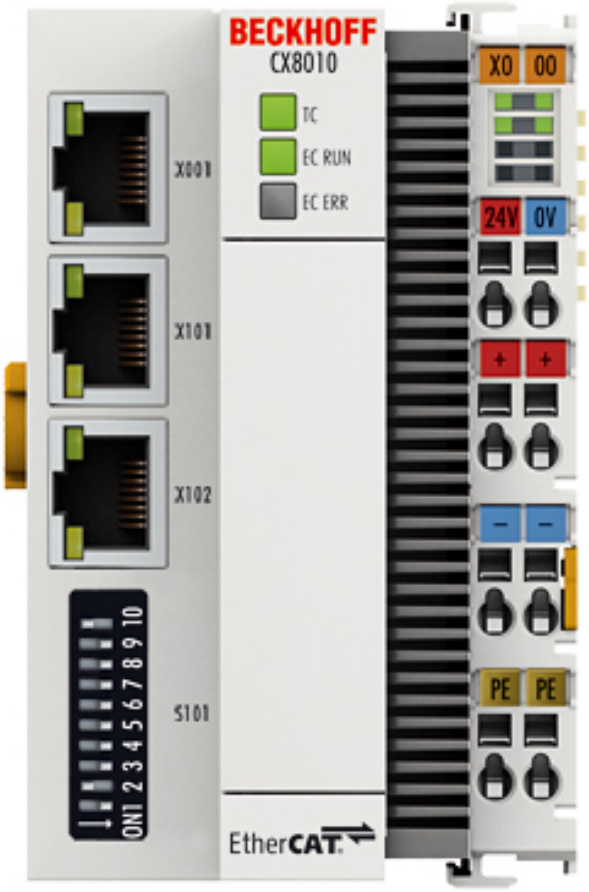



## Documentation for

# CX8010

## Embedded-PC with EtherCAT Slave Interface

EUROTEC
TECNOLOGIE IMPIANTISTICHE - AUTOMAZIONE - TELECONTROLLO

BECKHOFF

# Table of contents


**1 Notes on the documentation ... 5**
1.1 Explanation of symbols ... 6
1.2 Documentation issue status ... 7

**2 For your safety ... 8**
2.1 Intended use ... 8
2.2 Staff qualification ... 9
2.3 Safety instructions ... 9

**3 Transport and storage ... 11**

**4 Product overview ... 12**
4.1 CX80xx - System overview ... 12
4.2 CX8010 - Introduction ... 14
4.3 Technical data ... 15
4.4 Technical data - EtherCAT ... 16
4.5 CX80xx - MicroSD cards ... 17

**5 Mounting and wiring ... 18**
5.1 Mounting ... 18
5.1.1 Dimensions ... 18
5.1.2 Installation on mounting rails ... 18
5.2 Wiring ... 21
5.2.1 Power supply ... 21
5.2.2 Ethernet ... 23
5.3 Changing the battery ... 25

**6 Parameterization and commissioning ... 26**
6.1 DIP switch ... 26
6.2 Setting the IP address ... 27
6.2.1 IP address ... 27
6.2.2 Setting the address via DHCP server ... 27
6.2.3 Subnet mask ... 27
6.3 Configuration ... 29
6.3.1 CX80xx - Operating system ... 29
6.3.2 Power supply terminal ... 31
6.3.3 EtherCAT ... 33
6.3.4 Web Services ... 35
6.3.5 Real Time Clock (RTC) ... 38
6.3.6 1-second UPS (Uninterruptible Power Supply) ... 39
6.3.7 CPU load ... 40

**7 Programming ... 41**
7.1 Library for CX80xx ... 41
7.2 Seconds UPS ... 41
7.2.1 Function blocks ... 41
7.2.2 Data types ... 44
7.3 Diagnostics ... 45
7.3.1 FUNCTION F_CX80xx_ADDRESS ... 45

**8 Ethernet X001 Interface ... 46**
8.1 System introduction ... 46
8.1.1 Ethernet ... 46

8.1.2 Topology example ... 48
8.2 ModbusTCP ... 49
8.2.1 ModbusTCP Server - Overview ... 49
8.2.2 ModbusTCP Protocol ... 50
8.2.3 Mapping between Modbus and ADS ... 51
8.3 TCP/IP ... 53
8.4 ADS-Communication ... 54
**9 EtherCAT ... 56**
9.1 Limits of existing Ethernet communication approaches ... 56
9.2 New approach ... 56
9.3 System Properties ... 58
9.4 Application areas ... 62
9.5 Openness ... 63
9.6 Summary and Outlook ... 64
**10 Error handling diagnosis ... 65**
10.1 Diagnostic LEDs ... 65
**11 Appendix ... 68**
11.1 First steps ... 68
11.2 Image Update ... 73
11.3 Certification ... 75
11.3.1 Ex ... 75
11.3.2 FCC ... 76
11.3.3 UL ... 76
11.4 Support and Service ... 77

# 1 Notes on the documentation

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

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

### Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.
We reserve the right to revise and change the documentation at any time and without prior announcement.
No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

### Trademarks

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

### Patent Pending

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:
EP1590927, EP1789857, DE102004044764, DE102007017835
with corresponding applications or registrations in various other countries.

The TwinCAT Technology is covered, including but not limited to the following patent applications and patents:
EP0851348, US6167425 with corresponding applications or registrations in various other countries.

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

### Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.
The reproduction, distribution and utilization of this document as well as the communication of its contents to others without express authorization are prohibited.
Offenders will be held liable for the payment of damages. All rights reserved in the event of the grant of a patent, utility model or design.

## 1.1 Explanation of symbols

The following symbols with corresponding warnings or explanatory text are used in the documentation. Read and follow the warnings.

**Symbols that warn of personal injury:**

| | |
|---|---|
| DANGER | **Serious risk of injury**<br>Note this warning. Hazard with high risk of death or serious injury. |
| WARNING | **Risk of injury**<br>Note this warning. Hazard with medium risk of death or serious injury. |
| CAUTION | **Personal injuries**<br>Note this warning. Hazard with a low degree of risk, which could lead to minor or moderate injury. |

**Symbols that warn of damage to property or equipment:**

| | |
|---|---|
| Attention | **Damage to the devices or environment**<br>Note this warning. Risk of damage to the environment and equipment. |

**Symbols indicating further information or tips:**

| | |
|---|---|
| Note | **Tip or pointer**<br>This symbol indicates information that contributes to better understanding. |

## 1.2 Documentation issue status

| Version | Comment |
|---|---|
| 1.0 | • First release |
| 1.1 | • Foreword updated<br>• Chapter 1-second UPS added<br>• Chapter *Operating System* added |
| 1.2 | • Foreword reworked<br>• Chapter „For your safety“ added<br>• ATEX warnings added<br>• Technical data – EtherCAT reworked |
| 1.3 | • Chapter “Transport and storage“ added |
| 1.4 | • Chapter “FCC” added |
| 1.5 | • Warnings for Ex area revised<br>• Chapter “IECEx” added |

**CX8010 image version**

| Firmware | Description |
|---|---|
| Build 2244 | • Release (programming version TwinCAT 2.11 build 2245 or higher) |

# 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.
- Untrained personnel.
- Use of unauthorized replacement parts.

## 2.1 Intended use

The CX80xx Embedded PC is a control system designed for installation on a DIN rail in a control cabinet or terminal box.

The Embedded PC series is used in conjunction with Bus Terminals for recording digital or analog signals from sensors and transferring them to actuators or higher-level controllers.

The Embedded PC is designed for a working environment that meets the requirements of protection class IP 20. This involves finger protection and protection against solid foreign objects up to 12.5 mm in size, but not protection against water. Operation 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 CX80xx Embedded PC is only suitable for the following potentially explosive atmospheres:

1. For Zone 2 atmospheres in which gas occurs 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 occurs 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:

- The Embedded PC must 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 cabinets with suitable 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 Embedded 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**

| | |
|---|---|
| CAUTION | **Danger of explosion**<br>Gases or dusts can be ignited in potentially explosive areas. Read and follow the safety instructions to prevent deflagration 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 plug connectors, in order to prevent the plug connectors slipping out. Only use RJ45 connectors with an intact latch.

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**

- Limit physical and electronic access to all devices to an authorized group of persons.
- Change the default passwords to reduce the risk of unauthorized access. Regularly change the passwords.
- Install the devices behind 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**

- Use up-to-date security software. The safe function of the Embedded PC can be compromised by malicious software such as viruses or Trojans.
- The sensitivity of an Embedded 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 |
| --- |
| Documentation about IPC Security |

# 3 Transport and storage

### Transport

| | |
|---|---|
| **Attention** | **Short circuit due to moisture**<br>Moisture can form during transport in cold weather or in the event of large temperature fluctuations.<br>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.

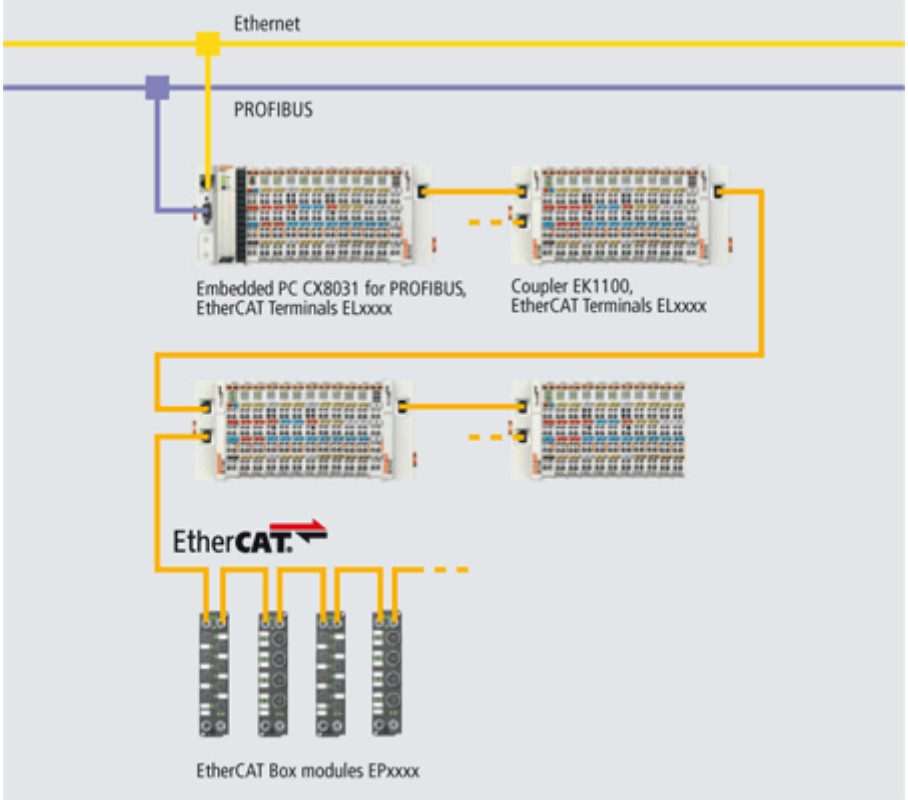
The devices from the CX80xx series represent a further development of the well-known and proven 16-bit microcontroller-based Bus Terminal Controllers from the BC and BX series including more efficient 32-bit processors. As with the BC/BX, it is also ensured in the case of the CX80xx that the control 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.




### 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

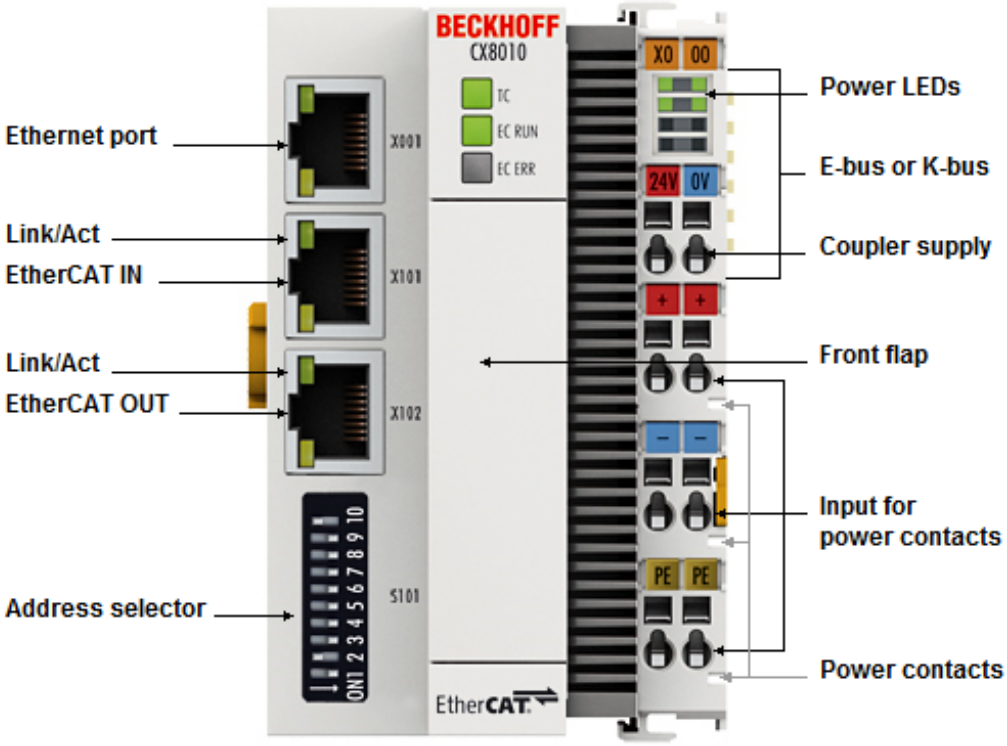
### Programming

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 CX8010 - Introduction




The CX8010 is a control system with EtherCAT slave interface. Alternatively K-bus or E-bus terminals can be series-connected; the CX8010 automatically detects which type of terminal is connected during the start-up phase. The controller is programmed with TwinCAT via the additional Ethernet interface.

In the basic version the CX80xx contains a 512 MB MicroSD card. The basic equipment includes two Ethernet interfaces as well as a K-bus or E-bus interface.

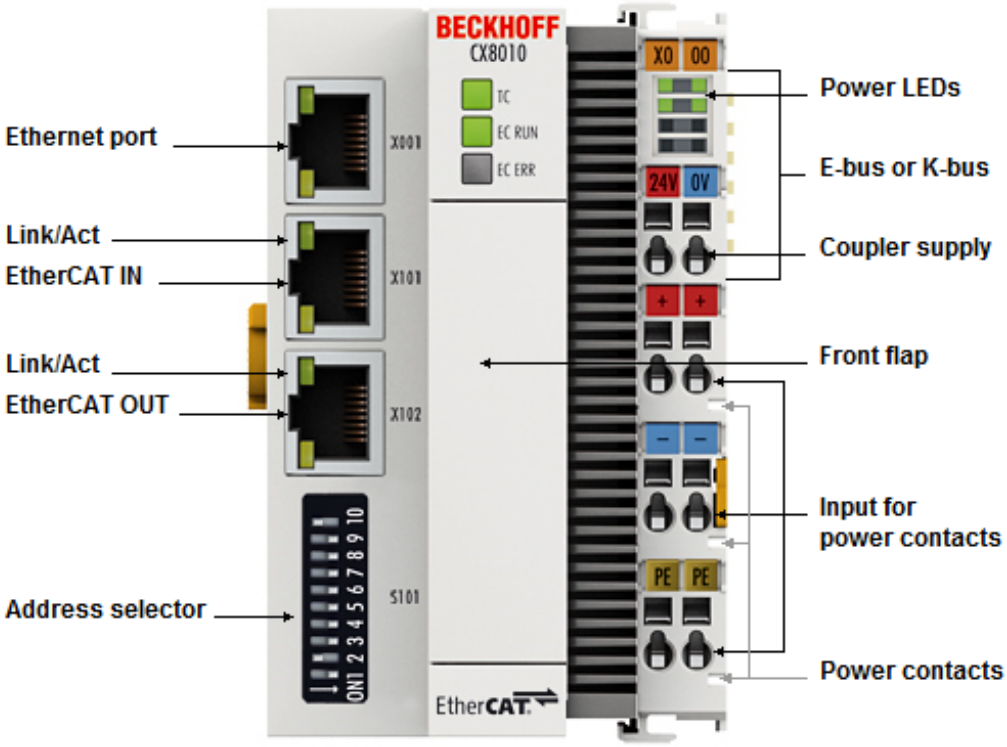
The smallest task time to be used is 1 ms (a task time of 1 to 50 ms is recommended for the I/O data, further tasks can also be set slower).
When using shorter cycle times, the total system load is to be observed. If too short a cycle time is selected, the Web visualization and remote desktop may operate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

## 4.3 Technical data

| Technical data | CX8010 |
|---|---|
| Processor | 32 bit, 400 MHz, ARM9 |
| Internal main memory | 64 MB RAM (internal, not extendable) |
| Operating system | Microsoft Windows CE 6.0 |
| Web-based Management | yes |
| Flash memory | MicroSD card (ATP) 512 MB (optionally 1, 2, 4, 8 GB) |
| Interfaces | 1 x USB device (behind the front flap)<br>1 x RJ45 Ethernet, 10/100 Mbit/s<br>2 x RJ45 switched, 100 Mbit/s |
| X001 protocols (non RT-Ethernet Port) | ADS, ModbusTCP, TCP/IP, UDP/IP |
| X101/X102 protocols | EtherCAT Slave Interface<br>max. 512 bytes of data in and out, max. 256 PDOs (variables) |
| Interface for I/O terminals | K-bus or E-bus, automatic recognition |
| Process data on the K-Bus | max. 2 KB input data<br>max. 2 KB output data |
| Diagnostics 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 |
| Configuration Software | TwinCAT PLC runtime (from version 2.11 R3 from build 2240) |
| Programming | TwinCAT PLC |
| Programming languages | IEC 61131-3 |
| Online Change | Yes |
| Up/download code | Yes/Yes |
| Power supply | 24 $V_{DC}$ (-15 %/+20 %) |
| UPS | 1-second UPS |
| Power supply for I/O terminals | max. 2 A |
| Max. power loss | 3.0 W (including system interfaces) |
| Dielectric strength | 500 V (supply / internal electronics) |
| Dimensions (W x H x D) | 64 mm x100 mm x 73 mm |
| Weight | approx. 180 g |
| Permissible ambient temperature during operation | 0 °C to +55 °C |
| Permissible ambient temperature during storage | -25 °C to +85 °C<br>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 class | IP20 |

| System data | Ethernet (CX8010) |
|---|---|
| Number of I/O modules | depending on controller |
| Number of I/O points | depending on controller |
| Transmission medium | 4 x 2 twisted pair copper cable; category 5 (100 Mbaud), shielded |
| Cable length | 100 m |
| Data transfer rate | 100 Mbaud |
| Topology | Star-form cabling, line topology |

