BECKHOFF New Automation Technology

Manual | EN

CXxxxx-M310/B310

Profibus Optional Interface for CX9020, CX5xx0 and CX20xx

2024-11-20 | Version: 1.0

Table of contents

1 Foreword

1.1 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®, EtherCAT P®, 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

1.2 Safety instructions

Safety regulations

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

Exclusion of liability

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

Personnel qualification

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

Description of symbols

In this documentation the following symbols are used with an accompanying safety instruction or note. The safety instructions must be read carefully and followed without fail!

⚠ DANGER

Serious risk of injury!

Failure to follow the safety instructions associated with this symbol directly endangers the life and health of persons.

⚠ WARNING

Risk of injury!

Failure to follow the safety instructions associated with this symbol endangers the life and health of persons.

⚠ CAUTION

Personal injuries!

Failure to follow the safety instructions associated with this symbol can lead to injuries to persons.

NOTICE

Damage to the environment or devices

Failure to follow the instructions associated with this symbol can lead to damage to the environment or equipment.

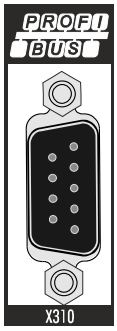
Tip or pointer

This symbol indicates information that contributes to better understanding.

1.3 Documentation issue state

Version	Modifications
1.0	First version

2 PROFIBUS system overview



The Beckhoff Embedded PCs can be ordered ex works with an optional interface, e.g. PROFIBUS, CANopen or RS232. Some of the optional interfaces can be delivered as master or slave.

The following Embedded PCs can be ordered with an optional interface:

- CX9020
- CX50x0
- CX51x0
- CX20x0

PROFIBUS master (M310)

The optional interface M310 is a PROFIBUS master and enables a segment-like construction of control structures in large plants and machines. Further Beckhoff fieldbus components such as Bus Couplers, Bus Terminal Controllers, drive components, etc. can be used with an Embedded PC for configuring control structures.

Fieldbus masters are used for decentralized collection of process data and signals in large machines and plants. The number of slaves that can be connected to the master is only limited by the respective bus system. Using master and slave connections makes it possible to link several Embedded PCs with each other via the fieldbus level.

The optional interfaces are detected, parameterized and configured in TwinCAT, and the connected I/O components are added. TwinCAT is also used for diagnostics.

PROFIBUS slave (B310)

The optional interface B310 is a PROFIBUS slave and enables an Embedded PC to be used as subordinate decentral controller for configuring complex or modular systems.

The PROFIBUS slave receives external process data from the master and processes them or returns data from its own process periphery to the master after processing.

Like the PROFIBUS master, the optional PROFIBUS slave interface is parameterized and configured in TwinCAT.

Functioning

PROFIBUS is a manufacturer - independent, open fieldbus standard with a wide range of applications in manufacturing and process automation. Manufacturer-independence and openness are guaranteed by the International standards EN 50170 and EN 50254.

Further Information
PROFIBUS User Organization (PNO): www.profibus.com

PROFIBUS allows devices from different manufacturers to communicate without the need for specially adapted interfaces. PROFIBUS is suitable both for fast, time-critical applications and for complex communication tasks.

In PROFIBUS DP, central control devices (e.g. Industrial PCs or PLCs) communicate via a fast serial connection with decentralized input and output modules. RS485 is the most frequently used transmission technique, using a screened twisted pair cable. Data is mainly exchanged cyclically, although acyclic services (DP-V1) are available for paramétrisation and diagnosis.

PROFIBUS DP offers short system response times: at a transmission rate of 12 Mbaud, less than 2 ms are required to transmit 512 bits each of input and output data to 32 devices.

All Beckhoff PROFIBUS devices feature a high-performance protocol implementation and are certified by the PROFIBUS user organization (PNO).

PROFIBUS distinguishes the following device types:

Master devices determine the data traffic on the bus. A master may transmit messages without having received an external request when it is in possession of the bus access authorization (token). Masters are also referred to as active devices.

Slave devices are peripheral devices such as input/output devices, valves, drives, measuring transducers and the Beckhoff PROFIBUS slaves from the BK3xx0, BC3xx0, IPxxxx-B310, IL230x-B310 and IL230x-C310 series. They do not receive any bus access authorization, so that they are only allowed to acknowledge messages that have been received, or to send messages in response to a request from master. Slaves are referred to as passive devices. They only require a small proportion of the bus protocol, which means that they can be implemented with little effort.

PROFIBUS-DP

PROFIBUS DP is designed for efficient data exchange at the field level. The central automation devices such as PLC/PCs or process control systems communicate here over a fast serial link with decentralized field devices such as I/O, drives, valves, etc. Data is primarily exchanged with these distributed devices cyclically. The communication functions required for this are specified by the basic DP functions conforms to EN 50170.

Beyond these basic functions, PROFIBUS DP offers advanced acyclic communication services for paramétrisation and operation, for example, which are supported by Beckhoff PROFIBUS slaves of the IPxxxx-B310, IL230x-B310 and IL230x-C310 series. A central controller (master) cyclically reads the input information from the slaves and cyclically writes the output information to the slaves. The bus cycle time here should be shorter than the central automation system's program cycle time, which lies around 10 ms in many applications.

2.1 Configuration options

PROFIBUS DP allows single master or multi-master systems to be implemented. This permits a high level of flexibility in system configuration. A maximum of 126 devices (master or slaves) can be connected to one bus. A station address between 0 and 99 can be chosen for the Beckhoff PROFIBUS slaves from the IPxxxx-B310, IL230x- B310 and IL230x-C310 series. The specifications for the system configuration contain the number of stations, the assignment of the station addresses to the I/O addresses, data consistency of the I/O data, the format of the diagnostics messages and the bus parameters being used. Every PROFIBUS DP system consists of different device types. Three types of device are distinguished:

Class	Description
DP master class 1 (DPM1)	This involves a central controller that exchanges information cyclically with the decentral stations (slaves) in a specified message cycle. Typical devices include, for instance, programmable logic controllers (PLCs) or PCs.
DP master class 2 (DPM2)	Devices of this type are engineering, project design or operating devices. They are used for commissioning, for servicing and diagnosis in order to configure the connected devices, to evaluate measured values and parameters and to interrogate the status of devices.
DP slave	A PROFIBUS DP slave is a peripheral device (I/O, drive, measuring transducer, etc.) that reads input information and passes output information on to the peripherals. It is also possible to have devices that only handle either input or output information. The quantity of input and output information is device-dependent, and may not exceed 240 bytes of input data and 240 bytes of output data.

