhoff offer an OPC server for this purpose.

Protocol

The ADS functions provide a method for accessing the Bus Coupler information directly from the PC. ADS function blocks can be used in TwinCAT PLC Control for this. The function blocks are contained in the *PLCSystem.lib* library. It is also equally possible to call the ADS functions from AdsOCX, ADSDLL or OPC.



AMSNetID
The AMSNetID provides a reference to the device that is to be addressed. This is taken from the MAC address of the first Ethernet port (X001) and is printed on the side of the CX80xx. For the AMSNetID the bytes 3 to 6 plus ".1.1" are typically used.
Example:
MAC address 00-01-**05-01-02-03**
AMSNetID 5.1.2.3.1.1

Port number
The port number distinguishes sub-elements in the connected device.
Port 801: local process data PLC runtime 1

Index group
The index group distinguishes different data within a port.

Index offset
Indicates the offset, the byte from which reading or writing is to start.

Len
Gives the length of the data, in bytes, that is to be read or written.

TCP port number
The TCP port number for the ADS protocol is 48898 or 0xBF02.

9 CAN

9.1 Introduction

CANopen is a widely used CAN application layer, developed by the CAN in Automation association (CiA, http://www.can-cia.org), which has meanwhile been adopted for international standardization.

Device Model

CANopen consists of the protocol definitions (communication profile) and of the device profiles that standardize the data contents for the various device classes. Process data objects (PDO) [▶ 70] are used for fast communication of input and output data. The CANopen device parameters and process data are stored in a structured object directory. Any data in this object directory is accessed via service data objects (SDO). There are, additionally, a few special objects (such as telegram types) for network management (NMT), synchronization, error messages and so on.

Communication Types

CANopen defines a number of communication classes for the input and output data (process data objects):

- Event driven [▶ 70]: Telegrams are sent as soon as their contents have changed. This means that the process image as a whole is not continuously transmitted, only its changes.
- Cyclic synchronous [▶ 70]: A SYNC telegram causes the modules to accept the output data that was previously received, and to send new input data.
- Requested [▶ 70]: A CAN data request telegram causes the modules to send their input data.

The desired communication type is set by the Transmission Type [▶ 70] parameter.

Device Profile

The BECKHOFF CANopen devices support all types of I/O communication, and correspond to the device profile for digital and analog input/output modules (DS401 Version 1). For reasons of backwards compatibility, the default mapping was not adapted to the DS401 V2 profile version.

Transmission Rates

Nine transmission rates from 10 kbaud up to 1 Mbaud are available for different bus lengths. The effective utilization of the bus bandwidth allows CANopen to achieve short system reaction times at relatively low data rates.

Topology

Topology [▶ 25]

CAN is based on a linear topology. The number of devices participating in each network is logically limited by CANopen to 128, 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 transit time required on the bus medium. For 1 Mbaud, for instance, the network may extend 25 m, whereas at 50 kbaud the network may reach up to 1000 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 CANopen parameters to be set conveniently. An "EDS" file (an electronic data sheet) is available on the BECKHOFF website (http://www.beckhoff.com) for the parameterization of BECKHOFF CANopen devices using configuration tools from other manufacturers.

Certification

The BECKHOFF CANopen devices have a powerful implementation of the protocol, and are certified by the CAN in Automation Association (http://www.can-cia.org).

9.2 Protocol description

9.2.1 Network Management

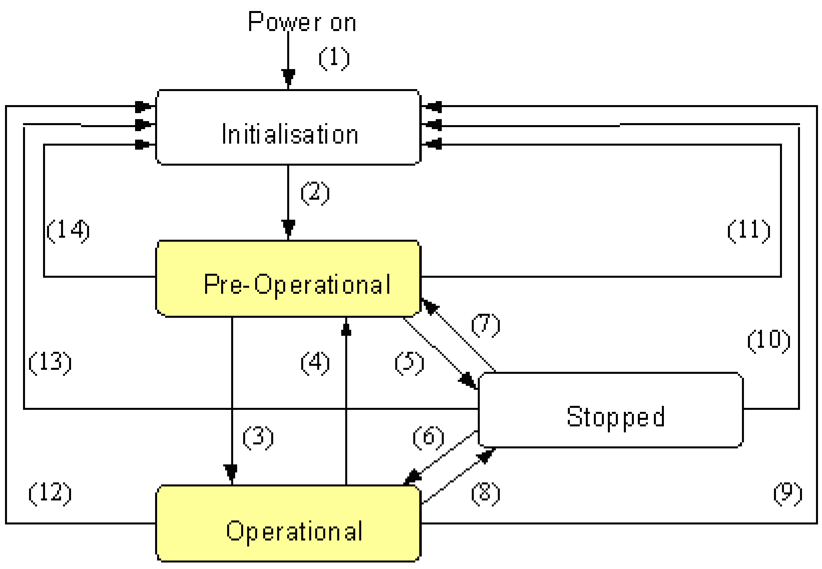
Simple Boot-Up

CANopen allows the distributed network to boot in a very simple way. After initialization, the modules are automatically in the *Pre-Operational* state. In this state it is already possible to access the object directory using service data objects (SDOs) with default identifiers, so that the modules can be configured. Since default settings exist for all the entries in the object directory, it is in most cases possible to omit any explicit configuration.

Only one CAN message is then required to start the module: Start_Remote_Node: Identifier *0*, two data bytes: 0x01, 0x00. It switches the node into the *Operational state*.

Network Status

The states and the state transitions involved as CANopen boots up can be seen from the state diagram:



Pre-Operational

After initialization the Bus Coupler goes automatically (i.e. without the need for any external command) into the *Pre-Operational* state. In this state it can be configured, since the service data objects (SDOs) are already active. The process data objects, on the other hand, are still locked.

Operational

In the *Operational state* the process data objects are also active.

If external influences (such as a CAN error, or absence of output voltage) or internal influences (such as a K-Bus error) mean that it is no longer possible for the Bus Coupler to set outputs, to read inputs or to communicate, it attempts to send an appropriate emergency message, goes into the fault state, and thus returns to the *Pre-Operational* state. In this way the NMT status machine in the network master can also immediately detect fatal errors.

Stopped

In the *Stopped state* (formerly: *Prepared*) data communication with the Coupler is no longer possible - only NMT messages are received. The outputs go into the fault state.

State Transitions

The network management messages have a very simple structure: CAN identifier *0*, with two bytes of data content. The first data byte contains what is known as the command specifier (cs), and the second data byte contains the node address, the node address *0* applying to all nodes (broadcast).

11 bit identifier	2 bytes of user data							
0x00	cs	Node-ID						

The following table gives an overview of all the CANopen state transitions and the associated commands (command specifier in the NMT master telegram):

Status transition	Command Specifier cs	Explanation
(1)	-	The initialization state is reached automatically at power-up
(2)	-	After initialization the pre-operational state is reached automatically - this involves sending the boot-up message.
(3), (6)	cs = 1 = 0x01	Start_Remote_Node. Starts the module, enables outputs, starts transmission of PDOs.
(4), (7)	cs = 128 = 0x80	Enter_Pre-Operational. Stops PDO transmission, SDO still active.
(5), (8)	cs = 2 = 0x02	Stop_Remote_Node. Outputs go into the fault state, SDO and PDO switched off.
(9), (10), (11)	cs = 129 = 0x81	Reset_Node. Carries out a reset. All objects are reset to their power-on defaults.
(12), (13), (14)	cs = 130 = 0x82	Reset_Communication. Carries out a reset of the communication functions. Objects 0x1000 - 0x1FFF are reset to their power-on defaults.

Example 1

The following telegram puts all the modules in the network into the error state (outputs in a safe state):

11 bit identifier	2 bytes of user data							
0x00	0x02	0x00						

Example 2

The following telegram resets node 17:

11 bit identifier	2 bytes of user data							
0x00	0x81	0x11						

Boot-up message

After the initialization phase and the self test, the Bus Coupler sends the boot-up message, a CAN message with no data bytes and with the identifier of the emergency message: CAN-ID = 0x700 + Node-ID. In this way temporary failure of a module during operation (e.g. due to a voltage interruption), or a module that is switched on at a later stage, can be reliably detected, even without Node Guarding. The sender can be determined from the message identifier (see default identifier allocation).

It is also possible, with the aid of the boot-up message, to recognize the nodes present in the network at start-up with a simple CAN monitor, without having to make write access to the bus (such as a scan of the network by reading out parameter 0x1000).

Finally, the boot-up message communicates the end of the initialization phase; the Bus Coupler signals that it can now be configured or started.

Firmware BA

Up to firmware status BA the emergency identifier was used for the boot up message.

Format of the Boot-up message

11 bit identifier	1 byte of user data							
0x700 (=1792) + Node-ID	0x00							

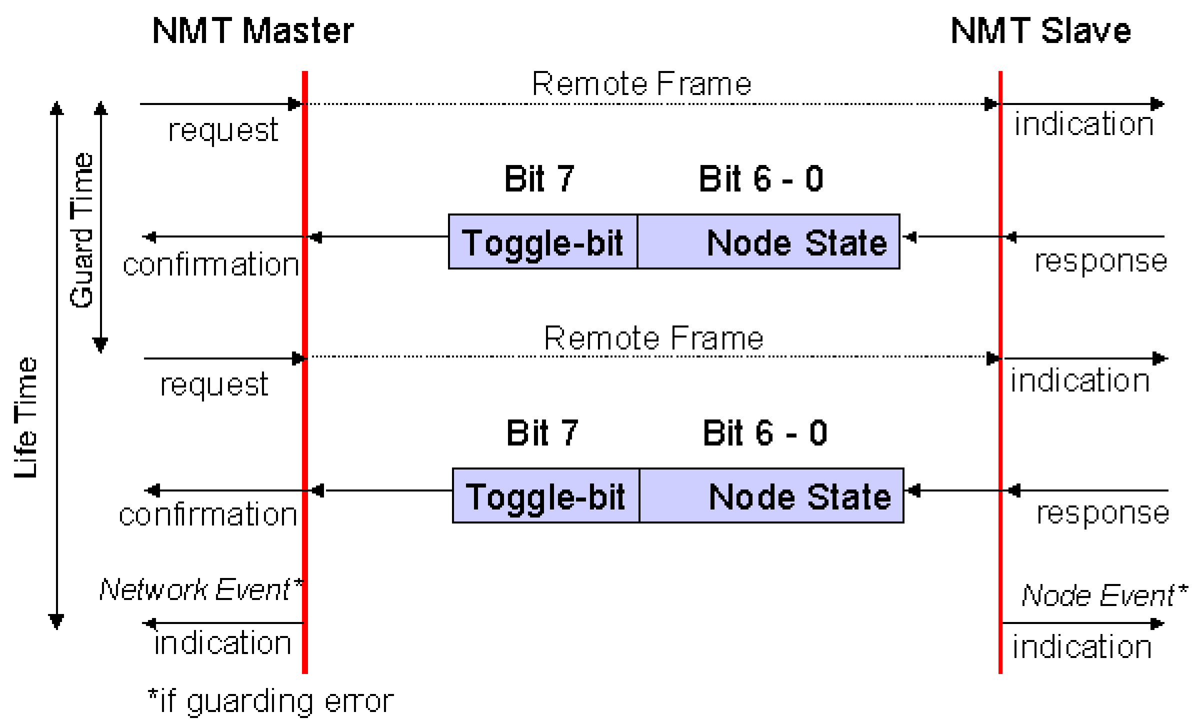
Node Monitoring

Heartbeat and guarding mechanisms are available to monitor failures in the CANopen network. These are of particular importance for CANopen, since modules do not regularly speak in the event-driven mode of operation. In the case of "guarding", the devices are cyclically interrogated about their status by means of a data request telegram (remote frame), whereas with "heartbeat" the nodes transmit their status on their own initiative.

Guarding: Node Guarding and Life Guarding

Node Guarding is used to monitor the non-central peripheral modules, while they themselves can use Life Guarding to detect the failure of the guarding master. Guarding involves the master sending remote frames (remote transmit requests) to the guarding identifier of the slaves that are to be monitored. These reply with the guarding message. This contains the slave’s status code and a toggle bit that has to change after every message. If either the status or the toggle bit do not agree with that expected by the NMT master, or if there is no answer at all, the master assumes that there is a slave fault.

Guarding procedure



Protocol

The toggle bit (t) transmitted in the first guarding telegram has the value *0*. After this, the bit must change (toggle) in every guarding telegram so that the loss of a telegram can be detected. The node uses the remaining seven bits to transmit its network status (s):

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

Example

The guarding message for node 27 (0x1B) must be requested by a remote frame having identifier 0x71B (1819_{dec}). If the node is *Operational*, the first data byte of the answer message alternates between 0x05 and 0x85, whereas in the *Pre-Operational* state it alternates between 0x7F and 0xFF.

Guard time and life time factor

If the master requests the guard messages in a strict cycle, the slave can detect the failure of the master. In this case, if the slave fails to receive a message request from the master within the set *Node Life Time* (a guarding error), it assumes that the master has failed (the watchdog function). It then puts its outputs into the error state, sends an emergency telegram, and returns to the pre-operational state. After a guarding time-out the procedure can be re-started by transmitting a guarding telegram again.

The node life time is calculated from the guard time (object 0x100C) and life time factor (object 0x100D) parameters:

Life time = guard time x life time factor

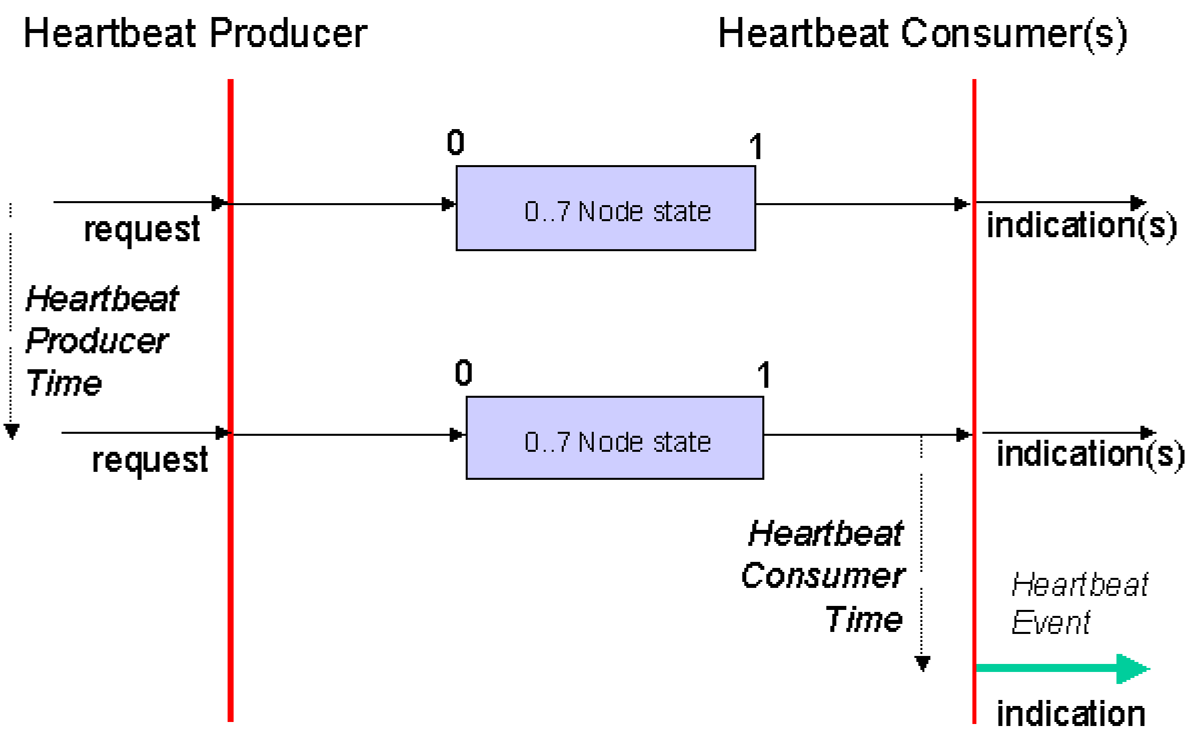
If either of these two parameters is "0" (the default setting), the master will not be monitored (no life guarding).

Heartbeat: Node Monitoring without Remote Frame

In the heart beat procedure, each node transmits its status message cyclically on its own initiative. There is therefore no need to use remote frames, and the bus is less heavily loaded than under the guarding procedure.

The master also regularly transmits its heartbeat telegram, so that the slaves are also able to detect failure of the master.

Heartbeat procedure



Protocol

The toggle bit is not used in the heart beat procedure. The nodes send their status cyclically (s). See Guarding.

9.2.2 Process Data Objects (PDO)

Introduction

In many fieldbus systems the entire process image is continuously transferred - usually in a more or less cyclic manner. CANopen is not limited to this communication principle, since the multi-master bus access protocol allows CAN to offer other methods. Under CANopen the process data is not transferred in a master/ slave procedure, but follows instead the producer-consumer model. In this model, a bus node transmits its data, as a producer, on its own accord. This might, for example, be triggered by an event. All the other nodes listen, and use the identifier to decide whether they are interested in this telegram, and handle it accordingly. These are the consumers.

The process data in CANopen is divided into segments with a maximum of 8 bytes. These segments are known as process data objects (PDOs). The PDOs each correspond to a CAN telegram, whose specific CAN identifier is used to allocate them and to determine their priority. Receive PDOs (RxPDOs) and transmit PDOs (TxPDOs) are distinguished, the name being chosen from the point of view of the device: an input/ output module sends its input data with TxPDOs and receives its output data in the RxPDOs. **This naming convention is retained in the TwinCAT System Manager.**

Communication parameters

The PDOs can be given different communication parameters according to the requirements of the application. Like all the CANopen parameters, these are also available in the device's object directory, and can be accessed by means of the service data objects. The parameters for the receive PDOs are at index 0x1400 (RxPDO1) onwards. There can be up to 512 RxPDOs (ranging up to index 0x15FF). In the same way, the entries for the transmit PDOs are located from index 0x1800 (TxPDO1) to 0x19FF (TxPDO512).

The BECKHOFF Bus Couplers or Fieldbus Coupler Box modules make 16 RxPDO and TxPDOs available for the exchange of process data (although the figure for Economy and LowCost BK5110 and LC5100 Couplers and the Fieldbus Boxes is 5 PDOs each, since these devices manage a lower quantity of process data). The FC510x CANopen master card supports up to 192 transmit and 192 receive PDOs for each channel - although this is restricted by the size of the DPRAM. Up to 32 TxPDOs and 32 RxPDOs can be handled in slave mode.

For each existing process data object there is an associated communication parameter object. The TwinCAT System Manager automatically assigns the set parameters to the relevant object directory entries. These entries and their significance for the communication of process data are explained below.

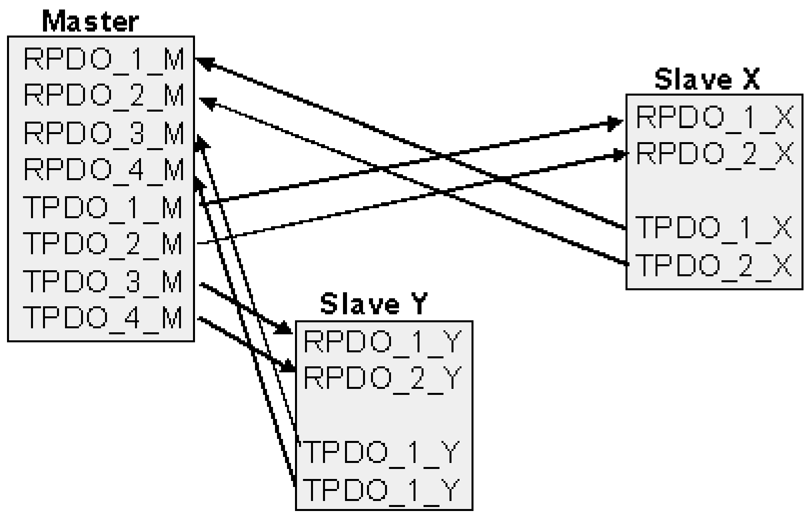
PDO Identifier

The most important communication parameter in a PDO is the CAN identifier (also know as the communication object identifier, or COB-ID). It is used to identify the data, and determines their priority for bus access. For each CAN data telegram there may only be one sender node (producer), although all messages sent in the CAN broadcast procedure can be received, as described, by any number of nodes (consumers). Thus a node can make its input information available to a number of bus devices at the same time - even without transferring them through a logical bus master. The identifier is located in sub-index 1 of the communication parameter set. It is coded as a 32-bit value in which the least significant 11 bits (bits 0...10) contain the identifier itself. The data width of the object of 32 bits also allows 29-bit identifiers in accordance with CAN 2.0B to be entered, although the default identifiers always refer to the more usual 11-bit versions. Generally speaking, CANopen is economical it its use of the available identifiers, so that the use of the 29-bit versions remains limited to unusual applications. It is therefore also not supported by a Beckhoff's CANopen devices. The highest bit (bit 31) can be used to activate the process data object or to turn it off.

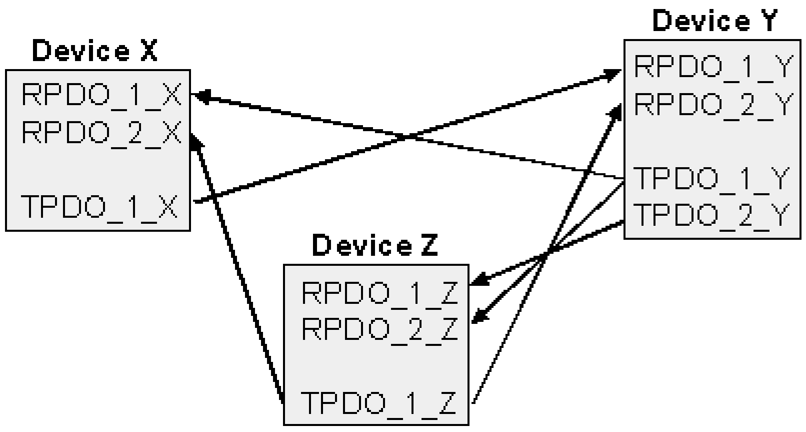
A complete identifier list [▶ 139] is provided in the appendix.

PDO linking

In the system of default identifiers, all the nodes (here: slaves) communicate with one central station (the master), since slave nodes do not listen by default to the transmit identifier of any other slave node.



Default identifier allocation: Master/Slave

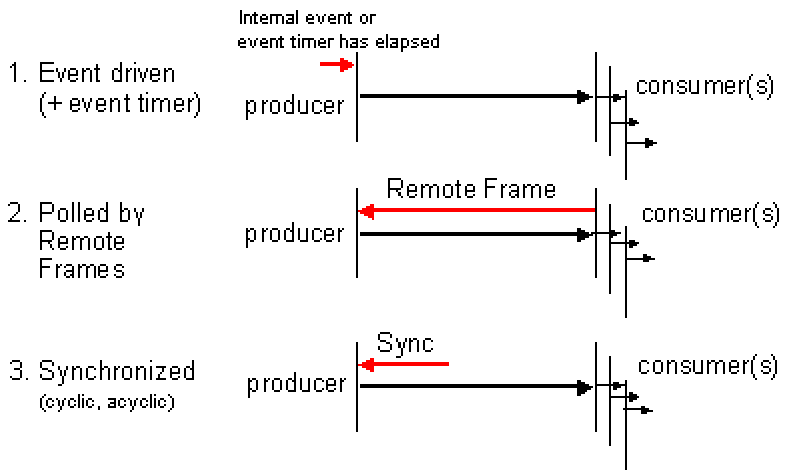


PDO linking: Peer to Peer

If the consumer-producer model of CANopen PDOs is to be used for direct data exchange between nodes (without a master), the identifier allocation must be appropriately adapted, so that the TxPDO identifier of the producer agrees with the RxPDO identifier of the consumer: This procedure is known as PDO linking. It permits, for example, easy construction of electronic drives in which several slave axes simultaneously listen to the actual value in the master axis TxPDO.

PDO Communication Types: Outline

CANopen offers a number of possible ways to transmit process data (see also: Notes on PDO Parameterization [▸ 77]).)



.

Event driven

The "event" is the alteration of an input value, the data being transmitted immediately after this change. The event-driven flow can make optimal use of the bus bandwidth, since instead of the whole process image it is only the changes in it that are transmitted. A short reaction time is achieved at the same time, since when an input value changes it is not necessary to wait for the next interrogation from a master.

As from CANopen Version 4 it is possible to combine the event driven type of communication with a cyclic update. Even if an event has not just occurred, event driven TxPDOs are sent after the event timer has elapsed. If an event does occur, the event timer is reset. For RxPDOs the event timer is used as a watchdog in order to monitor the arrival of event driven PDOs . If a PDO does not arrive within a set period of time, the bus node adopts the error state.

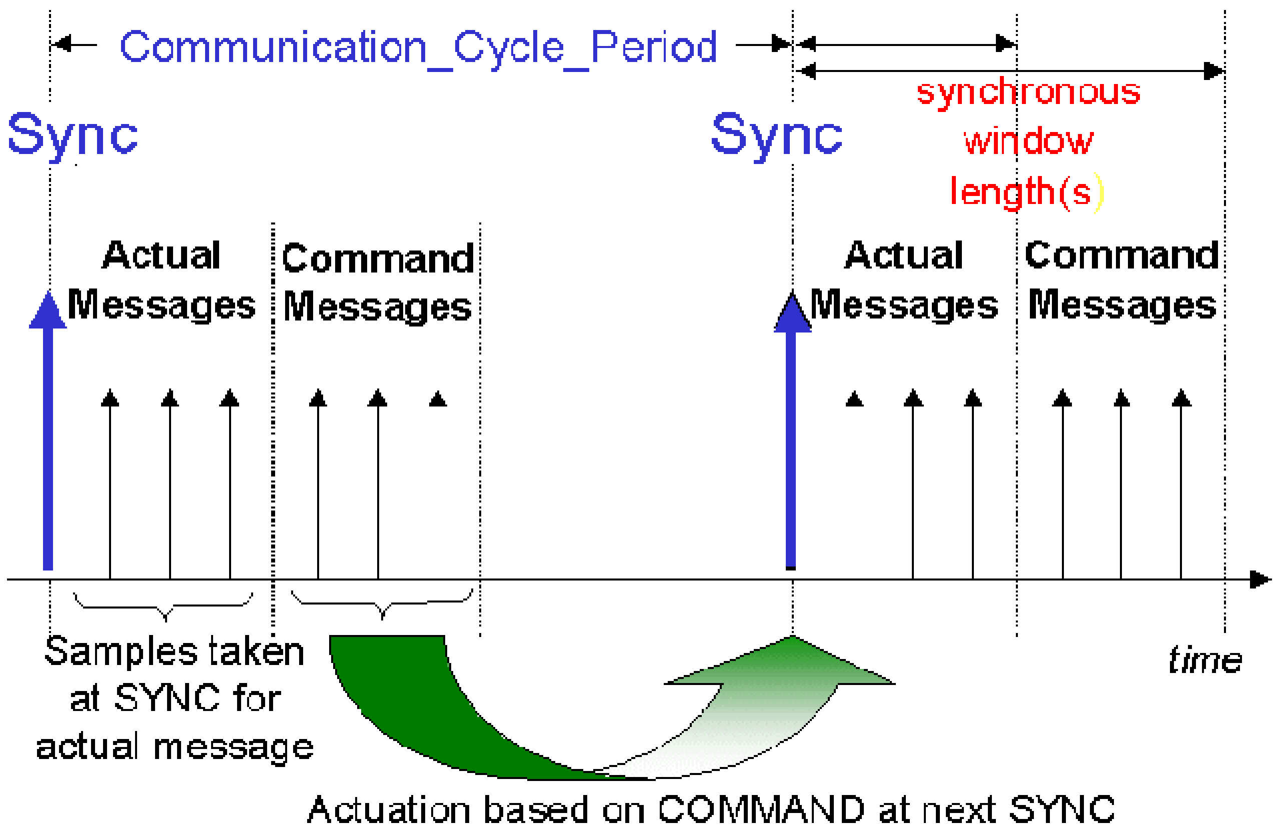
Polled

The PDOs can also be polled by data request telegrams (remote frames). In this way it is possible to get the input process image of event-driven inputs onto the bus, even when they do not change, for instance through a monitoring or diagnostic device brought into the network while it is running. The time behavior of remote frame and answer telegrams depends on what CAN controller is in use (Fig. 8). Components with full integrated message filtering ("FullCAN") usually answer a data request telegram immediately, transmitting data that is waiting in the appropriate transmit buffer - it is the responsibility of the application to see that the data there is continuously updated. CAN controllers with simple message filtering (BasicCAN) on the other hand pass the request on to the application which can now compose the telegram with the latest data. This does take longer, but does mean that the data is up-to-date. BECKHOFF use CAN controllers following the principle of Basic CAN.

Since this device behavior is usually not transparent to the user, and because there are CAN controllers still in use that do not support remote frames at all, polled communication can only with reservation be recommended for operative running.

Synchronized

It is not only for drive applications that it is worthwhile to synchronize the determination of the input information and the setting the outputs. For this purpose CANopen provides the SYNC object, a CAN telegram of high priority but containing no user data, whose reception is used by the synchronized nodes as a trigger for reading the inputs or for setting the outputs.