## 4.4 Technical data - EtherCAT




| System data X001 | Ethernet (CX8010) |
|---|---|
| Transmission medium | 4 x 2 copper cable (twisted pair); category 5 (100 Mbaud) |
| Cable length | 100 m from switch to CX8010 |
| Data transfer rate | 10/100 Mbaud |
| Topology | star wiring |
| Protocols | all non-real-time-capable protocols that are based on TCP or UDP and require no real-time extension |

| System data X101/102 | Ethernet (CX8010) EtherCAT interface |
|---|---|
| Transmission medium | 4 x 2 twisted pair copper cable category 5 (100 Mbaud) |
| max. data length INPUT | 480 bytes (256 PDOs - variables) |
| max. data length OUTPUT | 480 bytes (256 PDOs - variables) |
| Cable length | 100 m |
| Data transfer rate | 100 Mbaud |
| Protocols (real-time) | EtherCAT |
| Protocols (non-real-time) | - |

## 4.5 CX80xx - MicroSD cards

| | |
|---|---|
| CAUTION | **MicroSD card as ignition source in potentially explosive atmospheres**<br>Gases or dusts can be ignited by a spark discharge when the MicroSD card is inserted or removed.<br>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: http://www.beckhoff.de/CX8000

# 5 Mounting and wiring

## 5.1 Mounting

| | |
|---|---|
| CAUTION | **Application in potentially explosive atmospheres**<br>The Embedded PC must be fitted with a suitable housing and suitable cables for use in potentially explosive atmospheres.<br>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.*

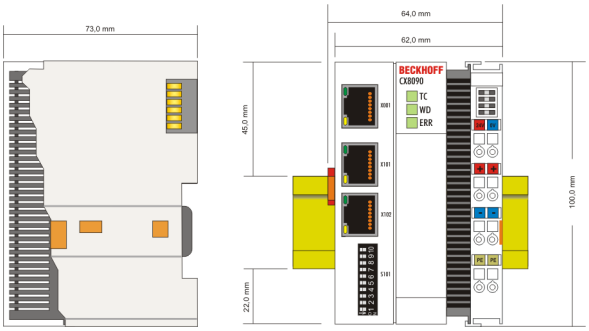
| Ex area | Flammable substance | Protection class |
|---|---|---|
| Zone 2 | Gas | IP 54, according to EN 60079-15 |
| Zone 22 | dust, non-conductive | IP 54, according to EN 60079-31 |
| | dust, conductive | IP 6x, according to EN 60079-31 |

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**

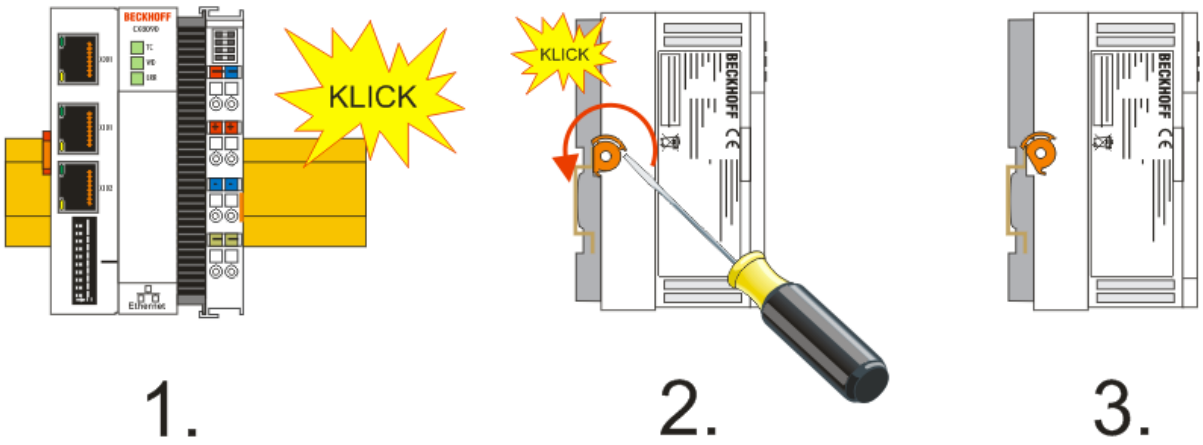



Drawings in various CAD formats can be found at: http://www.beckhoff.de/german/download/cx1000.htm

### 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.

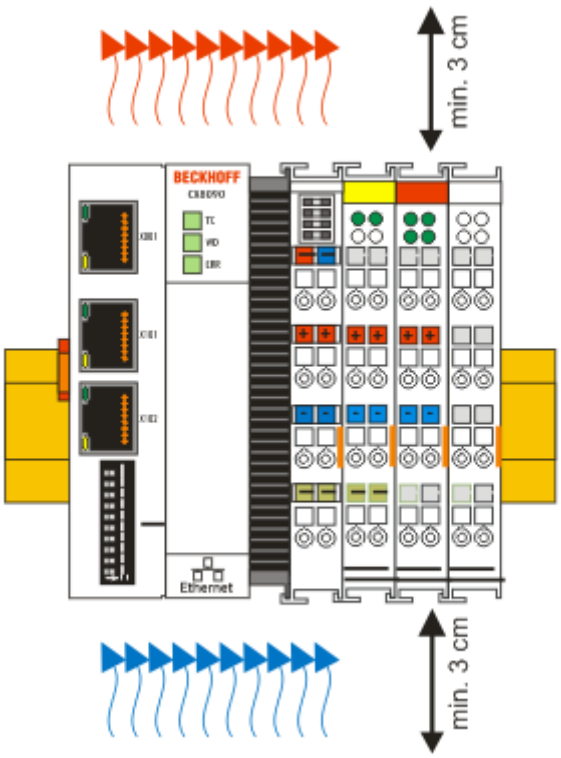



| | |
|---|---|
| **Attention** | **Avoid damage!**<br>Do not force the module or apply excessive pressure! |

**Permissible installation positions and minimum distances**

Installation positions

**Installation position up to 55 °C**

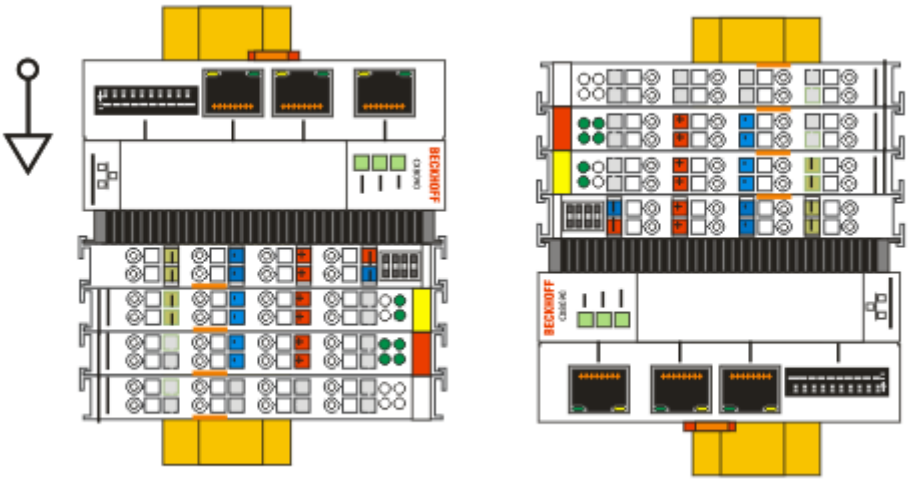


| | |
|---|---|
| **Attention** | **Comply with the permitted installation position and minimum distances!**<br>The maximum ambient temperature for CPU modules mounted on a DIN rail is 55°C. The orientation in which the device is fitted must be selected in such a way that cooling air can flow vertically 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

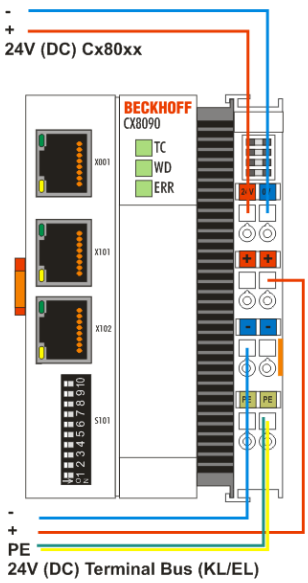
| | |
|---|---|
| WARNING | **Risk of injury through electric shock and damage to the device!**<br>Bring the CX80xx into a safe, de-energized state before starting assembly, disassembly or wiring! |

| | |
|---|---|
| CAUTION | **Connections as ignition source in potentially explosive atmospheres**<br>Gases or dusts can be ignited by a spark discharge when the Embedded PC is wired.<br>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**<br>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 (switched)

CX8010, CX809x: X101/102
EK9xxx: 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. transmission speed 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 (screened with copper braid)
Has a general screen of copper braid to reduce influence of external interference. This cable is recommended for use with Bus Couplers.

### FTP

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

### S/FTP

Screened/foiled-shielded twisted pair (screened with copper braid and aluminum foil)
Has a laminated aluminum screen 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 an outer screen, without defining the nature of the screen any more closely.

### S/STP

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

### ITP

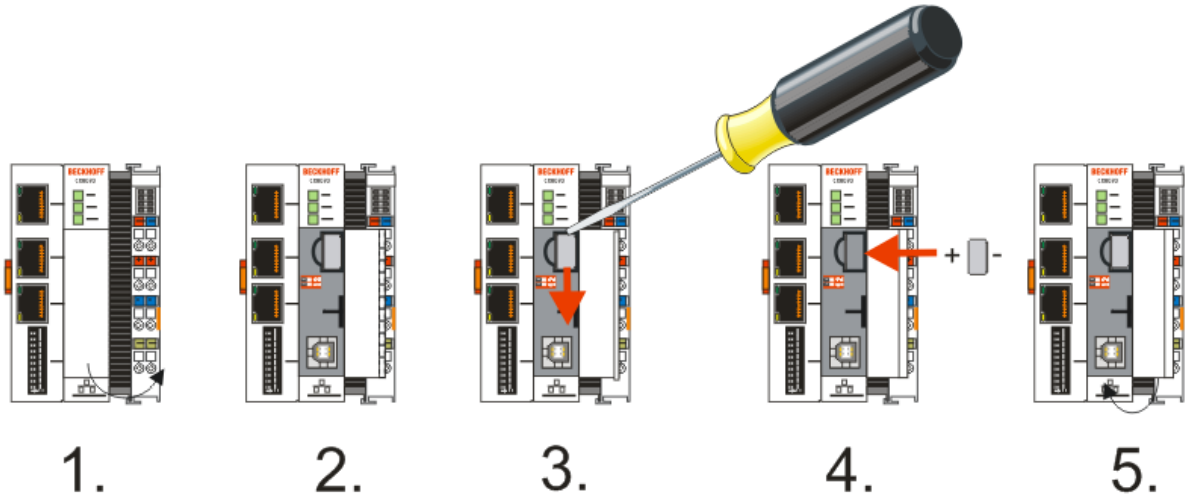
Industrial Twisted-Pair
The structure is similar to that of S/STP, but, in contrast to S/STP, it has only one pair of conductors.

## 5.3 Changing the battery

| | |
|---|---|
| CAUTION | **Battery as ignition source in potentially explosive atmospheres**<br>Gases or dusts can be ignited by a spark discharge when the battery is inserted or removed.<br>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. |

| | |
|---|---|
| Attention | **An incorrectly inserted battery may explode!**<br>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 |

| | |
|---|---|
| Note | **Battery maintenance**<br>The battery must be replaced every 5 years. Spare batteries can be ordered from Beckhoff Service. |

# 6 Parameterization and commissioning

## 6.1 DIP switch

| | |
|---|---|
| CAUTION | **DIP switches as ignition source in potentially explosive atmospheres**<br>Gases or dusts can be ignited by a spark discharge when DIP switches are used.<br>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. |

**10-pole DIP switch S101**

The DIP switch has only one meaning for the EtherCAT interfaces X101 and X102.

- left: OFF "0"
- right: ON "1".

| DIP switch S101 | Meaning |
|---|---|
| 9 | Reserved |
| 10 | Reserved |
| 1..8 | Hot-Connect address (see EtherCAT slave interface) |

**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 | reserved |

# 6.2 Setting the IP address

## 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.

Port 2 (X101 and X102) is to be configured by DIP switch (see DIP switch)

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.

**CX8090 example**

- 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 |

| | |
|---|---|
| Note | **Assignment of subnets, host numbers and IP addresses**<br>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.

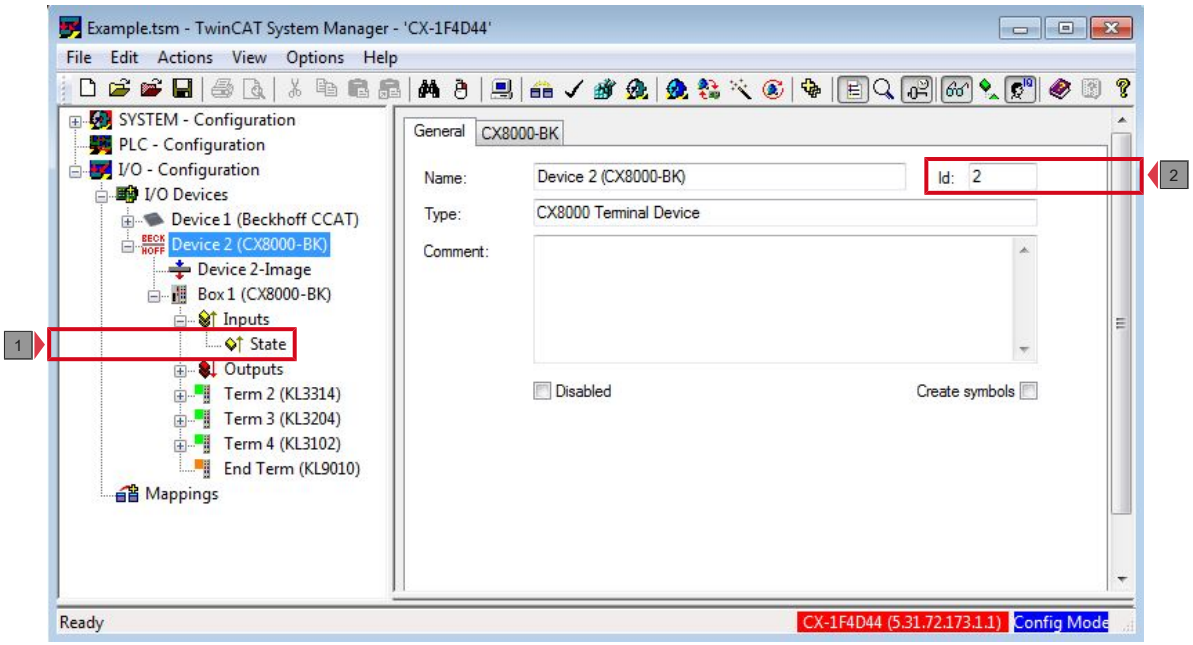



Fig. 1: K-Bus Interface

### 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)

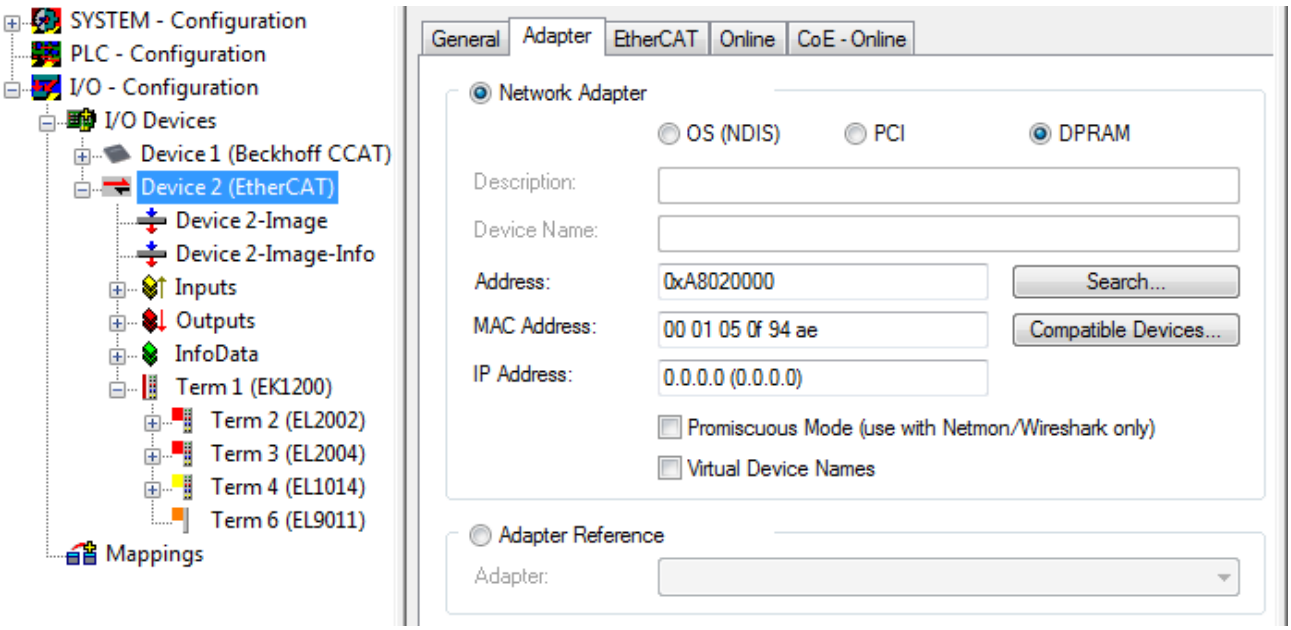
If there is a K-bus error, this can be reset via the IOF_DeviceReset function block (in the TcIoFunctions.lib).