Mono master systems

In single master systems only one master is active on the bus in the operating phase of the bus system. The PLC controller is the central control element. The decentralised slaves are coupled to the PLC controller via the transmission medium. The shortest bus cycle time is achieved with this system configuration.

Basic device files (GSD)

In PROFIBUS DP, the performance characteristics of the devices are documented by the manufacturers and made available to users in the form of a device data sheet and of a basic device file. The structure, content and coding of these basic device files (GSD) is standardised. They make it easy to plan a project with any PROFIBUS DP slaves using project planning devices from various manufacturers. The PROFIBUS User Organisation (Profibus Nutzer Organisation - PNO) archives this information for all manufacturers, and will provide information about the GSD from any manufacturer on request. The GSD files are read by a PROFIBUS master configuration software, and appropriate adjustments are transferred to the PROFIBUS master. Please see the appropriate software manual from the master manufacturer for a description.

The Beckhoff GSD files may be obtained from the internet under http://www.beckhoff.de.

Diagnostic functions

The extensive diagnostic functions of PROFIBUS DP allow rapid fault localisation. Diagnosis of the Beckhoff Bus Coupler is not activated in the default setting of the type file or the GSD file. The diagnostic messages are transmitted over the bus and collated by the master.

They are divided into three levels:

Diagnosis type	Description
Station-related	Messages relating to the general operational readiness of a device such as over-temperature or under-voltage
Module-related	These messages indicate that diagnostic signals are pending within a specific I/O sub range of the device (e.g. an 8 bit output module)
Channel-related	Here the cause of an error is related to a single input/output bit (channel), such as a short circuit on output 2

The Beckhoff PROFIBUS slaves from the IPxxxx-B310, IL230x-B310 and IL230x-C310 series support the PROFIBUS DP diagnostic functions. Assessment of the diagnostic data by means of the controller depends on the support for the PROFIBUS master. Please refer to the device manuals for the master interfaces for details of how to handle the diagnosis.

Sync and Freeze Mode

In addition to the user data traffic related to the device, which is automatically dealt with by DPM1, a DP master has the option of sending control commands to one DP slave, to a group of them or to all of them at the same time. These control commands are transmitted as multicasts. These control commands can be used to specify the sync and freeze operating modes, in order to synchronise the DP slave. They permit event-controlled synchronisation of the DP slaves.

The DP slaves start **sync mode** when they receive a sync control command from the assigned DP master. In this operating mode, the outputs of all the addressed DP slaves are frozen at their current values. In the following user data transmissions, the DP slaves store the output data, but the output states themselves nevertheless remain unchanged. Only when the next sync control command is received from the master the stored output data is switched through to the outputs. Sync operation is ended with an unsync control command.

A freeze control command similarly causes the addressed DP slaves to enter **freeze mode**. In this operating mode the states of the inputs are frozen at their current value. The input data is only updated again when the DP master has sent the next freeze control command to the devices concerned. Freeze operation is ended with an unfreeze command.

System behaviour

The system behaviour is also standardised in PROFIBUS DP, so that devices can to a large extent be interchanged. It is largely determined by the operating state of the DPM1. This can either be controlled locally, or over the bus by the project design device.

The following three major states are distinguished:

Operation mode	Description
Stop	There is no data traffic between the DPM1 and the DP slaves. The Bus Coupler only addresses the Bus Terminals once after the power has been switched on (none of the I/O LEDs are lit).
Clear	The DPM1 reads the input information from the DP slaves, and maintains the outputs of the DP slaves in a safe state (depending on the reaction to fieldbus errors, the green I/O LED is lit and the outputs are set).
Operate	The DPM1 is in a data transfer phase. In the course of cyclic data traffic the inputs of the DP slaves are read and the output information is transmitted to the DP slaves (the green I/O LED is lit).

The DPM1 sends its local status at a configurable time interval using a multicast command cyclically to all the DP slaves that have been assigned to it. The reaction that the system has to the occurrence of an error during the DPM1's data transfer phase, such as the failure of a DP slave, is specified in the *Auto-Clear* operating parameter. If this parameter is set to *True*, then the DPM1 switches the outputs of all the

associated DP slaves into a safe state as soon as one DP slave is no longer ready for the transfer of user data. The DPM1 then switches into the Clear state. If the parameter is *False* then the DPM1 remains in the operating state even after a fault, and the user can himself specify the system's reaction.

Data traffic between the DPM1 and the DP slaves

The data traffic between the DPM1 and the DP slaves that have been assigned to it is automatically executed by the DPM1 in a specified, continuously repeated sequence. The user specifies the assignment of a DP slave to the DPM1 when the bus system's project is being planned. Those DP slaves that are included in or excluded from the cyclic user data traffic are also defined.

The data traffic between the DPM1 and the DP slaves is divided into the paramétrisation, configuration and data transfer phases.

Before a DP slave is included in the data transfer phase, the DPM1 checks, in the paramétrisation and configuration phase, whether the theoretical configuration that has been planned agrees with the actual configuration of devices. The check requires the device type, the format and length information, as well as the number of inputs and outputs, to be in agreement. The user is thus provided with reliable protection against errors in paramétrisation. In addition to the transfer of user data, which is automatically carried out by the DPM1, it is possible to send new paramétrisation data to the DP slaves at the user's request.