Communication_Cycle_Period
Sync
Sync
synchronous window length(s)
Actual Messages
Command Messages
Actual Messages
Command Messages
Samples taken at SYNC for actual message
time
Actuation based on COMMAND at next SYNC

PDO transmission types: Parameterisation

The PDO transmission type parameter specifies how the transmission of the PDO is triggered, or how received PDOs are handled.

Transmission type	Cyclical	Acyclical	Synchronous	Asynchronous	Only RTR
0		X	X		
1-240	X		X		
241-251	- reserved -				
252			X		X
253				X	X
254, 255				X	

The type of transmission is parameterized for RxPDOs in the objects at 0x1400ff, sub-index 2, and for TxPDOs in the objects at 0x1800ff, sub-index 2.

Acyclic Synchronous

PDOs of transmission type 0 function synchronously, but not cyclically. An RxPDO is only evaluated after the next SYNC telegram has been received. In this way, for instance, axis groups can be given new target positions one after another, but these positions only become valid at the next SYNC - without the need to be constantly outputting reference points. A device whose TxPDO is configured for transmission type 0 acquires its input data when it receives the SYNC (synchronous process image) and then transmits it if the data correspond to an event (such as a change in input) having occurred. Transmission type 0 thus combines transmission for reasons that are event driven with a time for transmission (and, as far as possible, sampling) and processing given by the reception of "SYNC".

Cyclic Synchronous

In transmission types 1-240 the PDO is transmitted cyclically: after every ”nth" SYNC (n = 1...240). Since transmission types can be combined on a device as well as in the network, it is possible, for example, for a fast cycle to be agreed for digital inputs (n = 1), whereas the data for analog inputs is transmitted in a slower cycle (e.g. n = 10). RxPDOs do not generally distinguish between transmission types 0...240: a PDO that has been received is set to valid when the next SYNC is received. The cycle time (SYNC rate) can be monitored (object 0x1006), so that if the SYNC fails the device reacts in accordance with the definition in the device profile, and switches, for example, its outputs into the fault state.

The FC510x card provides full support for the synchronous type of communication: transmitting the SYNC telegram is coupled to the linked task, so that new input data is available every time the task begins. The card will recognize the absence of a synchronous PDO, and will report it to the application.

Only RTR

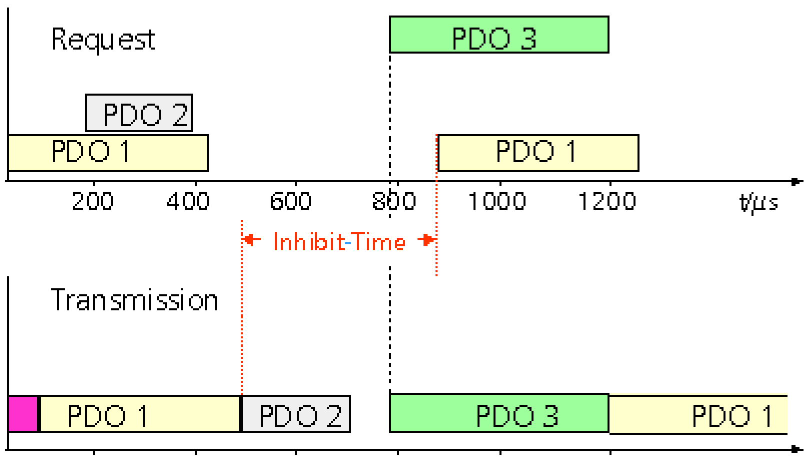
Transmission types 252 and 253 apply to process data objects that are transmitted exclusively on request by a remote frame. 252 is synchronous: when the SYNC is received the process data is acquired. It is only transmitted on request. 253 is asynchronous. The data here is acquired continuously, and transmitted on request. This type of transmission is not generally recommended, because fetching input data from some CAN controllers is only partially supported. Because, furthermore, the CAN controllers sometimes answer remote frames automatically (without first requesting up-to-date input data), there are circumstances in which it is questionable whether the polled data is up-to-date. Transmission types 252 and 253 are for this reason not supported by the BECKHOFF PC cards.

Asynchronous

The transmission types 254 + 255 are asynchronous, but may also be event-driven. In transmission type 254, the event is specific to the manufacturer, whereas for type 255 it is defined in the device profile. In the simplest case, the event is the change of an input value - this means that every change in the value is transmitted. The asynchronous transmission type can be coupled with the event timer, thus also providing input data when no event has just occurred.

Inhibit time

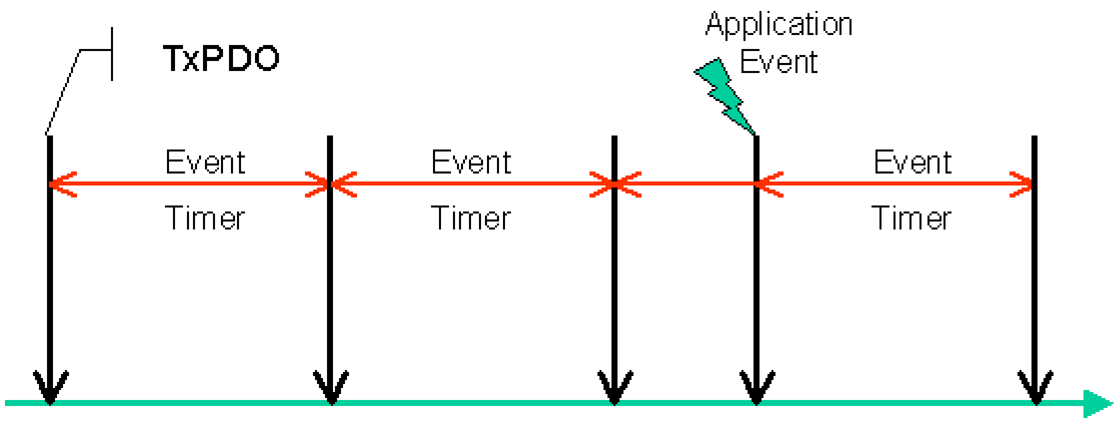
The "inhibit time" parameter can be used to implement a "transmit filter" that does not increase the reaction time for relatively new input alterations, but is active for changes that follow immediately afterwards. The inhibit time (transmit delay time) specifies the minimum length of time that must be allowed to elapse between the transmission of two of the same telegrams. If the inhibit time is used, the maximum bus loading can be determined, so that the worst case latency can then be found.



Although the BECKHOFF FC510x PC cards can parameterize the inhibit time on slave devices, they do not themselves support it. The transmitted PDOs become automatically spread out (transmit delay) as a result of the selected PLC cycle time - and there is little value in having the PLC run faster than the bus bandwidth permits. The bus loading, furthermore, can be significantly affected by the synchronous communication.

Event Timer

An event timer for transmit PDOs can be specified by sub-index 5 in the communication parameters. Expiry of this timer is treated as an additional event for the corresponding PDO, so that the PDO will then be transmitted. If the application event occurs during a timer period, it will also be transmitted, and the timer is reset.



In the case of receive PDOs, the timer is used to set a watchdog interval for the PDO: the application is informed if no corresponding PDO has been received within the set period. The FC510x can in this way monitor each individual PDO.

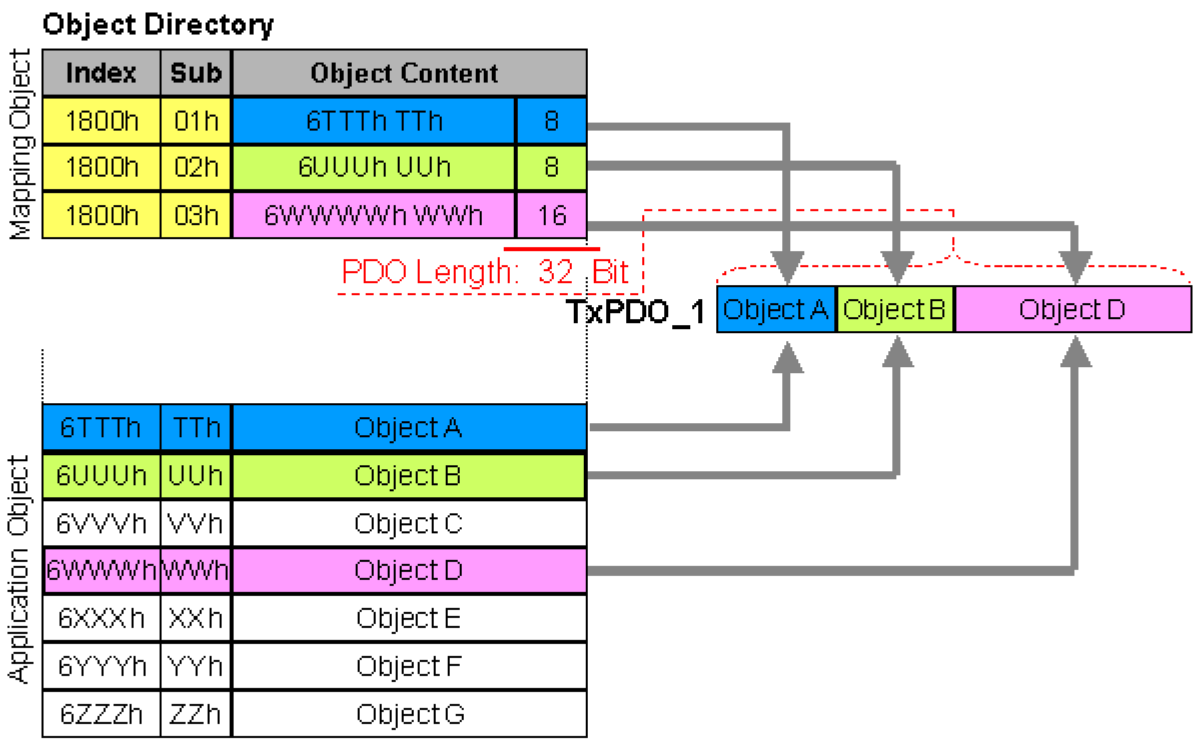
Notes on PDO Parameterization [▸ 77]

PDO Mapping

PDO mapping refers to mapping of the application objects (real time data) from the object directory to the process data objects. The CANopen device profile provide a default mapping for every device type, and this is appropriate for most applications. Thus the default mapping for digital I/O simply represents the inputs and outputs in their physical sequence in the transmit and receive process data objects.

The default PDOs for drives contain 2 bytes each of a control and status word and a set or actual value for the relevant axis.

The current mapping can be read by means of corresponding entries in the object directory. These are known as the mapping tables. The first location in the mapping table (sub-index 0) contains the number of mapped objects that are listed after it. The tables are located in the object directory at index 0x1600ff for the RxPDOs and at 0x1A00ff for the TxPDOs.



Digital and analog input/output modules: Read out the I/O number

The current number of digital and analog inputs and outputs can be determined or verified by reading out the corresponding application objects in the object directory:

Parameters	Object directory address
Number of digital input bytes	Index 0x6000, sub-index 0
Number of digital output bytes	Index 0x6200, sub-index 0
Number of analog inputs	Index 0x6401, sub-index 0
Number of analog outputs	Index 0x6411, sub-index 0

Variable mapping

As a rule, the default mapping of the process data objects already satisfies the requirements. For special types of application the mapping can nevertheless be altered: the Beckhoff CANopen Bus Couplers, for instance, thus support variable mapping, in which the application objects (input and output data) can be freely allocated to the PDOs. The mapping tables must be configured for this: as from Version 4 of CANopen, only the following procedure is permitted, and must be followed precisely:

1. First delete the PDO (set 0x1400ff, or 0x1800ff, sub-index 1, bit 31 to "1")
2. Set sub-index 0 in the mapping parameters (0x1600ff or 0x1A00ff) to "0"
3. Change mapping entries (0x1600ff or 0x1A00ff, SI 1..8)
4. Set sub-index 0 in the mapping parameters to the valid value. The device then checks the entries for consistency.
5. Create PDO by entering the identifier (0x1400ff or 0x1800ff, sub-index 1).

Dummy Mapping

A further feature of CANopen is the mapping of placeholders, or dummy entries. The data type entries stored in the object directory, which do not themselves have data, are used as placeholders. If such entries are contained in the mapping table, the corresponding data from the device is not evaluated. In this way, for instance, a number of drives can be supplied with new set values using a single CAN telegram, or outputs on a number of nodes can be set simultaneously, even in event-driven mode.

9.2.3 PDO Parameterization

Even though the majority of CANopen networks operate satisfactorily with the default settings, i.e. with the minimum of configuration effort, it is wise at least to check whether the existing bus loading is reasonable: 80% bus loading may be acceptable for a network operating purely in cyclic synchronous modes, but for a network with event-driven traffic this value would generally be too high, as there is hardly any bandwidth available for additional events.

Consider the Requirements of the Application

The communication of the process data must be optimized in the light of application requirements which are likely to be to some extent in conflict. These include

- Little work on parameterization - useable default values are optimal
- Guaranteed reaction time for specific events
- Cycle time for regulation processes over the bus
- Safety reserves for bus malfunctions (enough bandwidth for the repetition of messages)
- Maximum baud rate - depends on the maximum bus length
- Desired communication paths - who is speaking with whom

The determining factor often turns out to be the available bus bandwidth (bus load).

Determine the Baud Rate

We generally begin by choosing the highest baud rate that the bus will permit. It should be borne in mind that serial bus systems are fundamentally more sensitive to interference as the baud rate is increased. The following rule therefore applies: just as fast as necessary. 1000 kbit/s are not usually necessary, and only to be unreservedly recommended on networks within a control cabinet where there is no electrical isolation between the bus nodes. Experience also tends to show that estimates of the length of bus cable laid are often over-optimistic - the length actually laid tends to be longer.

Determine the Communication Type

Once the baud rate has been chosen it is appropriate to specify the PDO communication type(s). These have different advantages and disadvantages:

- Cyclic synchronous communication provides an accurately predictable bus loading, and therefore a defined time behavior - you could say that the standard case is the worst case. It is easy to configure: The SYNC rate parameter sets the bus loading globally. The process images are synchronized: Inputs are read at the same time, output data is set valid simultaneously, although the quality of the synchronization depends on the implementation. The Beckhoff FC510x PC cards are capable of synchronizing the CANopen bus system with the cycles of the application program (PLC or NC).
  The guaranteed reaction time under cyclic synchronous communication is always at least as long as the cycle time, and the bus bandwidth is not exploited optimally, since old data, i.e. data that has not changed, is continuously transmitted. It is however possible to optimize the network through the selection of different SYNC multiples (transmission types 1...240), so that data that changes slowly is transmitted less often than, for instance, time-critical inputs. It must, however, be borne in mind that input states that last for a time that is shorter than the cycle time will not necessarily be communicated. If it is necessary for such conditions to be registered, the associated PDOs for asynchronous communication should be provided.
- Event-driven asynchronous communication is optimal from the point of view of reaction time and the exploitation of bus bandwidth - it can be described as "pure CAN". Your choice must, however, also take account of the fact that it is not impossible for a large number of events to occur simultaneously, leading to corresponding delays before a PDO with a relatively low priority can be sent. Proper network

planning therefore necessitates a worst-case analysis. Through the use of, for instance, inhibit time [▸ 70], it is also necessary to prevent a constantly changing input with a high PDO priority from blocking the bus (technically known as a "babbling idiot"). It is for this reason that event driving is switched off by default in the device profile of analog inputs, and must be turned on specifically. Time windows for the transmit PDOs can be set using progress timers: the telegram is not sent again before the inhibit time [▸ 70] has elapsed, and not later than the time required for the progress timer to complete.

- The communication type is parameterized by means of the transmission type [▸ 70].

It is also possible to combine the two PDO principles. It can, for instance, be helpful to exchange the set and actual values of an axis controller synchronously, while limit switches, or motor temperatures with limit values are monitored with event-driven PDOs. This combines the advantages of the two principles: synchronicity for the axis communication and short reaction times for limit switches. In spite of being event-driven, the distributed limit value monitoring avoids a constant addition to the bus load from the analog temperature value.

In this example it can also be of value to deliberately manipulate the identifier allocation, in order to optimize bus access by means of priority allocation: the highest priority is given to the PDO with the limit switch data, and the lowest to that with the temperature values.

Optimization of bus access latency time through modification of the identifier allocation is not, however, normally required. On the other hand the identifiers must be altered if masterless communication is to be made possible (PDO linking [▸ 70]). In this example it would be possible for one RxPDO for each axis to be allocated the same identifier as the limit switch TxPDO, so that alterations of the input value can be received without delay.

Determining the Bus Loading

It is always worth determining the bus loading. But what bus loading values are permitted, or indeed sensible? It is first necessary to distinguish a short burst of telegrams in which a number of CAN messages follow one another immediately - a temporary 100% bus loading. This is only a problem if the sequence of receive interrupts that it caused at the CAN nodes can not be handled. This would constitute a data overflow (or CAN queue overrun). This can occur at very high baud rates (> 500 kbit/s) at nodes with software telegram filtering and relatively slow or heavily loaded microcontrollers if, for instance, a series of remote frames (which do not contain data bytes, and are therefore very short) follow each other closely on the bus (at 1 Mbit/s this can generate an interrupt every 40 µs; for example, an NMT master might transmit all its guarding requests in an unbroken sequence). This can be avoided through skilled implementation, and the user should be able to assume that the device suppliers have taken the necessary trouble. A burst condition is entirely normal immediately after the SYNC telegram, for instance: triggered by the SYNC, all the nodes that are operating synchronously try to send their data at almost the same time. A large number of arbitration processes take place, and the telegrams are sorted in order of priority for transmission on the bus. This is not usually critical, since these telegrams do contain some data bytes, and the telegrams trigger a sequence of receive interrupts at the CAN nodes which is indeed rapid, but is nevertheless manageable.

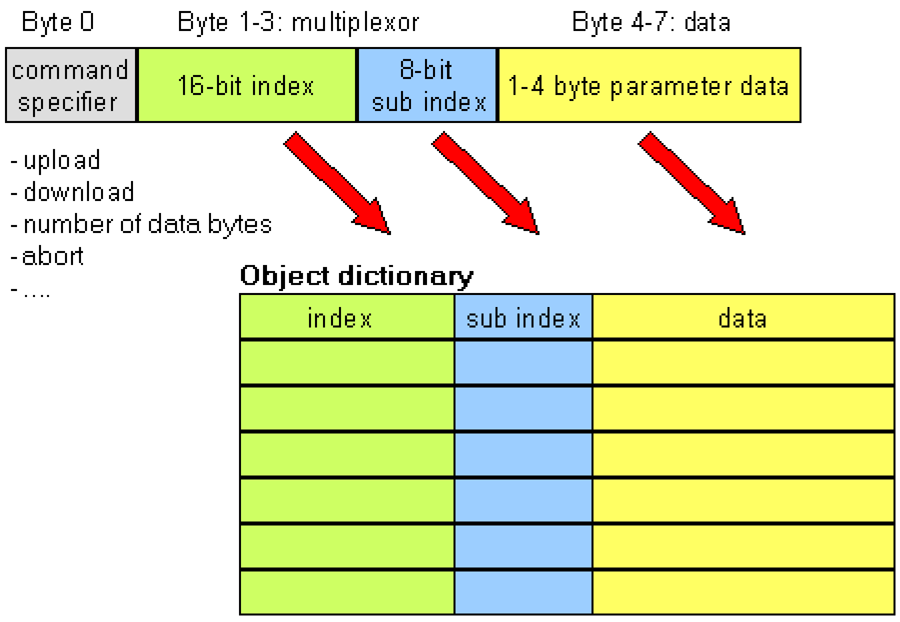
Bus loading most often refers to the value averaged over several primary cycles, that is the mean value over 100-500 ms. CAN, and therefore CANopen, is indeed capable of managing a bus loading of close to 100% over long periods, but this implies that no bandwidth is available for any repetitions that may be necessitated by interference, for asynchronous error messages, parameterization and so on. Clearly, the dominant type of communication will have a large influence on the appropriate level of bus loading: a network with entirely cyclic synchronous operation is always in any case near to the worst case state, and can therefore be operated with values in the 70-80% range. The figure is very hard to state for an entirely event-driven network: an estimate must be made of how many events additional to the current state of the system might occur, and of how long the resulting burst might last - in other words, for how long the lowest priority message will be delayed. If this value is acceptable to the application, then the current bus loading is acceptable. As a rule of thumb it can usually be assumed that an event-driven network running with a base loading of 30-40% has enough reserve for worst-case scenarios, but this assumption does not obviate the need for a careful analysis if delays could have critical results for the plant.

The BECKHOFF FC510x PC cards indicate the bus loading via the System Manager. This variable can also be processed in the PLC, or can be displayed in the visualization system.

The amount data in the process data objects is of course as relevant as the communication parameters: the PDO mapping [▸ 70].

9.2.4 Service Data Objects (SDO)

The parameters listed in the object directory are read and written by means of service data objects. These SDOs are *Multiplexed Domains*, i.e. data structures of any size that have a multiplexer (address). The multiplexer consists of a 16-bit index and an 8-bit sub-index that address the corresponding entries in the object directory.



SDO protocol: access to the object directory

The CANopen Bus Couplers are servers for the SDO, which means that at the request of a client (e.g. of the IPC or the PLC) they make data available (upload), or they receive data from the client (download). This involves a handshake between the client and the server.

When the size of the parameter to be transferred is not more than 4 bytes, a single handshake is sufficient (one telegram pair): For a download, the client sends the data together with its index and sub-index, and the server confirms reception. For an upload, the client requests the data by transmitting the index and sub-index of the desired parameter, and the server sends the parameter (including index and sub-index) in its answer telegram.

The same pair of identifiers is used for both upload and download. The telegrams, which are always 8 bytes long, encode the various services in the first data byte. All parameters with the exception of objects 1008h, 1009h and 100Ah (device name, hardware and software versions) are only at most 4 bytes long, so this description is restricted to transmission in expedited transfer.

Protocol

The structure of the SDO telegrams is described below.

Client -> Server, Upload Request

11 bit identifier	8 bytes of user data							
0x600 (=1536dez) + node ID	0x40	Index0	Index1	SubIdx	0x00	0x00	0x00	0x00

Parameters	Explanation
Index0	Index low byte (Unsigned16, LSB)
Index1	Index high byte (Unsigned16, MSB)
SubIdx	Sub-index (Unsigned8)

Client -> Server, Upload Response

11 bit identifier	8 bytes of user data							
0x580 (=1408dec) + node ID	0x4x	Index0	Index1	SubIdx	Data0	Data1	Data2	Data3

Parameters	Explanation
Index0	Index low byte (Unsigned16, LSB)
Index1	Index high byte (Unsigned16, MSB)
SubIdx	Sub-index (Unsigned8)
Data0	Data low low byte (LLSB)
Data3	Data high high byte (MMSB)

Parameters whose data type is Unsigned8 are transmitted in byte D0, parameters whose type is Unsigned16 use D0 and D1.

The number of valid data bytes is coded as follows in the first CAN data byte (0x4x):

Number of parameter bytes	1	2	3	4
First CAN data byte	0x4F	0x4B	0x47	0x43

Client -> Server, Download Request

11 bit identifier	8 bytes of user data							
0x600 (=1536dec) + node ID	0x22	Index0	Index1	SubIdx	Data0	Data1	Data2	Data3

Parameters	Explanation
Index0	Index low byte (Unsigned16, LSB)
Index1	Index high byte (Unsigned16, MSB)
SubIdx	Sub-index (Unsigned8)
Data0	Data low low byte (LLSB)
Data3	Data high high byte (MMSB)

It is optionally possible to give the number of valid parameter data bytes in the first CAN data byte

Number of parameter bytes	1	2	3	4
First CAN data byte	0x2F	0x2B	0x27	0x23

This is, however, not generally necessary, since only the less significant data bytes up to the length of the object directory entry that is to be written are evaluated. A download of data up to 4 bytes in length can therefore always be achieved in Beckhoff bus nodes with 22h in the first CAN data byte.

Client -> Server, Download Response

11 bit identifier	8 bytes of user data							
0x580 (=1408dec) + node ID	0x60	Index0	Index1	SubIdx	0x00	0x00	0x00	0x00

Parameters	Explanation
Index0	Index low byte (Unsigned16, LSB)
Index1	Index high byte (Unsigned16, MSB)
SubIdx	Sub-index (Unsigned8)

Breakdown of Parameter Communication

Parameter communication is interrupted if it is faulty. The client or server send an SDO telegram with the following structure for this purpose:

11 bit identifier	8 bytes of user data							
0x580 (client) or 0x600(server) + node ID	0x80	Index0	Index1	SubIdx	Error0	Error1	Error2	Error3

Parameters	Explanation
Index0	Index low byte (Unsigned16, LSB)
Index1	Index high byte (Unsigned16, MSB)
SubIdx	Sub-index (Unsigned8)
Error0	SDO error code low low byte (LLSB)
Error3	SDO error code high high byte (MMSB)

List of SDO error codes (reason for abortion of the SDO transfer):

SDO error code	Explanation
0x05 03 00 00	Toggle bit not changed
0x05 04 00 01	SDO command specifier invalid or unknown
0x06 01 00 00	Access to this object is not supported
0x06 01 00 02	Attempt to write to a Read_Only parameter
0x06 02 00 00	The object is not found in the object directory
0x06 04 00 41	The object can not be mapped into the PDO
0x06 04 00 42	The number and/or length of mapped objects would exceed the PDO length
0x06 04 00 43	General parameter incompatibility
0x06 04 00 47	General internal error in device
0x06 06 00 00	Access interrupted due to hardware error
0x06 07 00 10	Data type or parameter length do not agree or are unknown
0x06 07 00 12	Data type does not agree, parameter length too great
0x06 07 00 13	Data type does not agree, parameter length too short
0x06 09 00 11	Sub-index not present
0x06 09 00 30	General value range error
0x06 09 00 31	Value range error: parameter value too great
0x06 09 00 32	Value range error: parameter value too small
0x06 0A 00 23	Resource not available
0x08 00 00 21	Access not possible due to local application
0x08 00 00 22	Access not possible due to current device status

Further, manufacturer-specific error codes have been introduced for register communication (index 0x4500, 0x4501):

SDO error code	Explanation
0x06 02 00 11	Invalid table: Table or channel not present
0x06 02 00 10	Invalid register: table not present
0x06 01 00 22	Write protection still set
0x06 07 00 43	Incorrect number of function arguments
0x06 01 00 21	Function still active, try again later
0x05 04 00 40	General routing error
0x06 06 00 21	Error accessing BC table
0x06 09 00 10	General error communicating with terminal
0x05 04 00 47	Time-out communicating with terminal

9.3 Objekt dictionary

9.3.1 Object Directory - Structure

All the CANopen objects relevant for the Bus Coupler are entered into the CANopen object directory. The object directory is divided into three different regions:

1. communication-specific profile region (index 0x1000 – 0x1FFF).
   This contains the description of all the parameters specific to communication.
2. manufacturer-specific profile region (index 0x2000 – 0x5FFF).
   Contains the description of the manufacturer-specific entries.
3. standardized device profile region (0x6000 – 0x9FFF).
   Contains the objects for a device profile according to DS-401.

Every entry in the object directory is identified by a 16 bit index. If an object consists of several components (e.g. object type array or record), the components are identified by an 8-bit sub-index. The object name describes the function of an object, while the data type attribute specifies the data type of the entry. The access attribute specifies whether an entry may only be read, only written, or may be both read and written.

Communication-specific region

All the parameters and objects necessary for the CANopen Bus Coupler's communication are in this region of the object directory. The region from 0x1000 to 0x1018 contains various general communication-specific parameters (e.g. the device name).

The communication parameters (e.g. identifiers) for the receive PDOs are located in the region from 0x1400 to 0x140F (plus sub-index). The mapping parameters of the receive PDOs are in the region from 0x1600 to 0x160F (plus sub-index). The mapping parameters contain the cross-references to the application objects that are mapped into the PDOs and the data width of the corresponding object (see also the section dealing with PDO Mapping).

The communication and mapping parameters for the transmit PDOs are located in the regions from 0x1800 to 0x180F and from 0x1A00 to 0x1A0F.

Manufacturer-specific region

This region contains entries that are specific to BECKHOFF, e.g.:

- data objects for special terminals
- objects for register communication providing access to all the Bus Couplers' and Bus Terminals' internal registers
- objects for simplified configuration of the PDOs

Standardized device profile region

The standardized device profile region supports the device profile of CANopen DS-401, Version 1. Functions are available for analog inputs that can adapt communication in the event-driven operating mode to the requirements of the application and to minimize the loading of the bus:

- limit value monitoring
- Delta function
- activation/deactivation of event-driven mode

9.3.2 Object List

The objects in the object directory can be reached by SDO access, but not generally through the KS2000 configuration software. On the other hand, all the registers that can be configured with KS2000 can also be reached using SDO access to the object directory (objects 0x4500 and 0x4501) - even though this does not offer the same convenience as the KS2000 software.