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.




| | |
|---|---|
| **Note** | **DC Distributed Clocks**<br>The CX80xx series is not suitable for the use of EtherCAT slaves that use or need distributed clocks functionality. |

## 6.3.3 EtherCAT

### Slave Interface

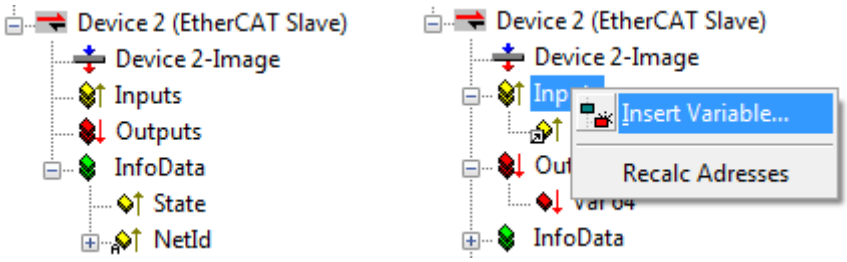
The incoming EtherCAT signal is connected to the interfaces X101. X102 goes to further EtherCAT Slave devices. The 10-pole DIP switch is intended exclusively for the “Hot Swap” (see Hot Swap). A maximum of 512 bytes of input and output data or 256 variables can be connected via the EtherCAT interface.

For example, the creation of 512 bytes cannot work, because 512 variables would be required for it.

### Process data

How is the process data created in TwinCAT?

The System Manager adds an “EtherCAT Slave” device as soon as a CX8010 is scanned. The interface has a very simple structure and usually consists only of “inputs”, “outputs” and InfoData. If you click on the inputs you can add data to the interface with the right mouse button. The system then automatically addresses them.




The same is done with the outputs. Now link them to your PLC program or to the K-bus and/or E-bus terminals. If you only link from terminals to the EtherCAT process image, make sure that both devices possess a task that they trigger and that the task is also automatically started.

InfoData: The current AMSNet ID of the interface is stored here. You can evaluate the state in order to determine the state of the EtherCAT slave interface.

### Process data with structures

In order to save a large number of links it is a good idea to use a data structure to save data that you wish to exchange. Note here, however, that an x86 system and an ARM processor will handle data structures with different variables in a different way. The ARM processor always places Word variables (2 bytes) at an even address and 4 DWORD variables (4 bytes) at an address that is divisible by 4.

### Example

Data structure

```
byTest :BYTE;

udTest:UDINT;
```

| ARM address | ARM variable | Address x86 | ARM variable |
|---|---|---|---|
| Byte Offset 0 | Byte | Byte Offset 0 | BYTE |
| Byte Offset 4 | UDINT | Byte Offset 1 | UDINT |
| Sum: 8 byte | | Sum 5 bytes | |

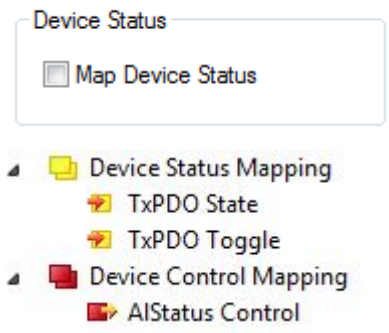
You can determine the length of a data structure on both systems using the command SIZEOF. If there is a difference here, this indicates that something is wrong with the data structure.

This problem can be solved by more skillful arrangement of the variables or by working with filler or dummy variables.

| ARM address | ARM variable | Address x86 | ARM variable |
|---|---|---|---|
| Byte Offset 0 | Byte | Byte Offset 0 | BYTE |
| | | Byte Offset 1 | BYTE (Dummy1) |
| | | Byte Offset 2 | BYTE (Dummy2) |
| | | Byte Offset 3 | BYTE (Dummy3) |
| Byte Offset 4 | UDINT | Byte Offset 4 | UDINT |
| Sum: 8 byte | | Sum 8 bytes | |

### Device status

The device status indicates whether the EtherCAT Slave interface is in OP mode and indicates the data communication via the toggle bit. The EtherCAT status can also be affected by the slave side in the output area.




TxPDO State “1”: node is in OP mode; “0”: node is no longer in OP mode

TxPDO Toggle “1/0”: node is in data exchange

AIStatus Control

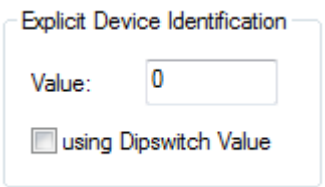
0x0000_0000 Normal mode (OP mode)
0x0001_0000 INIT mode
0x0002_0000 PREOP mode
0x0004_0000 SAFEOP mode
0x0008_0000 OP mode

All other bits are reserved!

### Hot Swap (Explicit Device Identification)

In applications in which a CX8010 is to be replaced by another CX8010 during operation (with different applications on the respective CX8010), the individual CX8010s can be distinguished from each other with *Explicit Device Identification*. These are sorted into a Hot-Connect group.

The *Explicit Device Identification Number* can be set using the DIP switch or by software in the System Manager. If it is to be used via the DIP switch, it must be activated once in the System Manager. When starting the system the CX8010 then leaves the switch on and operates with its setting.




### ADS Interface AMS NetId

Reserved for extensions.

**Advanced Settings**

Reserved for extensions.

**Distributed Clocks**

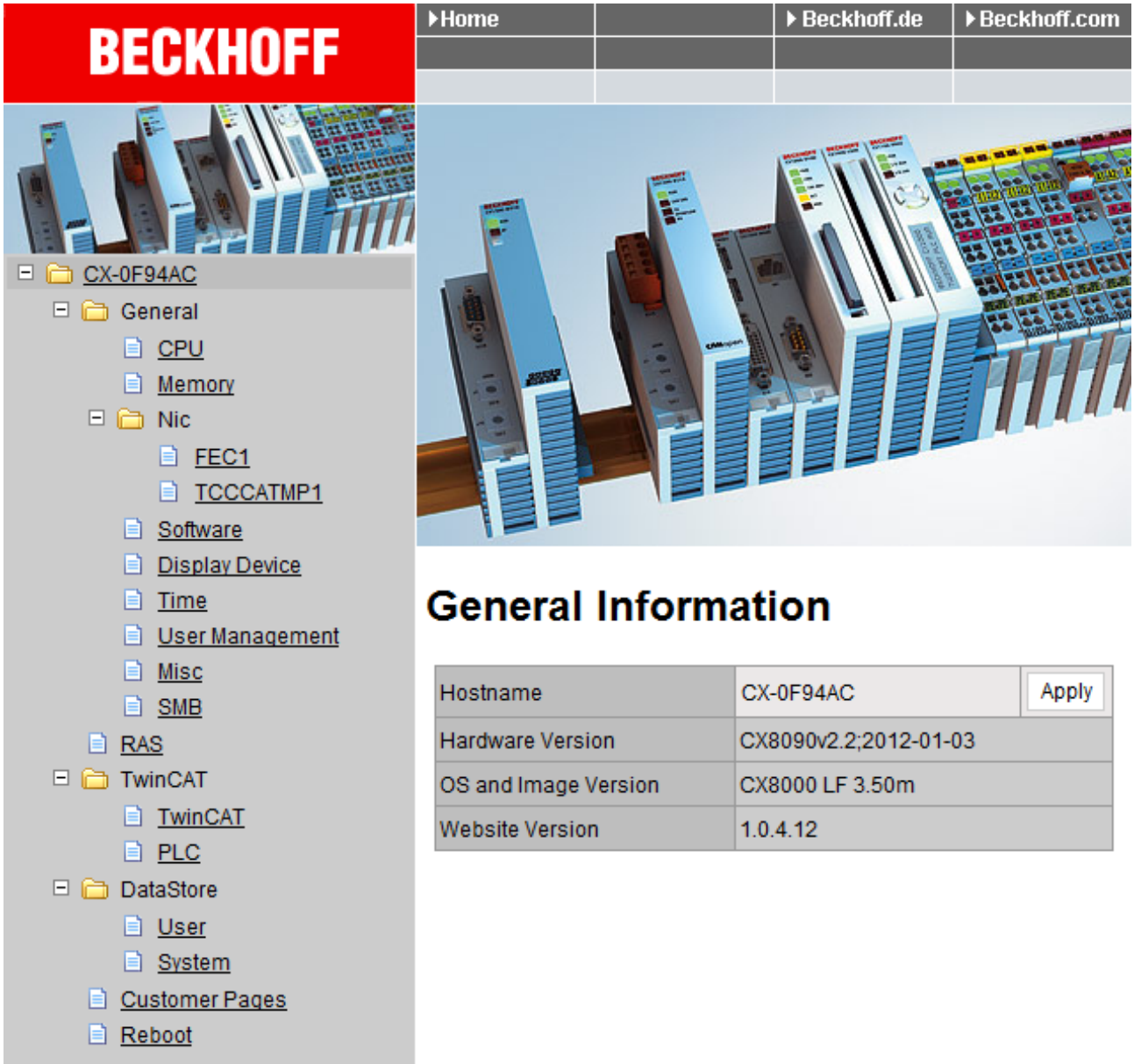
Not currently supported.

## 6.3.4 Web Services

**Upnp webpages**

There is a Upnp webpage on the CX80xx for diagnostics.

User name: guest
Password: 1



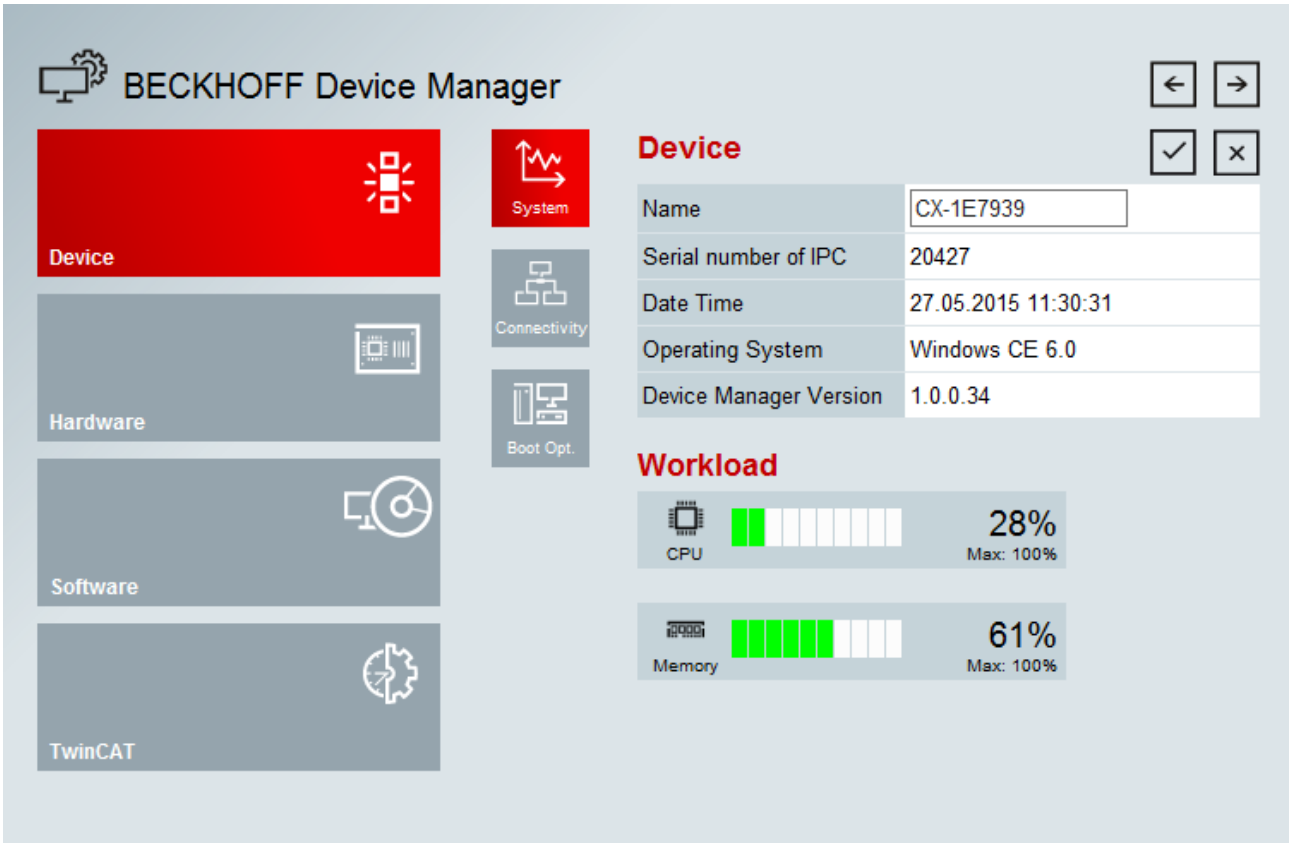

Enter the IP address or the device name.

**Example**

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

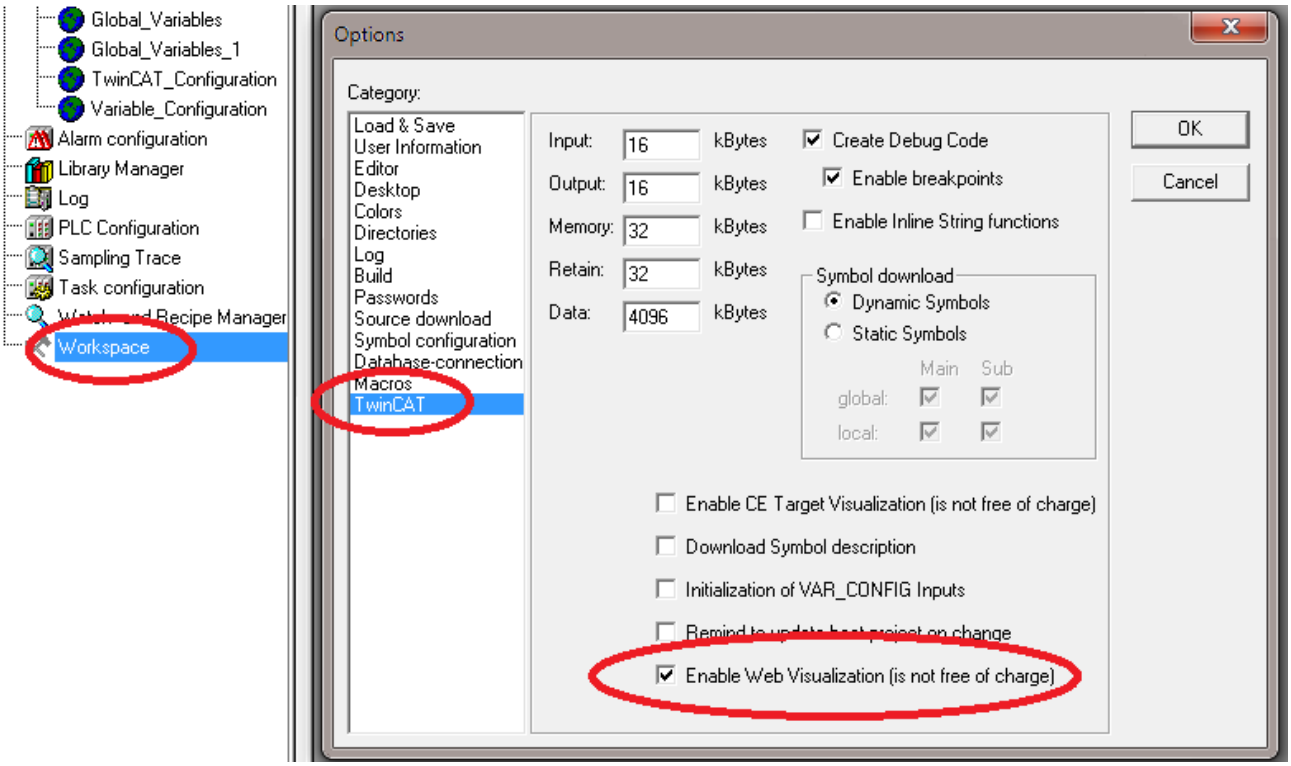
http://**172.16.17.55**/config

The diagnostic page was revised starting from image v354c.




### 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.



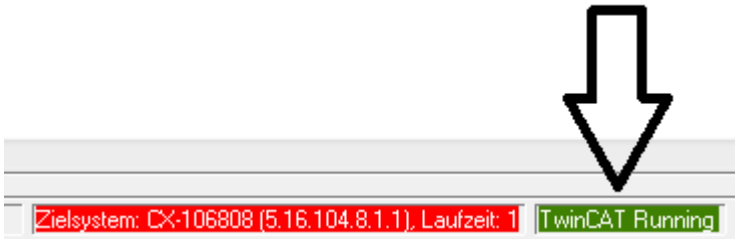
The call is made via the IP address or the device name in a web browser.
Further information can be taken from the documentation on the web visualization (see TwinCAT Supplements PLC HMI Web).

**Example**

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

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

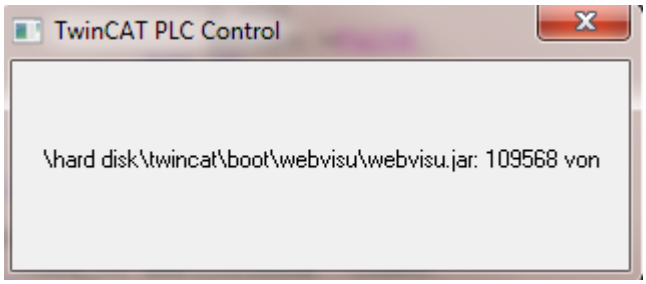
Ascertain before logging in (i.e. in the logged out condition) whether a ADS connection is established to the CX – "TwinCAT Running" in the bottom right-hand corner must be green. If that is not the case, please go onto Online/Selection of the target system again and call the CX once again.