Protection mechanisms

In the context of decentralised peripherals it is necessary, for reasons of safety and reliability, for the system to be given extremely effective functions to protect against incorrect paramétrisation or the failure of the transmission mechanisms. PROFIBUS DP uses monitoring mechanisms in the DP Master and in the DP Slaves. They are implemented in the form of time monitors. The monitoring interval is specified in when the DP system project is planned.

Protection mechanisms	Description
At the DP Master	The DPM1 monitors the slave's transfer of user data with the Data_Control_Timer. An individual monitoring timer is used for each assigned slave. The time monitor triggers if a proper transfer of user data does not take place within the monitoring interval. In this case the user is informed. If automatic error reaction is enabled (Auto_Clear = True) then the DPM1 leaves the *Operate* state, switches the outputs of the assigned slaves into a safe state, and then goes into the *Clear* operating mode.
At the DP Slave	The slave uses communication monitoring in order to detect errors of the master or in the transmission segment. If data is not transferred with the assigned master within the communication monitoring interval the slave switches the outputs into the safe state itself. The slave inputs and outputs further require access protection in multi-master systems, to ensure that direct access is only made from the authorized master. The slaves will make an image of the inputs and outputs available to other masters, and this can be read by any other master even if it does not have access authorization.

Ident number

Every DP slave and every DPM1 must have an individual identification number. This is required so that a DP master can identify the types of the connected devices without any significant protocol overhead. The master compares the identification numbers of the connected DP devices with the identification numbers in the project planning data specified by DPM2. The transfer of user data only starts if the correct device types are connected to the bus at the correct station addresses. This provides protection from project planning errors. Manufacturer-specific identification numbers are issued by the PROFIBUS User Organisation (PNO). The PNO administers the identification numbers along with the basic device data (GSD).

2.2 Communication protocols and services

In PROFIBUS DP systems a master (PLC, PC, etc.) usually communicates with many slaves (I/Os, drives, etc.); only the master actively accesses the bus (by sending unsolicited telegrams), while a DP slave only sends telegrams when requested by the master.

PROFIBUS DP

- **DP-StartUp**
  Before the master and the slave exchange data cyclically, parameter and configuration data are transferred from the master to the slaves during the DP startup. Once the parameter and configuration data have been sent, the master queries the diagnostic data of the slave until the slave indicates its readiness for data exchange. This process can take several seconds, depending on the scope of the calculations, which the slave has to carry out based on the parameter and configuration data it has received.
- **Parameter data**
  The master sends the parameter data to the slaves with the SetPrmLock request telegram. The SetPrmLock response telegram contains no data and only consists of one byte, which represents a short acknowledgement. The parameter data consist of DP parameters (e.g. the DP watchdog and the ID number), the DPV1/DPV2 parameters and application-specific parameters, which only have to be transferred once during startup. If an error is found in the parameter data, this is indicated in the diagnostic data, and the slave either remains in or enters the WAIT-PRM state.
- **Configuration data**
  The master sends the configuration data to the slaves with the ChkCfg request telegram. The ChkCfg response telegram contains no data and only consists of one byte, which represents a short acknowledgement. The configuration data describes the assignment of the DP modules to the cyclic I/O data that is to be exchanged between the master and slave via the Data_Exchange telegram in the cyclic data exchange phase. The order of the DP modules attached to a slave determines the order of the corresponding I/O data in the data exchange telegram.
- **Diagnostic data**
  The master requests the diagnostic data with a SlaveDiag request telegram. The slave sends the diagnostic data with a SlaveDiag response telegram. The diagnostic data consist of the standard DP diagnostic data (e.g. state of the slave, ID number) and application-specific diagnostic data.
- **Cyclic data exchange**
  At the core of the PROFIBUS DP protocol is the cyclic data exchange, during which the master exchanges I/O data with each slave within a PROFIBUS DP cycle. The master sends the outputs to each slave with a DataExchange request telegram. The slave returns the inputs in a DataExchange response telegram. This means that all the output and/or input data is transmitted in one telegram, in which the DP configuration (the sequence of DP modules) specifies the assignment of the output and/or input data to the slave's actual process data.
- **Diagnosis during cyclic data exchange**
  A slave can send a diagnostics signal to the master during cyclic data exchange. In this case, the slave sets a flag in the DataExchange response telegram, whereby the master recognizes that there is new diagnostic data at the slave. The master then fetches these data with a SlaveDiag telegram. The diagnostic data is therefore not available at the same time as the cyclic I/O data, but always delayed by at least one DP cycle.
- **Synchronisation with Sync and Freeze**
  The Sync and Freeze commands in the GlobalControl request telegram allow the master to synchronise the activation of the outputs (Sync) or the reading of the inputs (Freeze) in a number of slaves.
  If sync commands are used, the slaves are initially switched to sync mode (this is acknowledged in the diagnostic data). The I/O data are then exchanged sequentially exchanged with the slaves via DataExchange telegrams. When the sync command is sent in the GlobalControl telegram, the slaves issue the last received outputs. In Freeze operation a Freeze command is first sent in the GlobalControl telegram, in response to which all the slaves latch their inputs. These are then fetched sequentially by the master in the DataExchange telegram.
- **States in the master**
  The master distinguishes between the CLEAR state (all outputs are set to the Fail_Safe value) and the OPERATE state (all outputs have the process value). The Master is usually switched into the CLEAR mode when, for instance, the PLC enters STOP.

- **Class 1 and Class 2 DP Masters**
  The Class 1 master refers to the controller that carries out cyclic I/O data exchange with the slaves, while a Class 2 master is a B&B device that generally only has read access to the slaves' I/O data.

PROFIBUS DPV1

PROFIBUS DPV1 refers primarily to the acyclic read and write telegrams, with which data sets in the slave are acyclically accessed. A distinction between a Class 1 and a Class 2 master is also made for DPV1.