Parameter	Index	BK5120/ BK515x	BK5110	LC5100	BX5100/ BC5150	CX705x/ CX8051/B510
Device type [▶ 86]	0x1000	x	x	x		x
Error register [▶ 86]	0x1001	x	x	x	x	x *
Error memory [▶ 86]	0x1003	x	x	x		
Sync Identifier [▶ 86]	0x1005	x	x	x	x	x
Sync Interval [▶ 86]	0x1006	x	x	x	x	x
Device name [▶ 86]	0x1008	x	x	x	x	x *
Hardware version [▶ 86]	0x1009	x	x	x		
Software version [▶ 86]	0x100A	x	x	x	x	x
Node number [▶ 86]	0x100B	x	x	x		
Guard Time [▶ 86]	0x100C	x	x	x	x	x
Life Time Factor [▶ 86]	0x100D	x	x	x	x	x
Guarding Identifier [▶ 86]	0x100E	x	x	x		
Save parameters [▶ 86]	0x1010	x	x	x		
Load default values [▶ 86]	0x1011	x	x	x		
Emergency Identifier [▶ 86]	0x1014	x	x	x		
Consumer Heartbeat Time [▶ 86]	0x1016	x	x	x	x	x
Producer Heartbeat Time [▶ 86]	0x1017	x	x	x	x	x
Device identifier (identity object) [▶ 86]	0x1018	x	x	x	x	x *
Server SDO parameters [▶ 86]	0x1200	x	x	x		
Communication parameters for the 1st - 5th RxPDO [▶ 86]	0x1400 - 0x1404	x	x	x	x	x
Communication parameters for the 6th - 16th RxPDO [▶ 86]	0x1405 - 0x140F	x			x	x
Communication parameters for the 17th - 32nd RxPDO [▶ 86]	0x1410 - 0x141F				x only BX5100	x
Mapping 1st -5th RxPDO [▶ 86]	0x1600 - 0x1604	x	x	x	x	x
Mapping 6th -16th RxPDO [▶ 86]	0x1605 - 0x160F	x			x	x
Mapping 17th -32nd RxPDO [▶ 86]	0x1610 - 0x161F				x only BX5100	x
Communication parameters for the 1st - 5th TxPDO [▶ 86]	0x1800 - 0x1804	x	x	x	x	x
Communication parameters for the 6th - 16th TxPDO [▶ 86]	0x1805 - 0x180F	x			x	x
Communication parameters for the 17th - 32nd TxPDO [▶ 86]	0x1810 - 0x181F				x only BX5100	x
Mapping 1st -5th TxPDO [▶ 86]	0x1A00 - 0x1A04	x	x	x	x	x
Mapping 6th -16th TxPDO [▶ 86]	0x1A05 - 0x1A0F	x			x	x
Mapping 17th -32nd TxPDO [▶ 86]	0x1A10 - 0x1A1F				x only BX5100	x
Flag area %MB0-511	0x2F00				x	
Flag area %MB511-1023	0x2F01				x	
Flag area %MB1024-1535	0x2F02				x	
Flag area %MB1536-2047	0x2F03				x	
Flag area %MB2048-2559	0x2F04				x	

Parameter	Index	BK5120/ BK515x	BK5110	LC5100	BX5100/ BC5150	CX705x/ CX8051/B510
Flag area %MB2560-3071	0x2F05				x	
Flag area %MB3072-3584	0x2F06				x	
Flag area %MB3585-4095	0x2F07				x	
3-byte special terminals, input data [▸ 86]	0x2600	x				
3-byte special terminals, output data [▸ 86]	0x2700	x				
4-byte special terminals, input data [▸ 86]	0x2800	x				
4-byte special terminals, output data [▸ 86]	0x2900	x				
5-byte special terminals, input data [▸ 86]	0x2A00	x				
5-byte special terminals, output data [▸ 86]	0x2B00	x				
6-byte special terminals, input data [▸ 86]	0x2C00	x				
6-byte special terminals, output data [▸ 86]	0x2D00	x				
8-byte special terminals, input data [▸ 86]	0x3000	x				
8-byte special terminals, output data [▸ 86]	0x3100	x				
Register communication, bus node [▸ 86]	0x4500	x	x	x		
Register communication, bus terminal/extension box [▸ 86]	0x4501	x	x	x		
Enable PDOs [▸ 86]	0x5500	x	x	x		
NetId	0x5FFE				x	
Digital inputs [▸ 86]	0x6000	x	x	x		
Interrupt mask [▸ 86]	0x6126	x	x	x		
Digital outputs [▸ 86]	0x6200	x	x	x		
Analog inputs [▸ 86]	0x6401	x				
Analog outputs [▸ 86]	0x6411	x				
Event control, analog inputs [▸ 86]	0x6423	x				
Upper limit value, analog inputs [▸ 86]	0x6424	x				
Lower limit value, analog inputs [▸ 86]	0x6425	x				
Delta function, analog inputs [▸ 86]	0x6426	x				

* When an ADS server is registered, these objects are relayed to the PLC via ADS notification and have to be answered there.

9.3.3 Objects and Data

Device type

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x1000	0	Device type	Unsigned32	ro	N	0x00000000	Statement of device type

The 32 bit value is divided into two 16 bit fields:

MSB	LSB
Additional information	Device profile number
0000 0000 0000 wxyz	0x191 (401$_{dez}$)

The *additional information* contains data related to the signal type of the I/O device:
z=1 signifies digital inputs,
y=1 signifies digital outputs,
x=1 signifies analog inputs,
w=1 signifies analog outputs.
A BK5120 with digital and analog inputs, but with no outputs, thus returns 0x00 05 01 91.

Special terminals (such as serial interfaces, PWM outputs, incremental encoder inputs) are not considered. A Coupler that, for example, only has KL6001 serial interface terminals plugged in, thus returns 0x00 00 01 91.

The device type supplies only a rough classification of the device. The terminal identifier register of the Bus Coupler can be read for detailed identification of the Bus Couplers and the attached terminals (for details see register communication index 0x4500).

Error register

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x1001	0	Error register	Unsigned8	ro	N	0x00	Error register

The 8 bit value is coded as follows:

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

ManSpec. Manufacturer-specific error, specified more precisely in object 1003.

Comm. Communication error (CAN overrun)

Generic An error that is not more precisely specified has occurred (the flag is set at every error message)

Error store

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x1003	0x00	Predefined error field (Error store)	Unsigned8	rw	N	0x00	Object 1003h contains a description of the error that has occurred in the device - sub-index 0 has the number of error states stored.
	1	Actual error	Unsigned32	ro	N	None	Last error state to have occurred
	...	...	...	--	...	...	...
	10	Standard error field	Unsigned32	ro	N	None	A maximum of 10 error states are stored.

The 32 bit value in the error store is divided into two 16 bit fields:

MSB	LSB
Additional code	Error Code

The additional code contains the error trigger (see emergency object) and thereby a detailed error description.

New errors are always saved at sub-index 1, all the other sub-indices being appropriately incremented. The whole error store is cleared by writing a 0 to sub-index 0.

If there has not been an error since power up, then object 0x1003 only consists of sub-index 0 with a 0 entered into it. The error store is cleared by a reset or a power cycle.

As is usual in CANopen, the LSB is transferred first, followed by the MSB.

Sync Identifier

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x1005	0	COB-ID Sync Message	Unsigned32	rw	N	0x80000080	Identifier of the SYNC message

The bottom 11 bits of the 32 bit value contain the identifier (0x80=128 dec). Bit 30 indicates whether the device sends the SYNC telegram (1) or not (0). The CANopen I/O devices receive the SYNC telegram, and accordingly bit 30=0. For reasons of backwards compatibility, bit 31 has no significance.

Sync Interval

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x1006	0	Communication cycle period	Unsigned32	rw	N	0x00000000	Length of the SYNC interval in µs.

If a value other than zero is entered here, the bus node will go into the fault state if, during synchronous PDO operation, no SYNC telegram is received within the watchdog time. The watchdog time corresponds here to 1.5 times the communication cycle period that has been set - the planned SYNC interval can therefore be entered.

The I/O update is carried out at the Beckhoff CANopen bus nodes immediately after reception of the SYNC telegram, provided the following conditions are satisfied:

- Firmware status C0 or above (CANopen Version 4.01 or higher).

- All PDOs that have data are set to synchronous communication (0..240).

- The sync interval has been entered in object 0x1006 and (sync interval x lowest PDO transmission type) is less than 90ms.

The modules are then synchronised throughout.

Device name

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x1008	0	Manufacturer Device Name	Visible String	ro	N	BK51x0, LC5100, IPxxxx-B510 or ILxxxx-B510	Device name of the bus node

Since the returned value is longer than 4 bytes, the segmented SDO protocol is used for transmission.

Hardware version

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x1009	0	Manufacturer hardware-version	Visible String	ro	N	-	Hardware version number of the bus node

Since the returned value is longer than 4 bytes, the segmented SDO protocol is used for transmission.

Software version

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x100A	0	Manufacturer software-version	Visible String	ro	N	-	Software version number of the bus node

Since the returned value is longer than 4 bytes, the segmented SDO protocol is used for transmission.

Node number

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x100B	0	Node-ID	Unsigned32	ro	N	none	Set node number

The node number is supported for reasons of compatibility.

Guard time

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x100C	0	Guard time [ms]	Unsigned16	rw	N	0	Interval between two guard telegrams. Is set by the NMT master or configuratio n tool.

Life time factor

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x100D	0	Life time factor	Unsigned8	rw	N	0	Life time factor x guard time = life time (watchdog for life guarding)

If a guarding telegram is not received within the life time, the node enters the error state. If the life time factor and/or guard time = 0, the node does not carry out any life guarding, but can itself be monitored by the master (node guarding).

Guarding identifier

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x100 E	0	COB-ID guarding protocol	Unsigned32	ro	N	0x000007xy, xy = NodeID	Identifier of the guarding protocol

The guarding identifier is supported for reasons of compatibility. Changing the guarding identifier has no longer been permitted since version 4 of CANopen.

Save parameters

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x1010	0	Store Parameter	Unsigned8	ro	N	1	Number of store options
	1	store all parameters	Unsigned32	rw	N	1	Stores all (storable) parameters

By writing the string *save* in ASCII code (hexadecimal 0x65766173) to sub-index 1, the current parameters are placed into non-volatile storage. (The byte sequence on the bus including the SDO protocol: 0x23 0x10 0x10 0x01 0x73 0x61 0x76 0x65).

The storage process takes about 3 seconds, and is confirmed, if successful, by the corresponding TxSDO (0x60 in the first byte). Since the Bus Coupler is unable to send or receive any CAN telegrams during the storage process, saving is only possible when the node is in the pre-operational state. It is recommended that the entire network is placed into the pre-operational state before such storage. This avoids a buffer overflow.

Data saved includes:

- The terminals currently inserted (the number of each terminal category)
- All PDO parameters (identifier, transmission type, inhibit time, mapping).

[Gefahrinformation hier einfügen!]

NoteThe stored identifiers apply afterwards, not the default identifiers derived from the node addresses. Changes to the DIP switch setting no longer affects the PDOs!

- All SYNC parameters
- All guarding parameters
- Limit values, delta values and interrupt enables for analog inputs

Parameters directly stored in the terminals by way of register communication are immediately stored there in non-volatile form.

Load default values

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x1011	0	Restore Parameter	Unsigned8	ro	N	4	Number of reset options
	1	Restore all parameters	Unsigned32	rw	N	1	Resets all parameters to their default values
	4	Set manufacturer Defaults	Unsigned32	rw	N	1	Resets all coupler parameters to manufacturer's settings (including registers)

Writing the string *load* in ASCII code (hexadecimal 0x64616F6C) into sub-index 1 resets all parameters to default values (as initially supplied) **at the next boot (reset)**.

(The byte sequence on the bus including the SDO protocol: 0x23 0x11 0x10 0x01 0x6C 0x6F 0x61 0x64).

This makes the default identifiers for the PDOs active again.

Emergency identifier

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x1014	0	COB-ID Emergency	Unsigned32	rw	N	0x00000080, + NodeID	Identifier of the emergency telegram

The bottom 11 bits of the 32 bit value contain the identifier (0x80=128 dec). The MSBit can be used to set whether the device sends (1) the emergency telegram or not (0).

Alternatively, the bus node's diagnostic function can also be switched off using the *Device diagnostics* bit in the K-Bus configuration (see object 0x4500).

Consumer heartbeat time

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x1016	0	Number of elements	Unsigned8	ro	N	2	The consumer heartbeat time describes the expected heartbeat cycle time and the node ID of the monitored node
	1	Consumer heartbeat time	Unsigned32	rw	N	0	Watchdog time in ms and node ID of the monitored node

The 32-bit value is used as follows:

MSB		**LSB**
Bit 31...24	Bit 23...16	Bit 15...0
Reserved (0)	Node ID (unsigned8)	Heartbeat time in ms (unsigned16)

The monitored identifier can be obtained from the node ID by means of the default identifier allocation: Guard-ID = 0x700 + Node-ID.

As is usual in CANopen, the LSB is transferred first, followed by the MSB.

Producer heartbeat time

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x1017	0	Producer heartbeat time	Unsigned16	rw	N	0	Interval in ms between two transmitted heartbeat telegrams

Device identifier (identity object)

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x1018	0	Identity Object: Number of elements	Unsigned8	ro	N	4	The identity object contains general information about the type and version of the device.
	1	Vendor ID	Unsigned32	ro	N	0x00000002	Manufacturer identifier. Beckhoff has vendor ID 2
	2	Product Code	Unsigned32	ro	N	Depends on the product	Device identifier
	3	Revision Number	Unsigned32	ro	N	-	Version number
	4	Serial Number	Unsigned32	ro	N	-	Production date low word, high byte: calendar week (dec), low word, low byte: calendar year

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

Server SDO parameters

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x1200	0	Number of elements	Unsigned8	ro	N	2	Communication parameters of the server SDO. Sub-index 0: number of following parameters
	1	COB-ID Client ->Server	Unsigned32	ro	N	0x000006xy, xy=Node-ID	COB-ID RxSDO (Client -> Server)
	2	COB-ID Server ->Client	Unsigned32	ro	N	0x00000580 + Node-ID	COB-ID TxSDO (Client -> Server)

This is contained in the object directory for reasons of backwards compatibility.

Communication parameters for the 1st RxPDO

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x1400	0	Number of elements	Unsigned8	ro	N	5	Communication parameters for the first receive PDO. Sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0x000002xy, xy=Node-ID	COB-ID (Communication Object Identifier) RxPDO1
	2	Transmission Type	Unsigned8	rw	N	255	Transmission type of the PDO
	3	Inhibit Time	Unsigned16	rw	N	0	Present for reasons of backwards compatibility, but not used in the RxPDO.
	4	CMS Priority Group	Unsigned8	rw	N	-	Present for reasons of backwards compatibility, but not used.
	5	Event Timer	Unsigned16	rw	N	0	Event-Timer. Watchdog time defined for monitoring reception of the PDO.

Sub-index 1 (COB-ID): The bottom 11 bits of the 32 bit value (bits 0-10) contain the CAN identifier. The MSB (bit 31) indicates whether the PDO exists currently (0) or not (1). Bit 30 indicates whether an RTR access to this PDO is permissible (0) or not (1). Changing the identifier (bits 0-10) is not allowed while the object exists (bit 31=0). Sub-index 2 contains the type of the transmission (see introduction to PDOs).

Communication parameters for the 2nd RxPDO

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x1401	0	Number of elements	Unsigned8	ro	N	5	Communication parameter for the second receive PDO.
	1	COB-ID	Unsigned32	rw	N	0x000003xy, xy=Node-ID	COB-ID (Communication Object Identifier) RxPDO2
	2	Transmission Type	Unsigned8	rw	N	255	Transmission type of the PDO
	3	Inhibit Time	Unsigned16	rw	N	0	Present for reasons of backwards compatibility, but not used in the RxPDO.
	4	CMS Priority Group	Unsigned8	rw	N	-	Present for reasons of backwards compatibility, but not used.
	5	Event Timer	Unsigned16	rw	N	0	Event-Timer. Watchdog time defined for monitoring reception of the PDO.

Communication parameters for the 3rd RxPDO

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x1402	0	Number of elements	Unsigned8	ro	N	5	Communication parameter for the third receive PDO.
	1	COB-ID	Unsigned32	rw	N	0x000004xy, xy=Node-ID	COB-ID (Communication Object Identifier) RxPDO3
	2	Transmission Type	Unsigned8	rw	N	255	Transmission type of the PDO
	3	Inhibit Time	Unsigned16	rw	N	0	Present for reasons of backwards compatibility, but not used in the RxPDO.
	4	CMS Priority Group	Unsigned8	rw	N	-	Present for reasons of backwards compatibility, but not used.
	5	Event Timer	Unsigned16	rw	N	0	Event-Timer. Watchdog time defined for monitoring reception of the PDO.

Communication parameters for the 4th RxPDO

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x1403	0	Number of elements	Unsigned8	ro	N	5	Communication parameters for the fourth receive PDO.
	1	COB-ID	Unsigned32	rw	N	0x000005xy, xy=Node-ID	COB-ID (Communication Object Identifier) RxPDO4
	2	Transmission Type	Unsigned8	rw	N	255	Transmission type of the PDO
	3	Inhibit Time	Unsigned16	rw	N	0	Present for reasons of backwards compatibility, but not used in the RxPDO.
	4	CMS Priority Group	Unsigned8	rw	N	-	Present for reasons of backwards compatibility, but not used.
	5	Event Timer	Unsigned16	rw	N	0	Event-Timer. Watchdog time defined for monitoring reception of the PDO.

Communication parameters for the 5th-16th RxPDOs

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x1404 - 0x140F (depending on the device type)	0	Number of elements	Unsigned8	ro	N	5	Communication parameter for the 5th to 16th receive PDOs.
	1dth="5%"> 1	COB-ID	Unsigned32	rw	N	0x8000000	COB-ID (Communication Object Identifier) RxPDO5...16
	2	Transmission Type	Unsigned8	rw	N	255	Transmission type of the PDO
	3	Inhibit Time	Unsigned16	rw	N	0	Present for reasons of backwards compatibility, but not used in the RxPDO.
	4	CMS Priority Group	Unsigned8	rw	N	-	Present for reasons of backwards compatibility, but not used.
	5	Event Timer	Unsigned16	rw	N	0	Event-Timer. Watchdog time defined for monitoring reception of the PDO.

The number of RxPDOs for each bus node type can be found in the technical data.

Mapping parameters for the 1st RxPDO

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x1600	0	Number of elements	Unsigned8	rw	N	Depending on type and fittings	Mapping parameter of the first receive PDO; sub-index 0: number of mapped objects.
	1	1st mapped object	Unsigned32	rw	N	0x62000108	1st mapped application object (2 bytes index, 1 byte sub-index, 1 byte bit width)
	2	2nd mapped object	Unsigned32	rw	N	0x62000208	2nd mapped application object (2 bytes index, 1 byte sub-index, 1 byte bit width)
	...	...	...	...	...	...	...
	8	8th mapped object	Unsigned32	rw	N	0x62000808	8th mapped application object (2 bytes index, 1 byte sub-index, 1 byte bit width)

The first receive PDO (RxPDO1) is provided by default for digital output data. Depending on the number of outputs inserted, the necessary length of the PDO is automatically determined, and the corresponding objects are mapped. Since the digital outputs are organised in bytes, the length of the PDO in bytes can be found directly at sub-index 0.

Changes to the mapping

The following sequence must be observed in order to change the mapping (specified as from CANopen, version 4):

1. Delete PDO (set bit 31 in the identifier entry (sub-index 1) of the communication parameters to 1)
2. Deactivate mapping (set sub-index 0 of the mapping entry to 0)
3. Change mapping entries (sub-indices 1...8)
4. Activate mapping (set sub-index 0 of the mapping entry to the correct number of mapped objects)
5. Create PDO (set bit 31 in the identifier entry (sub-index 1) of the communication parameters to 0)

Mapping parameters for the 2nd RxPDO

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x1601	0	Number of elements	Unsigned8	rw	N	Depending on type and fittings	Mapping parameter of the second receive PDO; sub-index 0: number of mapped objects.
	1	1st mapped object	Unsigned32	rw	N	0x64110110	1st mapped application object (2 bytes index, 1 byte sub-index, 1 byte bit width)
	2	2nd mapped object	Unsigned32	rw	N	0x64110210	2nd mapped application object (2 bytes index, 1 byte sub-index, 1 byte bit width)
	...	...	...	...	...	...	...
	8	8th mapped object	Unsigned32	rw	N	0x00000000	8th mapped application object (2 bytes index, 1 byte sub-index, 1 byte bit width)

The second receive PDO (RxPDO2) is provided by default for analog outputs. Depending on the number of outputs inserted, the necessary length of the PDO is automatically determined, and the corresponding objects are mapped. Since the analog outputs are organised in words, the length of the PDO in bytes can be found directly at sub-index 0.

A specific sequence must be observed in order to change the mapping (see object index 0x1600).

Mapping parameters for the 3rd-16th RxPDO

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x1602-0x160F (depending on the device type)	0	Number of elements	Unsigned8	rw	N	Depending on type and fittings	Mapping parameters for the third to sixteenth receive PDOs; sub-index 0: number of mapped objects.
	1	1st mapped object	Unsigned32	rw	N	0x00000000 (see text)	1st mapped application object (2 bytes index, 1 byte sub-index, 1 byte bit width)
	2	2nd mapped object	Unsigned32	rw	N	0x00000000 (see text)	2nd mapped application object (2 bytes index, 1 byte sub-index, 1 byte bit width)
	...	...	...	...	...	...	...
	8	8th mapped object	Unsigned32	rw	N	0x00000000 (see text)	8th mapped application object (2 bytes index, 1 byte sub-index, 1 byte bit width)

The 3rd to 16th receive PDOs (RxPDO3ff) are automatically given a default mapping by the bus node depending on the attached terminals (or depending on the extension modules). The procedure is described in the section on PDO Mapping.

A specific sequence must be observed in order to change the mapping (see object index 0x1600).

[Gefahrinformation hier einfügen!]

NoteDS401 V2 specifies analog input and/or output data as the default mapping for PDOs 3+4. This corresponds to Beckhoff's default mapping when less than 65 digital inputs or outputs are present. In order to ensure backwards compatibility, the Beckhoff default mapping is retained - the mapping behaviour of the devices therefore corresponds to DS401 V1, where in all other respects they accord with DS401 V2.

Communication parameters for the 1st TxPDO

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x1800	0	Number of elements	Unsigned8	ro	N	5	Communication parameters for the first transmit PDO. Sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0x00000180 + Node-ID	COB-ID (Communication Object Identifier) TxPDO1
	2	Transmission Type	Unsigned8	rw	N	255	Transmission type of the PDO
	3	Inhibit Time	Unsigned16	rw	N	0	Repetition delay [value x 100 µs]
	4	CMS Priority Group	Unsigned8	rw	N	-	Present for reasons of backwards compatibility, but not used.
	5	Event Timer	Unsigned16	rw	N	0	Event-Timer

Sub-index 1 (COB-ID): The bottom 11 bits of the 32 bit value (bits 0-10) contain the CAN identifier. The MSB (bit 31) indicates whether the PDO exists currently (0) or not (1). Bit 30 indicates whether an RTR access to this PDO is permissible (0) or not (1). Changing the identifier (bits 0-10) is not allowed while the object exists (bit 31=0). Sub-index 2 contains the type of transmission, sub-index 3 the repetition delay between two PDOs of the same type, while sub-index 5 contains the event timer. Sub-index 4 is retained for reasons of compatibility, but is not used. (See also the introduction to PDOs.)

Communication parameters for the 2nd TxPDO

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x1801	0	Number of elements	Unsigned8	ro	N	5	Communication parameters for the second transmit PDO. Sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0x00000280 + Node-ID	COB-ID (Communication Object Identifier) TxPDO1
	2	Transmission Type	Unsigned8	rw	N	255	Transmission type of the PDO
	3	Inhibit Time	Unsigned16	rw	N	0	Repetition delay [value x 100 µs]
	4	CMS Priority Group	Unsigned8	rw	N	-	Present for reasons of backwards compatibility, but not used.
	5	Event Timer	Unsigned16	rw	N	0	Event-Timer

The second transmit PDO is provided by default for analog inputs, and is configured for event-driven transmission (transmission type 255). Event-driven mode must first be activated (see object 0x6423), otherwise the inputs can only be interrogated (polled) by remote transmission request (RTR).

Communication parameters for the 3rd TxPDO

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x1802	0	Number of elements	Unsigned8	ro	N	5	Communication parameters for the third transmit PDO. Sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0x00000380 + Node-ID	COB-ID (Communication Object Identifier) TxPDO1
	2	Transmission Type	Unsigned8	rw	N	255	Transmission type of the PDO
	3	Inhibit Time	Unsigned16	rw	N	0	Repetition delay [value x 100 µs]
	4	CMS Priority Group	Unsigned8	rw	N	-	Present for reasons of backwards compatibility, but not used.
	5	Event Timer	Unsigned16	rw	N	0	Event-Timer

The third transmit PDO contains analog input data as a rule (see Mapping). It is configured for event-driven transmission (transmission type 255). Event-driven mode must first be activated (see object 0x6423), otherwise the inputs can only be interrogated (polled) by remote transmission request (RTR).

Communication parameters for the 4th TxPDO

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x1803	0	Number of elements	Unsigned8	ro	N	5	Communication parameters for the fourth transmit PDO. Sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0x00000480 + Node-ID	COB-ID (Communication Object Identifier) TxPDO1
	2	Transmission Type	Unsigned8	rw	N	255	Transmission type of the PDO
	3	Inhibit Time	Unsigned16	rw	N	0	Repetition delay [value x 100 µs]
	4	CMS Priority Group	Unsigned8	rw	N	-	Present for reasons of backwards compatibility, but not used.
	5	Event Timer	Unsigned16	rw	N	0	Event-Timer

The fourth transmit PDO contains analog input data as a rule (see Mapping). It is configured for event-driven transmission (transmission type 255). Event-driven mode must first be activated (see object 0x6423), otherwise the inputs can only be interrogated (polled) by remote transmission request (RTR).

Communication parameters for the 5th-16th TxPDOs

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x1804-0x180F (depending on the device type)	0	Number of elements	Unsigned8	ro	N	5	Communication parameters for the 5th to 16th transmit PDOs. Sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0x0000000	COB-ID (Communication Object Identifier) TxPDO1
	2	Transmission Type	Unsigned8	rw	N	255	Transmission type of the PDO
	3	Inhibit Time	Unsigned16	rw	N	0	Repetition delay [value x 100 µs]
	4	CMS Priority Group	Unsigned8	rw	N	-	Present for reasons of backwards compatibility, but not used.
	5	Event Timer	Unsigned16	rw	N	0	Event-Timer

Mapping 1st TxPDO

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x1A00	0	Number of elements	Unsigned8	rw	N	Depending on type and fittings	Mapping parameter of the first transmit PDO; sub-index 0: number of mapped objects.
	1	1st mapped object	Unsigned32	rw	N	0x60000108	1st mapped application object (2 bytes index, 1 byte sub-index, 1 byte bit width)
	2	2nd mapped object	Unsigned32	rw	N	0x60000208	2nd mapped application object (2 bytes index, 1 byte sub-index, 1 byte bit width)
	...	...	...	...	...	...	...
	8	8th mapped object	Unsigned32	rw	N	0x60000808	8th mapped application object (2 bytes index, 1 byte sub-index, 1 byte bit width)

The first transmit PDO (TxPDO1) is provided by default for digital input data. Depending on the number of inputs inserted, the necessary length of the PDO is automatically determined, and the corresponding objects are mapped. Since the digital inputs are organised in bytes, the length of the PDO in bytes can be found directly at sub-index 0.

A specific sequence must be observed in order to change the mapping (see object index 0x1600).

Mapping 2nd TxPDO

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x1A01	0	Number of elements	Unsigned8	rw	N	Depending on type and fittings	Mapping parameter of the second transmit PDO; sub-index 0: number of mapped objects.
	1	1st mapped object	Unsigned32	rw	N	0x64010110	1st mapped application object (2 bytes index, 1 byte sub-index, 1 byte bit width)
	2	2nd mapped object	Unsigned32	rw	N	0x64010210	2nd mapped application object (2 bytes index, 1 byte sub-index, 1 byte bit width)
	...	...	...	...	...	...	...
	8	8th mapped object	Unsigned32	rw	N		8th mapped application object (2 bytes index, 1 byte sub-index, 1 byte bit width)

The second transmit PDO (TxPDO2) is provided by default for analog input data. Depending on the number of inputs inserted, the necessary length of the PDO is automatically determined, and the corresponding objects are mapped. Since the analog inputs are organised in words, the length of the PDO in bytes can be found directly at sub-index 0.

A specific sequence must be observed in order to change the mapping (see object index 0x1600).

Mapping 3rd-16th TxPDO

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x1A02-0x1A0F (depending on the device type)	0	Number of elements	Unsigned8	rw	N	Depending on type and fittings	Mapping parameters for the third to sixteenth transmit PDOs; sub-index 0: number of mapped objects.
	1	1st mapped object	Unsigned32	rw	N	0x00000000 (see text)	1st mapped application object (2 bytes index, 1 byte sub-index, 1 byte bit width)
	2	2nd mapped object	Unsigned32	rw	N	0x00000000 (see text)	2nd mapped application object (2 bytes index, 1 byte sub-index, 1 byte bit width)
	...	...	...	...	...	...	...
	8	8th mapped object	Unsigned32	rw	N	0x00000000 (see text)	8th mapped application object (2 bytes index, 1 byte sub-index, 1 byte bit width)

The 3rd to 16th transmit PDOs (TxPDO3ff) are automatically given a default mapping by the bus node depending on the attached terminals (or depending on the extension modules). The procedure is described in the section on PDO Mapping.