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.




**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/cx8010_hw/Resources/zip/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.

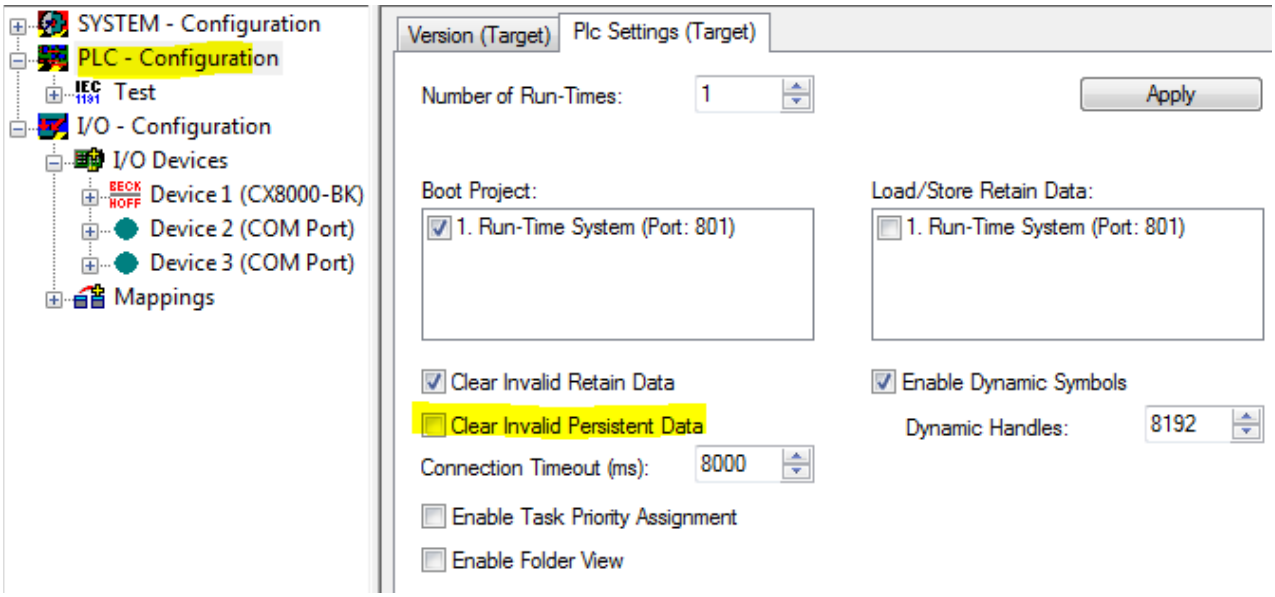
Saving of the persistent data only takes place in conjunction with the function block FB_S_UPS_CX80xx. This block must be called cyclically. We strongly recommend using the default values for the block.

### 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.




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
```

| | |
|---|---|
| **Note** | **Purpose of the 1-second UPS**<br>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.

| | |
|---|---|
| **Note** | **CPU load**<br>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/cx8010_hw/Resources/zip/1608565003.zip

## 7.2 Seconds UPS

### 7.2.1 Function blocks

**FUNCTION_BLOCK FB_S_UPS_CX80xx**

```
                    FB_S_UPS_CX80XX

sNetID : T_AmsNetId                 bPowerFailDetect : BOOL
iPLCPort : UINT                      eState : E_S_UPS_State
tTimeout : TIME
eUpsMode : E_S_UPS_Mode
ePersistentMode : E_PersistentMode
tRecoverTime : TIME
```

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.

| | |
|---|---|
| **Attention** | **Loss of data**<br>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.

| | |
|---|---|
| **Attention** | **Caution when using files:**<br>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 persis-
tent 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 persis-
tent 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 fail-
ure 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 QuickShut-
down)
```

### 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.

```
F_CX80XX_ADDRESS
iCX_Typ : INT F_CX80xx_Address : INT
```

### 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 |

# 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.

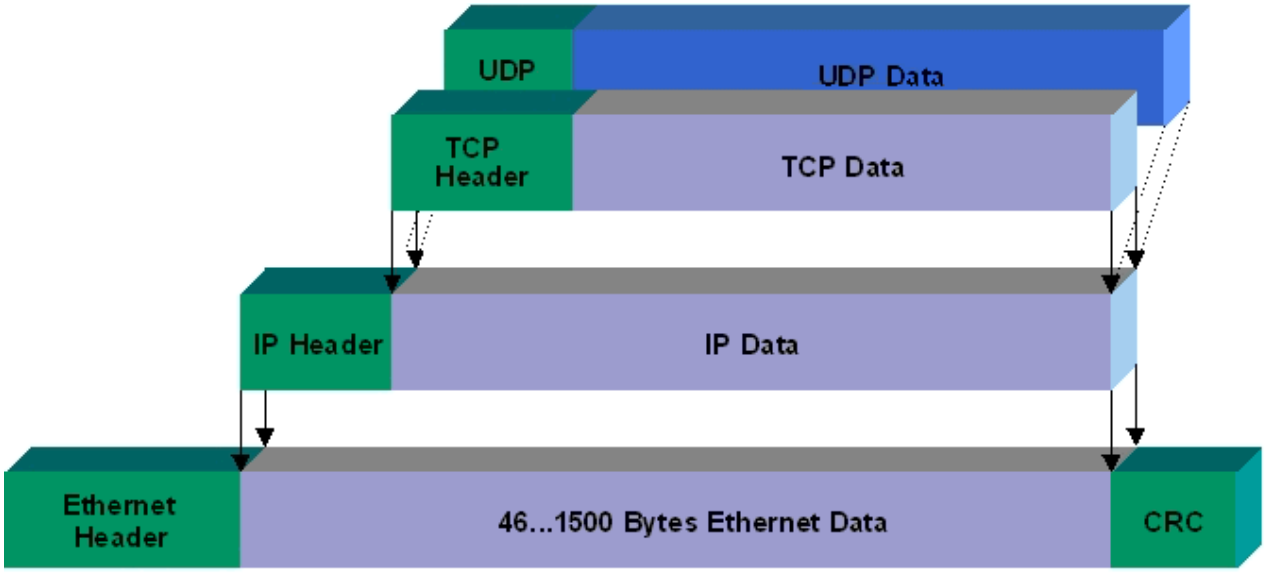


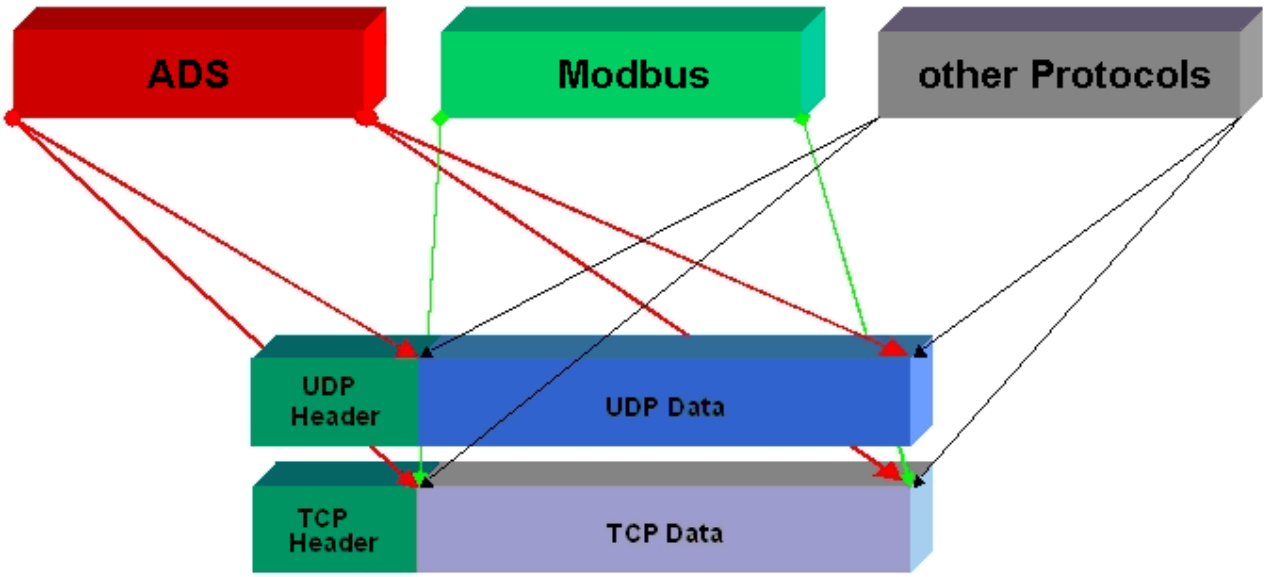

Fig. 2: Ethernet protocol

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

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

- ADS
- ModbusTCP

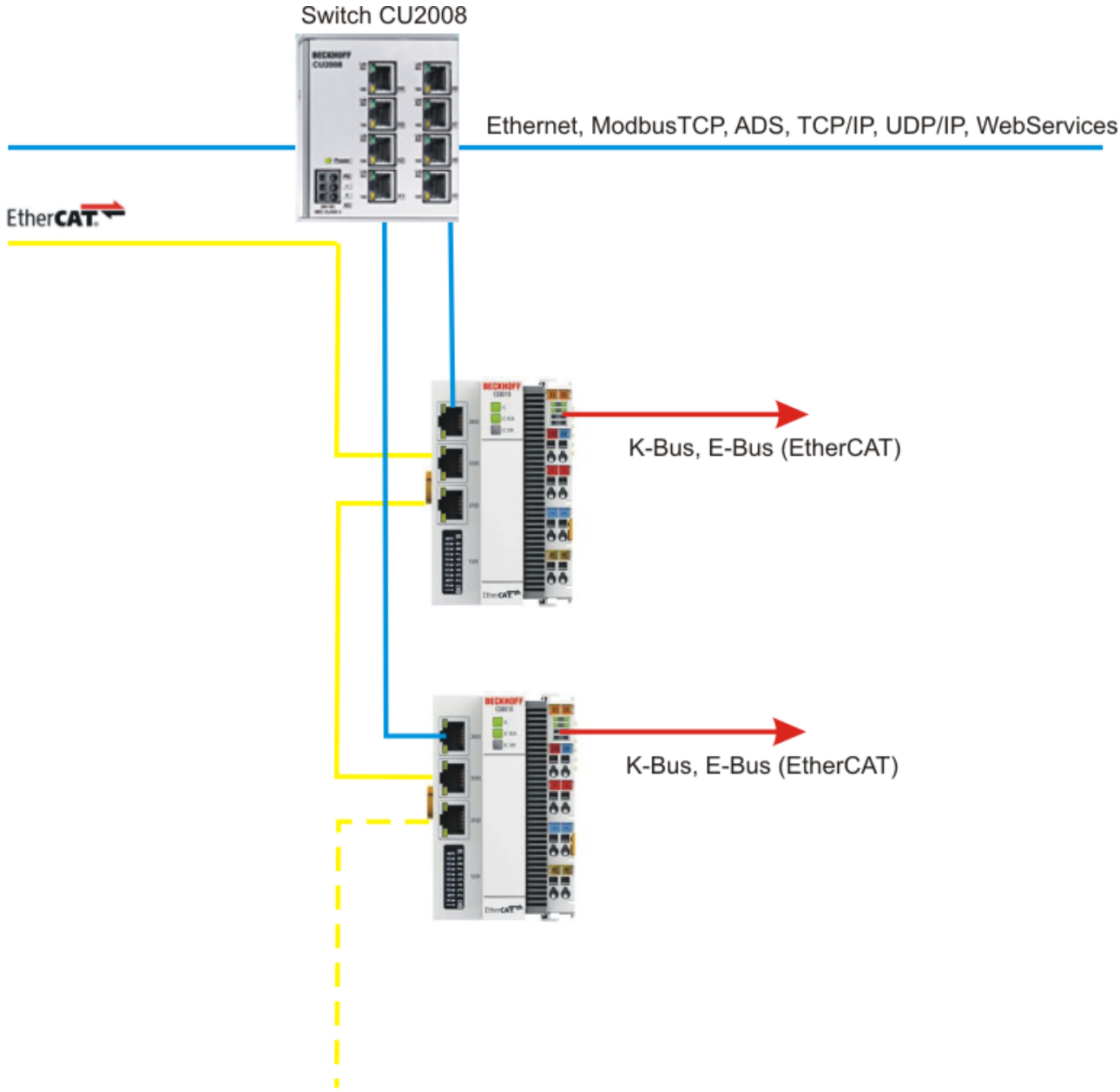
Both of these protocols are implemented in parallel on the Bus Coupler, so that no configuration is needed to activate the protocols.




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

## 8.1.2 Topology example

**CX8010**



# 8.2 ModbusTCP

## 8.2.1 ModbusTCP Server - Overview

On the CX80xx you can exchange data over a very widespread protocol with the aid of the ModbusTCP. Use the documentation *TwinCAT ModbusTCP server* for this. The server is already installed on the CX80xx. The X001 or X101/102 interfaces may be used.

German: http://infosys.beckhoff.com/content/1031/tcmodbussrv/html/tcmodbussrv_overview.htm

English: http://infosys.beckhoff.com/content/1033/tcmodbussrv/html/tcmodbussrv_overview.htm

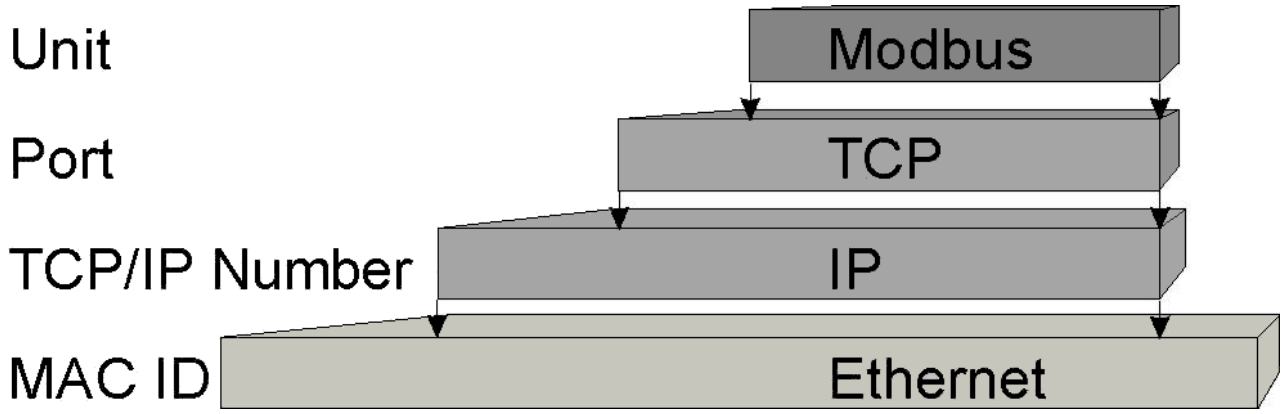
| | |
|---|---|
| Note | **CX8091**<br>The Modbus/TCP server is not part of the image of the CX8091. |

| | |
|---|---|
| Note | **Required libraries**<br>The corresponding libraries are located in the TwinCAT\Lib directory on the MicroSD card and can be copied to the programming system using an SD card reader. |

| | |
|---|---|
| Note | **CX8010, CX803x, CX805x, CX8080, CX8093, CX8095**<br>The Modbus TCP can only be used with the X001 interface. |

## 8.2.2 ModbusTCP Protocol

The Ethernet protocol is addressed by means of the MAC-ID. The user does not normally need to be concerned about this address. The IP number has a length of 4 bytes, and must be parameterized by the user on the Bus Coupler and in the application. In ModbusTCP, the TCP port is set to 502. The UNIT can be freely selected under ModbusTCP, and does not have to be configured by the user.




**TCP port number**

The TCP port number for ModbusTCP has been standardized to 502.

**Modbus-Unit**

The unit is returned by the slave.

**ModbusTCP Protocol**

| Byte | Name | Description |
|---|---|---|
| 0 | Transaction identifier | Is returned by the slave |
| 1 | Transaction identifier | Is returned by the slave |
| 2 | Protocol identifier | always 0 |
| 3 | Protocol identifier | always 0 |
| 4 | Length field | 0 (if the message is less than 256 bytes in length) |
| 5 | Length field | Number of following bytes |
| 6 | UNIT identifier | returned by the slave |
| 7 | Modbus | Modbus protocol with the function follows |

### 8.2.3 Mapping between Modbus and ADS

The standard mapping of the server is illustrated in the following tables for the first runtime system:

| Modbus ranges | Modbus address | ADS range | |
|---|---|---|---|
| Digital inputs | 0x0000 - 0x7FFF | **Index group** | **Index offset** |
| | | 0xF021 - process image of the physical inputs (bit access) | 0x0 |
| | 0x8000 - 0x80FF | **Name of the variables in the PLC program** | **Data type** |
| | | .mb_Input_Coils | ARRAY [0..255] OF BOOL |
| Digital outputs (coils) | 0x0000 - 0x7FFF | **Index group** | **Index offset** |
| | | 0xF031 - process image of the physical outputs (bit access) | 0x0 |
| | 0x8000 - 0x80FF | **Name of the variables in the PLC program** | **Data type** |
| | | .mb_Output_Coils | ARRAY [0..255] OF BOOL |
| Input registers | 0x0000 - 0x7FFF | **Index group** | **Index offset** |
| | | 0xF020 - process image of the physical inputs | 0x0 |
| | 0x8000 - 0x80FF | **Name of the variables in the PLC program** | **Data type** |
| | | .mb_Input_Registers | ARRAY [0..255] OF WORD |
| Output registers | 0x0000 - 0x2FFF | **Index group** | **Index offset** |
| | | 0xF030 - process image of the physical outputs | 0x0 |
| | 0x3000 - 0x5FFF | 0x4020 - PLC memory area | 0x0 |
| | 0x6000 - 0x7FFF | 0x4040 - PLC data area | 0x0 |
| | 0x8000 - 0x80FF | **Name of the variables in the PLC program** | **Data type** |
| | | .mb_Output_Registers | ARRAY [0..255] OF WORD |

The server maps this to the individual ADS ranges and enables access to the physical process image and the PLC flag ranges.