- **Class 1 (C1)**
  The acyclic C1 connection is established with the DP startup during cyclic DP operation. Acyclic DPV1 C1 read and write telegrams can be sent from the master to the slave from the state WAIT-CFG of the slave.
- **Class 2 (C2)**
  In the case of C2, a second C2 master usually establishes a separate connection, independent of the cyclic DP connection, so that a manufacturer-specific project planning and diagnostic tool can access the slave data, for example.
  If two masters are used, please note that they share the bus access, and the temporal conditions are therefore less favorable compared with a single master.

2.3 Technical data - PROFIBUS

Optional interface M310

Technical data	M310
Fieldbus	PROFIBUS DP, DP-V1; DP-V2 (MC)
Data transfer rate	9,6k; 19,2k; 93,75k; 187,5k; 500k; 1,5M; 3M; 6M; 12 MBaud
Bus interface	1 x D sub-socket, 9-pin
Bus devices	max. 125 with repeater
max. process image	30.5 kbytes in / 30.5 kbytes out
Properties	PROFIBUS – different DP cycle times possible for each slave; error management for each device freely configurable

Optional interface B310

Technical data	B310
Fieldbus	PROFIBUS DP, DP-V1
Data transfer rate	9,6k; 19,2k; 93,75k; 187,5k; 500k; 1,5M; 3M; 6M; 12 MBaud
Bus interface	1 x D sub-socket, 9-pin
Extendable process image	Up to 15 virtual slaves in addition
max. process image	16 slaves x (240 bytes in / 240 bytes out)

3 Connection technology

The field of application of a fieldbus system is essentially determined by the choice of transmission medium and the physical bus interface. In addition to the requirements for transmission security, the expense and work involved in acquiring and installing the bus cable is of crucial significance. The PROFIBUS standard therefore allows for a variety of implementations of the transmission technology while retaining a uniform bus protocol.

Cable-based transmission:

This version, which accords with the American EIA RS-485 standard, was specified as a basic version for applications in production engineering, building management and drive technology. A twisted copper cable with one pair of conductors is used. Depending on the intended application area (EMC aspects should be considered) the shielding may be omitted.

Two types of conductor are available, with differing maximum conductor lengths (see the RS485 table).

RS-485 transmission according to the PROFIBUS standard	
Network topology	Linear bus, active bus terminator at both ends, drop lines are possible
Medium	Shielded twisted cable, shielding may be omitted, depending upon the environmental conditions (EMC)
Number of stations	32 stations in each segment without repeater. Can be extended to 125 stations with repeater
Max. bus length without repeater	100 m at 12 Mbit/s 200 m at 1500 kbit/s, up to 1.2 km at 93.75 kbit/s
Max. bus length with repeater	Repeaters can increase the bus length up to 10 km. The number of possible repeaters is at least 3 and can be up to 10 depending on the manufacturer.
Data transfer rate (adjustable in steps)	9.6 kbit/s; 19.2 kbit/s; 93.75 kbit/s; 187.5 kbit/s; 500 kbit/s; 1500 kbit/s; 12 Mbit/s
Connector	9-pin D-sub connector for IP20 M12 round connector for IP65/67

Cable-related faults

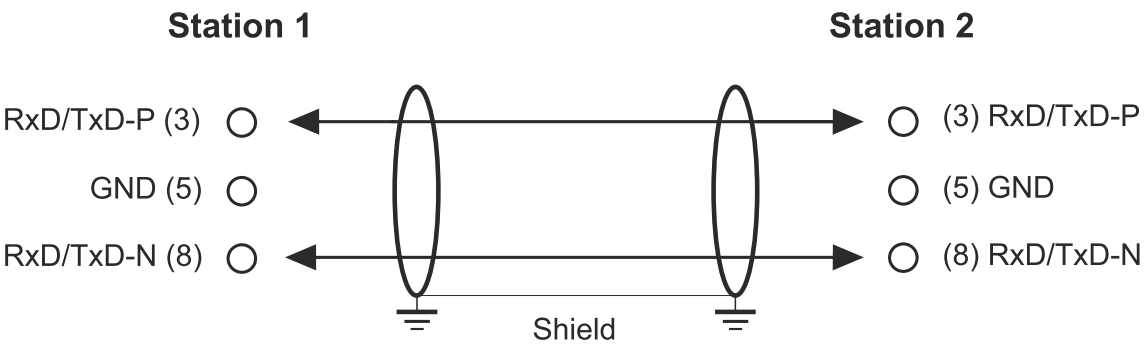
Note the special requirements on the data cable for baud rates greater than 1.5 Mbaud. The correct cable is a basic requirement for correct operation of the bus system. If a simple 1.5 Mbaud cable is used, reflections and excessive attenuation can lead to some surprising phenomena. For example, a connected PROFIBUS station does not establish a connection, but can do so again after disconnecting the neighboring station. Or there may be transmission errors when a specific bit pattern is transmitted. The result of this can be that when the equipment is not operating, PROFIBUS works without faults, but that there are apparently random bus errors after start-up. Reducing the baud rate (< 93.75 kbaud) corrects this faulty behavior.

If reducing the baud rate does not correct the error, then in many cases this can indicate a wiring fault. The two data lines may be crossed over at one or more connectors, or the termination resistors may not be active, or they may be active at the wrong locations.

Pre-assembled cable from BECKHOFF

The pre-assembled cables from BECKHOFF simplify installation considerably. Wiring faults are avoided, and commissioning is more rapidly completed. The BECKHOFF range includes fieldbus cables, power supply cables, sensor cables and accessories such as termination resistors and T-pieces. However, field-assembled plugs and cables are also available.

The following diagram shows the cabling between two stations and how the D-sub connections are assigned:



Termination resistors

In systems with more than two stations all devices are wired in parallel. The bus cable must always be terminated with resistors at the cable ends in order to avoid reflections and thus transmission problems.

Distances

The bus line is specified in EN 50170. This yields the following lengths for a bus segment.