A specific sequence must be observed in order to change the mapping (see object index 0x1600).

[Gefahrinformation hier einfügen!]

NoteDS401 V2 specifies analog input and/or output data as the default mapping for PDOs 3+4. This corresponds to Beckhoff's default mapping when less than 65 digital inputs or outputs are present. In order to ensure backwards compatibility, the Beckhoff default mapping is retained - the mapping behavior of the devices therefore corresponds to DS401 V1, where in all other respects they accord with DS401 V2.

For the sake of completeness, the following object entries are also contained in the object directory (and therefore also in the EDS files):

Index	Meaning
0x2000	Digital inputs (function identical to object 0x6000)
0x2100	Digital outputs (function identical to object 0x6100)
0x2200	1-byte special terminals, inputs (at present no terminals corresponding to this type are included in the product range)
0x2300	1-byte special terminals, outputs (at present no terminals corresponding to this type are included in the product range)
0x2400	2-byte special terminals, inputs (at present no terminals corresponding to this type are included in the product range)
0x2500	2-byte special terminals, outputs (at present no terminals corresponding to this type are included in the product range)
0x2E00	7-byte special terminals, inputs (at present no terminals corresponding to this type are included in the product range)
0x2F00	7-byte special terminals, outputs (at present no terminals corresponding to this type are included in the product range)

3-byte special terminals, input data

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x2600	0	Number of elements	Unsigned8	ro	N	Depending on type and fittings	Number of available 3-byte special channels, inputs
	1	1st input block	Unsigned24	ro	Y	0x000000	1st input channel
	...	...	...	...	...	...	...
	0X80	128th input block	Unsigned24	ro	Y	0x000000	128th input channel

Example of special terminals with 3-byte input data (in the default setting): KL2502 (PWM outputs, 2 x 3 bytes)

3-byte special terminals, output data

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x2700	0	Number of elements	Unsigned8	ro	N	Depending on type and fittings	Number of available 3-byte special channels, outputs
	1	1st output block	Unsigned24	rww	Y	0x000000	1st output channel
	...	...	...	...	...	...	...
	0X80	128th output block	Unsigned24	rww	Y	0x000000	128th output channel

Example of special terminals with 3-byte output data (in the default setting): KL2502 (PWM outputs, 2 x 3 bytes)

4-byte special terminals, input data

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x2800	0	Number of elements	Unsigned8	ro	N	Depending on type and fittings	Number of available 4-byte special channels, inputs
	1	1st input block	Unsigned32	ro	Y	0x00000000	1st input channel
	...	...	...	...	...	...	...
	0X80	128th input block	Unsigned32	ro	Y	0x00000000	128th input channel

Examples of special terminals with 4-byte input data (in the default setting): KL5001, KL6001, KL6021, KL6051

4-byte special terminals, output data

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x2900	0	Number of elements	Unsigned8	ro	N	Depending on type and fittings	Number of available 4-byte special channels, outputs
	1	1st output block	Unsigned32	rww	Y	0x00000000	1st output channel
	...	...	...	...	...	...	...
	0X80	128th output block	Unsigned32	rww	Y	0x00000000	128th output channel

Examples of special terminals with 4-byte output data (in the default setting): KL5001, KL6001, KL6021, KL6051

5-byte special terminals, input data

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x2A00	0	Number of elements	Unsigned8	ro	N	Depending on type and fittings	Number of available 5-byte special channels, inputs
	1	1st input block	Unsigned40	ro	Y	0x0000000000	1st input channel
	...	...	...	...	...	...	...
	0X40	64th input block	Unsigned40	ro	Y	0x0000000000	64th input channel

Example of special terminals with 5-byte input data (in the default setting): KL1501

5-byte special terminals, output data

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x2B00	0	Number of elements	Unsigned8	ro	N	Depending on type and fittings	Number of available 5-byte special channels, outputs
	1	1st output block	Unsigned40	rww	Y	0x0000000000	1st output channel
	...	...	...	...	...	...	...
	0X40	64th output block	Unsigned40	rww	Y	0x0000000000	64th output channel

Example of special terminals with 5-byte output data (in the default setting): KL1501

6-byte special terminals, input data

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x2C00	0	Number of elements	Unsigned8	ro	N	Depending on type and fittings	Number of available 6-byte special channels, inputs
	1	1st input block	Unsigned48	ro	Y	0x0000000000	1st input channel
	...	...	...	...	...	...	...
	0X40	64th input block	Unsigned48	ro	Y	0x0000000000	64th input channel

Example of special terminals with 6-byte input data (in the default setting): KL5051, KL5101, KL5111

6-byte special terminals, output data

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x2D00	0	Number of elements	Unsigned8	ro	N	Depending on type and fittings	Number of available 6-byte special channels, outputs
	1	1st output block	Unsigned48	rww	Y	0x0000000000	1st output channel
	...	...	...	...	...	...	...
	0X40	64th output block	Unsigned48	rww	Y	0x0000000000	64th output channel

Example of special terminals with 6-byte output data (in the default setting): KL5051, KL5101, KL5111

8-byte special terminals, input data

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x3000	0	Number of elements	Unsigned8	ro	N	Depending on type and fittings	Number of available 6-byte special channels, inputs
	1	1st input block	Unsigned64	ro	Y	0x0000000000	1st input channel
	...	...	...	...	...	...	...
	0x40	64th input block	Unsigned64	ro	Y	0x0000000000	64th input channel

Example for special terminals with 8-byte input data: KL5101 (with word alignment, not according to the default setting)

8-byte special terminals, output data

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x3100	0	Number of elements	Unsigned8	ro	N	Depending on type and fittings	Number of available 6-byte special channels, outputs
	1	1st output block	Unsigned64	rww	Y	0x0000000000	1st output channel
	...	...	...	...	...	...	...
	0X40	64th output block	Unsigned64	rww	Y	0x0000000000	64th output channel

Example for special terminals with 8-byte output data: KL5101 (with word alignment, not according to the default setting)

Bus node register communication

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x4500	0	Register Access	Unsigned32	rw	N	none	Access to internal bus node registers

The 32 bit value is composed as follows:

MSB			**LSB**
Access (bit 7) + table number (bits 6...0)	Register number	High byte register value	Low byte register value
[0..1] + [0...0x7F]	[0...0xFF]	[0...0xFF]	[0...0xFF]

As is usual in CANopen, the LSB is transferred first, followed by the MSB.

Accessing index 0x4500 allows any registers in the bus station to be written or read. The channel number and the register are addressed here with a 32 bit data word.

Reading the register value

The coupler must first be informed of which register is to be read. This requires an SDO write access to the appropriate index/sub-index combination, with:
- table number (access bit = 0) in byte 3
- register address in byte 2 of the 32-bit data value.

Bytes 1 and 0 are not evaluated if the access bit (MSB of byte 3) equals 0. The register value can then be read with the same combination of index and sub-index.

After the writing of the register address to be read, the coupler sets the access bit to 1 until the correct value is available. Thus an SDO read access must check that the table number lies in the range from 0...0x7F.

An access error during register communication is indicated by the corresponding return value in the SDO protocol (see the SDO section, Breakdown of parameter communication).

An example of reading register values

It is necessary to determine which baud rate index has been assigned to switch setting 1,1 (DIP 7,8). (See the section covering *Network addresses and baud rates*). To do this, the value in table 100, register 3, must be read. This means that the following SDO telegrams must be sent:

Write access (download request) to index 4500, sub-index 0, with the 32 bit data value 0x64 03 00 00.

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

Then a read access (upload request) to the same index/sub-index. The data value sent here is irrelevant (00 is used here).

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

The coupler responds with the upload response telegram:

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

This tells us that the value contained in this register is 4, and this baud rate index corresponds to 125 kbit/s (the default value).

Writing register values

SDO write access to the corresponding combination of index and sub-index with:
- table number + 0x80 (access bit = 1) in byte 3
- register address in byte 2
- high byte register value in byte 1
- low byte register value in byte 0 of the 32-bit data value.

Remove coupler write protection

Before the registers of the Bus Coupler can be written, the write protection must first be removed. In order to do this, the following values must be written in the given sequence to the corresponding registers:

Step	Table	Register	Value	Corresponding SDO download value (0x4500/0)
1.	99	2	45054 (0xAFFE)	0xE3 02 AF FE (0xE3=0x63(=99)+ 0x80)
2.	99	1	1 (0x0001)	0xE3 01 00 01
3.	99	0	257 (0x0101)	0xE3 00 01 01

Remove coupler write protection (CAN representation)

In order to remove the coupler write protection, the following SDO telegrams (download requests) must thus be sent to the coupler:

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

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

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

An example of writing register values

After the write protection has been removed, the baud rate index for DIP switch setting 1,1 is to be set to the value 7. This will assign a baud rate of 20 kbaud to this switch setting.

This requires the value 7 to be written into table 100, register 3. This is done with an SDO write access (download request) to index 0x4500, sub-index 0 with the 32 bit value E4 03 00 07 (0xE4 = 0x64+0x80):

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

Identify terminals

The identifier of the coupler (or of the bus station) and of the attached Bus Terminals can be read from the Bus Coupler's table 9. Register 0 then contains the identifier of the Bus Coupler itself, register 1 the identifier of the first terminal and register n the identification of the nth terminal:

Table number	Register number	Description	Value range
9	0	Bus station identifier	0 - 65535
9	1-255	Identifier of the extension module/bus terminal	0 - 65535

The Bus Coupler description in register number 0 contains 5120 = 0x1400 for the BK5120, 5110 = 0x13F6 for the BK5110 and 5100 = 0x13EC for the LC5100. The Fieldbus Box modules contain the identifier 510 dec = 0x1FE in register 0.

In the case of analog and special terminals, the terminal identifier (dec) is contained in the extension module identifier or the terminal description.
Example: if a KL3042 is plugged in as the third terminal, then register 3 contains the value 3042$_{dec}$ (0x0BE2).

The following bit identifier is used for digital terminals:

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

s6...s1: data width in bits; a=1: output terminal; e=1: input terminal

This identifier scheme results in the terminal descriptions listed below:

Extension module identifier	Meaning
0x8201	2 bit digital input terminal, e.g. KL1002, KL1052, Kl9110, KL9260
0x8202	2 bit digital output terminal, e.g. KL2034, KL2612, KL2702
0x8401	4 bit digital input terminal, e.g. KL1104, KL1124, KL1194
0x8402	4 bit digital output terminal, e.g. KL2124, KL2134, KL2184
0x8403	4 bit digital input/output terminal, e.g. KL2212

General coupler configuration (table 0)

Table 0 of the Bus Coupler contains the data for the general coupler configuration. It is not, as a general rule, necessary to change this; however, for special applications it is possible to change the settings using the KS2000 configuration software, or through direct access via register communication. The write protection must first be removed in order to do this (see above).

The relevant register entries are described below:

K-Bus configuration

Table 0, register 2, contains the K-Bus configuration, and is coded as follows (default value: 0x0006):

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

A: Auto-reset

If there is a K-Bus error, attempts are made cyclically to start the K-Bus up again through a reset. If emergency telegrams and guarding are not evaluated, activation of auto-reset can lead to output and input information being lost without that loss being noticed.

0: No auto-reset (default)

1: Auto-reset active

G: Device diagnostics

Reporting (by means of emergency telegram), that, for example
- a current input is open circuit (with diagnostics)
- 10 V exceeded at a 1-10V input terminal

0: Device diagnostics switched off

1: Device diagnostics active (default)

D: Diagnostic data

from digital terminals is included in the process image (e.g. KL2212). This flag is only evaluated when device diagnostics is active (see above).

0: Do not display

1: Display (default)

Process image description

Table 0, register 3, contains the process image description, and is coded as follows (default value: 0x0903):

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

k0...k1: Reaction to K-Bus errors

0,2: Inputs remain unchanged (default = 2);

1: Set inputs to 0 (TxPDO with zeros is sent)

f0...f1: Reaction to fieldbus error

0: Stop the K-Bus cycles, watchdog in the terminals triggers, fault output values become active. The old output values are initially set during a restart.

1: Set outputs to 0, then stop the K-Bus cycles (default). 2: Outputs remain unchanged.

a: Word alignment (of analog and special terminals)

0: No alignment (default)

1: Map data to word boundaries (process data always starts on an even address in the PDO)

d: Data format for complex terminals (analog and special terminals)

0: Intel format (default)

1: Motorola format

k: Evaluation of complex terminals (analog and special terminals)

0: User data only (default)

1: Complete evaluation (note: analog channels then, for example, need 3 input and 3 output bytes instead of, e.g., 2 input bytes; instead of 4 channels per PDO, 2 channels require a RxPDO and a TxPDO)

Bus Terminal / Extension Box register communication

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x4501	0	Access Terminal Register	Unsigned8	ro	N	none	Index 0x4501 allows access to all the registers in the bus terminal or extension module. Sub-index 0 contains the number of attached bus terminals.
	1	Access Reg. Terminal 1	Unsigned32	rw	N	none	Access to bus terminal or extension module register 1
	...	...	...	...	...	...	...
	0XFE	Access Reg. Terminal 254	Unsigned32	rw	N	none	Access to bus terminal or extension module register 254

The 32 bit value is composed as follows:

MSB			**LSB**
Access (bit 7) + channel number (bits 6...0)	Register number	High byte register value	Low byte register value
[0..1] + [0...0x7F]	[0...0xFF]	[0...0xFF]	[0...0xFF]

As is usual in CANopen, the LSB is transferred first, followed by the MSB.

Accessing index 0x4501 allows the user registers in the bus terminal or extension module to be written or read. The modules have a set of registers for each input or output channel. The modules are addressed by means of the sub-index; the channel number and register are addressed in the 32-bit data value. Channel number 0 corresponds here to the first channel, 1 to the second channel, and so forth.

Reading the register value

The coupler must first be informed of which register is to be read. This requires an SDO write access to the appropriate index/sub-index combination, with:
- channel number (access bit = 0) in byte 3
- register address in byte 2 of the 32-bit data value.

Bytes 1 and 0 are not evaluated if the access bit (MSB of byte 3) equals 0. The register value can then be read with the same combination of index and sub-index.

After the writing of the register address to be read, the coupler sets the access bit to 1 until the correct value is available. Thus an SDO read access must check that the table number lies in the range from 0...0x7F.

An access error during register communication is indicated by the corresponding return value in the SDO protocol (see the SDO section, Breakdown of parameter communication).

An example of reading register values

The thermocouple type to which the second input channel of a KL3202 Thermocouple Input Terminal has been set is to be determined. This requires feature register 32 to be read. The terminal is located in the fifth slot, next to the Bus Coupler. This means that the following SDO telegrams must be sent:

Write access (download request) to index 4501, sub-index 5 with 32 bit data value 01 20 00 00 (0x01 = 2nd channel, 0x20 = register 32)
Id=0x600+Node-ID DLC=8; Data=23 01 45 05 00 00 20 01

Then a read access (upload request) to the same index/sub-index. The data value sent here is irrelevant (0x00 is used here).
Id=0x600+Node-ID DLC=8; Data=40 01 45 05 00 00 00 00

The coupler responds with the upload response telegram:
Id=0x580+Node-ID DLC=8; Data=43 01 45 05 06 31 20 01

This means that the feature register contains the value 31 06. The upper 4 bits indicate the thermocouple type. Their value here is 3, which means that PT500 is the type that has been set for this channel (see the KL3202 documentation).

Writing register values

SDO write access to the corresponding combination of index and sub-index with:
- channel number + 0x80 (access bit = 1) in byte 3
- register address in byte 2
- high byte register value in byte 1
- low byte register value in byte 0 of the 32-bit data value.

NOTICE

[Gefahrinformation hier einfügen!]

WarningIf the write protection is not removed (as a result, for instance, of a faulty codeword), then although a write access to the terminal register will be confirmed (SDO download response), the value is not in fact entered into the register. It is therefore recommended that the value is read back after writing and compared.

Remove terminal write protection

Before the user registers in the Bus Terminal (register 32-xx, depending on terminal type or extension module) can be written to, it is first necessary for write protection to be removed. The following codeword is written for this purpose into register 31 of the channel concerned:

Write protection	Channel	Register	Value	Corresponding SDO download value (0x4500/0)
	1,2, 3 or 4	31 (0x1F)	4661 (0x1235)	8y 1F 12 35 (y = channel number)

Remove terminal write protection (CAN representation)

In order to remove the terminal's write protection, the following SDO telegram must thus be sent to the coupler:

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

where xx is the terminal's slot, and y indicates the channel.

An example of removing write protection

Suppose that a KL3202 Thermocouple Input Terminal is inserted into slot 5 of a BK5120 that has node address 3, then the write protection for the first channel can be removed as follows:

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

The following telegram is sent for the second channel:

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

An example of writing register values

The type of thermocouple attached to the second channel of the KL3202 Terminal in slot 5 is now to be changed to PT1000. For this purpose, the value 2 must be written into the upper 4 bits (the upper nibble) of the feature register. It is assumed to that the default values are to be supplied for all the other bits in the feature register. Once the write protection has been removed, SDO write access (download request) is used to write the following 32 bit value into index 0x4501, sub-index 05: 81 20 21 06 (0x81=01+0x80; 0x20=32;0x2106 = register value).

The corresponding telegram on the bus looks like this:

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

Activate PDOs

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x5500	0	Activate PDO Defaults	Unsigned32	rw	N	0x00000000	sets PDO communication parameters for PDOs 2...11

CANopen defines default identifiers for 4 transmit (Tx) and 2 receive (Rx) PDOs, all other PDOs being initially deactivated after the nodes have started up. Index 0x5500 can activate all the PDOs that, in accordance with the terminals inserted, are filled with process data (manufacturer-specific default mapping). A manufacturer-specific default identifier allocation is carried out here for PDO5…11, while the transmission type and a uniform inhibit time is set for PDO2…11. PDOs that do not have process data (and which are thus superfluous in the present configuration) are not activated.

[Gefahrinformation hier einfügen!]

NoteThis object can only be written in the pre-operational state!

The 32-bit value is used as follows:

MSB			LSB
Transmission Type RxPDOs	Transmission Type TxPDOs	High byte inhibit time	Low byte inhibit time

As is usual in CANopen, the LSB is transferred first, followed by the MSB.

Example

Activate PDOs for bus node number 1, set inhibit time to 10ms (=100 x 100µs), set transmission type for TxPDOs to 255, and set transmission type for RxPDOs to 1. The following telegram must be sent:
Id=0x601 DLC=8; Data=23 00 55 00 64 00 FF 01

The node responds with the following telegram:
Id=0x601 DLC=8; Data=60 00 55 00 00 00 00 00

Identifiers used

The default identifier allocation for the additional PDOs leaves the pre-defined regions for guarding, SDOs etc. free, assumes a maximum of 64 nodes in the network with PDO6 as the next node, and proceeds according to the following scheme:

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

NOTICE

[Gefahrinformation hier einfügen!]

WarningEnsure that index 0x5500 is not used if Bus Couplers with more than 5 PDOs are present in networks with node addresses > 64, otherwise identification overlaps can occur. In that case, the PDO identifiers must be set individually.

For the sake of clarity, the default identifiers defined according to CANopen are also listed here:

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

The identifiers that result from the DIP switch settings on the coupler are given, as are the identifier regions for the node addresses 64...127 (not settable in Bus Couplers BK5110, BK5120 and LC5100) in square brackets. Addresses 1…99 can be set for the Fieldbus Box modules and the BK515x Bus Couplers.

The appendix [▶ 139] contains a tabular summary of all the identifiers.

Digital inputs

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x6000	0	Number of elements	Unsigned8	ro	N	Depending on type and fittings	Number of available digital 8-bit input data blocks
	1	1st input block	Unsigned8	ro	Y	0x00	1st input channel
	...	...	...	...	...	...	...
	0XFE	254th input block	Unsigned8	ro	Y	0x00	254th input channel

Interrupt mask

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x6126	0	Number of elements	Unsigned8	ro	N	Depending on type	The number of 32-bit interrupt masks = 2 x the number of TxDPOs
	1	IR-Mask0 TxPDO1	Unsigned32	rw	N	0xFFFFFFFF	IR-mask bytes 0...3 TxPDO1
	2	IR-Mask1 TxPDO1	Unsigned32	rw	N	0xFFFFFFFF	IR-mask bytes 4...7 TxPDO1
	3	IR-Mask0 TxPDO2	Unsigned32	rw	N	0xFFFFFFFF	IR-mask bytes 0...3 TxPDO2
	...	...	...	...	...	...	...
	0x20	IR-Mask1 TxPDO16	Unsigned32	rw	N	0xFFFFFFFF	IR-mask bytes 4...7 TxPDO16

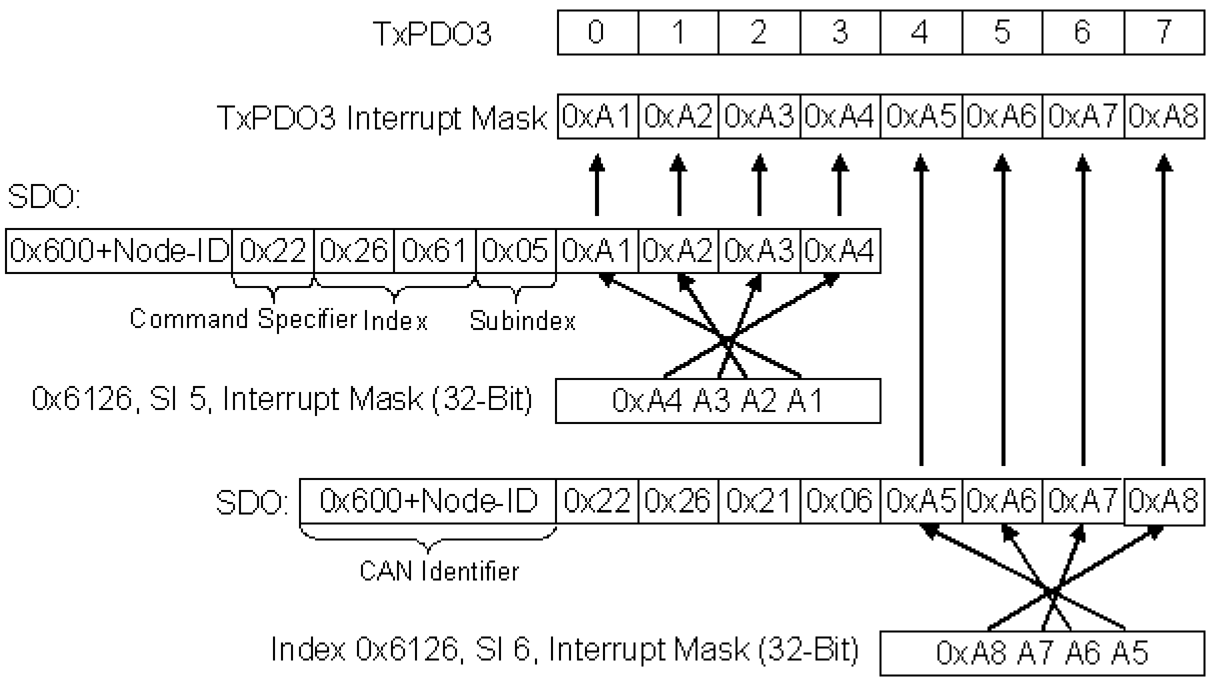
By default, every change in the value in an event-driven PDO causes a telegram to be sent. The interrupt mask makes it possible to determine which data changes are evaluated for this purpose. By clearing the appropriate ranges within the PDOs they are masked out for event-driving purposes (interrupt control). The interrupt mask does not just govern all the PDOs with digital inputs, but all the TxPDOs that are present. If the TxPDOs are shorter than 8 bytes, then the superfluous part of the IR mask is not evaluated.

The interrupt mask only has an effect on TxPDOs with transmission types 254 and 255. It is not stored in the device (not even through object 0x1010). Changes to the mask at runtime (when the status is operational) are possible, and are evaluated starting from the next change of input data.

The interrupt mask for TxPDOs with analog input data is not evaluated if either limit values (0x6424, 0x6425) or the delta function (0x6426) have been activated for the inputs.

This entry has been implemented in firmware C3 and above.

Example of data assignment



Application example

The value contained in a fast counter input is only to be transmitted when bits in the status word (the latch input, for instance) have changed. This requires the 32 bit counter value to be masked out (zeroed) in the interrupt mask. The status is located in byte 0, while the counter value is, by default, contained in bytes or 1..4 of the corresponding PDOs (TxPDO3 in this example, because < 65 digital and < 5 analog inputs are present).
This means that index 0x6126, sub-index5 must receive the value 0x0000 00FF and that sub-index6 must have 0xFFFF FF00 written into it.

The corresponding SDOs therefore appear as follows:

11 bit identifier	8 bytes of user data							
0x600+ node ID	0x22	0x26	0x61	0x05	0xFF	0x00	0x00	0x00

11 bit identifier	8 bytes of user data							
0x600+ node ID	0x22	0x26	0x61	0x06	0x00	0xFF	0xFF	0xFF

Digital outputs

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x6200	0	Number of elements	Unsigned8	ro	N	Depending on type and fittings	Number of available digital 8-bit output data blocks
	1	1st input block	Unsigned8	rw	Y	0x00	1st output channel
	...	...	...	...	...	...	...
	0XFE	254th input block	Unsigned8	rw	Y	0x00	254th output channel

Analog inputs

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x6401	0	Number of elements	Unsigned8	ro	N	Depending on type and fittings	Number of analog input channels available
	1	1st input	Unsigned16	ro	Y	0x0000	1st input channel
	...	...	...	...	...	...	...
	0XFE	254th input	Unsigned16	ro	Y	0x0000	254th input channel

The analog signals are displayed left aligned. The representation in the process image is therefore independent of the actual resolution. Detailed information on the data format can be found at the relevant signal type.

Analog outputs

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x6411	0	Number of elements	Unsigned8	ro	N	Depending on type and fittings	Number of analog output channels available
	1	1st input block	Unsigned16	rw	Y	0x0000	1st output channel
	...	...	...	...	...	...	...
	0XFE	254th input block	Unsigned16	rw	Y	0x0000	254th output channel

The analog signals are displayed left aligned. The representation in the process image is therefore independent of the actual resolution. Detailed information on the data format can be found at the relevant signal type.

Event driven analog inputs

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x6423	0	Global Interrupt Enable	Boolean	rw	N	FALSE (0)	Activates the event-driven transmission of PDOs with analog inputs.

Although, in accordance with CANopen, the analog inputs in TxPDO2..4 are by default set to transmission type 255 (event driven), the event (the alteration of an input value) is suppressed by the event control in object 0x6423, in order to prevent the bus from being swamped with analog signals. It is recommended that the flow of data associated with the analog PDOs is controlled either through synchronous communication or through using the event timer. In event-driven operation, the transmission behavior of the analog PDOs can be parameterized before activation by setting the inhibit time (object 0x1800ff, sub-index 3) and/or limit value monitoring (objects 0x6424 + 0x6425) and/or delta function (object 0x6426).

Upper limit value analog inputs

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x6424	0	Number of elements	Unsigned8	ro	N	Depending on type and fittings	Number of analog input channels available
	1	upper limit 1st input	Unsigned16	rw	Y	0x0000	Upper limit value for 1st input channel
	...	...	...	...	...	...	...
	0XFE	upper limit 254th input	Unsigned16	rw	Y	0x0000	Upper limit value for 254th input channel

Values different from 0 activate the upper limit value for this channel. A PDO is then transmitted if this limit value is exceeded. In addition, the event driven mode must be activated (object 0x6423). The data format corresponds to that of the analog inputs.

Lower limit value analog inputs

Index	Sub-index	Name	Type	Attribute	Mapping	Default value	Meaning
0x6425	0	Number of elements	Unsigned8	ro	N	Depending on type and fittings	Number of analog input channels available
	1	lower limit 1st input	Unsigned16	rw	Y	0x0000	Lower limit value for 1st input channel
	...	...	...	...	...	...	...
	0XFE	lower limit 254th input	Unsigned16	rw	Y	0x0000	Lower limit value for 254th input channel

Values different from 0 activate the lower limit value for this channel. A PDO is then transmitted if the value falls below this limit value. In addition, the event driven mode must be activated (object 0x6423). The data format corresponds to that of the analog inputs.