The configurator enables the adaptation of the setting.

#### Default XML

The standard configuration looks like this:

```
<Configuration>
    <!-- Modbus TCP port, default = 502-->
    <Port>502</Port>
    <!-- optional IP configuration for Modbus TCP server-->
    <IpAddr/>
    <Mapping>
    <InputCoils>
        <MappingInfo>
        <!-- AdsPort: TwinCAT2 PLC1 = 801, PLC2 = 811...-->
        <AdsPort>801</AdsPort>
        <StartAddress>0</StartAddress>
        <EndAddress>32767</EndAddress>
        <!-- IndexGroup 61473 = 0xF021 -> physical plc inputs %IX -->
        <IndexGroup>61473</IndexGroup>
        <!-- Bit offset-->
        <IndexOffset>0</IndexOffset>
        </MappingInfo>
        <MappingInfo>
        <AdsPort>801</AdsPort>
        <!-- Modbus input coils -->
        <StartAddress>32768</StartAddress>
        <EndAddress>33023</EndAddress>
        <VarName>.mb_Input_Coils</VarName>
```

```
        </MappingInfo>
    </InputCoils>
    <OutputCoils>
        <MappingInfo>
        <AdsPort>801</AdsPort>
        <EndAddress>32767</EndAddress>
        <!-- IndexGroup 61489 = 0xF031 -> physical plc outputs %QX -->
        <IndexGroup>61489</IndexGroup>
        <!-- Bit offset-->
        <IndexOffset>0</IndexOffset>
        </MappingInfo>
        <MappingInfo>
        <AdsPort>801</AdsPort>
        <!-- Modbus output coils-->
        <StartAddress>32768</StartAddress>
        <EndAddress>33023</EndAddress>
        <VarName>.mb_Output_Coils</VarName>
        </MappingInfo>
    </OutputCoils>
    <InputRegisters>
        <MappingInfo>
        <AdsPort>801</AdsPort>
        <StartAddress>0</StartAddress>
        <EndAddress>32767</EndAddress>
        <!-- IndexGroup 61472 = 0xF020 -> physical plc input register %I -->
        <IndexGroup>61472</IndexGroup>
        <!-- Byte offset-->
        <IndexOffset>0</IndexOffset>
        </MappingInfo>
        <MappingInfo>
        <AdsPort>801</AdsPort>
        <!-- Modbus input registers -->
        <StartAddress>32768</StartAddress>
        <EndAddress>33023</EndAddress>
        <VarName>.mb_Input_Registers</VarName>
        </MappingInfo>
    </InputRegisters>
    <OutputRegisters>
        <MappingInfo>
        <AdsPort>801</AdsPort>
        <StartAddress>0</StartAddress>
        <EndAddress>12287</EndAddress>
        <!-- IndexGroup 61488 = 0xF030 -> physical plc output register %Q -->
        <IndexGroup>61488</IndexGroup>
        <!-- Byte offset-->
        <IndexOffset>0</IndexOffset>
        </MappingInfo>
        <MappingInfo>
        <AdsPort>801</AdsPort>
        <StartAddress>12288</StartAddress>
        <EndAddress>24575</EndAddress>
        <!-- IndexGroup 16416 = 0x4020 -> plc memory area %M -->
        <IndexGroup>16416</IndexGroup>
        <!-- Byte offset-->
        <IndexOffset>0</IndexOffset>
        </MappingInfo>
        <MappingInfo>
        <AdsPort>801</AdsPort>
        <StartAddress>24576</StartAddress>
        <EndAddress>32767</EndAddress>
        <!-- IndexGroup 16448 = 0x4040 -> plc data area -->
        <IndexGroup>16448</IndexGroup>
        <!-- Byte offset-->
        <IndexOffset>0</IndexOffset>
        </MappingInfo>
        <MappingInfo>
        <AdsPort>801</AdsPort>
        <!-- Modbus output registers -->
        <StartAddress>32768</StartAddress>
        <EndAddress>33023</EndAddress>
        <VarName>.mb_Output_Registers</VarName>
        </MappingInfo>
    </OutputRegisters>
    </Mapping>
</Configuration>
```

## 8.3 TCP/IP

**Server overview**

Via the CX80x0 you can also implement your own protocols with the help of the TCP/IP server. Use the documentation *TwinCAT TCP/IP Server* for this.
The TCP/IP server is already pre-installed on the CX80x0. Only the X001 or X101/102 interfaces may be used.

German: http://infosys.beckhoff.com/content/1031/tcpipserver/html/tcpipserver_overview.htm

English: http://infosys.beckhoff.com/content/1033/tcpipserver/html/tcpipserver_overview.htm

| | |
|---|---|
| Note | **CX8091**<br>The TCP/IP server is not part of the image of the CX8091. |

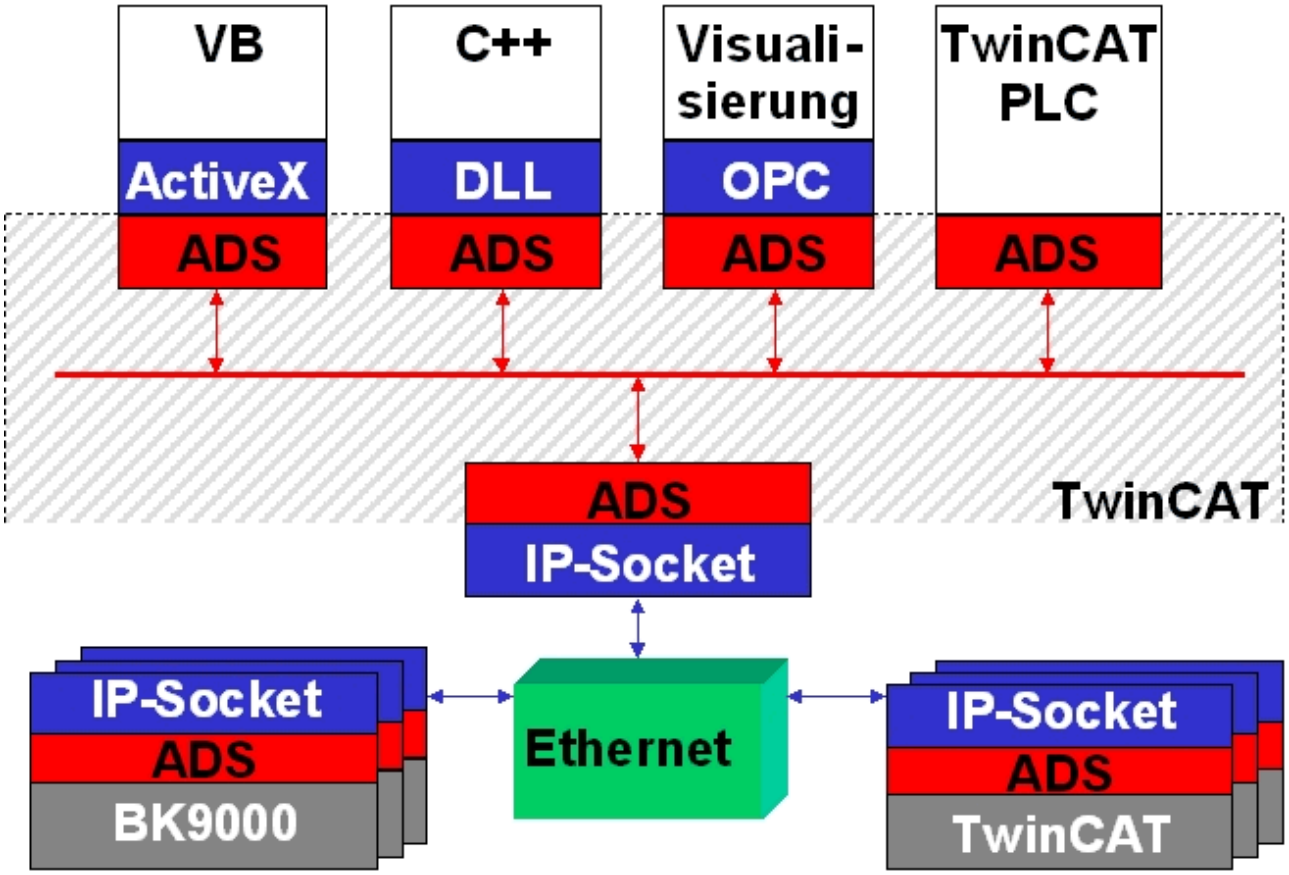
| | |
|---|---|
| Note | **Required libraries**<br>The corresponding libraries are located in the TwinCAT\Lib directory on the MicroSD card and can be copied to the programming system using an SD card reader. |

| | |
|---|---|
| Note | **CX8010, CX803x, CX805x, CX8080, CX8093, CX8095**<br>The TCP/IP server can only be used with the X001 interface. |

## 8.4 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**

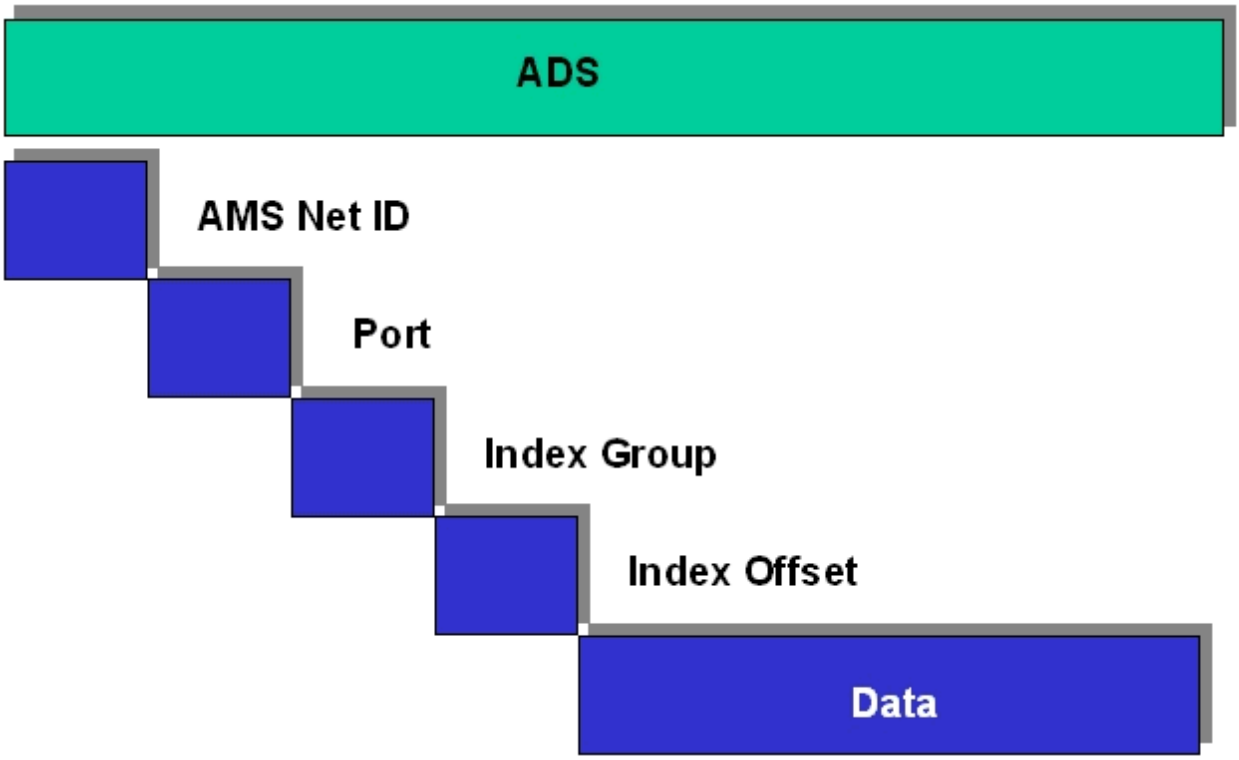
**ADS-OCX**
The ADS-OCX is an Active-X component. It offers a standard interface to, for instance, Visual Basic, Delphi, etc.

**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 EtherCAT

## 9.1 Limits of existing Ethernet communication approaches

There are many different approaches that try and provide real-time capability for Ethernet: for example, the CSMA/CD access procedure is disabled via higher level protocol layers and replaced by the time slices procedure or polling; other propositions use special switches that distribute Ethernet packets in a precisely controlled timely manner. Whilst these solutions may be able to transport data packets more or less quickly and accurately to the connected Ethernet nodes, the times required for the redirection to the outputs or drive controllers and for reading the input data strongly depend on the implementation. A sub-bus is usually also required, particularly in modular I/O systems, which, like the Beckhoff K-Bus, can be synchronized and fast, but nevertheless always adds small delays to the communication that cannot be avoided.

If individual Ethernet frames are used for each device, the usable data rate is very low in principle: The shortest Ethernet frame is 84 bytes long (incl. inter-packet gap IPG). If, for example, a drive cyclically sends 4 bytes of actual value and state information and accordingly receives 4 bytes of set value and control word information, at 100% bus load (i.e. with infinitely short response time of the drive) a usable data rate of only 4/84= 4.7% is achieved. At an average response time of 10 µs, the rate drops to 1.9%. These limitations apply to all real-time Ethernet approaches that send an Ethernet frame to each device (or expect a frame from each device), irrespective of the protocols used within the Ethernet frame.

## 9.2 New approach

EtherCAT technology overcomes these inherent limitations of other Ethernet solutions: the Ethernet packet is no longer received, then interpreted and copied as process data at every connection. The newly developed FMMU (fieldbus memory management unit) in each I/O terminal reads the data addressed to it, whilst the telegram continues through the device. Similarly, input data are inserted while the telegram passes through. The telegrams are only delayed by a few nanoseconds.

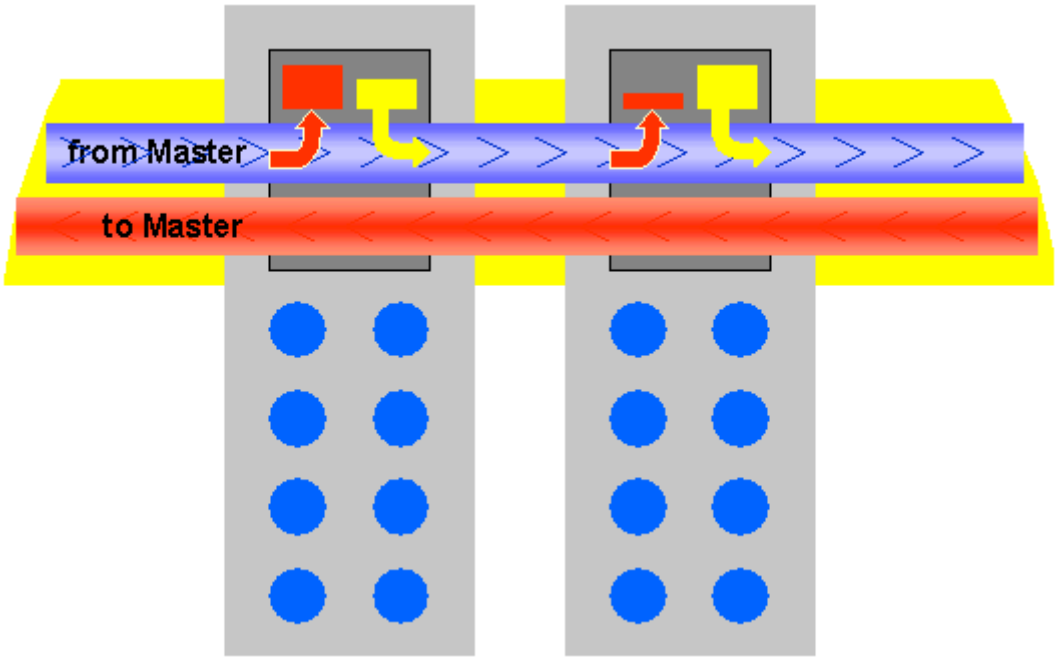



Fig. 1: Dynamic telegram processing

Since an Ethernet frame reaches the data of many devices both in send and receive direction, the usable data rate increases to approx. 80%. The full-duplex features of 100BaseTx are fully utilized, so that effective data rates of > 100 Mbit/s (up to 80% of 2 x 100 Mbit/s) can be achieved.

### Ethernet up to the terminal

The Ethernet protocol conforms to IEEE 802.3 remains intact right up to the individual terminals; no sub-bus is required. Only the transfer physics is converted in the coupler from twisted pair or optical fiber physics to E-bus, in order to meet the requirements of the electronic terminal block.