Baud rate in kbits/ sec	9.6	19.2	93.75	187.5	500	1500	12000
Cable length in m	1200	1200	1200	1000	400	200	100

Drop lines up to 1500 kbaud <6.6 m; drop lines should not be used at 12 Mbaud.

Bus segment

A bus segment consists of at most 32 devices. 125 devices are permitted in a PROFIBUS network. Repeaters are required to refresh the signal in order to achieve this number. Each repeater is counted as one device.

IP-Link is the subsidiary bus system of the Fieldbus Boxes whose topology is a ring structure. The coupler modules (IP230x-Bxxx or IP230x-Cxxx) contain an IP-Link master to which up to 120 extension modules (IExxxx) can be connected. The distance between two modules may not exceed 5 m. When planning and installing the modules, note that because of the ring structure the IP-Link master must be connected again to the last module.

Installation guidelines

When installing the modules and laying the cable, observe the technical guidelines of the PROFIBUS user organization (PROFIBUS-Nutzerorganisation e.V.) for PROFIBUS-DP/FMS (see: https://www.profibus.com).

Check the PROFIBUS cable

A PROFIBUS cable (or a cable segment when using repeaters) can be checked with a few simple resistance measurements. The cable should meanwhile be disconnected from all stations:

1. Resistance between A and B at the beginning of the line: approx. 110 Ohm
2. Resistance between A and B at the end of the line: approx. 110 Ohm
3. Resistance between A at the beginning and A at the end of the line: approx. 0 Ohm
4. Resistance between B at the beginning and B at the end of the line: approx. 0 Ohm
5. Resistance between shield at the beginning and shield at the end of the line: approx. 0 Ohm

If these measurements are successful, the cable is okay. If, in spite of this, bus malfunctions still occur, this is usually a result of EMC interference. Observe the installation instructions of the PROFIBUS User Organization (www.profibus.com).

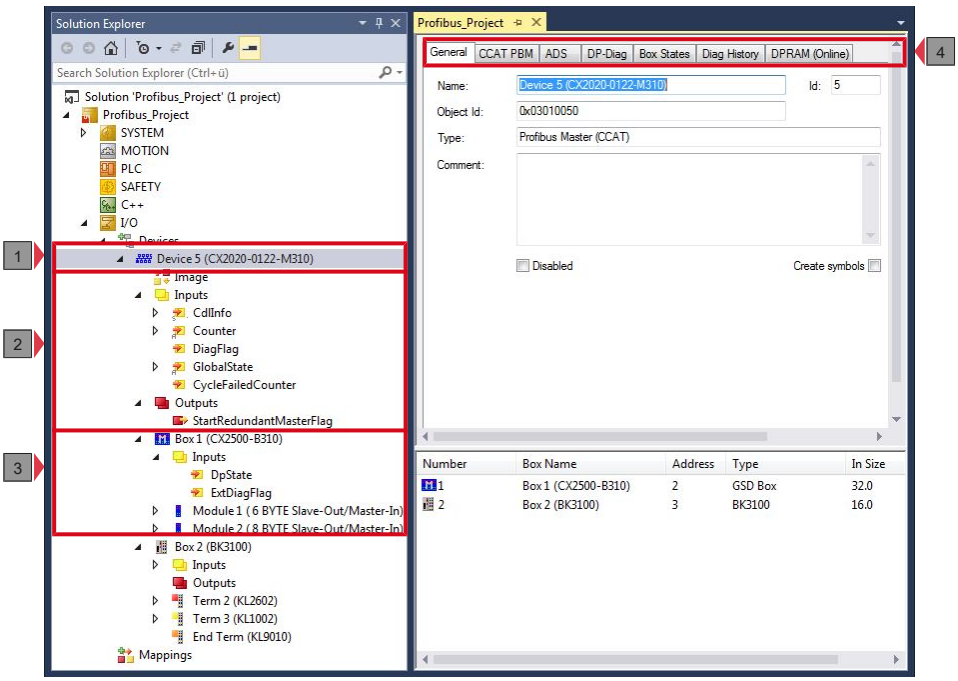
4 TwinCAT tabs

In TwinCAT, information and settings for the Profibus interface are added under tabs. The main TwinCAT tabs are described in this section. In addition, the section illustrates how the Profibus interface is displayed in the tree view under TwinCAT.

The tree view and the tabs for a Profibus interface are identical under TwinCAT2 and TwinCAT3.

4.1 Tree view

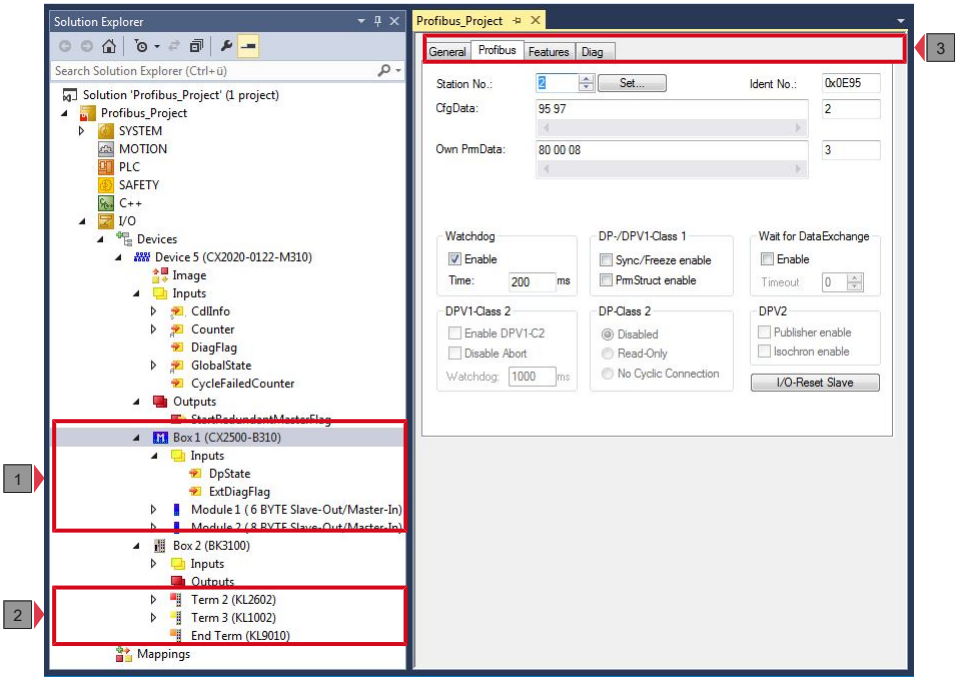
A Profibus master and a Profibus slave are displayed as follows in the tree view:



In this example the slave was linked to the master. TwinCAT was then scanned for the master, and the master was added in TwinCAT together with the slave.