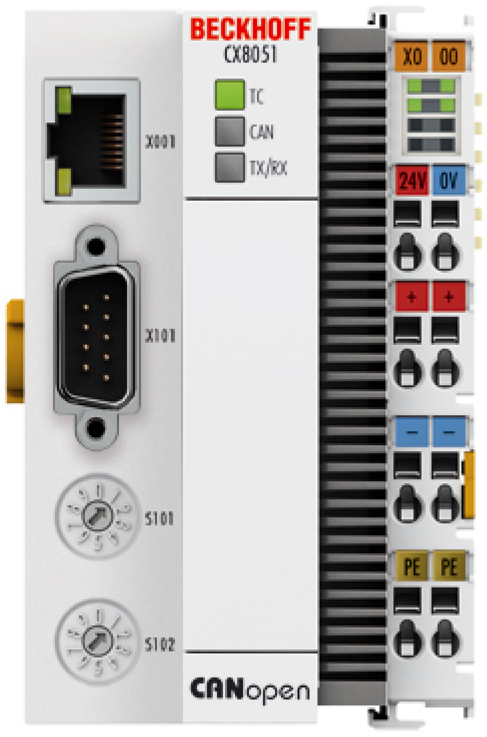
Delta function for analog inputs

Index	**Sub-index**	**Name**	**Type**	**Attribute**	**Mapping**	**Default value**	**Meaning**
0x6426	0	Number of elements	Unsigned8	ro	N	Depending on type and fittings	Number of analog input channels available
	1	delta value 1st input	Unsigned16	rw	Y	0x0000	Delta value for the 1st input channel
	...	...	...	...	...	...	...
	0XFE	delta value 254th input	Unsigned16	rw	Y	0x0000	Delta value for the 254th input channel

Values different from 0 activate the delta function for this channel. A PDO is then transmitted if the value has changed by more than the delta value since the last transmission. In addition, the event driven mode must be activated (object 0x6423). The data format corresponds to that of the analog inputs (delta value: can only have positive values).

10 Error handling and siagosis

10.1 LED displays



Ethernet interface X001

Interface X001	Ethernet (CX805x)	Meaning
LED green	on	Link present
LED yellow	flashing	Activity

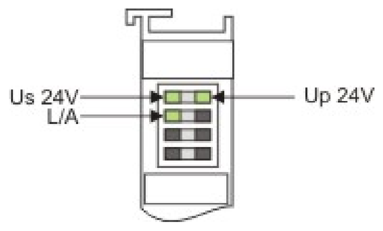
CAN master LEDs of the CX8050

Labeling	Meaning	Color	Meaning
TC	Indicates the status of the coupler	red	TwinCAT is in "stop" mode
		Green	TwinCAT is in "run" mode
		Blue (If red DIP switch 1 is set to on when starting the coupler)	TwinCAT is in "config" mode
CAN	Shows the CAN status	Green on / Red off	CAN is OK
		Green off / Red on	CAN in bus off
		Green 200 ms / Red 200 ms	CAN Warning
		Green off / Red on	CAN not configured
TX/RX	Indicates CAN errors	Green on / Red off	All nodes have NodeState = 0
		Green 200 ms / Red 200 ms	All boxes in OP state, but the tasks have not yet started
		Green off / Red 200 ms	Not all nodes in OP
		Green off / Red on	No boxes configured

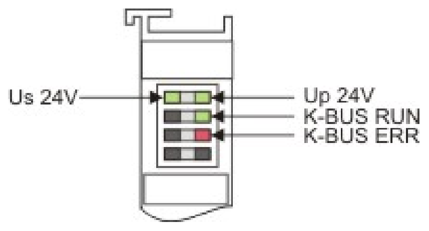
CANopen slave LEDs of the CX8051

Labeling	Meaning	Color	Meaning
TC	Indicates the status of the coupler	red	TwinCAT is in "stop" mode
		Green	TwinCAT is in "run" mode
		Blue (If red DIP switch 1 is set to on when starting the coupler)	TwinCAT is in "config" mode
CAN	Shows the CAN status	Green on / Red off	CAN is OK
		Green off / Red on	CAN in bus off
		Green 200 ms / Red 200 ms	CAN Warning
		Green and red flashing rapidly	Baud rate search active
		Green off / Red on	CAN not configured
TX/RX	Indicates CAN errors	Green on	Everything OK
		Green 200 ms / Red 200 ms	All boxes in OP state, but the tasks have not yet started
		Green off / Red 200 ms	Not all boxes in OP
		Green off / Red on	No boxes configured

Power supply terminal LEDs



Operation with E-bus terminals



Operation with K-bus terminals

Display LED	Description	Meaning
1 Us 24 V (top left, 1st row)	CX80xx supply voltage	connected to -24 V
2 Up 24 V (top right, 1st row)	Power contacts supply voltage	connected to -24 V
3 L/A (left centre, 2nd row)	EtherCAT LED	flashing green: EtherCAT communication active connected to E-bus / no data traffic not connected to E-bus
4 K-BUS RUN (right centre, 2nd row)	K-bus LED RUN	Lights up green: K-bus running, everything OK
6 K-BUS ERR (bottom right, 3rd row)	K-bus LED ERR	Lights up red: K-bus error - see K-bus error code

K-bus error codes

Error code	Error code argument	Description	Remedy
Persistent, continuous flashing		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)
3 pulses	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 pulses	0	K-Bus data error, break behind the Bus Coupler	Check whether the n+1 Bus Terminal is correctly connected; replace if necessary.
	n	Break behind Bus Terminal n	Check whether the Bus End Terminal 9010 is connected.
5 pulses	n	K-bus error in register communication with Bus Terminal n	Exchange the nth bus terminal
6 pulses	0	Error at initialisation	Exchange Bus Coupler
	1	Internal data error	Perform a hardware reset on the Bus Coupler (switch off and on again)
	8	Internal data error	Perform a hardware reset on the Bus Coupler (switch off and on again)
7 pulses	0	Process data lengths do not correspond to the configuration	Check the Bus Terminals for the configured Bus Terminals
	1..n	K-bus reset failed	Check the Bus Terminals

11 Appendix

11.1 First steps

The following components are necessary for the first steps

- PC with TwinCAT 2.11 R3
- Ethernet cable
- Power supply (24 V_{DC}), cabling material
- a KL2xxx or an EL2xxx, digital output terminal, end terminal

Required TwinCAT version

TwinCAT 2.11 R3 is required for the programming of the CX80xx series. Older TwinCAT versions and TwinCAT 3.x are not supported!

1. Connect K-bus or E-bus terminals to the controller.

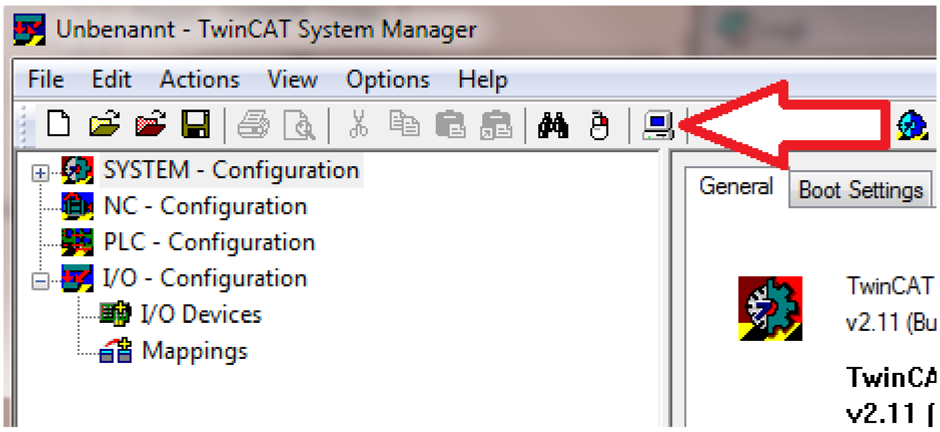
2. Connect voltage to the CX80xx (see power supply [▶ 22]).

3. Connect Ethernet (CX80xx X001) to your network or a direct connection to your PC (make sure in the case of a peer-to-peer connection that the IP addressing in your PC is set to DHCP).

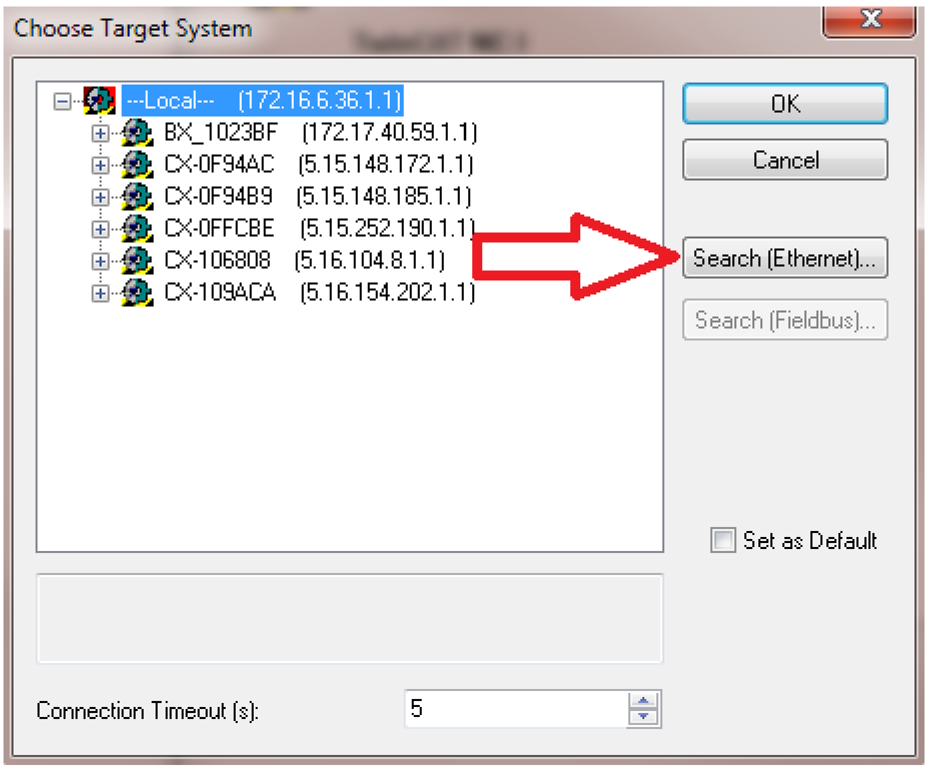
4. Wait a while, approx. 1 to 2 minutes; either the CX80xx will be assigned an address by the DHCP server (usually fast) or, if it does not find a DHCP server, it uses a local IP address.

5. Switch on TC on the PC in Config Mode (TwinCAT icon blue) and start the System Manager

6. In the System Manager, click on the PC symbol (Choose Target System) or press >F8<



7. The following dialog box opens; click on Search (Ethernet).

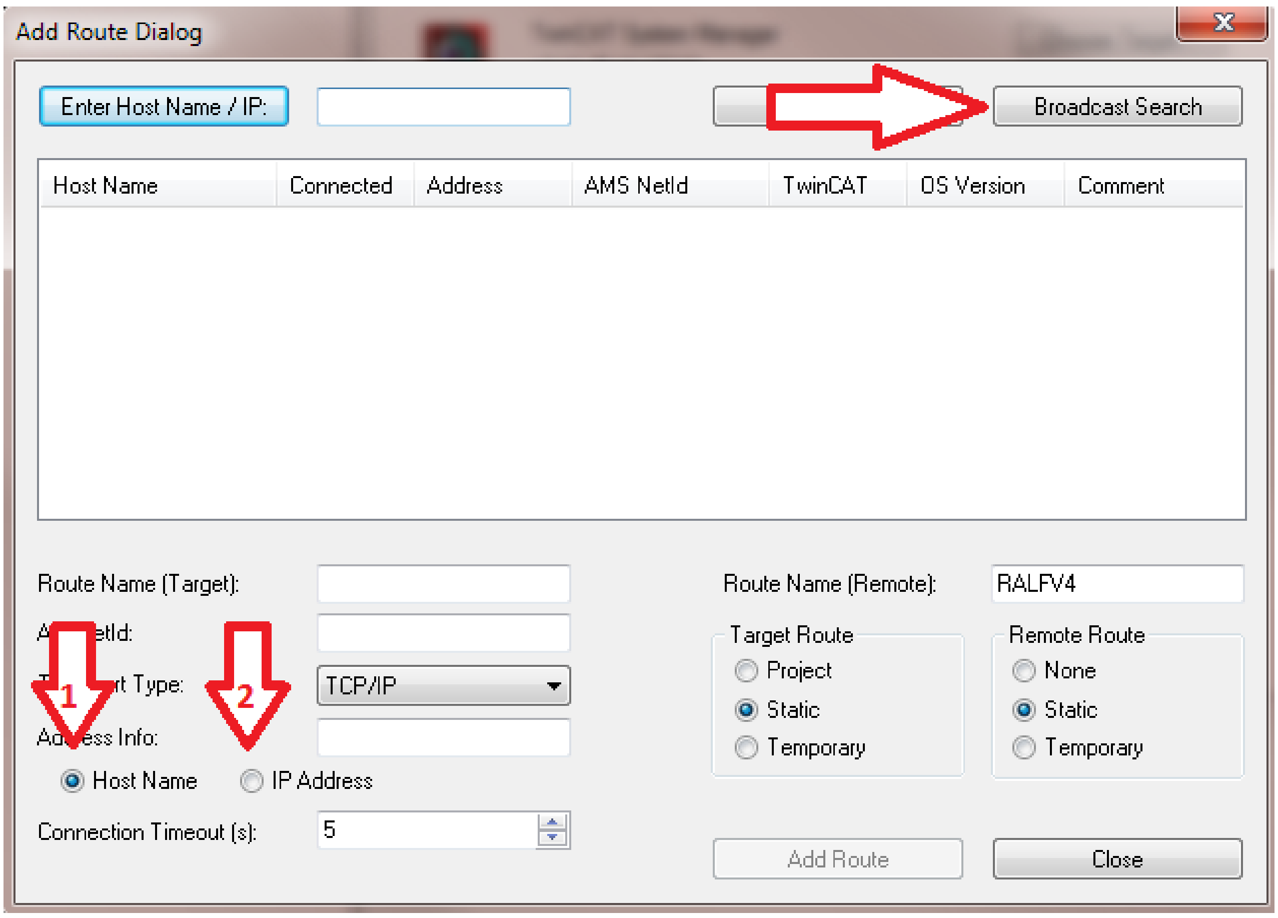

Choose Target System

---Local--- (172.16.6.36.1.1)
BX_1023BF (172.17.40.59.1.1)
CX-0F94AC (5.15.148.172.1.1)
CX-0F94B9 (5.15.148.185.1.1)
CX-0FFCBE (5.15.252.190.1.1)
CX-106808 (5.16.104.8.1.1)
CX-109ACA (5.16.154.202.1.1)

Set as Default

Connection Timeout (s): 5

8. Select Option 1 if you have addressed via DHCP or Option 2 in case of DHCP or local IP address. Then click on "Broadcast search".


Add Route Dialog

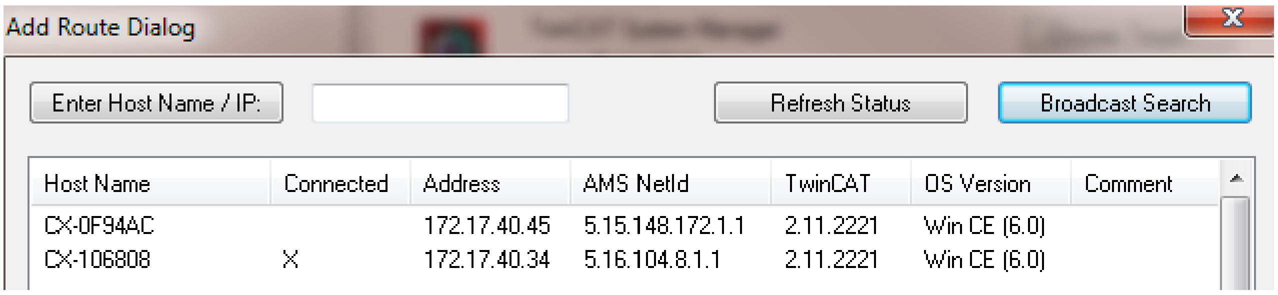
Host Name	Connected	Address	AMS NetId	TwinCAT	OS Version	Comment

Route Name (Target):
AmsNetId:
Transport Type: TCP/IP
Address Info:
[✓ Host Name] IP Address
Connection Timeout (s): 5

Route Name (Remote): RALFV4
Target Route: Project [✓ Static] Temporary
Remote Route: None [✓ Static] Temporary

Your network is scanned for Beckhoff controllers. If none is found, this means that the DHCP of the controller is not yet completed or the network settings on your PC are incorrect. A network cable that has not been connected can naturally also be the cause, but this should not be the case if point 3 has been done.

9. The host name is composed by default of "CX" and the last 3 bytes of the MAC address. You can find the MAC address on the side of the CX80xx. The MAC address is always 6 bytes long and the first three bytes are the vendor ID, which is always 00 01 05 in the case of Beckhoff devices.

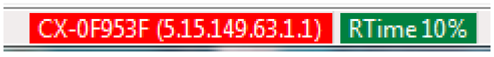


Host Name	Connected	Address	AMS NetId	TwinCAT	OS Version	Comment
CX-0F94AC		172.17.40.45	5.15.148.172.1.1	2.11.2221	Win CE (6.0)	
CX-106808	X	172.17.40.34	5.16.104.8.1.1	2.11.2221	Win CE (6.0)	

An "X" next to *Connected* means that the CX is already known in the system and can be used. To make it known, click in the list on the CX with which you want to connect and then click on "Add route". An input mask opens with "User name" and "Password". By default there is no password, simply confirm by clicking on OK. Afterwards the "X" should appear next to *connected*.

10. Next, the CX should appear in the list of the devices; select it and confirm by clicking on OK.

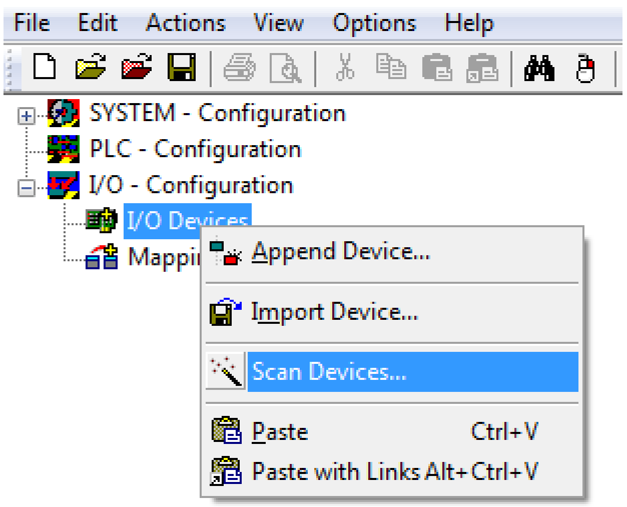
11. Check whether the connection is there. In the System Manager in the bottom right-hand corner. It must be blue or green and may **not** be yellow.



12. If the setting is green, switch the CX to Config Mode with "Shift F4" or click on the blue TC icon in the System Manager. The System Manager now asks you whether you really want to switch to Config Mode; confirm by clicking on OK.

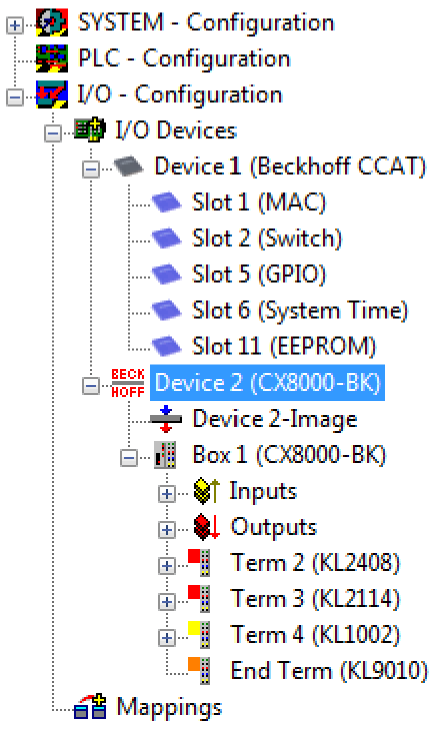
13. The setting at the bottom right must now change to blue and the TC LED on the CX80xx must now also light up blue.

14. Now click on I/O Devices and then on Scan Devices...

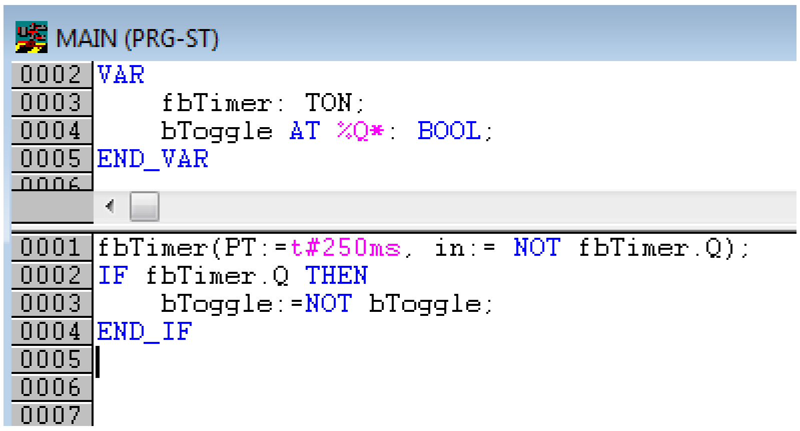


15. A message appears, informing you that not everything will be automatically detected.

16. The CCAT interface is usually found (CX8090) or the corresponding fieldbus interface (other CX80xx devices) and either a K-bus interface or an EtherCAT interface, depending now on which terminals you have connected to the CX. The CCAT interface must be present in the System Manager file and may not be deleted. If an error message should appear when scanning, check the revision level of your TwinCAT version and perform an update if necessary.



17. Now we come to the programming. To do this, open the PLC Control and select File -> New. The PLC Control asks you for the target system. Select CX (ARM). Afterwards it asks you for the function block; set the ST language (structured text). Now write a small program...


MAIN (PRG-ST)

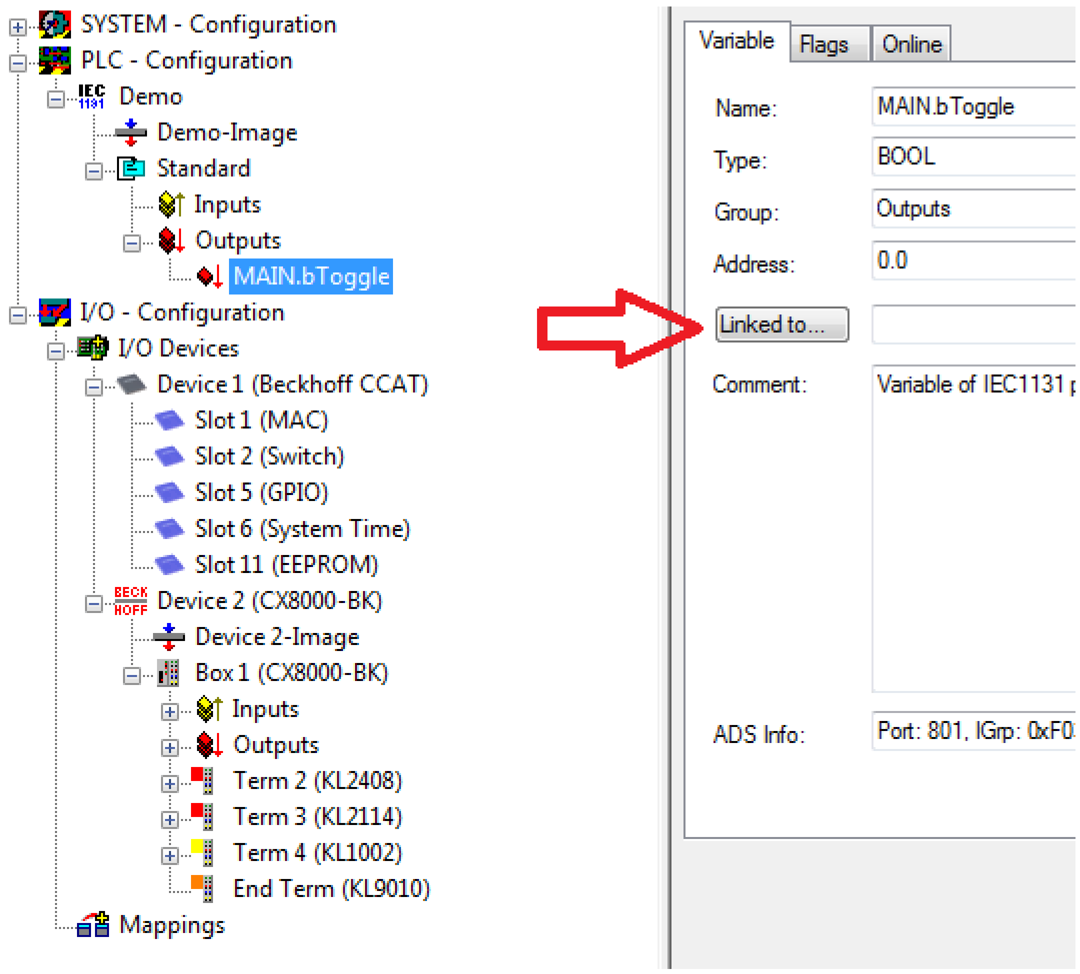
```
VAR
    fbTimer: TON;
    bToggle AT %Q*: BOOL;
END_VAR
```

```
fbTimer(PT:=t#250ms, in:= NOT fbTimer.Q);
IF fbTimer.Q THEN
    bToggle:=NOT bToggle;
END_IF
```

Translate the program. If it is error free (a warning must come, that it is OK) save the project under an arbitrary name, translate it again and save it once **again**.

18: Switch once again to the System Manager and add the program under PLC - Configuration. A FileName.typ file is sought.

19: Now open the project, then the task and then outputs, in which there must be a variable MAIN.bToggle. You can link this with the hardware. To do this, click on "Linked to...".



Name: MAIN.bToggle
Type: BOOL
Group: Outputs
Address: 0.0
Comment: Variable of IEC1131 p
ADS Info: Port: 801, IGrp: 0xF0

Select a digital output. Now you can download the configuration to the CX and switch the CX to Run Mode. To do this, click on the ‘cube’ or press Ctrl + Shift + F4. The TC LED on the CX must then light up green.

20. Switch back to PLC Control, go to "Online/Selection of the target system", select the correct CX, click on it and select runtime system 1. Now "Online/login" again (F11) transfer the PLC program, then "Online/Start" (F5). The LED on your terminal must now flash.

21. As a final step Online/Generate a boot project. This completes the project and your first program is stored on the CX.

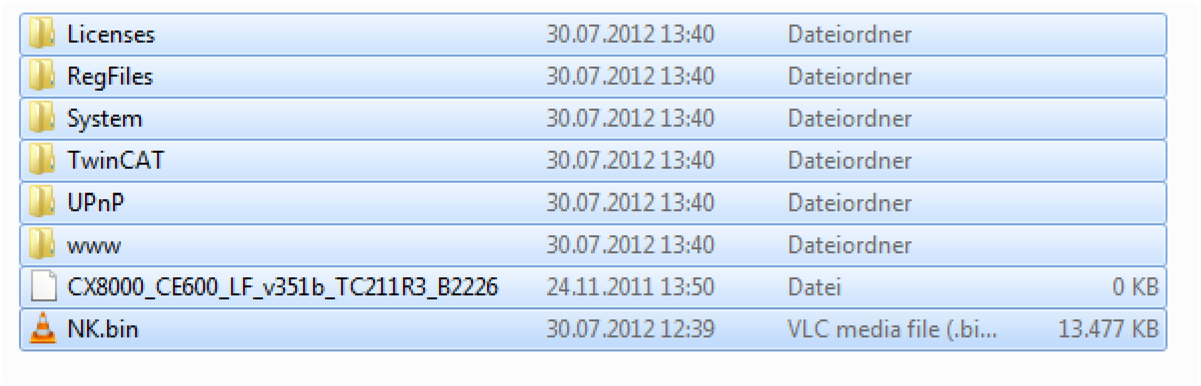
Congratulations, the foundations have now been laid and are ready to be built on. We wish you every success!

11.2 Image Update

There are two different possibilities to update the image of the CX80xx.

Prerequisites
- Please make sure before the update that your CX80xx supports the image that you want to load.
- When updating the image, please first update all existing files and only then copy the new image.

Always copy all files and directories in order to update a CX80xx.



Licenses	30.07.2012 13:40	Dateiordner	
RegFiles	30.07.2012 13:40	Dateiordner	
System	30.07.2012 13:40	Dateiordner	
TwinCAT	30.07.2012 13:40	Dateiordner	
UPnP	30.07.2012 13:40	Dateiordner	
www	30.07.2012 13:40	Dateiordner	
CX8000_CE600_LF_v351b_TC211R3_B2226	24.11.2011 13:50	Datei	0 KB
NK.bin	30.07.2012 12:39	VLC media file (.bi...	13.477 KB

Update via USB

⚠ CAUTION

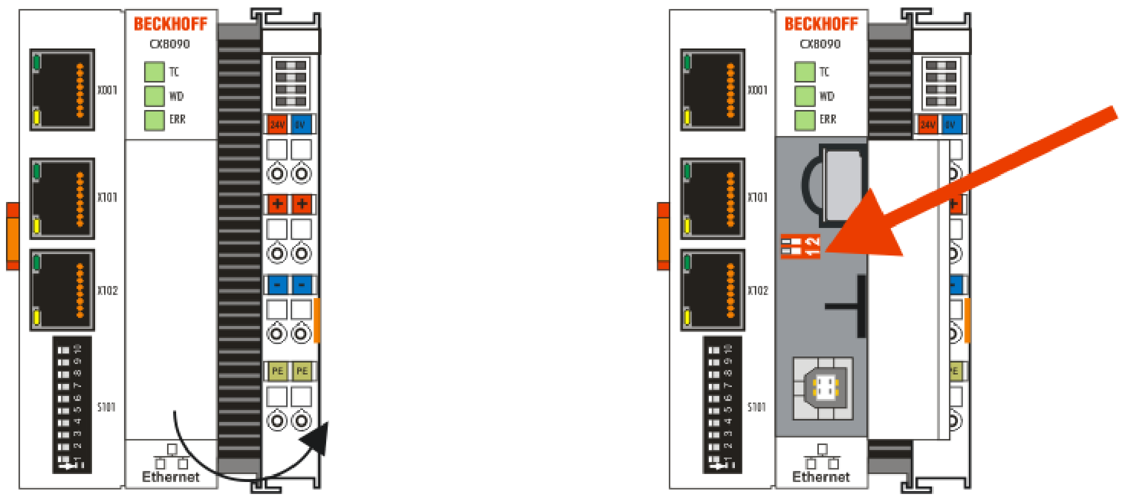
USB port as ignition source in potentially explosive atmospheres

Gases or dusts can be ignited by a spark discharge when the USB port is used.