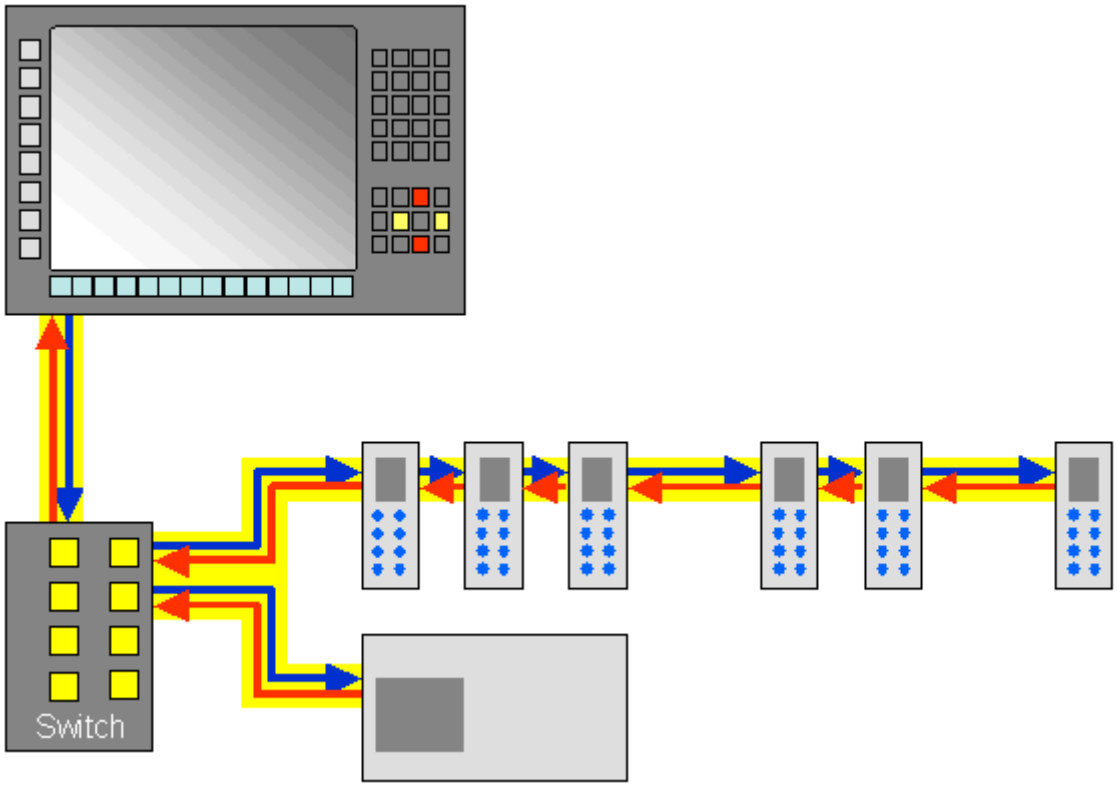



Fig. 2: Full duplex Ethernet in the ring, one frame for many devices. The EtherCAT system architecture increases the communication efficiency.

On the control side, the TwinCAT Y driver for Ethernet complements the FMMU technology. This integrates transparently into the system, so that it appears as an operating system-compatible network driver, and additionally as a TwinCAT fieldbus card. An internal prioritization and buffer is provided at the transmitter end which always finds a free transmission channel for Ethernet frames from the real-time system that may be queuing. The operating system's Ethernet frames are only a later transmitted in the gaps if sufficient time is available.

At the receiving end, all the Ethernet frames received are examined by the TwinCAT I/O system, and those with real-time relevance are filtered out. All other frames are passed on to the operating system after examination, outside the context of the real-time system.

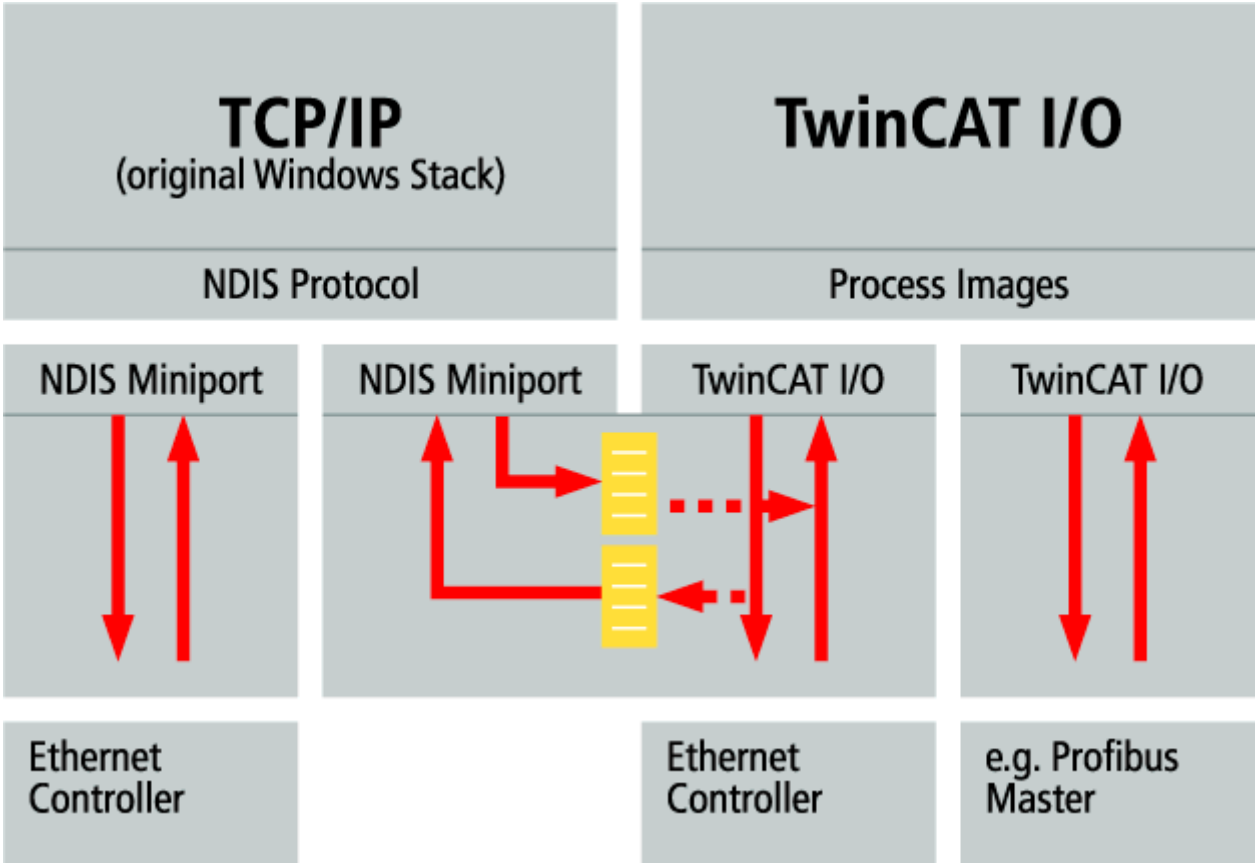



Fig. 3: Operating system-compatible TwinCAT Y driver:

Very inexpensive, commercially available standard network interface cards (NIC) are used as hardware in the controller. The data transfer to the PC takes place via DMA (direct memory access). Therefore, no CPU performance is lost for network access.

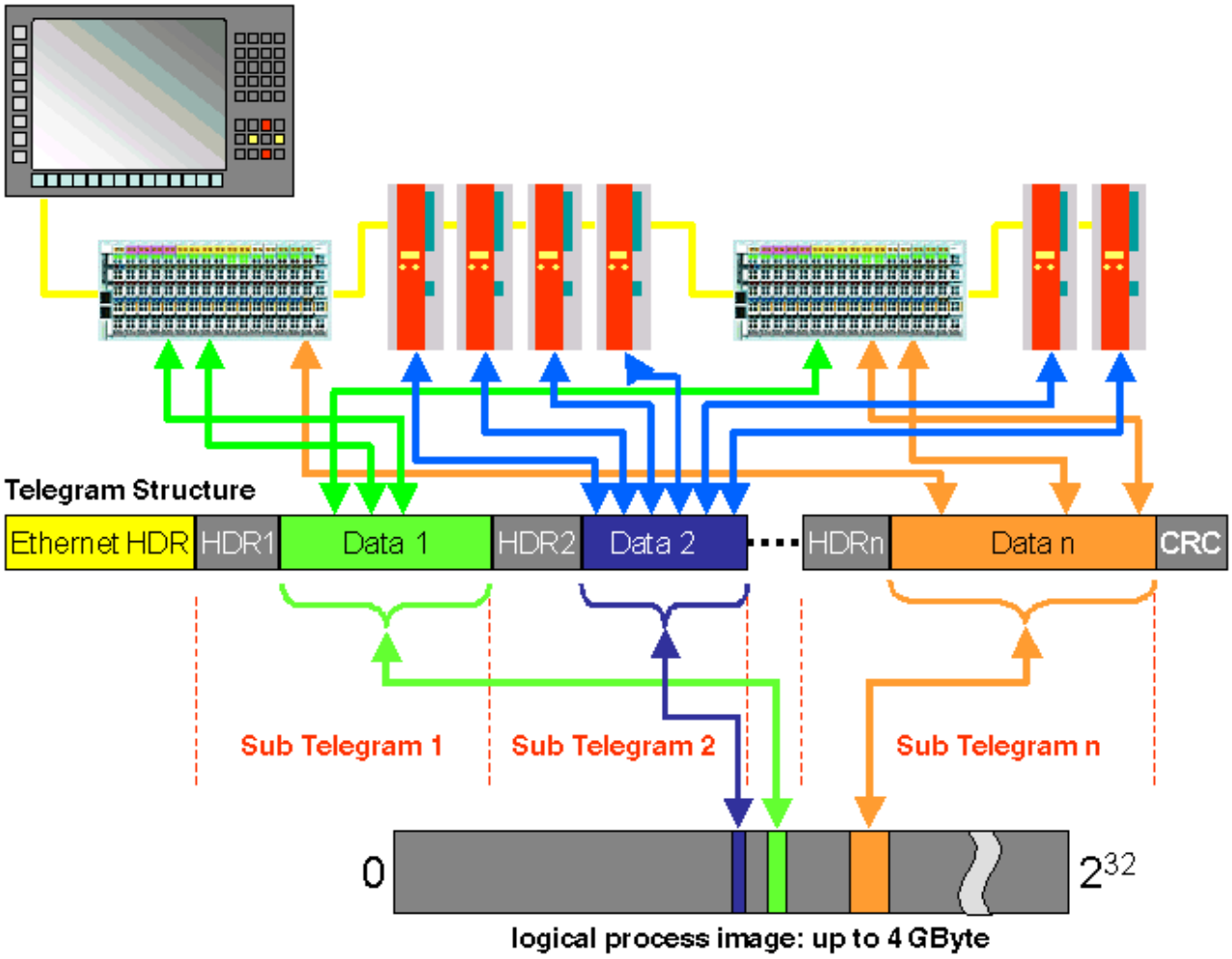
Since the Ethernet functionality of the operating system is fully maintained, all operating system-compatible protocols can be operated in parallel on the same physical network. This not only includes standard IT-protocols such as TCP/IP, HTTP, FTP or SOAP, but also practically all Industrial-Ethernet-protocols such as Modbus TCP, ProfiNet or Ethernet/IP.

## 9.3 System Properties

**Protocol**

The EtherCAT protocol is optimized for process data and is transported directly within the Ethernet frame thanks to a special Ether-type. It may consist of several sub-telegrams, each serving a particular memory area of the logical process images that can be up to 4 gigabytes in size. The data sequence is independent of the physical order of the Ethernet terminals in the network; addressing can be in any order. Broadcast, Multicast and communication between slaves are possible. Transfer directly in the Ethernet frame is used in cases where EtherCAT components are operated in the same subnet as the control computer.

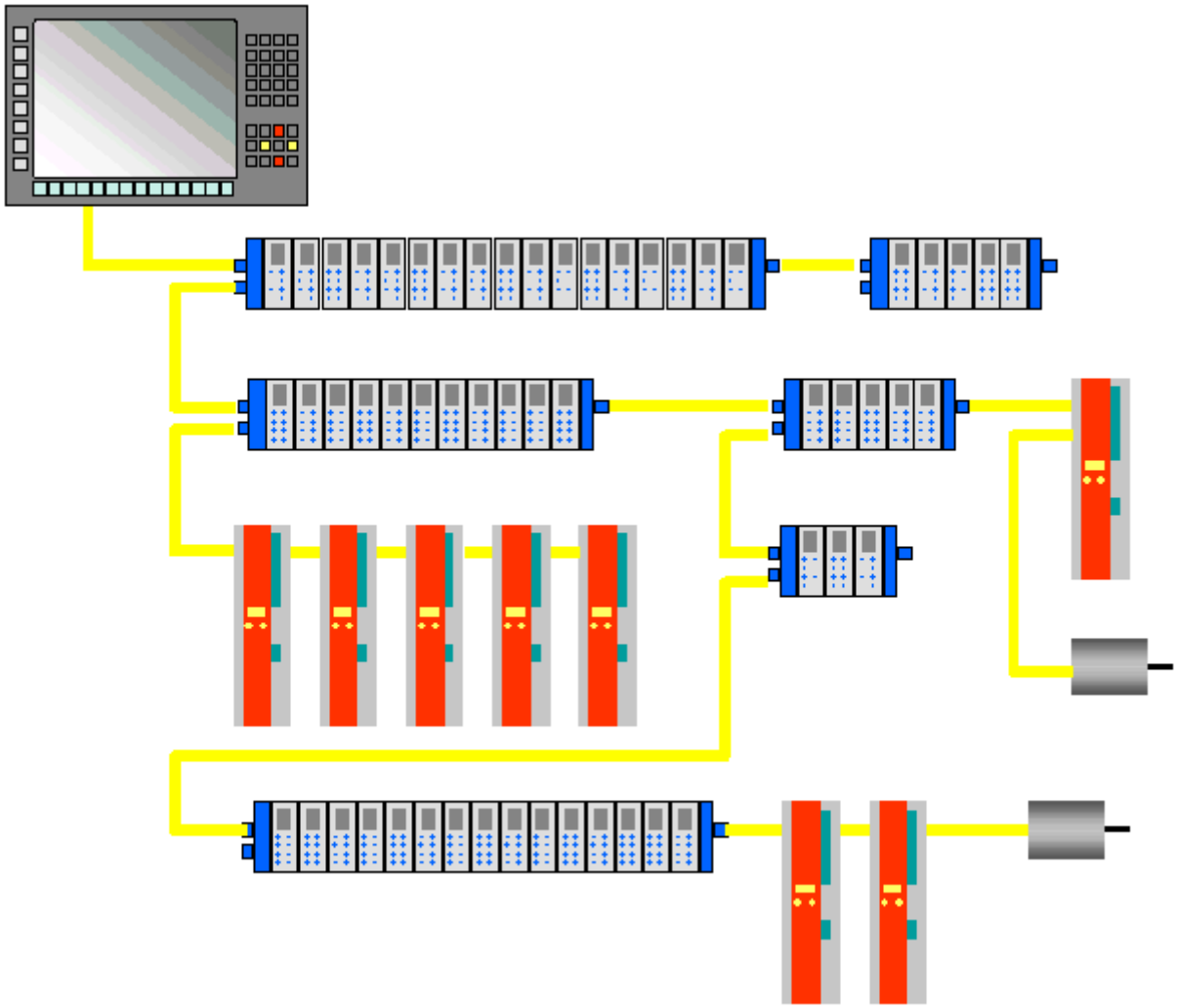
However, EtherCAT applications are not limited to a subnet: EtherCAT UDP packs the EtherCAT protocol into UDP/IP datagrams. This enables any control with Ethernet protocol stack to address EtherCAT systems. Even communication across routers into other subnets is possible. In this variant, system performance obviously depends on the real-time characteristics of the control and its Ethernet protocol implementation. The response times of the EtherCAT network itself are hardly restricted at all: the UDP datagram only has to be unpacked in the first station.




Protocol structure: The process image allocation is freely configurable. Data are copied directly in the I/O terminal to the desired location within the process image: no additional mapping is required. The available logical address space is with very large (4 GB).

### Topology

Line, tree or star: EtherCAT supports almost any topology. The bus or line structure known from the fieldbusses thus also becomes available for Ethernet. Particularly useful for system wiring is the combination of line and branches or stubs. The required interfaces exist on the couplers; no additional switches are required. Naturally, the classic switch-based Ethernet star topology can also be used.



Maximum wiring flexibility:
with or without switch, line or tree topologies, can be freely selected and combined. The complete bandwidth of the Ethernet network - such as different optical fibers and copper cables - can be used in combination with switches or media converters.

**Distributed Clocks**

Accurate synchronization is particularly important in cases where spatially distributed processes require simultaneous actions. This may be the case, for example, in applications where several servo axes carry out coordinated movements simultaneously.

The most powerful approach for synchronization is the accurate alignment of distributed clocks, as described in the new IEEE 1588 standard. In contrast to fully synchronous communication, where synchronization quality suffers immediately in the event of a communication fault, distributed aligned clocks have a high degree of tolerance vis-à-vis possible fault-related delays within the communication system.

With EtherCAT, the data exchange is fully based on a pure hardware machine. Since the communication utilizes a logical (and thanks to full-duplex Fast Ethernet also physical) ring structure, the mother clock can determine the run-time offset to the individual daughter clocks simply and accurately - and vice versa. The distributed clocks are adjusted based on this value, which means that a very precise network-wide timebase with a jitter of significantly less then 1 microsecond is available.

However, high-resolution distributed clocks are not only used for synchronization, but can also provide accurate information about the local timing of the data acquisition. For example, controls frequently calculate velocities from sequentially measured positions. Particularly with very short sampling times, even a small temporal jitter in the displacement measurement leads to large step changes in velocity. With EtherCAT new, extended data types are introduced as a logical extension (time stamp and oversampling data type). The local time is linked to the measured value with a resolution of up to 10 ns, which is made possible by the

large bandwidth offered by Ethernet. The accuracy of a velocity calculation then no longer depends on the jitter of the communication system. It is orders of magnitude better than that of measuring techniques based on jitter-free communication.

### Performance

EtherCAT reaches new dimensions in network performance. Protocol processing is purely hardware-based through an FMMU chip in the terminal and DMA access to the network card of the master. It is thus independent of protocol stack run-times, CPU performance and software implementation. The update time for 1000 I/Os is only 30 µs - including terminal cycle time. Up to 1486 bytes of process data can be exchanged with a single Ethernet frame - this is equivalent to almost 12000 digital inputs and outputs. The transfer of this data quantity only takes 300 µs.

The communication with 100 servo axes only takes 100 µs. During this time, all axes are provided with set values and control data and report their actual position and status. Distributed clocks enable the axes to be synchronised with a deviation of significantly less than 1 microsecond.

The extremely high performance of the EtherCAT technology enables control concepts that could not be realized with classic fieldbus systems. For example, the Ethernet system can now not only deal with velocity control, but also with the current control of distributed drives. The tremendous bandwidth enables status information to be transferred with each data item. With EtherCAT, a communication technology is available that matches the superior computing capacity of modern Industrial PCs. The bus system is no longer the bottleneck of the control concept. Distributed I/Os are recorded faster than is possible with most local I/O interfaces. The EtherCAT technology principle is scalable and not bound to the baud rate of 100 MBaud – extension to GBit Ethernet is possible.