Number	Description
1	The device name of the master is shown in brackets. All Profibus slaves are added under the master.
2	Under the Profibus master, status messages are listed as input variables and output variables. The variables can be linked with the PLC and used for diagnostic purposes (e.g. error codes, counters, etc.).
3	Profibus slaves are added under the master, labelled as box and numbered consecutively. The device name appears in brackets after it. Each Profibus slave has its own input variables for diagnostic purposes, which indicate the state of the communication (DPState, ExtDiagFlag).
4	Further settings for the Profibus master or slave can be implemented under the tabs. Other tabs are displayed, depending on whether the master or slave is selected in the tree view.

A Profibus slave and the corresponding tabs are shown as follows in the tree view:



Number	Description
1	Under the Profibus slave, status messages are listed as input variables. The variables can be linked with the PLC and used for diagnostic purposes. The inputs DpState and ExtDiagFlag are created for each slave box. In addition, process data for the data exchange are added under the slave.
2	All slaves are added under the master. This example shows the B3100 with connected Bus Terminals. All available terminals are displayed with their inputs and outputs.
3	Further settings for the Profibus slave can be implemented under the tabs. Other tabs are displayed, depending on whether the master or slave is selected in the tree view.

When the PLC process image is read, the variables for status messages and the variables under the process data can be linked with the variables from the PLC program. Double-click on a variable name in the tree view to open the link dialog. The link variables are identified with a small arrow icon.

Further information about TwinCAT can be found in the TwinCAT documentation on the Beckhoff website: www.beckhoff.com

4.2 Profibus master

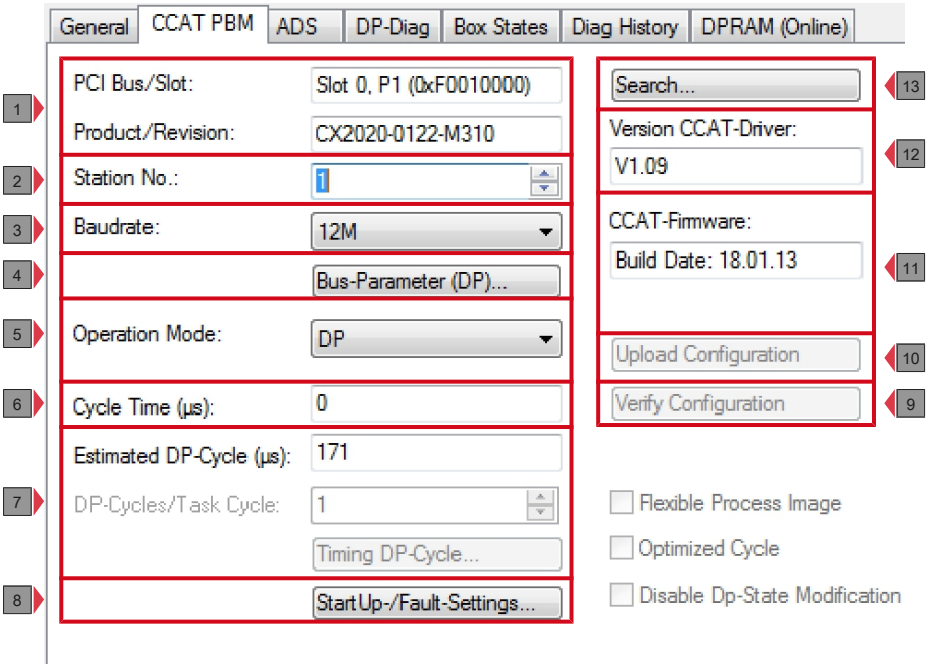
4.2.1 General

The General tab contains general information for a Profibus device, including name, type and ID.

No.	Description
1	Name of the Profibus device
2	Type of the Profibus device
3	Here you can add a comment (e.g. notes relating to the system component)
4	Here you can disable the Profibus device
5	Running No.

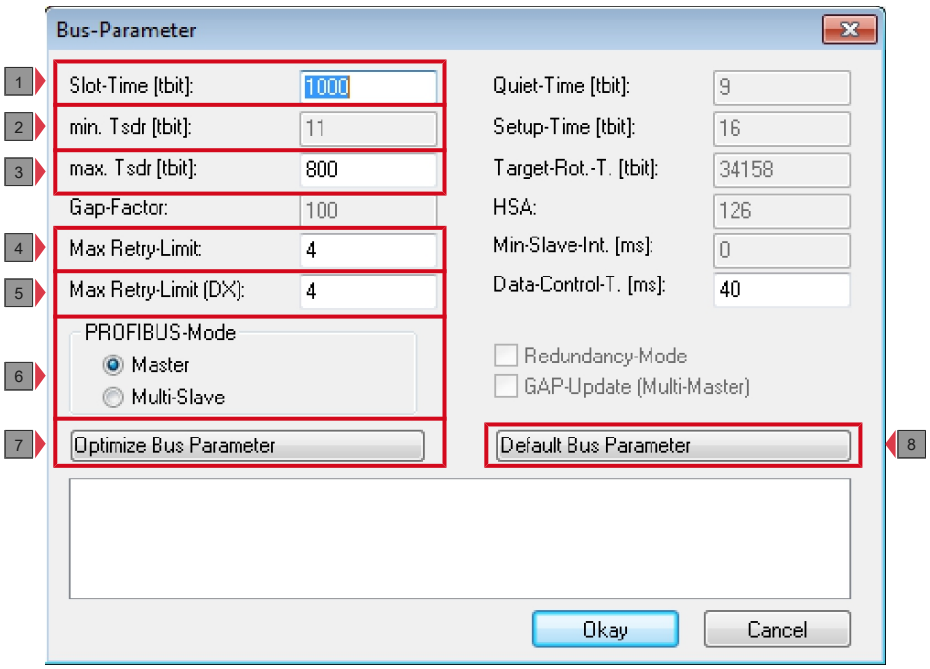
The Profibus device can be switched off via this tab. A comment field offers the option to add a label, in order to provide additional information on the device.