Switch off the power supply and wait until the 1-second UPS has discharged. Ensure that there is no explosive atmosphere before you use the USB port.

A USB cable is required for this!

- Switch off the CX80xx



- Set red Dip switch (under the flap) DIP 1 to ON
- Switch on the CX
- Connect the PC with USB
- Delete all files (we recommend that you backup all files first), no formatting

- Wait until copying has finished, then remove the USB cable
- Switch DIP switch 1 to OFF
- Switch off the CX80xx
- Switch on the CX80xx; it may take a little longer the first time

Update the MicroSD card

A MicroSD card reader is required for this!

- Remove the MicroSD card from the switched-off CX device.
- Insert the MicroSD card into the reader
- Delete all files (we recommend that you backup all files first), no formatting
- Load the new image
- Wait until copying has finished, then remove the MicroSD card
- Insert the MicroSD card into the SD slot of the switched-off CX again
- Now switch on the CX again; it may take a little longer the first time

11.3 Certification

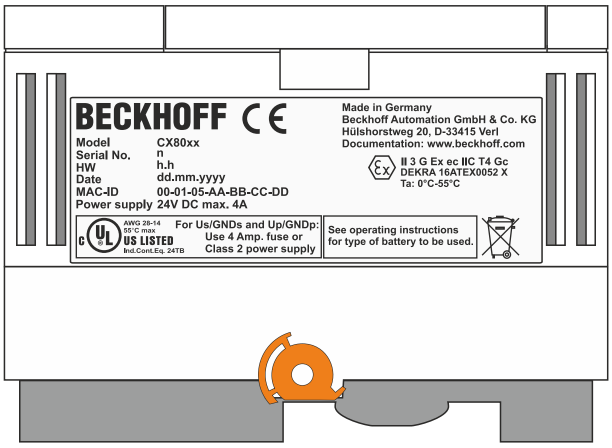
11.3.1 Ex

The CX8xxx Embedded PCs, which are certified for use in hazardous areas, have the following IDs:

II 3 G Ex ec IIC T4 Gc
DEKRA 16ATEX0052 X
Ta: 0 °C - 55 °C

Serial number

The name plate of the CX8xxx Embedded PCs shows a consecutive serial number, a hardware version and a date of manufacture:



Key:

n:	Serial number, consecutive number
h:	Hardware version, ascending number
dd:	Production day
mm:	Production month
yyyy:	Year of production

11.3.2 FCC

FCC Approvals for the United States of America

FCC: Federal Communications Commission Radio Frequency Interference Statement

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

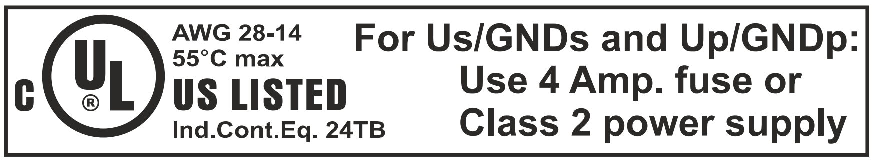
FCC Approval for Canada

FCC: Canadian Notice

This equipment does not exceed the Class A limits for radiated emissions as described in the Radio Interference Regulations of the Canadian Department of Communications.

11.3.3 UL

The UL-certified CX8xxx Embedded PCs have the following IDs:



c UL ® AWG 28-14 55°C max US LISTED Ind.Cont.Eq. 24TB

For Us/GNDs and Up/GNDp: Use 4 Amp. fuse or Class 2 power supply

Compliance with UL requirements:

Compliance with the following UL requirements is required, in order to guarantee the UL certification for the CX8xxx Embedded PC:

- The Embedded PCs must not be connected to unlimited voltage sources.
- Embedded PCs may only be supplied from a 24 V DV voltage source. The voltage source must be insulated and protected with a fuse of maximum 4 A (corresponding to UL248).
- Or the power supply must originate from a voltage source that corresponds to NEC class 2. An NEC class 2 voltage source must not be connected in series or parallel with another NEC class 2 voltage source.

11.4 CAN Identifier List

The list provided here should assist in identifying and assigning CANopen messages. All the identifiers allocated by the CANopen default identifier allocation are listed, as well as the manufacturer-specific default identifiers issued by BECKHOFF via object 0x5500 [▸ 86] (only to be used in networks with node addresses less than 64).

The following values can be used as search aids and "entry points" in the extensive identifier table in the *chm edition of the documentation:

Decimal: 400 500 600 700 800 900 1000 1100 1200 1300 1400 1500 1600 1700 1800 1900

Hexadecimal: 0x181 0x1C1 0x201 0x301 0x401 0x501 0x601 0x701

Identifier allocation via object 0x5500 follows this scheme:

Object	**Resulting COB ID (hex)**	**Resulting COB ID (dec)**
Emergency	0x81 - 0xBF [0xFF]	129 - 191 [255]
TxPDO1	0x181 - 0x1BF [0x1FF]	385 - 447 [511]
RxPDO1	0x201 - 0x23F [0x27F]	513 - 575 [639]
TxPDO2	0x281 - 0x2BF [0x2FF]	641 - 676 [767]
RxPDO2	0x301 - 0x33F [0x37F]	769 - 831 [895]
TxDPO3	0x381 - 0x3BF [0x3FF]	897 - 959 [1023]
RxPDO3	0x401 - 0x43F [0x47F]	1025 - 1087 [1151]
TxPDO4	0x481 - 0x4BF [0x4FF]	1153 - 1215 [1279]
RxPDO4	0x501 - 0x53F [0x57F]	1281- 1343 [1407]
TxPDO5	0x681 - 0x6BF	1665 - 1727
RxPDO5	0x781 - 0x7BF	1921- 1983
TxPDO6	0x1C1 - 0x1FF	449 - 511
RxPDO6	0x241 - 0x27F	577 - 639
TxDPO7	0x2C1 - 0x2FF	705 - 767
RxPDO7	0x341 - 0x37F	833 - 895
TxPDO8	0x3C1- 0x3FF	961 - 1023
RxPDO8	0x441 - 0x47F	1089 - 1151
TxPDO9	0x4C1 - 0x4FF	1217 - 1279
RxPDO9	0x541 - 0x57F	1345 - 1407
TxDPO10	0x5C1 - 0x5FF	1473 - 1535
RxPDO10	0x641 - 0x67F	1601- 1663
TxPDO11	0x6C1 - 0x6FF	1729 - 1791
RxPDO11	0x741 - 0x77F	1857 - 1919
SDO (Tx)	0x581 - 0x5BF [0x5FF]	1409 - 1471 [1535]

Object	Resulting COB ID (hex)	Resulting COB ID (dec)
SDO (Rx)	0x601 - 0x63F [0x67F]	1537 - 1599 [1663]
Guarding / Heartbeat/ Bootup	0x701 - 0x73F [0x77F]	1793 - 1855 [1919]

Identifier List

Identifiers marked with * are given manufacturer-specific assignments on the Bus Couplers after writing index 0x5500

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
0	0	NMT	874	36A	RxPDO7*, Nd.42	1430	596	SDO Tx Nd.22
128	80	SYNC	875	36B	RxPDO7*, Nd.43	1431	597	SDO Tx Nd.23
129	81	EMCY Nd.1	876	36C	RxPDO7*, Nd.44	1432	598	SDO Tx Nd.24
130	82	EMCY Nd.2	877	36D	RxPDO7*, Nd.45	1433	599	SDO Tx Nd.25
131	83	EMCY Nd.3	878	36E	RxPDO7*, Nd.46	1434	59A	SDO Tx Nd.26
132	84	EMCY Nd.4	879	36F	RxPDO7*, Nd.47	1435	59B	SDO Tx Nd.27
133	85	EMCY Nd.5	880	370	RxPDO7*, Nd.48	1436	59C	SDO Tx Nd.28
134	86	EMCY Nd.6	881	371	RxPDO7*, Nd.49	1437	59D	SDO Tx Nd.29
135	87	EMCY Nd.7	882	372	RxPDO7*, Nd.50	1438	59E	SDO Tx Nd.30
136	88	EMCY Nd.8	883	373	RxPDO7*, Nd.51	1439	59F	SDO Tx Nd.31
137	89	EMCY Nd.9	884	374	RxPDO7*, Nd.52	1440	5A0	SDO Tx Nd.32
138	8A	EMCY Nd.10	885	375	RxPDO7*, Nd.53	1441	5A1	SDO Tx Nd.33
139	8B	EMCY Nd.11	886	376	RxPDO7*, Nd.54	1442	5A2	SDO Tx Nd.34
140	8C	EMCY Nd.12	887	377	RxPDO7*, Nd.55	1443	5A3	SDO Tx Nd.35
141	8D	EMCY Nd.13	888	378	RxPDO7*, Nd.56	1444	5A4	SDO Tx Nd.36
142	8E	EMCY Nd.14	889	379	RxPDO7*, Nd.57	1445	5A5	SDO Tx Nd.37
143	8F	EMCY Nd.15	890	37A	RxPDO7*, Nd.58	1446	5A6	SDO Tx Nd.38
144	90	EMCY Nd.16	891	37B	RxPDO7*, Nd.59	1447	5A7	SDO Tx Nd.39
145	91	EMCY Nd.17	892	37C	RxPDO7*, Nd.60	1448	5A8	SDO Tx Nd.40
146	92	EMCY Nd.18	893	37D	RxPDO7*, Nd.61	1449	5A9	SDO Tx Nd.41
147	93	EMCY Nd.19	894	37E	RxPDO7*, Nd.62	1450	5AA	SDO Tx Nd.42
148	94	EMCY Nd.20	895	37F	RxPDO7*, Nd.63	1451	5AB	SDO Tx Nd.43
149	95	EMCY Nd.21	897	381	TxPDO3*,	1452	5AC	SDO Tx Nd.44
150	96	EMCY Nd.22	898	382	TxPDO3*, Nd.2	1453	5AD	SDO Tx Nd.45
151	97	EMCY Nd.23	899	383	TxPDO3*, Nd.3	1454	5AE	SDO Tx Nd.46
152	98	EMCY Nd.24	900	384	TxPDO3*, Nd.4	1455	5AF	SDO Tx Nd.47

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
153	99	EMCY Nd.25	901	385	TxPDO3*, Nd.5	1456	5B0	SDO Tx Nd.48
154	9A	EMCY Nd.26	902	386	TxPDO3*, Nd.6	1457	5B1	SDO Tx Nd.49
155	9B	EMCY Nd.27	903	387	TxPDO3*, Nd.7	1458	5B2	SDO Tx Nd.50
156	9C	EMCY Nd.28	904	388	TxPDO3*, Nd.8	1459	5B3	SDO Tx Nd.51
157	9D	EMCY Nd.29	905	389	TxPDO3*, Nd.9	1460	5B4	SDO Tx Nd.52
158	9E	EMCY Nd.30	906	38A	TxPDO3*, Nd.10	1461	5B5	SDO Tx Nd.53
159	9F	EMCY Nd.31	907	38B	TxPDO3*, Nd.11	1462	5B6	SDO Tx Nd.54
160	A0	EMCY Nd.32	908	38C	TxPDO3*, Nd.12	1463	5B7	SDO Tx Nd.55
161	A1	EMCY Nd.33	909	38D	TxPDO3*, Nd.13	1464	5B8	SDO Tx Nd.56
162	A2	EMCY Nd.34	910	38E	TxPDO3*, Nd.14	1465	5B9	SDO Tx Nd.57
163	A3	EMCY Nd.35	911	38F	TxPDO3*, Nd.15	1466	5BA	SDO Tx Nd.58
164	A4	EMCY Nd.36	912	390	TxPDO3*, Nd.16	1467	5BB	SDO Tx Nd.59
165	A5	EMCY Nd.37	913	391	TxPDO3*, Nd.17	1468	5BC	SDO Tx Nd.60
166	A6	EMCY Nd.38	914	392	TxPDO3*, Nd.18	1469	5BD	SDO Tx Nd.61
167	A7	EMCY Nd.39	915	393	TxPDO3*, Nd.19	1470	5BE	SDO Tx Nd.62
168	A8	EMCY Nd.40	916	394	TxPDO3*, Nd.20	1471	5BF	SDO Tx Nd.63
169	A9	EMCY Nd.41	917	395	TxPDO3*, Nd.21	1473	5C1	TxPDO10
170	AA	EMCY Nd.42	918	396	TxPDO3*, Nd.22	1474	5C2	TxPDO10 *, Nd.2
171	AB	EMCY Nd.43	919	397	TxPDO3*, Nd.23	1475	5C3	TxPDO10 *, Nd.3
172	AC	EMCY Nd.44	920	398	TxPDO3*, Nd.24	1476	5C4	TxPDO10 *, Nd.4
173	AD	EMCY Nd.45	921	399	TxPDO3*, Nd.25	1477	5C5	TxPDO10 *, Nd.5
174	AE	EMCY Nd.46	922	39A	TxPDO3*, Nd.26	1478	5C6	TxPDO10 *, Nd.6
175	AF	EMCY Nd.47	923	39B	TxPDO3*, Nd.27	1479	5C7	TxPDO10 *, Nd.7
176	B0	EMCY Nd.48	924	39C	TxPDO3*, Nd.28	1480	5C8	TxPDO10 *, Nd.8
177	B1	EMCY Nd.49	925	39D	TxPDO3*, Nd.29	1481	5C9	TxPDO10 *, Nd.9
178	B2	EMCY Nd.50	926	39E	TxPDO3*, Nd.30	1482	5CA	TxPDO10 *, Nd.10

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
179	B3	EMCY Nd.51	927	39F	TxPDO3*, Nd.31	1483	5CB	TxPDO10 *, Nd.11
180	B4	EMCY Nd.52	928	3A0	TxPDO3*, Nd.32	1484	5CC	TxPDO10 *, Nd.12
181	B5	EMCY Nd.53	929	3A1	TxPDO3*, Nd.33	1485	5CD	TxPDO10 *, Nd.13
182	B6	EMCY Nd.54	930	3A2	TxPDO3*, Nd.34	1486	5CE	TxPDO10 *, Nd.14
183	B7	EMCY Nd.55	931	3A3	TxPDO3*, Nd.35	1487	5CF	TxPDO10 *, Nd.15
184	B8	EMCY Nd.56	932	3A4	TxPDO3*, Nd.36	1488	5D0	TxPDO10 *, Nd.16
185	B9	EMCY Nd.57	933	3A5	TxPDO3*, Nd.37	1489	5D1	TxPDO10 *, Nd.17
186	BA	EMCY Nd.58	934	3A6	TxPDO3*, Nd.38	1490	5D2	TxPDO10 *, Nd.18
187	BB	EMCY Nd.59	935	3A7	TxPDO3*, Nd.39	1491	5D3	TxPDO10 *, Nd.19
188	BC	EMCY Nd.60	936	3A8	TxPDO3*, Nd.40	1492	5D4	TxPDO10 *, Nd.20
189	BD	EMCY Nd.61	937	3A9	TxPDO3*, Nd.41	1493	5D5	TxPDO10 *, Nd.21
190	BE	EMCY Nd.62	938	3AA	TxPDO3*, Nd.42	1494	5D6	TxPDO10 *, Nd.22
191	BF	EMCY Nd.63	939	3AB	TxPDO3*, Nd.43	1495	5D7	TxPDO10 *, Nd.23
385	181	TxPDO1	940	3AC	TxPDO3*, Nd.44	1496	5D8	TxPDO10 *, Nd.24
386	182	TxPDO1, DI, Nd.2	941	3AD	TxPDO3*, Nd.45	1497	5D9	TxPDO10 *, Nd.25
387	183	TxPDO1, DI, Nd.3	942	3AE	TxPDO3*, Nd.46	1498	5DA	TxPDO10 *, Nd.26
388	184	TxPDO1, DI, Nd.4	943	3AF	TxPDO3*, Nd.47	1499	5DB	TxPDO10 *, Nd.27
389	185	TxPDO1, DI, Nd.5	944	3B0	TxPDO3*, Nd.48	1500	5DC	TxPDO10 *, Nd.28
390	186	TxPDO1, DI, Nd.6	945	3B1	TxPDO3*, Nd.49	1501	5DD	TxPDO10 *, Nd.29
391	187	TxPDO1, DI, Nd.7	946	3B2	TxPDO3*, Nd.50	1502	5DE	TxPDO10 *, Nd.30
392	188	TxPDO1, DI, Nd.8	947	3B3	TxPDO3*, Nd.51	1503	5DF	TxPDO10 *, Nd.31
393	189	TxPDO1, DI, Nd.9	948	3B4	TxPDO3*, Nd.52	1504	5E0	TxPDO10 *, Nd.32
394	18A	TxPDO1, DI, Nd.10	949	3B5	TxPDO3*, Nd.53	1505	5E1	TxPDO10 *, Nd.33
395	18B	TxPDO1, DI, Nd.11	950	3B6	TxPDO3*, Nd.54	1506	5E2	TxPDO10 *, Nd.34
396	18C	TxPDO1, DI, Nd.12	951	3B7	TxPDO3*, Nd.55	1507	5E3	TxPDO10 *, Nd.35
397	18D	TxPDO1, DI, Nd.13	952	3B8	TxPDO3*, Nd.56	1508	5E4	TxPDO10 *, Nd.36

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
398	18E	TxPDO1, DI, Nd.14	953	3B9	TxPDO3*, Nd.57	1509	5E5	TxPDO10 *, Nd.37
399	18F	TxPDO1, DI, Nd.15	954	3BA	TxPDO3*, Nd.58	1510	5E6	TxPDO10 *, Nd.38
400	190	TxPDO1, DI, Nd.16	955	3BB	TxPDO3*, Nd.59	1511	5E7	TxPDO10 *, Nd.39
401	191	TxPDO1, DI, Nd.17	956	3BC	TxPDO3*, Nd.60	1512	5E8	TxPDO10 *, Nd.40
402	192	TxPDO1, DI, Nd.18	957	3BD	TxPDO3*, Nd.61	1513	5E9	TxPDO10 *, Nd.41
403	193	TxPDO1, DI, Nd.19	958	3BE	TxPDO3*, Nd.62	1514	5EA	TxPDO10 *, Nd.42
404	194	TxPDO1, DI, Nd.20	959	3BF	TxPDO3*, Nd.63	1515	5EB	TxPDO10 *, Nd.43
405	195	TxPDO1, DI, Nd.21	961	3C1	TxPDO8	1516	5EC	TxPDO10 *, Nd.44
406	196	TxPDO1, DI, Nd.22	962	3C2	TxPDO8*, Nd.2	1517	5ED	TxPDO10 *, Nd.45
407	197	TxPDO1, DI, Nd.23	963	3C3	TxPDO8*, Nd.3	1518	5EE	TxPDO10 *, Nd.46
408	198	TxPDO1, DI, Nd.24	964	3C4	TxPDO8*, Nd.4	1519	5EF	TxPDO10 *, Nd.47
409	199	TxPDO1, DI, Nd.25	965	3C5	TxPDO8*, Nd.5	1520	5F0	TxPDO10 *, Nd.48
410	19A	TxPDO1, DI, Nd.26	966	3C6	TxPDO8*, Nd.6	1521	5F1	TxPDO10 *, Nd.49
411	19B	TxPDO1, DI, Nd.27	967	3C7	TxPDO8*, Nd.7	1522	5F2	TxPDO10 *, Nd.50
412	19C	TxPDO1, DI, Nd.28	968	3C8	TxPDO8*, Nd.8	1523	5F3	TxPDO10 *, Nd.51
413	19D	TxPDO1, DI, Nd.29	969	3C9	TxPDO8*, Nd.9	1524	5F4	TxPDO10 *, Nd.52
414	19E	TxPDO1, DI, Nd.30	970	3CA	TxPDO8*, Nd.10	1525	5F5	TxPDO10 *, Nd.53
415	19F	TxPDO1, DI, Nd.31	971	3CB	TxPDO8*, Nd.11	1526	5F6	TxPDO10 *, Nd.54
416	1A0	TxPDO1, DI, Nd.32	972	3CC	TxPDO8*, Nd.12	1527	5F7	TxPDO10 *, Nd.55
417	1A1	TxPDO1, DI, Nd.33	973	3CD	TxPDO8*, Nd.13	1528	5F8	TxPDO10 *, Nd.56
418	1A2	TxPDO1, DI, Nd.34	974	3CE	TxPDO8*, Nd.14	1529	5F9	TxPDO10 *, Nd.57
419	1A3	TxPDO1, DI, Nd.35	975	3CF	TxPDO8*, Nd.15	1530	5FA	TxPDO10 *, Nd.58
420	1A4	TxPDO1, DI, Nd.36	976	3D0	TxPDO8*, Nd.16	1531	5FB	TxPDO10 *, Nd.59
421	1A5	TxPDO1, DI, Nd.37	977	3D1	TxPDO8*, Nd.17	1532	5FC	TxPDO10 *, Nd.60
422	1A6	TxPDO1, DI, Nd.38	978	3D2	TxPDO8*, Nd.18	1533	5FD	TxPDO10 *, Nd.61
423	1A7	TxPDO1, DI, Nd.39	979	3D3	TxPDO8*, Nd.19	1534	5FE	TxPDO10 *, Nd.62

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
424	1A8	TxPDO1, DI, Nd.40	980	3D4	TxPDO8*, Nd.20	1535	5FF	TxPDO10*, Nd.63
425	1A9	TxPDO1, DI, Nd.41	981	3D5	TxPDO8*, Nd.21	1537	601	SDO Rx
426	1AA	TxPDO1, DI, Nd.42	982	3D6	TxPDO8*, Nd.22	1538	602	SDO Rx Nd.2
427	1AB	TxPDO1, DI, Nd.43	983	3D7	TxPDO8*, Nd.23	1539	603	SDO Rx Nd.3
428	1AC	TxPDO1, DI, Nd.44	984	3D8	TxPDO8*, Nd.24	1540	604	SDO Rx Nd.4
429	1AD	TxPDO1, DI, Nd.45	985	3D9	TxPDO8*, Nd.25	1541	605	SDO Rx Nd.5
430	1AE	TxPDO1, DI, Nd.46	986	3DA	TxPDO8*, Nd.26	1542	606	SDO Rx Nd.6
431	1AF	TxPDO1, DI, Nd.47	987	3DB	TxPDO8*, Nd.27	1543	607	SDO Rx Nd.7
432	1B0	TxPDO1, DI, Nd.48	988	3DC	TxPDO8*, Nd.28	1544	608	SDO Rx Nd.8
433	1B1	TxPDO1, DI, Nd.49	989	3DD	TxPDO8*, Nd.29	1545	609	SDO Rx Nd.9
434	1B2	TxPDO1, DI, Nd.50	990	3DE	TxPDO8*, Nd.30	1546	60A	SDO Rx Nd.10
435	1B3	TxPDO1, DI, Nd.51	991	3DF	TxPDO8*, Nd.31	1547	60B	SDO Rx Nd.11
436	1B4	TxPDO1, DI, Nd.52	992	3E0	TxPDO8*, Nd.32	1548	60C	SDO Rx Nd.12
437	1B5	TxPDO1, DI, Nd.53	993	3E1	TxPDO8*, Nd.33	1549	60D	SDO Rx Nd.13
438	1B6	TxPDO1, DI, Nd.54	994	3E2	TxPDO8*, Nd.34	1550	60E	SDO Rx Nd.14
439	1B7	TxPDO1, DI, Nd.55	995	3E3	TxPDO8*, Nd.35	1551	60F	SDO Rx Nd.15
440	1B8	TxPDO1, DI, Nd.56	996	3E4	TxPDO8*, Nd.36	1552	610	SDO Rx Nd.16
441	1B9	TxPDO1, DI, Nd.57	997	3E5	TxPDO8*, Nd.37	1553	611	SDO Rx Nd.17
442	1BA	TxPDO1, DI, Nd.58	998	3E6	TxPDO8*, Nd.38	1554	612	SDO Rx Nd.18
443	1BB	TxPDO1, DI, Nd.59	999	3E7	TxPDO8*, Nd.39	1555	613	SDO Rx Nd.19
444	1BC	TxPDO1, DI, Nd.60	1000	3E8	TxPDO8*, Nd.40	1556	614	SDO Rx Nd.20
445	1BD	TxPDO1, DI, Nd.61	1001	3E9	TxPDO8*, Nd.41	1557	615	SDO Rx Nd.21
446	1BE	TxPDO1, DI, Nd.62	1002	3EA	TxPDO8*, Nd.42	1558	616	SDO Rx Nd.22
447	1BF	TxPDO1, DI, Nd.63	1003	3EB	TxPDO8*, Nd.43	1559	617	SDO Rx Nd.23
449	1C1	TxPDO6	1004	3EC	TxPDO8*, Nd.44	1560	618	SDO Rx Nd.24
450	1C2	TxPDO6*, Nd.2	1005	3ED	TxPDO8*, Nd.45	1561	619	SDO Rx Nd.25

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
451	1C3	TxPDO6*, Nd.3	1006	3EE	TxPDO8*, Nd.46	1562	61A	SDO Rx Nd.26
452	1C4	TxPDO6*, Nd.4	1007	3EF	TxPDO8*, Nd.47	1563	61B	SDO Rx Nd.27
453	1C5	TxPDO6*, Nd.5	1008	3F0	TxPDO8*, Nd.48	1564	61C	SDO Rx Nd.28
454	1C6	TxPDO6*, Nd.6	1009	3F1	TxPDO8*, Nd.49	1565	61D	SDO Rx Nd.29
455	1C7	TxPDO6*, Nd.7	1010	3F2	TxPDO8*, Nd.50	1566	61E	SDO Rx Nd.30
456	1C8	TxPDO6*, Nd.8	1011	3F3	TxPDO8*, Nd.51	1567	61F	SDO Rx Nd.31
457	1C9	TxPDO6*, Nd.9	1012	3F4	TxPDO8*, Nd.52	1568	620	SDO Rx Nd.32
458	1CA	TxPDO6*, Nd.10	1013	3F5	TxPDO8*, Nd.53	1569	621	SDO Rx Nd.33
459	1CB	TxPDO6*, Nd.11	1014	3F6	TxPDO8*, Nd.54	1570	622	SDO Rx Nd.34
460	1CC	TxPDO6*, Nd.12	1015	3F7	TxPDO8*, Nd.55	1571	623	SDO Rx Nd.35
461	1CD	TxPDO6*, Nd.13	1016	3F8	TxPDO8*, Nd.56	1572	624	SDO Rx Nd.36
462	1CE	TxPDO6*, Nd.14	1017	3F9	TxPDO8*, Nd.57	1573	625	SDO Rx Nd.37
463	1CF	TxPDO6*, Nd.15	1018	3FA	TxPDO8*, Nd.58	1574	626	SDO Rx Nd.38
464	1D0	TxPDO6*, Nd.16	1019	3FB	TxPDO8*, Nd.59	1575	627	SDO Rx Nd.39
465	1D1	TxPDO6*, Nd.17	1020	3FC	TxPDO8*, Nd.60	1576	628	SDO Rx Nd.40
466	1D2	TxPDO6*, Nd.18	1021	3FD	TxPDO8*, Nd.61	1577	629	SDO Rx Nd.41
467	1D3	TxPDO6*, Nd.19	1022	3FE	TxPDO8*, Nd.62	1578	62A	SDO Rx Nd.42
468	1D4	TxPDO6*, Nd.20	1023	3FF	TxPDO8*, Nd.63	1579	62B	SDO Rx Nd.43
469	1D5	TxPDO6*, Nd.21	1025	401	RxPDO3	1580	62C	SDO Rx Nd.44
470	1D6	TxPDO6*, Nd.22	1026	402	RxPDO3*, Nd.2	1581	62D	SDO Rx Nd.45
471	1D7	TxPDO6*, Nd.23	1027	403	RxPDO3*, Nd.3	1582	62E	SDO Rx Nd.46
472	1D8	TxPDO6*, Nd.24	1028	404	RxPDO3*, Nd.4	1583	62F	SDO Rx Nd.47
473	1D9	TxPDO6*, Nd.25	1029	405	RxPDO3*, Nd.5	1584	630	SDO Rx Nd.48
474	1DA	TxPDO6*, Nd.26	1030	406	RxPDO3*, Nd.6	1585	631	SDO Rx Nd.49
475	1DB	TxPDO6*, Nd.27	1031	407	RxPDO3*, Nd.7	1586	632	SDO Rx Nd.50
476	1DC	TxPDO6*, Nd.28	1032	408	RxPDO3*, Nd.8	1587	633	SDO Rx Nd.51