### Diagnostics

Experience with fieldbus systems shows that availability and commissioning times crucially depend on the diagnostic capability. Only faults that are detected quickly and accurately and which can be precisely located can be corrected quickly. Therefore, special attention was paid to exemplary diagnostic features during the development of EtherCAT.

During commissioning, the actual configuration of the I/O terminals should be checked for consistency with the specified configuration. The topology should also match the saved configuration. Due to the built-in topology recognition down to the individual terminals, this verification can not only take place during system start-up, automatic reading in of the network is also possible (configuration upload).

Bit faults during the transfer are reliably detected through evaluation of the CRC checksum: The 32 bit CRC polynomial has a minimum hamming distance of 4. Apart from breaking point detection and localization, the protocol, physical transfer behavior and topology of the EtherCAT system enable individual quality monitoring of each individual transmission segment. The automatic evaluation of the associated error counters enables precise localization of critical network sections. Gradual or changing sources of error such as EMC influences, defective push-in connectors or cable damage are detected and located, even if they do not yet overstrain the self-healing capacity of the network.

### Integration of standard Bus Terminals from Beckhoff

In addition to the new Bus Terminals with E-Bus connection (ELxxxx), all Bus Terminals from the familiar standard range with K-Bus connection (KLxxxx) can be connected via the BK1120 or BK1250 Bus Coupler. This ensures compatibility and continuity with the existing Beckhoff Bus Terminal systems. Existing investments are protected.

## 9.4 Application areas

### Control loops via the bus

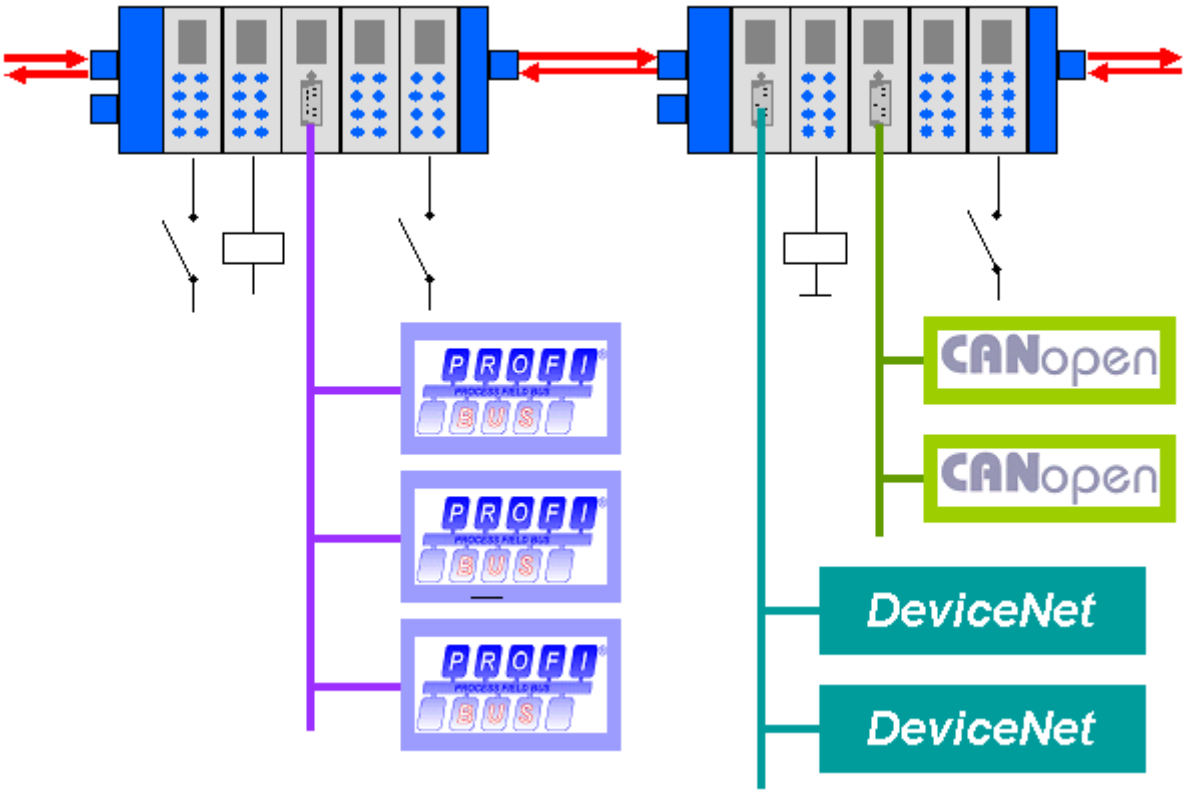
In conjunction with fast machine control systems the outstanding performance of EtherCAT can be demonstrated particularly well. The bus system offers access speeds similar to local I/Os. EtherCAT enables not only the position control loop, but also the velocity control loop or even the current control loop to be closed via the bus, resulting in cost-effective drive controllers. EtherCAT therefore takes full advantage of the performance of fast PC-based control technology. The IPCs become small and economical, since slots are avoided: Expansion cards are also controlled via EtherCAT.

EtherCAT is not only suitable for very fast applications. Since all protocol processing takes place in hardware, the technology only places low demands on the master connection: a commercially available Ethernet Controller is sufficient. The master only has to send EtherCAT frames cyclically or as required – mapping, addressing, node monitoring etc. takes place in the slave ASICs. The input process image arrives at the master fully sorted and therefore ready prepared for the control application. For simple cyclic applications with a process image size of up to 1486 bytes, only one telegram has to be processed in the master, consisting almost exclusively of the process data. Larger process images are distributed over several frames.

EtherCAT may therefore be the fieldbus technology placing the lowest demands on the master.

### EtherCAT instead of PCI

With increasing miniaturisation of the PC-components, the physical size of Industrial PCs is increasingly determined by the number of required slots. The bandwidth of Fast Ethernet, together with the data width of the EtherCAT communication hardware (FMMU chip) enables new directions: Interfaces that are conventionally located in the IPC are transferred to intelligent interface terminals at the EtherCAT. Apart from the decentralized I/Os, axes and operating devices, complex systems such as fieldbus masters, fast serial interfaces, gateways and other communication interfaces can be addressed. Even further Ethernet devices without restriction on protocol variants can be connected via decentralized switch port terminals. The central IPC becomes smaller and therefore cost-effective. An Ethernet interface is sufficient for complete communication with the periphery.



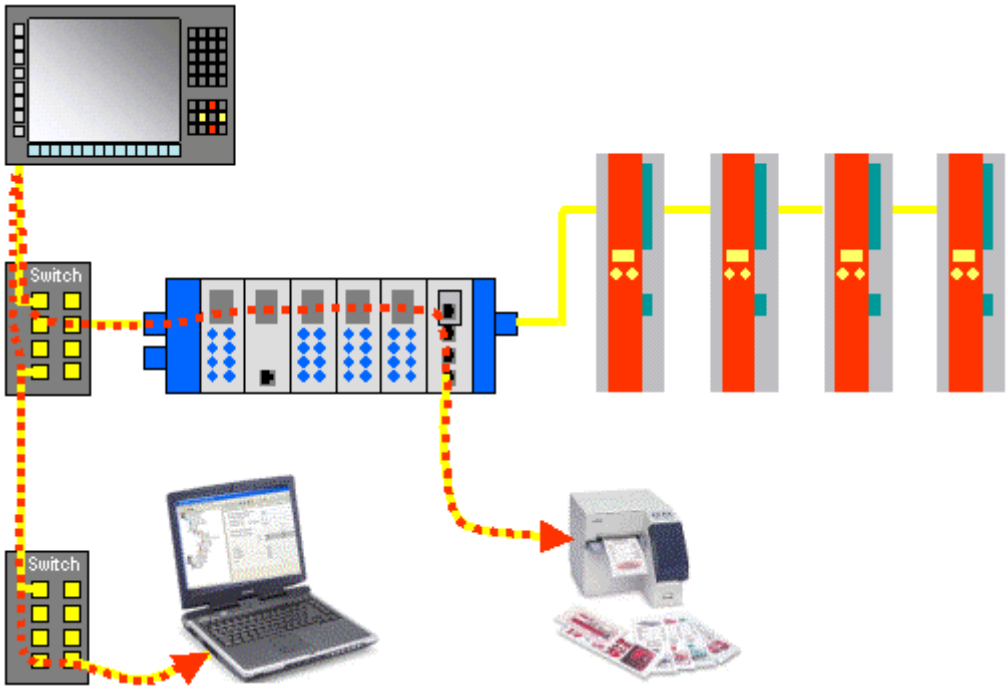
Fieldbus devices (e.g. Profibus, CANopen, DeviceNet, AS Interface etc.)
are integrated via decentralised fieldbus master terminals.

## 9.5 Openness

### EtherCAT is open

The EtherCAT technology is not only fully Ethernet-compatible, but also characterized by a particular open design: the protocol tolerates other Ethernet-based services and protocols on the same physical network - usually only with minimum loss of performance. There is no restriction on the type of Ethernet device that can be connected within the EtherCAT strand via a switch port. Devices with fieldbus interface are integrated via EtherCAT fieldbus master terminals. The UDP protocol variant can be implemented on each socket interface.

EtherCAT devices can additionally have a TCP/IP stack and therefore externally behave like a normal Ethernet device. The master serves as a switch that transfers the frames to the respective devices according to the address information. Instead of switch ports, only the EtherCAT addresses automatically assigned during startup are used.




The switch port terminals work accordingly: Any Ethernet devices can be connected. The Ethernet frames are tunneled through the EtherCAT protocol, as usual in the Internet (e.g. VPN, PPPoE (DSL) etc.). The EtherCAT network is fully transparent for the Ethernet device, and the real-time characteristics of the EtherCAT are not impaired.

### EtherCAT Technology Group

The EtherCAT Technology Group (ETG, http://www.ethercat.org) is the international manufacturer and user association with a mission to disseminate, further develop and support EtherCAT. All companies are invited to become members.

### Internationally standardised

The EtherCAT Technology Group is an official standardization partner of IEC (International Electrotechnical Commission). The EtherCAT specification was published by the IEC as IEC/PAS 62407 and is available from the IEC (http://www.iec.ch). It is currently integrated in the following fieldbus standards: IEC 61158, IEC 61800-7 and ISO 15745.

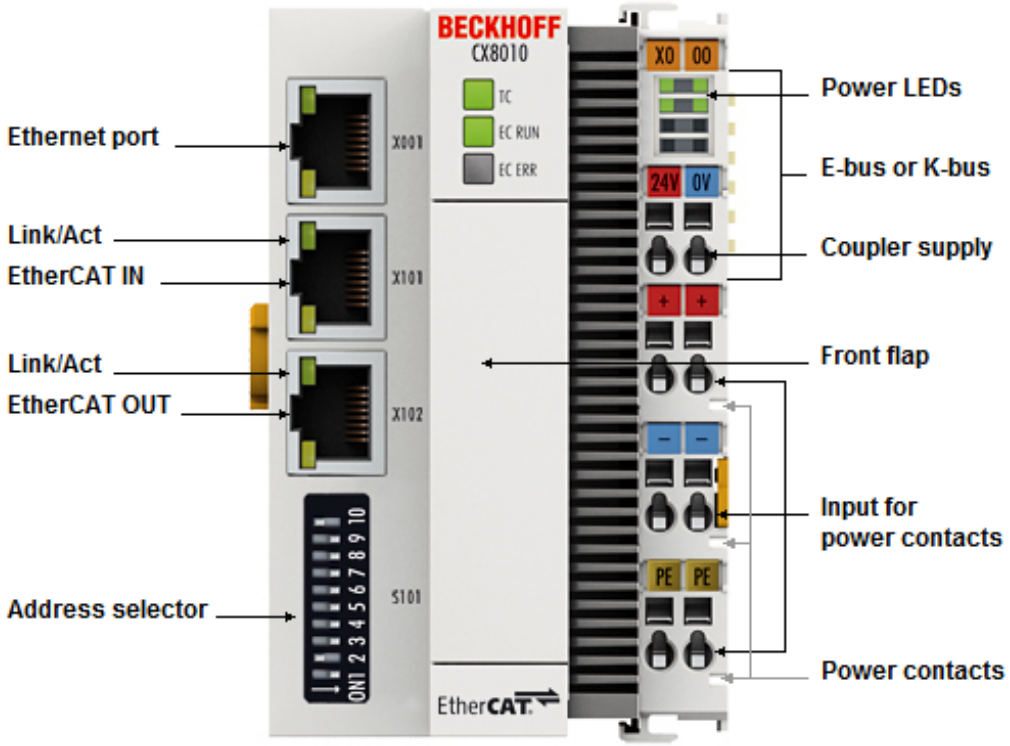
## 9.6 Summary and Outlook

EtherCAT is characterized by outstanding performance, very simple wiring and openness for other protocols. EtherCAT sets new standards where conventional fieldbus systems reach their limits: 1000 I/O in 30 µs, optionally twisted pair cable or optical fibre and, thanks to Ethernet and Internet-technologies, optimum vertical integration. With EtherCAT, the costly Ethernet star topology can be replaced with a simple line structure - no expensive infrastructure components are required. Optionally, EtherCAT may also be wired in the classic way using switches, in order to integrate other Ethernet devices. Where other real-time-Ethernet approaches require special connections in the controller, EtherCAT manages with very cost-effective standard Ethernet cards (NIC).

With EtherCAT, Ethernet down to the terminal block becomes technically feasible and economically sensible. Full Ethernet compatibility, Internet technologies even in very simple devices, maximum utilization of the large bandwidth offered by Ethernet, outstanding real-time characteristics at low costs are outstanding features of this new network. As a high-speed bus and I/O bus for Industrial PCs or in combination with smaller control technology, EtherCAT is expected to be widely used in a wide range of applications.

# 10 Error handling diagnosis

## 10.1 Diagnostic LEDs




**Ethernet interface X001**

| Interface X001 | Ethernet (CX8010) | Meaning |
|---|---|---|
| LED green | on | Link present |
| LED yellow | flashing | Activity |

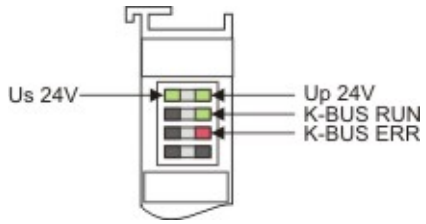
**Ethernet interface X101 and X102**

| Interface X101-102 | Ethernet (CX8010) | Meaning |
|---|---|---|
| LED green | flashing | Activity |
| LED yellow | is not used | - |

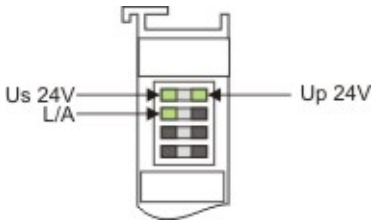
**LED coupler**

| Labeling | Meaning | Color | Meaning |
|---|---|---|---|
| TC | Displays the TwinCAT mode | red | TwinCAT in Stop |
| | | Green | TwinCAT in Running Mode |
| | | Blue | TwinCAT in Config Mode |
| EC RUN | EtherCAT status | on (green) | CX8010 in OP mode |
| | | flashing (green) 200 ms on/off | CX8010 in PRE-OP mode |
| | | flashing (green) 200 ms on/ 1000 ms off | CX8010 in SAFE-OP mode |
| | | off | INIT |
| EC ERR | | off | OK |
| | | on (red) | not configured |

### LED power supply terminal




Operation with K-bus terminals




Operation with E-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<br>connected to E-bus / no data traffic<br>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 code**

| Error code | Error code argument | Description | Remedy |
|---|---|---|---|
| Persistent, continuous flashing | | EMC problems | • Check power supply for undervoltage or overvoltage peaks<br>• Implement EMC measures<br>• 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<br>- 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 initialization | 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

| | |
|---|---|
| **Note** | **Required TwinCAT version**<br>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.

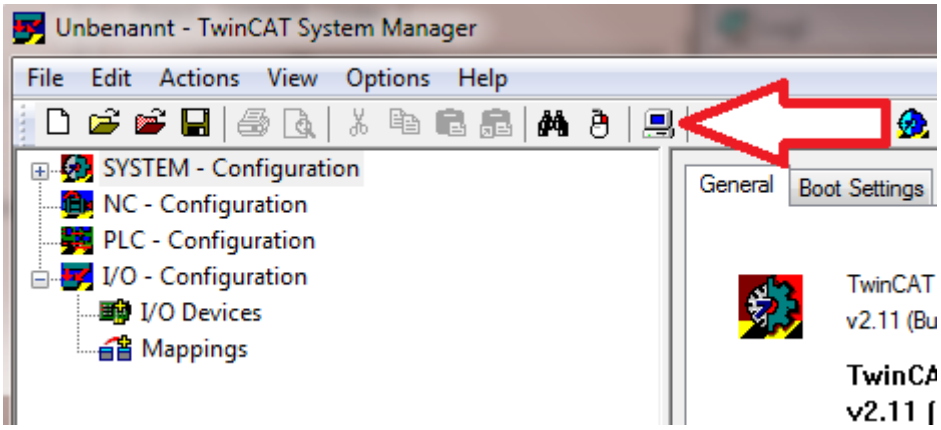
2. Connect voltage to the CX80xx (see power supply [▶ 21]).

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).