4.2.2 CCAT PBM



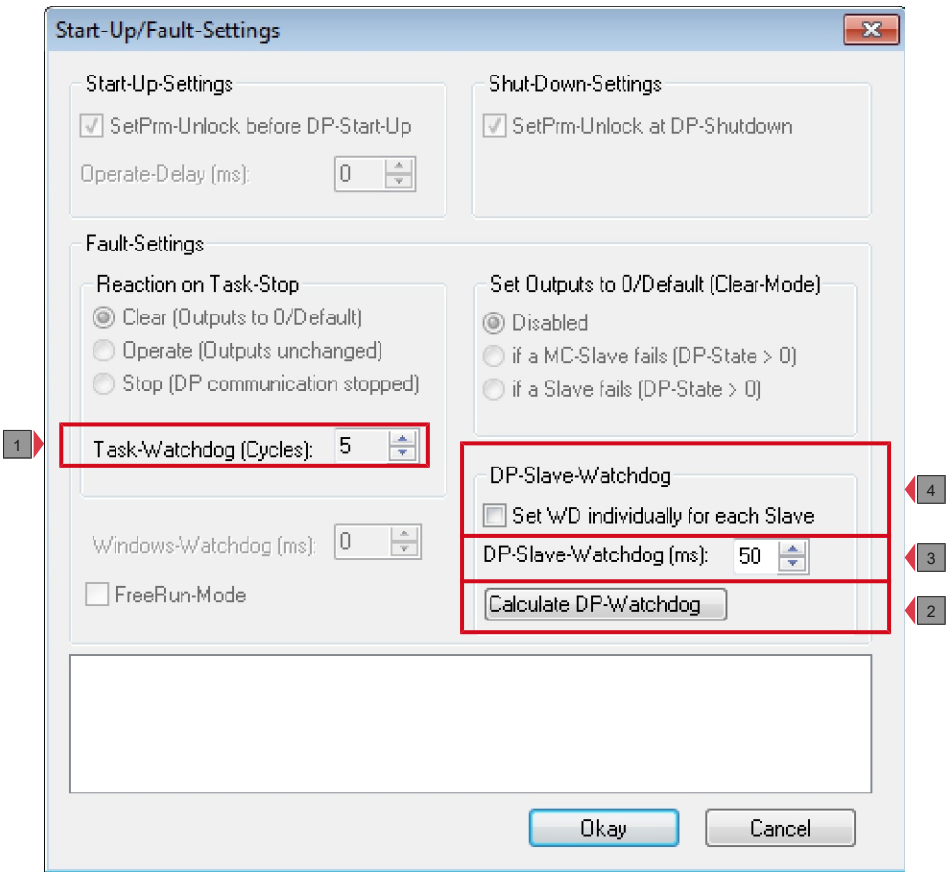
No.	Description
1	Name of the physical interface. Name and type of the Profibus device.
2	Unique station no. of the master.
3	The baud rate is set here.
4	Use this button to open the Bus Parameters window (see: Bus Parameters (DP) [▶ 22]).
5	Here you can select from the following three operating modes: DP, DP (equidistant/no GC) and DP/MC (equidistant). In all three operating modes the task with the highest priority linked to the corresponding device controls the PROFIBUS cycle and is therefore synchronized with the DP cycle.
6	Displays the cycle time of the corresponding highest priority task.
7	The expected Profibus cycle time is displayed here.
8	This button is used to open the Startup/Fault Settings window (see: Startup/Fault settings [▶ 23])
9	Here the Profibus is scanned and compared with the currently added boxes. Any deviations are displayed.
10	Here the PROFIBUS is scanned, and all devices that are found are added to the master. For Beckhoff boxes the configuration is read exactly, for external devices the system searches for the corresponding device master file.
11	Shows the current firmware version.
12	Shows the current version of the CCAT drivers.
13	The **Search** button is used to find and select the required physical interface, if not done automatically already.