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
477	1DD	TxPDO6*, Nd.29	1033	409	RxPDO3*, Nd.9	1588	634	SDO Rx Nd.52
478	1DE	TxPDO6*, Nd.30	1034	40A	RxPDO3*, Nd.10	1589	635	SDO Rx Nd.53
479	1DF	TxPDO6*, Nd.31	1035	40B	RxPDO3*, Nd.11	1590	636	SDO Rx Nd.54
480	1E0	TxPDO6*, Nd.32	1036	40C	RxPDO3*, Nd.12	1591	637	SDO Rx Nd.55
481	1E1	TxPDO6*, Nd.33	1037	40D	RxPDO3*, Nd.13	1592	638	SDO Rx Nd.56
482	1E2	TxPDO6*, Nd.34	1038	40E	RxPDO3*, Nd.14	1593	639	SDO Rx Nd.57
483	1E3	TxPDO6*, Nd.35	1039	40F	RxPDO3*, Nd.15	1594	63A	SDO Rx Nd.58
484	1E4	TxPDO6*, Nd.36	1040	410	RxPDO3*, Nd.16	1595	63B	SDO Rx Nd.59
485	1E5	TxPDO6*, Nd.37	1041	411	RxPDO3*, Nd.17	1596	63C	SDO Rx Nd.60
486	1E6	TxPDO6*, Nd.38	1042	412	RxPDO3*, Nd.18	1597	63D	SDO Rx Nd.61
487	1E7	TxPDO6*, Nd.39	1043	413	RxPDO3*, Nd.19	1598	63E	SDO Rx Nd.62
488	1E8	TxPDO6*, Nd.40	1044	414	RxPDO3*, Nd.20	1599	63F	SDO Rx Nd.63
489	1E9	TxPDO6*, Nd.41	1045	415	RxPDO3*, Nd.21	1601	641	RxPDO10
490	1EA	TxPDO6*, Nd.42	1046	416	RxPDO3*, Nd.22	1602	642	RxPDO10 *, Nd.2
491	1EB	TxPDO6*, Nd.43	1047	417	RxPDO3*, Nd.23	1603	643	RxPDO10 *, Nd.3
492	1EC	TxPDO6*, Nd.44	1048	418	RxPDO3*, Nd.24	1604	644	RxPDO10 *, Nd.4
493	1ED	TxPDO6*, Nd.45	1049	419	RxPDO3*, Nd.25	1605	645	RxPDO10 *, Nd.5
494	1EE	TxPDO6*, Nd.46	1050	41A	RxPDO3*, Nd.26	1606	646	RxPDO10 *, Nd.6
495	1EF	TxPDO6*, Nd.47	1051	41B	RxPDO3*, Nd.27	1607	647	RxPDO10 *, Nd.7
496	1F0	TxPDO6*, Nd.48	1052	41C	RxPDO3*, Nd.28	1608	648	RxPDO10 *, Nd.8
497	1F1	TxPDO6*, Nd.49	1053	41D	RxPDO3*, Nd.29	1609	649	RxPDO10 *, Nd.9
498	1F2	TxPDO6*, Nd.50	1054	41E	RxPDO3*, Nd.30	1610	64A	RxPDO10 *, Nd.10
499	1F3	TxPDO6*, Nd.51	1055	41F	RxPDO3*, Nd.31	1611	64B	RxPDO10 *, Nd.11
500	1F4	TxPDO6*, Nd.52	1056	420	RxPDO3*, Nd.32	1612	64C	RxPDO10 *, Nd.12
501	1F5	TxPDO6*, Nd.53	1057	421	RxPDO3*, Nd.33	1613	64D	RxPDO10 *, Nd.13
502	1F6	TxPDO6*, Nd.54	1058	422	RxPDO3*, Nd.34	1614	64E	RxPDO10 *, Nd.14

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
503	1F7	TxPDO6*, Nd.55	1059	423	RxPDO3*, Nd.35	1615	64F	RxPDO10 *, Nd.15
504	1F8	TxPDO6*, Nd.56	1060	424	RxPDO3*, Nd.36	1616	650	RxPDO10 *, Nd.16
505	1F9	TxPDO6*, Nd.57	1061	425	RxPDO3*, Nd.37	1617	651	RxPDO10 *, Nd.17
506	1FA	TxPDO6*, Nd.58	1062	426	RxPDO3*, Nd.38	1618	652	RxPDO10 *, Nd.18
507	1FB	TxPDO6*, Nd.59	1063	427	RxPDO3*, Nd.39	1619	653	RxPDO10 *, Nd.19
508	1FC	TxPDO6*, Nd.60	1064	428	RxPDO3*, Nd.40	1620	654	RxPDO10 *, Nd.20
509	1FD	TxPDO6*, Nd.61	1065	429	RxPDO3*, Nd.41	1621	655	RxPDO10 *, Nd.21
510	1FE	TxPDO6*, Nd.62	1066	42A	RxPDO3*, Nd.42	1622	656	RxPDO10 *, Nd.22
511	1FF	TxPDO6*, Nd.63	1067	42B	RxPDO3*, Nd.43	1623	657	RxPDO10 *, Nd.23
513	201	RxPDO1	1068	42C	RxPDO3*, Nd.44	1624	658	RxPDO10 *, Nd.24
514	202	RxPDO1, DO, Nd.2	1069	42D	RxPDO3*, Nd.45	1625	659	RxPDO10 *, Nd.25
515	203	RxPDO1, DO, Nd.3	1070	42E	RxPDO3*, Nd.46	1626	65A	RxPDO10 *, Nd.26
516	204	RxPDO1, DO, Nd.4	1071	42F	RxPDO3*, Nd.47	1627	65B	RxPDO10 *, Nd.27
517	205	RxPDO1, DO, Nd.5	1072	430	RxPDO3*, Nd.48	1628	65C	RxPDO10 *, Nd.28
518	206	RxPDO1, DO, Nd.6	1073	431	RxPDO3*, Nd.49	1629	65D	RxPDO10 *, Nd.29
519	207	RxPDO1, DO, Nd.7	1074	432	RxPDO3*, Nd.50	1630	65E	RxPDO10 *, Nd.30
520	208	RxPDO1, DO, Nd.8	1075	433	RxPDO3*, Nd.51	1631	65F	RxPDO10 *, Nd.31
521	209	RxPDO1, DO, Nd.9	1076	434	RxPDO3*, Nd.52	1632	660	RxPDO10 *, Nd.32
522	20A	RxPDO1, DO, Nd.10	1077	435	RxPDO3*, Nd.53	1633	661	RxPDO10 *, Nd.33
523	20B	RxPDO1, DO, Nd.11	1078	436	RxPDO3*, Nd.54	1634	662	RxPDO10 *, Nd.34
524	20C	RxPDO1, DO, Nd.12	1079	437	RxPDO3*, Nd.55	1635	663	RxPDO10 *, Nd.35
525	20D	RxPDO1, DO, Nd.13	1080	438	RxPDO3*, Nd.56	1636	664	RxPDO10 *, Nd.36
526	20E	RxPDO1, DO, Nd.14	1081	439	RxPDO3*, Nd.57	1637	665	RxPDO10 *, Nd.37
527	20F	RxPDO1, DO, Nd.15	1082	43A	RxPDO3*, Nd.58	1638	666	RxPDO10 *, Nd.38
528	210	RxPDO1, DO, Nd.16	1083	43B	RxPDO3*, Nd.59	1639	667	RxPDO10 *, Nd.39
529	211	RxPDO1, DO, Nd.17	1084	43C	RxPDO3*, Nd.60	1640	668	RxPDO10 *, Nd.40

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
530	212	RxPDO1, DO, Nd.18	1085	43D	RxPDO3*, Nd.61	1641	669	RxPDO10 *, Nd.41
531	213	RxPDO1, DO, Nd.19	1086	43E	RxPDO3*, Nd.62	1642	66A	RxPDO10 *, Nd.42
532	214	RxPDO1, DO, Nd.20	1087	43F	RxPDO3*, Nd.63	1643	66B	RxPDO10 *, Nd.43
533	215	RxPDO1, DO, Nd.21	1089	441	RxPDO8	1644	66C	RxPDO10 *, Nd.44
534	216	RxPDO1, DO, Nd.22	1090	442	RxPDO8*, Nd.2	1645	66D	RxPDO10 *, Nd.45
535	217	RxPDO1, DO, Nd.23	1091	443	RxPDO8*, Nd.3	1646	66E	RxPDO10 *, Nd.46
536	218	RxPDO1, DO, Nd.24	1092	444	RxPDO8*, Nd.4	1647	66F	RxPDO10 *, Nd.47
537	219	RxPDO1, DO, Nd.25	1093	445	RxPDO8*, Nd.5	1648	670	RxPDO10 *, Nd.48
538	21A	RxPDO1, DO, Nd.26	1094	446	RxPDO8*, Nd.6	1649	671	RxPDO10 *, Nd.49
539	21B	RxPDO1, DO, Nd.27	1095	447	RxPDO8*, Nd.7	1650	672	RxPDO10 *, Nd.50
540	21C	RxPDO1, DO, Nd.28	1096	448	RxPDO8*, Nd.8	1651	673	RxPDO10 *, Nd.51
541	21D	RxPDO1, DO, Nd.29	1097	449	RxPDO8*, Nd.9	1652	674	RxPDO10 *, Nd.52
542	21E	RxPDO1, DO, Nd.30	1098	44A	RxPDO8*, Nd.10	1653	675	RxPDO10 *, Nd.53
543	21F	RxPDO1, DO, Nd.31	1099	44B	RxPDO8*, Nd.11	1654	676	RxPDO10 *, Nd.54
544	220	RxPDO1, DO, Nd.32	1100	44C	RxPDO8*, Nd.12	1655	677	RxPDO10 *, Nd.55
545	221	RxPDO1, DO, Nd.33	1101	44D	RxPDO8*, Nd.13	1656	678	RxPDO10 *, Nd.56
546	222	RxPDO1, DO, Nd.34	1102	44E	RxPDO8*, Nd.14	1657	679	RxPDO10 *, Nd.57
547	223	RxPDO1, DO, Nd.35	1103	44F	RxPDO8*, Nd.15	1658	67A	RxPDO10 *, Nd.58
548	224	RxPDO1, DO, Nd.36	1104	450	RxPDO8*, Nd.16	1659	67B	RxPDO10 *, Nd.59
549	225	RxPDO1, DO, Nd.37	1105	451	RxPDO8*, Nd.17	1660	67C	RxPDO10 *, Nd.60
550	226	RxPDO1, DO, Nd.38	1106	452	RxPDO8*, Nd.18	1661	67D	RxPDO10 *, Nd.61
551	227	RxPDO1, DO, Nd.39	1107	453	RxPDO8*, Nd.19	1662	67E	RxPDO10 *, Nd.62
552	228	RxPDO1, DO, Nd.40	1108	454	RxPDO8*, Nd.20	1663	67F	RxPDO10 *, Nd.63
553	229	RxPDO1, DO, Nd.41	1109	455	RxPDO8*, Nd.21	1665	681	TxPDO5
554	22A	RxPDO1, DO, Nd.42	1110	456	RxPDO8*, Nd.22	1666	682	TxPDO5*, Nd.2
555	22B	RxPDO1, DO, Nd.43	1111	457	RxPDO8*, Nd.23	1667	683	TxPDO5*, Nd.3

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
556	22C	RxPDO1, DO, Nd.44	1112	458	RxPDO8*, Nd.24	1668	684	TxPDO5*, Nd.4
557	22D	RxPDO1, DO, Nd.45	1113	459	RxPDO8*, Nd.25	1669	685	TxPDO5*, Nd.5
558	22E	RxPDO1, DO, Nd.46	1114	45A	RxPDO8*, Nd.26	1670	686	TxPDO5*, Nd.6
559	22F	RxPDO1, DO, Nd.47	1115	45B	RxPDO8*, Nd.27	1671	687	TxPDO5*, Nd.7
560	230	RxPDO1, DO, Nd.48	1116	45C	RxPDO8*, Nd.28	1672	688	TxPDO5*, Nd.8
561	231	RxPDO1, DO, Nd.49	1117	45D	RxPDO8*, Nd.29	1673	689	TxPDO5*, Nd.9
562	232	RxPDO1, DO, Nd.50	1118	45E	RxPDO8*, Nd.30	1674	68A	TxPDO5*, Nd.10
563	233	RxPDO1, DO, Nd.51	1119	45F	RxPDO8*, Nd.31	1675	68B	TxPDO5*, Nd.11
564	234	RxPDO1, DO, Nd.52	1120	460	RxPDO8*, Nd.32	1676	68C	TxPDO5*, Nd.12
565	235	RxPDO1, DO, Nd.53	1121	461	RxPDO8*, Nd.33	1677	68D	TxPDO5*, Nd.13
566	236	RxPDO1, DO, Nd.54	1122	462	RxPDO8*, Nd.34	1678	68E	TxPDO5*, Nd.14
567	237	RxPDO1, DO, Nd.55	1123	463	RxPDO8*, Nd.35	1679	68F	TxPDO5*, Nd.15
568	238	RxPDO1, DO, Nd.56	1124	464	RxPDO8*, Nd.36	1680	690	TxPDO5*, Nd.16
569	239	RxPDO1, DO, Nd.57	1125	465	RxPDO8*, Nd.37	1681	691	TxPDO5*, Nd.17
570	23A	RxPDO1, DO, Nd.58	1126	466	RxPDO8*, Nd.38	1682	692	TxPDO5*, Nd.18
571	23B	RxPDO1, DO, Nd.59	1127	467	RxPDO8*, Nd.39	1683	693	TxPDO5*, Nd.19
572	23C	RxPDO1, DO, Nd.60	1128	468	RxPDO8*, Nd.40	1684	694	TxPDO5*, Nd.20
573	23D	RxPDO1, DO, Nd.61	1129	469	RxPDO8*, Nd.41	1685	695	TxPDO5*, Nd.21
574	23E	RxPDO1, DO, Nd.62	1130	46A	RxPDO8*, Nd.42	1686	696	TxPDO5*, Nd.22
575	23F	RxPDO1, DO, Nd.63	1131	46B	RxPDO8*, Nd.43	1687	697	TxPDO5*, Nd.23
577	241	RxPDO6	1132	46C	RxPDO8*, Nd.44	1688	698	TxPDO5*, Nd.24
578	242	RxPDO6*, Nd.2	1133	46D	RxPDO8*, Nd.45	1689	699	TxPDO5*, Nd.25
579	243	RxPDO6*, Nd.3	1134	46E	RxPDO8*, Nd.46	1690	69A	TxPDO5*, Nd.26
580	244	RxPDO6*, Nd.4	1135	46F	RxPDO8*, Nd.47	1691	69B	TxPDO5*, Nd.27
581	245	RxPDO6*, Nd.5	1136	470	RxPDO8*, Nd.48	1692	69C	TxPDO5*, Nd.28
582	246	RxPDO6*, Nd.6	1137	471	RxPDO8*, Nd.49	1693	69D	TxPDO5*, Nd.29

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
583	247	RxPDO6*, Nd.7	1138	472	RxPDO8*, Nd.50	1694	69E	TxPDO5*, Nd.30
584	248	RxPDO6*, Nd.8	1139	473	RxPDO8*, Nd.51	1695	69F	TxPDO5*, Nd.31
585	249	RxPDO6*, Nd.9	1140	474	RxPDO8*, Nd.52	1696	6A0	TxPDO5*, Nd.32
586	24A	RxPDO6*, Nd.10	1141	475	RxPDO8*, Nd.53	1697	6A1	TxPDO5*, Nd.33
587	24B	RxPDO6*, Nd.11	1142	476	RxPDO8*, Nd.54	1698	6A2	TxPDO5*, Nd.34
588	24C	RxPDO6*, Nd.12	1143	477	RxPDO8*, Nd.55	1699	6A3	TxPDO5*, Nd.35
589	24D	RxPDO6*, Nd.13	1144	478	RxPDO8*, Nd.56	1700	6A4	TxPDO5*, Nd.36
590	24E	RxPDO6*, Nd.14	1145	479	RxPDO8*, Nd.57	1701	6A5	TxPDO5*, Nd.37
591	24F	RxPDO6*, Nd.15	1146	47A	RxPDO8*, Nd.58	1702	6A6	TxPDO5*, Nd.38
592	250	RxPDO6*, Nd.16	1147	47B	RxPDO8*, Nd.59	1703	6A7	TxPDO5*, Nd.39
593	251	RxPDO6*, Nd.17	1148	47C	RxPDO8*, Nd.60	1704	6A8	TxPDO5*, Nd.40
594	252	RxPDO6*, Nd.18	1149	47D	RxPDO8*, Nd.61	1705	6A9	TxPDO5*, Nd.41
595	253	RxPDO6*, Nd.19	1150	47E	RxPDO8*, Nd.62	1706	6AA	TxPDO5*, Nd.42
596	254	RxPDO6*, Nd.20	1151	47F	RxPDO8*, Nd.63	1707	6AB	TxPDO5*, Nd.43
597	255	RxPDO6*, Nd.21	1153	481	TxPDO4	1708	6AC	TxPDO5*, Nd.44
598	256	RxPDO6*, Nd.22	1154	482	TxPDO4*, Nd.2	1709	6AD	TxPDO5*, Nd.45
599	257	RxPDO6*, Nd.23	1155	483	TxPDO4*, Nd.3	1710	6AE	TxPDO5*, Nd.46
600	258	RxPDO6*, Nd.24	1156	484	TxPDO4*, Nd.4	1711	6AF	TxPDO5*, Nd.47
601	259	RxPDO6*, Nd.25	1157	485	TxPDO4*, Nd.5	1712	6B0	TxPDO5*, Nd.48
602	25A	RxPDO6*, Nd.26	1158	486	TxPDO4*, Nd.6	1713	6B1	TxPDO5*, Nd.49
603	25B	RxPDO6*, Nd.27	1159	487	TxPDO4*, Nd.7	1714	6B2	TxPDO5*, Nd.50
604	25C	RxPDO6*, Nd.28	1160	488	TxPDO4*, Nd.8	1715	6B3	TxPDO5*, Nd.51
605	25D	RxPDO6*, Nd.29	1161	489	TxPDO4*, Nd.9	1716	6B4	TxPDO5*, Nd.52
606	25E	RxPDO6*, Nd.30	1162	48A	TxPDO4*, Nd.10	1717	6B5	TxPDO5*, Nd.53
607	25F	RxPDO6*, Nd.31	1163	48B	TxPDO4*, Nd.11	1718	6B6	TxPDO5*, Nd.54
608	260	RxPDO6*, Nd.32	1164	48C	TxPDO4*, Nd.12	1719	6B7	TxPDO5*, Nd.55

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
609	261	RxPDO6*, Nd.33	1165	48D	TxPDO4*, Nd.13	1720	6B8	TxPDO5*, Nd.56
610	262	RxPDO6*, Nd.34	1166	48E	TxPDO4*, Nd.14	1721	6B9	TxPDO5*, Nd.57
611	263	RxPDO6*, Nd.35	1167	48F	TxPDO4*, Nd.15	1722	6BA	TxPDO5*, Nd.58
612	264	RxPDO6*, Nd.36	1168	490	TxPDO4*, Nd.16	1723	6BB	TxPDO5*, Nd.59
613	265	RxPDO6*, Nd.37	1169	491	TxPDO4*, Nd.17	1724	6BC	TxPDO5*, Nd.60
614	266	RxPDO6*, Nd.38	1170	492	TxPDO4*, Nd.18	1725	6BD	TxPDO5*, Nd.61
615	267	RxPDO6*, Nd.39	1171	493	TxPDO4*, Nd.19	1726	6BE	TxPDO5*, Nd.62
616	268	RxPDO6*, Nd.40	1172	494	TxPDO4*, Nd.20	1727	6BF	TxPDO5*, Nd.63
617	269	RxPDO6*, Nd.41	1173	495	TxPDO4*, Nd.21	1729	6C1	TxPDO11
618	26A	RxPDO6*, Nd.42	1174	496	TxPDO4*, Nd.22	1730	6C2	TxPDO11 *, Nd.2
619	26B	RxPDO6*, Nd.43	1175	497	TxPDO4*, Nd.23	1731	6C3	TxPDO11 *, Nd.3
620	26C	RxPDO6*, Nd.44	1176	498	TxPDO4*, Nd.24	1732	6C4	TxPDO11 *, Nd.4
621	26D	RxPDO6*, Nd.45	1177	499	TxPDO4*, Nd.25	1733	6C5	TxPDO11 *, Nd.5
622	26E	RxPDO6*, Nd.46	1178	49A	TxPDO4*, Nd.26	1734	6C6	TxPDO11 *, Nd.6
623	26F	RxPDO6*, Nd.47	1179	49B	TxPDO4*, Nd.27	1735	6C7	TxPDO11 *, Nd.7
624	270	RxPDO6*, Nd.48	1180	49C	TxPDO4*, Nd.28	1736	6C8	TxPDO11 *, Nd.8
625	271	RxPDO6*, Nd.49	1181	49D	TxPDO4*, Nd.29	1737	6C9	TxPDO11 *, Nd.9
626	272	RxPDO6*, Nd.50	1182	49E	TxPDO4*, Nd.30	1738	6CA	TxPDO11 *, Nd.10
627	273	RxPDO6*, Nd.51	1183	49F	TxPDO4*, Nd.31	1739	6CB	TxPDO11 *, Nd.11
628	274	RxPDO6*, Nd.52	1184	4A0	TxPDO4*, Nd.32	1740	6CC	TxPDO11 *, Nd.12
629	275	RxPDO6*, Nd.53	1185	4A1	TxPDO4*, Nd.33	1741	6CD	TxPDO11 *, Nd.13
630	276	RxPDO6*, Nd.54	1186	4A2	TxPDO4*, Nd.34	1742	6CE	TxPDO11 *, Nd.14
631	277	RxPDO6*, Nd.55	1187	4A3	TxPDO4*, Nd.35	1743	6CF	TxPDO11 *, Nd.15
632	278	RxPDO6*, Nd.56	1188	4A4	TxPDO4*, Nd.36	1744	6D0	TxPDO11 *, Nd.16
633	279	RxPDO6*, Nd.57	1189	4A5	TxPDO4*, Nd.37	1745	6D1	TxPDO11 *, Nd.17
634	27A	RxPDO6*, Nd.58	1190	4A6	TxPDO4*, Nd.48	1746	6D2	TxPDO11 *, Nd.18

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
635	27B	RxPDO6*, Nd.59	1191	4A7	TxPDO4*, Nd.49	1747	6D3	TxPDO11 *, Nd.19
636	27C	RxPDO6*, Nd.60	1192	4A8	TxPDO4*, Nd.40	1748	6D4	TxPDO11 *, Nd.20
637	27D	RxPDO6*, Nd.61	1193	4A9	TxPDO4*, Nd.41	1749	6D5	TxPDO11 *, Nd.21
638	27E	RxPDO6*, Nd.62	1194	4AA	TxPDO4*, Nd.42	1750	6D6	TxPDO11 *, Nd.22
639	27F	RxPDO6*, Nd.63	1195	4AB	TxPDO4*, Nd.43	1751	6D7	TxPDO11 *, Nd.23
641	281	TxPDO2	1196	4AC	TxPDO4*, Nd.44	1752	6D8	TxPDO11 *, Nd.24
642	282	TxPDO2, AI, Nd.2	1197	4AD	TxPDO4*, Nd.45	1753	6D9	TxPDO11 *, Nd.25
643	283	TxPDO2, AI, Nd.3	1198	4AE	TxPDO4*, Nd.46	1754	6DA	TxPDO11 *, Nd.26
644	284	TxPDO2, AI, Nd.4	1199	4AF	TxPDO4*, Nd.47	1755	6DB	TxPDO11 *, Nd.27
645	285	TxPDO2, AI, Nd.5	1200	4B0	TxPDO4*, Nd.48	1756	6DC	TxPDO11 *, Nd.28
646	286	TxPDO2, AI, Nd.6	1201	4B1	TxPDO4*, Nd.49	1757	6DD	TxPDO11 *, Nd.29
647	287	TxPDO2, AI, Nd.7	1202	4B2	TxPDO4*, Nd.50	1758	6DE	TxPDO11 *, Nd.30
648	288	TxPDO2, AI, Nd.8	1203	4B3	TxPDO4*, Nd.51	1759	6DF	TxPDO11 *, Nd.31
649	289	TxPDO2, AI, Nd.9	1204	4B4	TxPDO4*, Nd.52	1760	6E0	TxPDO11 *, Nd.32
650	28A	TxPDO2, AI, Nd.10	1205	4B5	TxPDO4*, Nd.53	1761	6E1	TxPDO11 *, Nd.33
651	28B	TxPDO2, AI, Nd.11	1206	4B6	TxPDO4*, Nd.54	1762	6E2	TxPDO11 *, Nd.34
652	28C	TxPDO2, AI, Nd.12	1207	4B7	TxPDO4*, Nd.55	1763	6E3	TxPDO11 *, Nd.35
653	28D	TxPDO2, AI, Nd.13	1208	4B8	TxPDO4*, Nd.56	1764	6E4	TxPDO11 *, Nd.36
654	28E	TxPDO2, AI, Nd.14	1209	4B9	TxPDO4*, Nd.57	1765	6E5	TxPDO11 *, Nd.37
655	28F	TxPDO2, AI, Nd.15	1210	4BA	TxPDO4*, Nd.58	1766	6E6	TxPDO11 *, Nd.38
656	290	TxPDO2, AI, Nd.16	1211	4BB	TxPDO4*, Nd.59	1767	6E7	TxPDO11 *, Nd.39
657	291	TxPDO2, AI, Nd.17	1212	4BC	TxPDO4*, Nd.60	1768	6E8	TxPDO11 *, Nd.40
658	292	TxPDO2, AI, Nd.18	1213	4BD	TxPDO4*, Nd.61	1769	6E9	TxPDO11 *, Nd.41
659	293	TxPDO2, AI, Nd.19	1214	4BE	TxPDO4*, Nd.62	1770	6EA	TxPDO11 *, Nd.42
660	294	TxPDO2, AI, Nd.20	1215	4BF	TxPDO4*, Nd.63	1771	6EB	TxPDO11 *, Nd.43
661	295	TxPDO2, AI, Nd.21	1217	4C1	TxPDO9	1772	6EC	TxPDO11 *, Nd.44

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
662	296	TxPDO2, AI, Nd.22	1218	4C2	TxPDO9*, Nd.2	1773	6ED	TxPDO11 *, Nd.45
663	297	TxPDO2, AI, Nd.23	1219	4C3	TxPDO9*, Nd.3	1774	6EE	TxPDO11 *, Nd.46
664	298	TxPDO2, AI, Nd.24	1220	4C4	TxPDO9*, Nd.4	1775	6EF	TxPDO11 *, Nd.47
665	299	TxPDO2, AI, Nd.25	1221	4C5	TxPDO9*, Nd.5	1776	6F0	TxPDO11 *, Nd.48
666	29A	TxPDO2, AI, Nd.26	1222	4C6	TxPDO9*, Nd.6	1777	6F1	TxPDO11 *, Nd.49
667	29B	TxPDO2, AI, Nd.27	1223	4C7	TxPDO9*, Nd.7	1778	6F2	TxPDO11 *, Nd.50
668	29C	TxPDO2, AI, Nd.28	1224	4C8	TxPDO9*, Nd.8	1779	6F3	TxPDO11 *, Nd.51
669	29D	TxPDO2, AI, Nd.29	1225	4C9	TxPDO9*, Nd.9	1780	6F4	TxPDO11 *, Nd.52
670	29E	TxPDO2, AI, Nd.30	1226	4CA	TxPDO9*, Nd.10	1781	6F5	TxPDO11 *, Nd.53
671	29F	TxPDO2, AI, Nd.31	1227	4CB	TxPDO9*, Nd.11	1782	6F6	TxPDO11 *, Nd.54
672	2A0	TxPDO2, AI, Nd.32	1228	4CC	TxPDO9*, Nd.12	1783	6F7	TxPDO11 *, Nd.55
673	2A1	TxPDO2, AI, Nd.33	1229	4CD	TxPDO9*, Nd.13	1784	6F8	TxPDO11 *, Nd.56
674	2A2	TxPDO2, AI, Nd.34	1230	4CE	TxPDO9*, Nd.14	1785	6F9	TxPDO11 *, Nd.57
675	2A3	TxPDO2, AI, Nd.35	1231	4CF	TxPDO9*, Nd.15	1786	6FA	TxPDO11 *, Nd.58
676	2A4	TxPDO2, AI, Nd.36	1232	4D0	TxPDO9*, Nd.16	1787	6FB	TxPDO11 *, Nd.59
677	2A5	TxPDO2, AI, Nd.37	1233	4D1	TxPDO9*, Nd.17	1788	6FC	TxPDO11 *, Nd.60
678	2A6	TxPDO2, AI, Nd.38	1234	4D2	TxPDO9*, Nd.18	1789	6FD	TxPDO11 *, Nd.61
679	2A7	TxPDO2, AI, Nd.39	1235	4D3	TxPDO9*, Nd.19	1790	6FE	TxPDO11 *, Nd.62
680	2A8	TxPDO2, AI, Nd.40	1236	4D4	TxPDO9*, Nd.20	1791	6FF	TxPDO11 *, Nd.63
681	2A9	TxPDO2, AI, Nd.41	1237	4D5	TxPDO9*, Nd.21	1793	701	Guarding
682	2AA	TxPDO2, AI, Nd.42	1238	4D6	TxPDO9*, Nd.22	1794	702	Guarding Nd.2
683	2AB	TxPDO2, AI, Nd.43	1239	4D7	TxPDO9*, Nd.23	1795	703	Guarding Nd.3
684	2AC	TxPDO2, AI, Nd.44	1240	4D8	TxPDO9*, Nd.24	1796	704	Guarding Nd.4
685	2AD	TxPDO2, AI, Nd.45	1241	4D9	TxPDO9*, Nd.25	1797	705	Guarding Nd.5
686	2AE	TxPDO2, AI, Nd.46	1242	4DA	TxPDO9*, Nd.26	1798	706	Guarding Nd.6
687	2AF	TxPDO2, AI, Nd.47	1243	4DB	TxPDO9*, Nd.27	1799	707	Guarding Nd.7