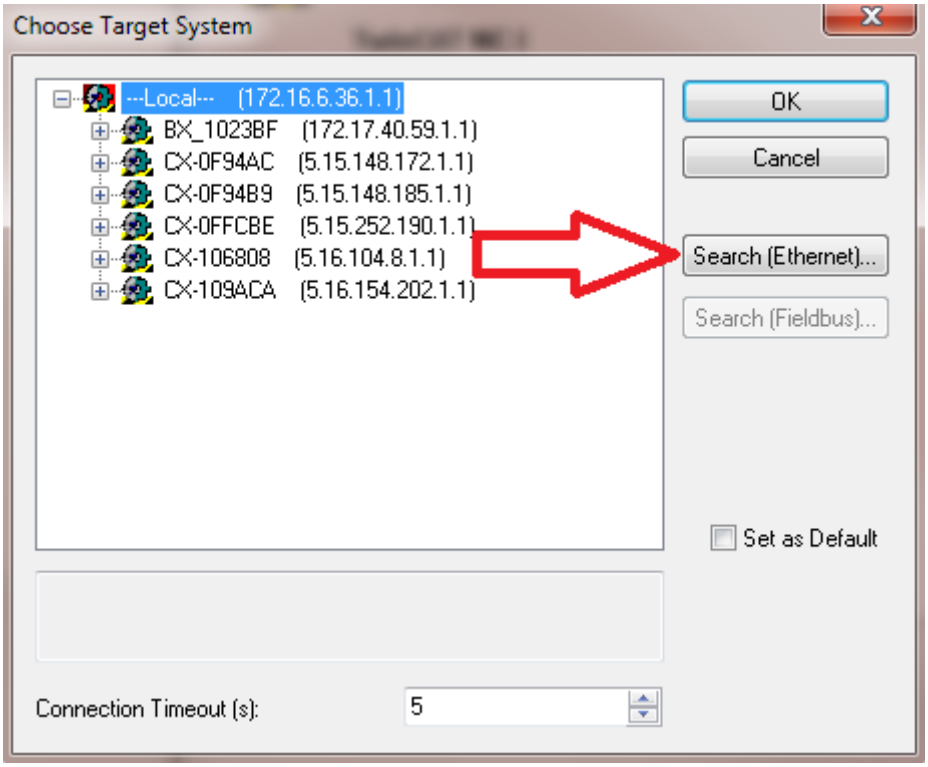
4. Wait a while, approx. 1 to 2 minutes; either the CX80xx will be assigned an address by the DHCP server (usually fast) or, if it does not find a DHCP server, it uses a local IP address.

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

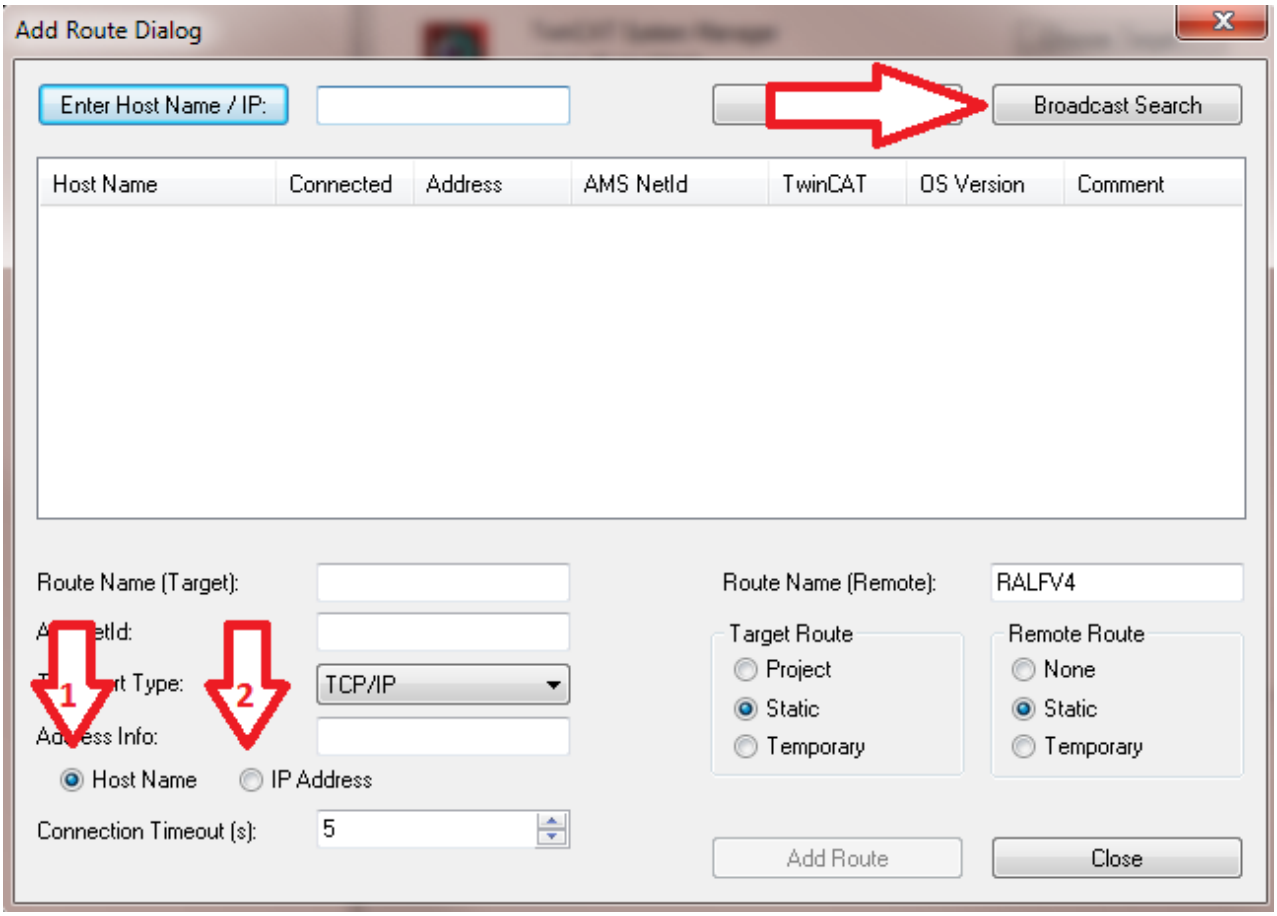
6. In the System Manager, click on the PC symbol (Choose Target System) or press >F8<




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

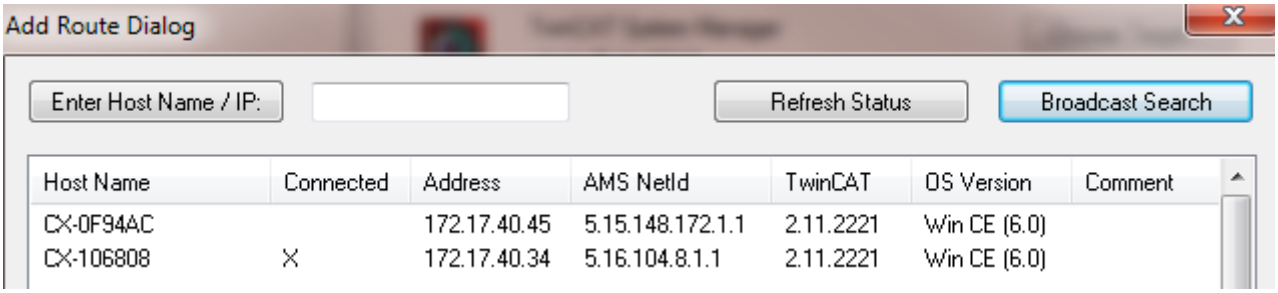



8. Select Option 1 if you have addressed via DHCP or Option 2 in case of DHCP or local IP address. Then click on "Broadcast search".




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.




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*.

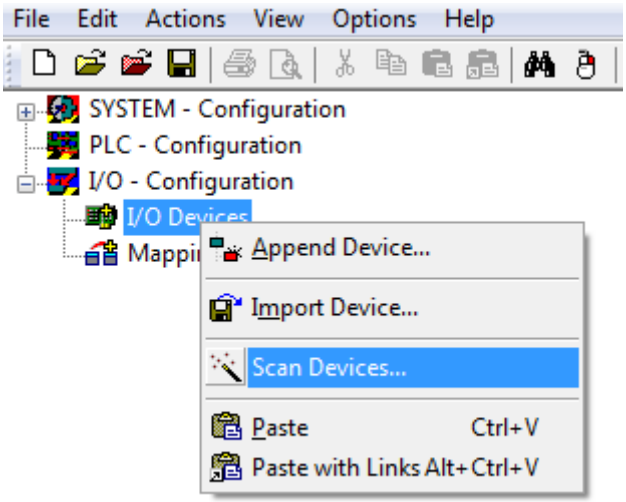
10. Next, the CX should appear in the list of the devices; select it and confirm by clicking on OK.

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.

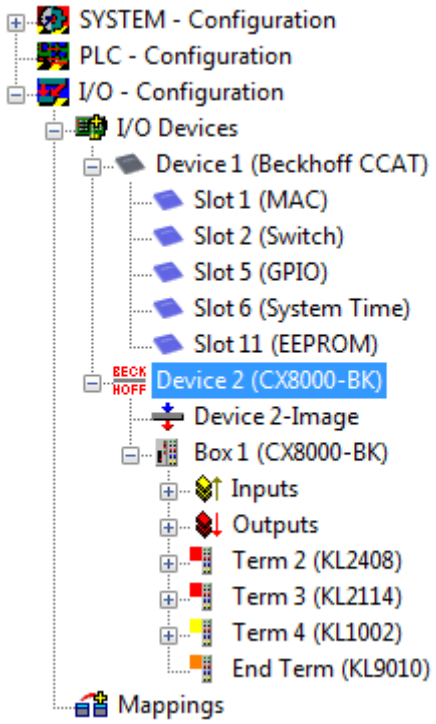
13. The setting at the bottom right must now change to blue and the TC LED on the CX80xx must now also light up blue.

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

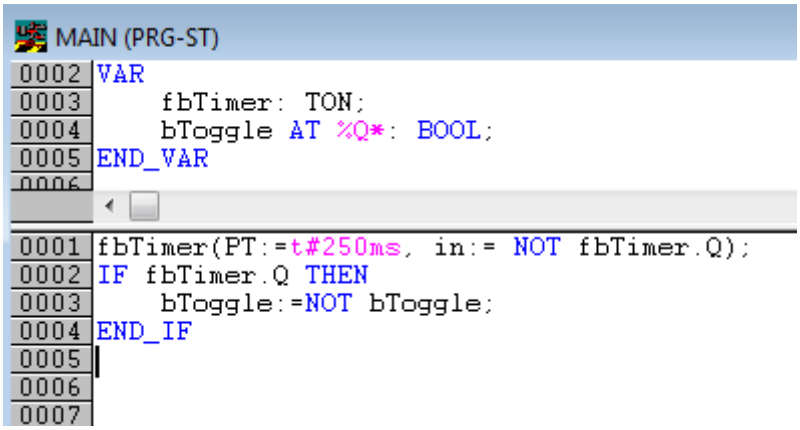



15. A message appears, informing you that not everything will be automatically detected.

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.



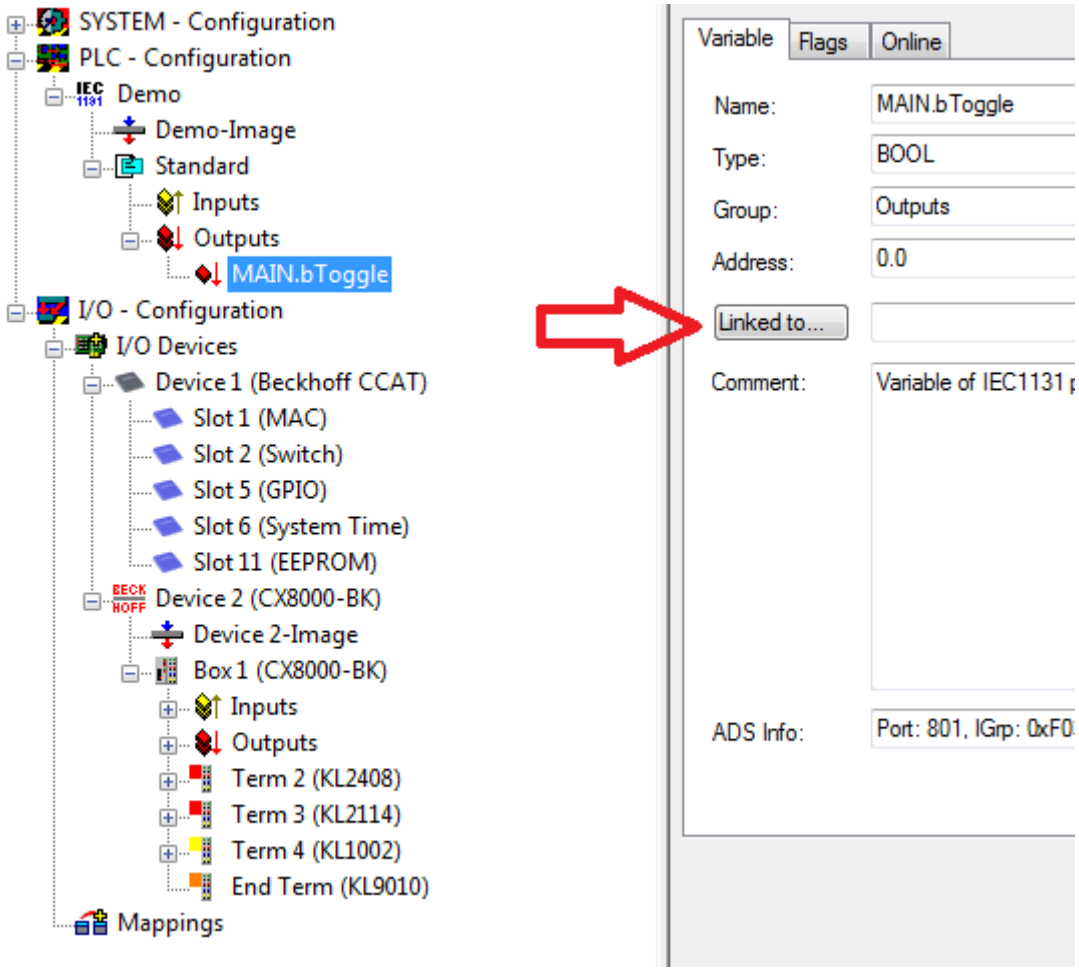

17. Now we come to the programming. To do this, open the PLC Control and select File -> New. The PLC Control asks you for the target system. Select CX (ARM). Afterwards it asks you for the function block; set the ST language (structured text). Now write a small program...




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...".




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.

| | |
|---|---|
| **Note** | **Prerequisites**<br>• Please make sure before the update that your CX80xx supports the image that you want to load.<br>• 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.

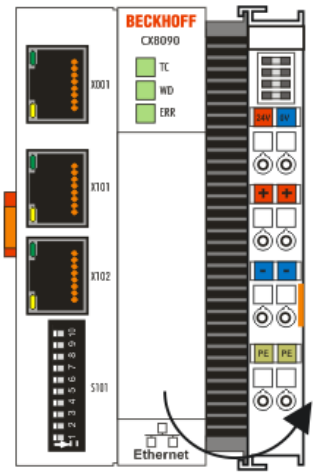
| | | | |
|---|---|---|---|
| Licenses | 30.07.2012 13:40 | Dateiordner | |
| RegFiles | 30.07.2012 13:40 | Dateiordner | |
| System | 30.07.2012 13:40 | Dateiordner | |
| TwinCAT | 30.07.2012 13:40 | Dateiordner | |
| UPnP | 30.07.2012 13:40 | Dateiordner | |
| www | 30.07.2012 13:40 | Dateiordner | |
| 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 |

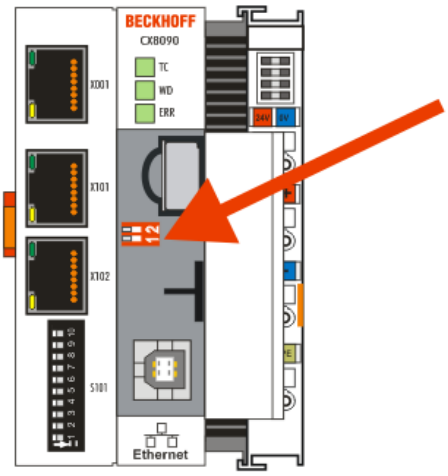
### Update via USB

| | |
|---|---|
| **CAUTION** | **USB port as ignition source in potentially explosive atmospheres**<br>Gases or dusts can be ignited by a spark discharge when the USB port is used.<br>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 potentially explosive atmospheres, have the following IDs:

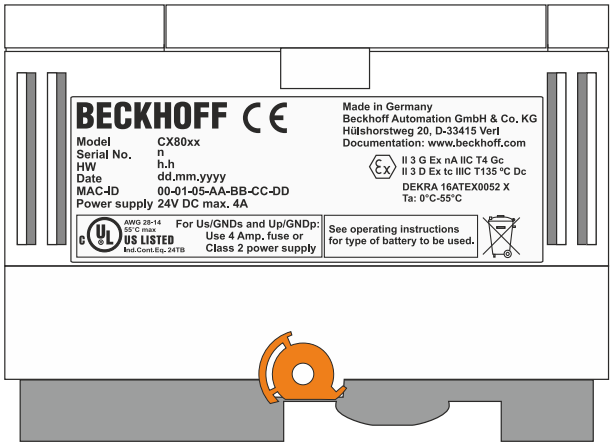
II 3 G Ex nA IIC T4 Gc

II 3 D Ex tc IIIC T135 ºC Dc

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:




Legend:

| | |
|---|---|
| n: | Serial number, consecutive number |
| h: | Hardware version, ascending number |
| dd: | Production day |
| mm: | Production month |
| yyyy: | Production year |

## 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:

**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 Support and Service

Beckhoff and their partners around the world offer comprehensive support and service, making available fast and competent assistance with all questions related to Beckhoff products and system solutions.

**Beckhoff's branch offices and representatives**

Please contact your Beckhoff branch office or representative for local support and service on Beckhoff products!

The addresses of Beckhoff's branch offices and representatives round the world can be found on her internet pages:
http://www.beckhoff.com

You will also find further documentation for Beckhoff components there.

Castel Maggiore (BO) 40013
Via Lirone 60/i

e-mail: info@eurotecsrl.com
Tel: +39 051/6321595
Website: www.eurotecsrl.com

C.F.: 04103780377
P.I.: 01596581205