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
688	2B0	TxPDO2, AI, Nd.48	1244	4DC	TxPDO9*, Nd.28	1800	708	Guarding Nd.8
689	2B1	TxPDO2, AI, Nd.49	1245	4DD	TxPDO9*, Nd.29	1801	709	Guarding Nd.9
690	2B2	TxPDO2, AI, Nd.50	1246	4DE	TxPDO9*, Nd.30	1802	70A	Guarding Nd.10
691	2B3	TxPDO2, AI, Nd.51	1247	4DF	TxPDO9*, Nd.31	1803	70B	Guarding Nd.11
692	2B4	TxPDO2, AI, Nd.52	1248	4E0	TxPDO9*, Nd.32	1804	70C	Guarding Nd.12
693	2B5	TxPDO2, AI, Nd.53	1249	4E1	TxPDO9*, Nd.33	1805	70D	Guarding Nd.13
694	2B6	TxPDO2, AI, Nd.54	1250	4E2	TxPDO9*, Nd.34	1806	70E	Guarding Nd.14
695	2B7	TxPDO2, AI, Nd.55	1251	4E3	TxPDO9*, Nd.35	1807	70F	Guarding Nd.15
696	2B8	TxPDO2, AI, Nd.56	1252	4E4	TxPDO9*, Nd.36	1808	710	Guarding Nd.16
697	2B9	TxPDO2, AI, Nd.57	1253	4E5	TxPDO9*, Nd.37	1809	711	Guarding Nd.17
698	2BA	TxPDO2, AI, Nd.58	1254	4E6	TxPDO9*, Nd.38	1810	712	Guarding Nd.18
699	2BB	TxPDO2, AI, Nd.59	1255	4E7	TxPDO9*, Nd.39	1811	713	Guarding Nd.19
700	2BC	TxPDO2, AI, Nd.60	1256	4E8	TxPDO9*, Nd.40	1812	714	Guarding Nd.20
701	2BD	TxPDO2, AI, Nd.61	1257	4E9	TxPDO9*, Nd.41	1813	715	Guarding Nd.21
702	2BE	TxPDO2, AI, Nd.62	1258	4EA	TxPDO9*, Nd.42	1814	716	Guarding Nd.22
703	2BF	TxPDO2, AI, Nd.63	1259	4EB	TxPDO9*, Nd.43	1815	717	Guarding Nd.23
705	2C1	TxPDO7	1260	4EC	TxPDO9*, Nd.44	1816	718	Guarding Nd.24
706	2C2	TxPDO7*, Nd.2	1261	4ED	TxPDO9*, Nd.45	1817	719	Guarding Nd.25
707	2C3	TxPDO7*, Nd.3	1262	4EE	TxPDO9*, Nd.46	1818	71A	Guarding Nd.26
708	2C4	TxPDO7*, Nd.4	1263	4EF	TxPDO9*, Nd.47	1819	71B	Guarding Nd.27
709	2C5	TxPDO7*, Nd.5	1264	4F0	TxPDO9*, Nd.48	1820	71C	Guarding Nd.28
710	2C6	TxPDO7*, Nd.6	1265	4F1	TxPDO9*, Nd.49	1821	71D	Guarding Nd.29
711	2C7	TxPDO7*, Nd.7	1266	4F2	TxPDO9*, Nd.50	1822	71E	Guarding Nd.30
712	2C8	TxPDO7*, Nd.8	1267	4F3	TxPDO9*, Nd.51	1823	71F	Guarding Nd.31
713	2C9	TxPDO7*, Nd.9	1268	4F4	TxPDO9*, Nd.52	1824	720	Guarding Nd.32
714	2CA	TxPDO7*, Nd.10	1269	4F5	TxPDO9*, Nd.53	1825	721	Guarding Nd.33

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
715	2CB	TxPDO7*, Nd.11	1270	4F6	TxPDO9*, Nd.54	1826	722	Guarding Nd.34
716	2CC	TxPDO7*, Nd.12	1271	4F7	TxPDO9*, Nd.55	1827	723	Guarding Nd.35
717	2CD	TxPDO7*, Nd.13	1272	4F8	TxPDO9*, Nd.56	1828	724	Guarding Nd.36
718	2CE	TxPDO7*, Nd.14	1273	4F9	TxPDO9*, Nd.57	1829	725	Guarding Nd.37
719	2CF	TxPDO7*, Nd.15	1274	4FA	TxPDO9*, Nd.58	1830	726	Guarding Nd.38
720	2D0	TxPDO7*, Nd.16	1275	4FB	TxPDO9*, Nd.59	1831	727	Guarding Nd.39
721	2D1	TxPDO7*, Nd.17	1276	4FC	TxPDO9*, Nd.60	1832	728	Guarding Nd.40
722	2D2	TxPDO7*, Nd.18	1277	4FD	TxPDO9*, Nd.61	1833	729	Guarding Nd.41
723	2D3	TxPDO7*, Nd.19	1278	4FE	TxPDO9*, Nd.62	1834	72A	Guarding Nd.42
724	2D4	TxPDO7*, Nd.20	1279	4FF	TxPDO9*, Nd.63	1835	72B	Guarding Nd.43
725	2D5	TxPDO7*, Nd.21	1281	501	RxPDO4	1836	72C	Guarding Nd.44
726	2D6	TxPDO7*, Nd.22	1282	502	RxPDO4*, Nd.2	1837	72D	Guarding Nd.45
727	2D7	TxPDO7*, Nd.23	1283	503	RxPDO4*, Nd.3	1838	72E	Guarding Nd.46
728	2D8	TxPDO7*, Nd.24	1284	504	RxPDO4*, Nd.4	1839	72F	Guarding Nd.47
729	2D9	TxPDO7*, Nd.25	1285	505	RxPDO4*, Nd.5	1840	730	Guarding Nd.48
730	2DA	TxPDO7*, Nd.26	1286	506	RxPDO4*, Nd.6	1841	731	Guarding Nd.49
731	2DB	TxPDO7*, Nd.27	1287	507	RxPDO4*, Nd.7	1842	732	Guarding Nd.50
732	2DC	TxPDO7*, Nd.28	1288	508	RxPDO4*, Nd.8	1843	733	Guarding Nd.51
733	2DD	TxPDO7*, Nd.29	1289	509	RxPDO4*, Nd.9	1844	734	Guarding Nd.52
734	2DE	TxPDO7*, Nd.30	1290	50A	RxPDO4*, Nd.10	1845	735	Guarding Nd.53
735	2DF	TxPDO7*, Nd.31	1291	50B	RxPDO4*, Nd.11	1846	736	Guarding Nd.54
736	2E0	TxPDO7*, Nd.32	1292	50C	RxPDO4*, Nd.12	1847	737	Guarding Nd.55
737	2E1	TxPDO7*, Nd.33	1293	50D	RxPDO4*, Nd.13	1848	738	Guarding Nd.56
738	2E2	TxPDO7*, Nd.34	1294	50E	RxPDO4*, Nd.14	1849	739	Guarding Nd.57
739	2E3	TxPDO7*, Nd.35	1295	50F	RxPDO4*, Nd.15	1850	73A	Guarding Nd.58
740	2E4	TxPDO7*, Nd.36	1296	510	RxPDO4*, Nd.16	1851	73B	Guarding Nd.59

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
741	2E5	TxPDO7*, Nd.37	1297	511	RxPDO4*, Nd.17	1852	73C	Guarding Nd.60
742	2E6	TxPDO7*, Nd.38	1298	512	RxPDO4*, Nd.18	1853	73D	Guarding Nd.61
743	2E7	TxPDO7*, Nd.39	1299	513	RxPDO4*, Nd.19	1854	73E	Guarding Nd.62
744	2E8	TxPDO7*, Nd.40	1300	514	RxPDO4*, Nd.20	1855	73F	Guarding Nd.63
745	2E9	TxPDO7*, Nd.41	1301	515	RxPDO4*, Nd.21	1857	741	RxPDO11
746	2EA	TxPDO7*, Nd.42	1302	516	RxPDO4*, Nd.22	1858	742	RxPDO11 *, Nd.2
747	2EB	TxPDO7*, Nd.43	1303	517	RxPDO4*, Nd.23	1859	743	RxPDO11 *, Nd.3
748	2EC	TxPDO7*, Nd.44	1304	518	RxPDO4*, Nd.24	1860	744	RxPDO11 *, Nd.4
749	2ED	TxPDO7*, Nd.45	1305	519	RxPDO4*, Nd.25	1861	745	RxPDO11 *, Nd.5
750	2EE	TxPDO7*, Nd.46	1306	51A	RxPDO4*, Nd.26	1862	746	RxPDO11 *, Nd.6
751	2EF	TxPDO7*, Nd.47	1307	51B	RxPDO4*, Nd.27	1863	747	RxPDO11 *, Nd.7
752	2F0	TxPDO7*, Nd.48	1308	51C	RxPDO4*, Nd.28	1864	748	RxPDO11 *, Nd.8
753	2F1	TxPDO7*, Nd.49	1309	51D	RxPDO4*, Nd.29	1865	749	RxPDO11 *, Nd.9
754	2F2	TxPDO7*, Nd.50	1310	51E	RxPDO4*, Nd.30	1866	74A	RxPDO11 *, Nd.10
755	2F3	TxPDO7*, Nd.51	1311	51F	RxPDO4*, Nd.31	1867	74B	RxPDO11 *, Nd.11
756	2F4	TxPDO7*, Nd.52	1312	520	RxPDO4*, Nd.32	1868	74C	RxPDO11 *, Nd.12
757	2F5	TxPDO7*, Nd.53	1313	521	RxPDO4*, Nd.33	1869	74D	RxPDO11 *, Nd.13
758	2F6	TxPDO7*, Nd.54	1314	522	RxPDO4*, Nd.34	1870	74E	RxPDO11 *, Nd.14
759	2F7	TxPDO7*, Nd.55	1315	523	RxPDO4*, Nd.35	1871	74F	RxPDO11 *, Nd.15
760	2F8	TxPDO7*, Nd.56	1316	524	RxPDO4*, Nd.36	1872	750	RxPDO11 *, Nd.16
761	2F9	TxPDO7*, Nd.57	1317	525	RxPDO4*, Nd.37	1873	751	RxPDO11 *, Nd.17
762	2FA	TxPDO7*, Nd.58	1318	526	RxPDO4*, Nd.38	1874	752	RxPDO11 *, Nd.18
763	2FB	TxPDO7*, Nd.59	1319	527	RxPDO4*, Nd.39	1875	753	RxPDO11 *, Nd.19
764	2FC	TxPDO7*, Nd.60	1320	528	RxPDO4*, Nd.40	1876	754	RxPDO11 *, Nd.20
765	2FD	TxPDO7*, Nd.61	1321	529	RxPDO4*, Nd.41	1877	755	RxPDO11 *, Nd.21
766	2FE	TxPDO7*, Nd.62	1322	52A	RxPDO4*, Nd.42	1878	756	RxPDO11 *, Nd.22

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
767	2FF	TxPDO7*, Nd.63	1323	52B	RxPDO4*, Nd.43	1879	757	RxPDO11 *, Nd.23
769	301	RxPDO2	1324	52C	RxPDO4*, Nd.44	1880	758	RxPDO11 *, Nd.24
770	302	RxPDO2, AO, Nd.2	1325	52D	RxPDO4*, Nd.45	1881	759	RxPDO11 *, Nd.25
771	303	RxPDO2, AO, Nd.3	1326	52E	RxPDO4*, Nd.46	1882	75A	RxPDO11 *, Nd.26
772	304	RxPDO2, AO, Nd.4	1327	52F	RxPDO4*, Nd.47	1883	75B	RxPDO11 *, Nd.27
773	305	RxPDO2, AO, Nd.5	1328	530	RxPDO4*, Nd.48	1884	75C	RxPDO11 *, Nd.28
774	306	RxPDO2, AO, Nd.6	1329	531	RxPDO4*, Nd.49	1885	75D	RxPDO11 *, Nd.29
775	307	RxPDO2, AO, Nd.7	1330	532	RxPDO4*, Nd.50	1886	75E	RxPDO11 *, Nd.30
776	308	RxPDO2, AO, Nd.8	1331	533	RxPDO4*, Nd.51	1887	75F	RxPDO11 *, Nd.31
777	309	RxPDO2, AO, Nd.9	1332	534	RxPDO4*, Nd.52	1888	760	RxPDO11 *, Nd.32
778	30A	RxPDO2, AO, Nd.10	1333	535	RxPDO4*, Nd.53	1889	761	RxPDO11 *, Nd.33
779	30B	RxPDO2, AO, Nd.11	1334	536	RxPDO4*, Nd.54	1890	762	RxPDO11 *, Nd.34
780	30C	RxPDO2, AO, Nd.12	1335	537	RxPDO4*, Nd.55	1891	763	RxPDO11 *, Nd.35
781	30D	RxPDO2, AO, Nd.13	1336	538	RxPDO4*, Nd.56	1892	764	RxPDO11 *, Nd.36
782	30E	RxPDO2, AO, Nd.14	1337	539	RxPDO4*, Nd.57	1893	765	RxPDO11 *, Nd.37
783	30F	RxPDO2, AO, Nd.15	1338	53A	RxPDO4*, Nd.58	1894	766	RxPDO11 *, Nd.38
784	310	RxPDO2, AO, Nd.16	1339	53B	RxPDO4*, Nd.59	1895	767	RxPDO11 *, Nd.39
785	311	RxPDO2, AO, Nd.17	1340	53C	RxPDO4*, Nd.60	1896	768	RxPDO11 *, Nd.40
786	312	RxPDO2, AO, Nd.18	1341	53D	RxPDO4*, Nd.61	1897	769	RxPDO11 *, Nd.41
787	313	RxPDO2, AO, Nd.19	1342	53E	RxPDO4*, Nd.62	1898	76A	RxPDO11 *, Nd.42
788	314	RxPDO2, AO, Nd.20	1343	53F	RxPDO4*, Nd.63	1899	76B	RxPDO11 *, Nd.43
789	315	RxPDO2, AO, Nd.21	1345	541	RxPDO9	1900	76C	RxPDO11 *, Nd.44
790	316	RxPDO2, AO, Nd.22	1346	542	RxPDO9*, Nd.2	1901	76D	RxPDO11 *, Nd.45
791	317	RxPDO2, AO, Nd.23	1347	543	RxPDO9*, Nd.3	1902	76E	RxPDO11 *, Nd.46
792	318	RxPDO2, AO, Nd.24	1348	544	RxPDO9*, Nd.4	1903	76F	RxPDO11 *, Nd.47
793	319	RxPDO2, AO, Nd.25	1349	545	RxPDO9*, Nd.5	1904	770	RxPDO11 *, Nd.48

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
794	31A	RxPDO2, AO, Nd.26	1350	546	RxPDO9*, Nd.6	1905	771	RxPDO11*, Nd.49
795	31B	RxPDO2, AO, Nd.27	1351	547	RxPDO9*, Nd.7	1906	772	RxPDO11*, Nd.50
796	31C	RxPDO2, AO, Nd.28	1352	548	RxPDO9*, Nd.8	1907	773	RxPDO11*, Nd.51
797	31D	RxPDO2, AO, Nd.29	1353	549	RxPDO9*, Nd.9	1908	774	RxPDO11*, Nd.52
798	31E	RxPDO2, AO, Nd.30	1354	54A	RxPDO9*, Nd.10	1909	775	RxPDO11*, Nd.53
799	31F	RxPDO2, AO, Nd.31	1355	54B	RxPDO9*, Nd.11	1910	776	RxPDO11*, Nd.54
800	320	RxPDO2, AO, Nd.32	1356	54C	RxPDO9*, Nd.12	1911	777	RxPDO11*, Nd.55
801	321	RxPDO2, AO, Nd.33	1357	54D	RxPDO9*, Nd.13	1912	778	RxPDO11*, Nd.56
802	322	RxPDO2, AO, Nd.34	1358	54E	RxPDO9*, Nd.14	1913	779	RxPDO11*, Nd.57
803	323	RxPDO2, AO, Nd.35	1359	54F	RxPDO9*, Nd.15	1914	77A	RxPDO11*, Nd.58
804	324	RxPDO2, AO, Nd.36	1360	550	RxPDO9*, Nd.16	1915	77B	RxPDO11*, Nd.59
805	325	RxPDO2, AO, Nd.37	1361	551	RxPDO9*, Nd.17	1916	77C	RxPDO11*, Nd.60
806	326	RxPDO2, AO, Nd.38	1362	552	RxPDO9*, Nd.18	1917	77D	RxPDO11*, Nd.61
807	327	RxPDO2, AO, Nd.39	1363	553	RxPDO9*, Nd.19	1918	77E	RxPDO11*, Nd.62
808	328	RxPDO2, AO, Nd.40	1364	554	RxPDO9*, Nd.20	1919	77F	RxPDO11*, Nd.63
809	329	RxPDO2, AO, Nd.41	1365	555	RxPDO9*, Nd.21	1921	781	RxPDO5
810	32A	RxPDO2, AO, Nd.42	1366	556	RxPDO9*, Nd.22	1922	782	RxPDO5*, Nd.2
811	32B	RxPDO2, AO, Nd.43	1367	557	RxPDO9*, Nd.23	1923	783	RxPDO5*, Nd.3
812	32C	RxPDO2, AO, Nd.44	1368	558	RxPDO9*, Nd.24	1924	784	RxPDO5*, Nd.4
813	32D	RxPDO2, AO, Nd.45	1369	559	RxPDO9*, Nd.25	1925	785	RxPDO5*, Nd.5
814	32E	RxPDO2, AO, Nd.46	1370	55A	RxPDO9*, Nd.26	1926	786	RxPDO5*, Nd.6
815	32F	RxPDO2, AO, Nd.47	1371	55B	RxPDO9*, Nd.27	1927	787	RxPDO5*, Nd.7
816	330	RxPDO2, AO, Nd.48	1372	55C	RxPDO9*, Nd.28	1928	788	RxPDO5*, Nd.8
817	331	RxPDO2, AO, Nd.49	1373	55D	RxPDO9*, Nd.29	1929	789	RxPDO5*, Nd.9
818	332	RxPDO2, AO, Nd.50	1374	55E	RxPDO9*, Nd.30	1930	78A	RxPDO5*, Nd.10
819	333	RxPDO2, AO, Nd.51	1375	55F	RxPDO9*, Nd.31	1931	78B	RxPDO5*, Nd.11

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
820	334	RxPDO2, AO, Nd.52	1376	560	RxPDO9*, Nd.32	1932	78C	RxPDO5*, Nd.12
821	335	RxPDO2, AO, Nd.53	1377	561	RxPDO9*, Nd.33	1933	78D	RxPDO5*, Nd.13
822	336	RxPDO2, AO, Nd.54	1378	562	RxPDO9*, Nd.34	1934	78E	RxPDO5*, Nd.14
823	337	RxPDO2, AO, Nd.55	1379	563	RxPDO9*, Nd.35	1935	78F	RxPDO5*, Nd.15
824	338	RxPDO2, AO, Nd.56	1380	564	RxPDO9*, Nd.36	1936	790	RxPDO5*, Nd.16
825	339	RxPDO2, AO, Nd.57	1381	565	RxPDO9*, Nd.37	1937	791	RxPDO5*, Nd.17
826	33A	RxPDO2, AO, Nd.58	1382	566	RxPDO9*, Nd.38	1938	792	RxPDO5*, Nd.18
827	33B	RxPDO2, AO, Nd.59	1383	567	RxPDO9*, Nd.39	1939	793	RxPDO5*, Nd.19
828	33C	RxPDO2, AO, Nd.60	1384	568	RxPDO9*, Nd.40	1940	794	RxPDO5*, Nd.20
829	33D	RxPDO2, AO, Nd.61	1385	569	RxPDO9*, Nd.41	1941	795	RxPDO5*, Nd.21
830	33E	RxPDO2, AO, Nd.62	1386	56A	RxPDO9*, Nd.42	1942	796	RxPDO5*, Nd.22
831	33F	RxPDO2, AO, Nd.63	1387	56B	RxPDO9*, Nd.43	1943	797	RxPDO5*, Nd.23
833	341	RxPDO7	1388	56C	RxPDO9*, Nd.44	1944	798	RxPDO5*, Nd.24
834	342	RxPDO7*, Nd.2	1389	56D	RxPDO9*, Nd.45	1945	799	RxPDO5*, Nd.25
835	343	RxPDO7*, Nd.3	1390	56E	RxPDO9*, Nd.46	1946	79A	RxPDO5*, Nd.26
836	344	RxPDO7*, Nd.4	1391	56F	RxPDO9*, Nd.47	1947	79B	RxPDO5*, Nd.27
837	345	RxPDO7*, Nd.5	1392	570	RxPDO9*, Nd.48	1948	79C	RxPDO5*, Nd.28
838	346	RxPDO7*, Nd.6	1393	571	RxPDO9*, Nd.49	1949	79D	RxPDO5*, Nd.29
839	347	RxPDO7*, Nd.7	1394	572	RxPDO9*, Nd.50	1950	79E	RxPDO5*, Nd.30
840	348	RxPDO7*, Nd.8	1395	573	RxPDO9*, Nd.51	1951	79F	RxPDO5*, Nd.31
841	349	RxPDO7*, Nd.9	1396	574	RxPDO9*, Nd.52	1952	7A0	RxPDO5*, Nd.32
842	34A	RxPDO7*, Nd.10	1397	575	RxPDO9*, Nd.53	1953	7A1	RxPDO5*, Nd.33
843	34B	RxPDO7*, Nd.11	1398	576	RxPDO9*, Nd.54	1954	7A2	RxPDO5*, Nd.34
844	34C	RxPDO7*, Nd.12	1399	577	RxPDO9*, Nd.55	1955	7A3	RxPDO5*, Nd.35
845	34D	RxPDO7*, Nd.13	1400	578	RxPDO9*, Nd.56	1956	7A4	RxPDO5*, Nd.36
846	34E	RxPDO7*, Nd.14	1401	579	RxPDO9*, Nd.57	1957	7A5	RxPDO5*, Nd.37

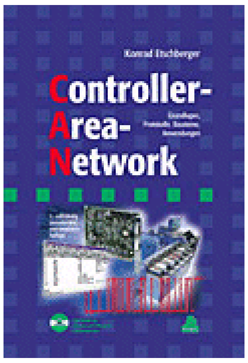
dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
847	34F	RxPDO7*, Nd.15	1402	57A	RxPDO9*, Nd.58	1958	7A6	RxPDO5*, Nd.38
848	350	RxPDO7*, Nd.16	1403	57B	RxPDO9*, Nd.59	1959	7A7	RxPDO5*, Nd.39
849	351	RxPDO7*, Nd.17	1404	57C	RxPDO9*, Nd.60	1960	7A8	RxPDO5*, Nd.40
850	352	RxPDO7*, Nd.18	1405	57D	RxPDO9*, Nd.61	1961	7A9	RxPDO5*, Nd.41
851	353	RxPDO7*, Nd.19	1406	57E	RxPDO9*, Nd.62	1962	7AA	RxPDO5*, Nd.42
852	354	RxPDO7*, Nd.20	1407	57F	RxPDO9*, Nd.63	1963	7AB	RxPDO5*, Nd.43
853	355	RxPDO7*, Nd.21	1409	581	SDO Tx	1964	7AC	RxPDO5*, Nd.44
854	356	RxPDO7*, Nd.22	1410	582	SDO Tx Nd.2	1965	7AD	RxPDO5*, Nd.45
855	357	RxPDO7*, Nd.23	1411	583	SDO Tx Nd.3	1966	7AE	RxPDO5*, Nd.46
856	358	RxPDO7*, Nd.24	1412	584	SDO Tx Nd.4	1967	7AF	RxPDO5*, Nd.47
857	359	RxPDO7*, Nd.25	1413	585	SDO Tx Nd.5	1968	7B0	RxPDO5*, Nd.48
858	35A	RxPDO7*, Nd.26	1414	586	SDO Tx Nd.6	1969	7B1	RxPDO5*, Nd.49
859	35B	RxPDO7*, Nd.27	1415	587	SDO Tx Nd.7	1970	7B2	RxPDO5*, Nd.50
860	35C	RxPDO7*, Nd.28	1416	588	SDO Tx Nd.8	1971	7B3	RxPDO5*, Nd.51
861	35D	RxPDO7*, Nd.29	1417	589	SDO Tx Nd.9	1972	7B4	RxPDO5*, Nd.52
862	35E	RxPDO7*, Nd.30	1418	58A	SDO Tx Nd.10	1973	7B5	RxPDO5*, Nd.53
863	35F	RxPDO7*, Nd.31	1419	58B	SDO Tx Nd.11	1974	7B6	RxPDO5*, Nd.54
864	360	RxPDO7*, Nd.32	1420	58C	SDO Tx Nd.12	1975	7B7	RxPDO5*, Nd.55
865	361	RxPDO7*, Nd.33	1421	58D	SDO Tx Nd.13	1976	7B8	RxPDO5*, Nd.56
866	362	RxPDO7*, Nd.34	1422	58E	SDO Tx Nd.14	1977	7B9	RxPDO5*, Nd.57
867	363	RxPDO7*, Nd.35	1423	58F	SDO Tx Nd.15	1978	7BA	RxPDO5*, Nd.58
868	364	RxPDO7*, Nd.36	1424	590	SDO Tx Nd.16	1979	7BB	RxPDO5*, Nd.59
869	365	RxPDO7*, Nd.37	1425	591	SDO Tx Nd.17	1980	7BC	RxPDO5*, Nd.60
870	366	RxPDO7*, Nd.38	1426	592	SDO Tx Nd.18	1981	7BD	RxPDO5*, Nd.61
871	367	RxPDO7*, Nd.39	1427	593	SDO Tx Nd.19	1982	7BE	RxPDO5*, Nd.62
872	368	RxPDO7*, Nd.40	1428	594	SDO Tx Nd.20	1983	7BF	RxPDO5*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
873	369	RxPDO7*, Nd.41	1429	595	SDO Tx Nd.21			

11.5 Bibliography

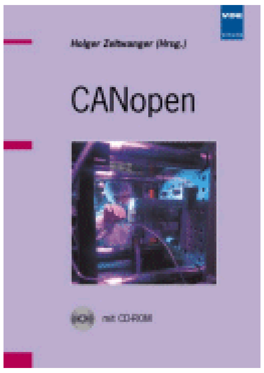
English books

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



German books

- Holger Zeltwander (Pub.):
  CANopen,
  VDE Verlag, 2001. 197 pages,
  ISBN 3-800-724480



- Konrad Etschberger:
  Controller Area Network, Grundlagen, Protokolle, Bausteine, Anwendungen. (Principles, protocols, components, applications.)
  Hanser Verlag, 2000. 431 pages.
  ISBN 3-446-19431-2

General fieldbus technology

- Gerhard Gruhler (Pub.):
  Feldbusse und Geräte-Kommunikationssysteme, Praktisches Know-How mit Vergleichsmöglichkeiten. (Fieldbus and Device Communication Systems, Practical Know-how with Comparative Resources)
  Franzis Verlag, 2001. 244 pages.
  ISBN 3-7723-5745-8

Standards

- ISO 11898:
  Road Vehicles - Interchange of digital information - Controller Area Network (CAN) for high speed communication.
- CiA DS 301:
  CANopen Application Layer and Communication Profile. Available from the CAN in Automation Association (http://www.can-cia.org).
- CiA DS 401:
  CANopen Device Profile for Generic I/O Modules. Available from the CAN in Automation Association (http://www.can-cia.org).

11.6 List of Abbreviations

CAN

Controller Area Network. A serial bus system standardized in ISO 11898. The technology on which CANopen is based.

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 initialise 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).

11.7 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.

Download finder

Our download finder contains all the files that we offer you for downloading. You will find application reports, technical documentation, technical drawings, configuration files and much more.

The downloads are available in various formats.

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 our internet page: 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
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
e-mail:	service@beckhoff.com

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

Phone:	+49 5246 963-0
e-mail:	info@beckhoff.com
web:	www.beckhoff.com

More Information:
www.beckhoff.com/CX8050

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
Phone: +49 5246 9630
info@beckhoff.com
www.beckhoff.com