4.2.3 Bus Parameters (DP)



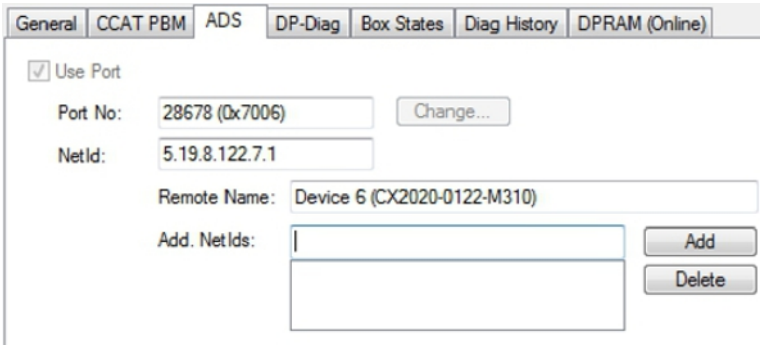
No.	Description
1	The Slot Time indicates how long the DP master will wait for a response from the DP slave before it sends either a repetition or the next telegram.
2	The minimum Tsdr indicates the minimum length of time for which the DP slave will wait with a response. This time is set for all the DP slaves during the DP start-up (the value range is 11-255 bit periods). The minimum Tsdr must be smaller than the maximum Tsdr.
3	The maximum Tsdr indicates the maximum length of time for which the DP slave may wait with a response. This time is set according to the DP slave's GSD file entries. The maximum Tsdr must be smaller than the slot time.
4	The Max Retry limit indicates how often a telegram is resent, if the addressed device does not respond. A value of at least 1 should be set, to ensure that for acyclic telegrams the telegram is repeated at least once in the event of an error.
5	Since the Data_Exchange telegram is repeated cyclically, a value of 0 could be used for the repetition of the Data_Exchange telegram here, in order to keep the cycle relatively constant in equidistant mode, even if there is no response from a device. However, in this case it would make sense to set the Features tab for the box such that lack of response of the slave would not lead to DATA EXCH being exited (see: Features [▶ 28]). If a device has not responded, this is indicated by the DpState, which in this case is not 0 for one cycle.
6	A distinction can be made between master functionality (default) and multi-slave.
7	This button is used to optimize the bus parameters.
8	This button is used to set the standard bus parameters.

4.2.4 Startup/Fault settings



No.	Description
1	The DP master changes automatically into the clear mode (the outputs of the slaves are set either to 0 or to the fail-safe value) when it ceases to receive an interrupt from the associated task (e.g. a PLC breakpoint has been reached, or the system has crashed). Here you can specify how many missing task cycles are tolerated before the master switches to Clear mode.
2	This button is used to set the DP watchdog for all DP slaves to an optimum value.
3	If the checkbox **Set WD individually for each Slave** is not ticked, the DP watchdog can be set to a uniform value for all slaves.
4	Here you can select whether the watchdog is to be set individually for each slave. If the checkbox is ticked, the watchdog can be set for each slave on the Profibus tab (see: Profibus [▶ 27]).

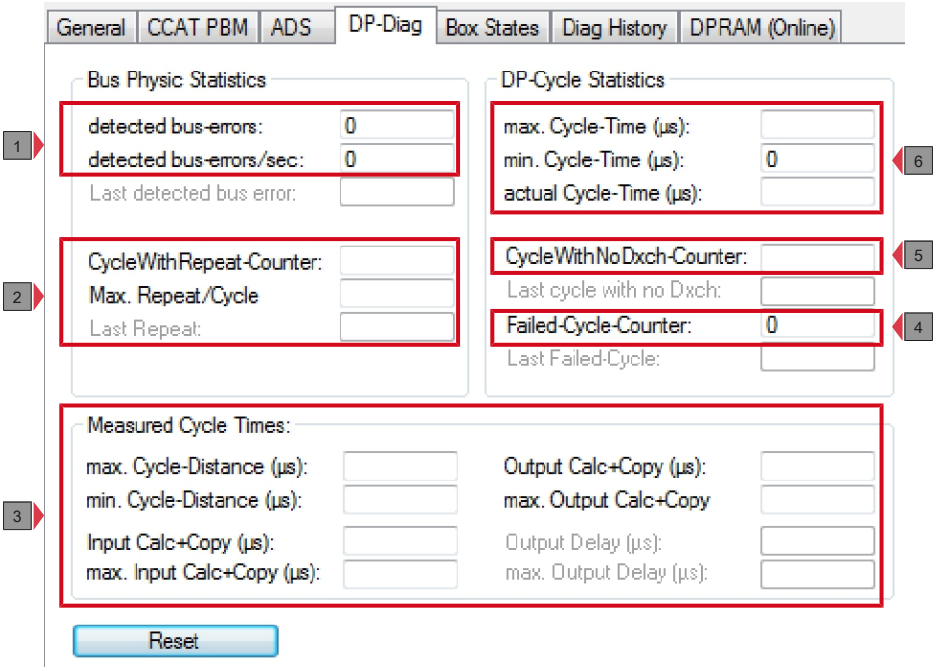
4.2.5 ADS



The Profibus master is an ADS device with its own Net ID, which can be modified here. All ADS services (diagnostics, acyclic communication) sent to the Profibus master must use this Net ID and port no.

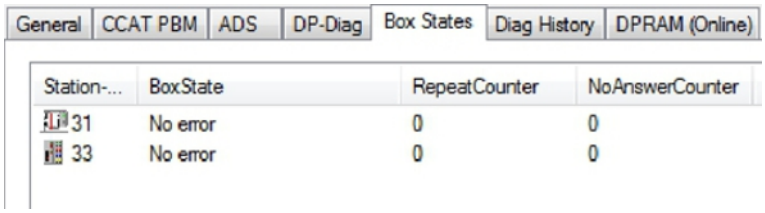
4.2.6 DP diag

Any problems with the cabling and the DP cycle times are displayed here.



No.	Description
1	detected bus errors: Displays the number of bus errors detected. If this counter is not 0, the cabling should be checked, if no PROFIBUS plug connectors were unplugged or connected. Unplugging of PROFIBUS connectors usually results in brief bus interference.
2	CycleWithRepeatCounter: Here, the number of PROFIBUS cycles is displayed, in which a telegram was repeated at least once. Repetitions indicate that there is something wrong with the bus hardware. Max. Repeat/Cycle: Here, the maximum number of repetitions within a cycle is displayed.
3	Measured Cycle Times: Displays the measured cycle times.
4	Failed-Cycle-Counter: The counter increments if the DP cycle was not complete before the next task cycle was started and all slaves are in data exchange (i.e. they have a DpState of 0).
5	CycleWithNoDxch-Counter: Increments if not all slaves exchange data (i.e. DpState is not 0).
6	max./min./actual Cycle-Time: Displays the maximum, minimum and current DP cycle times; only those cycles are taken into account, in which all slaves have exchanged data and no repetitions have occurred.

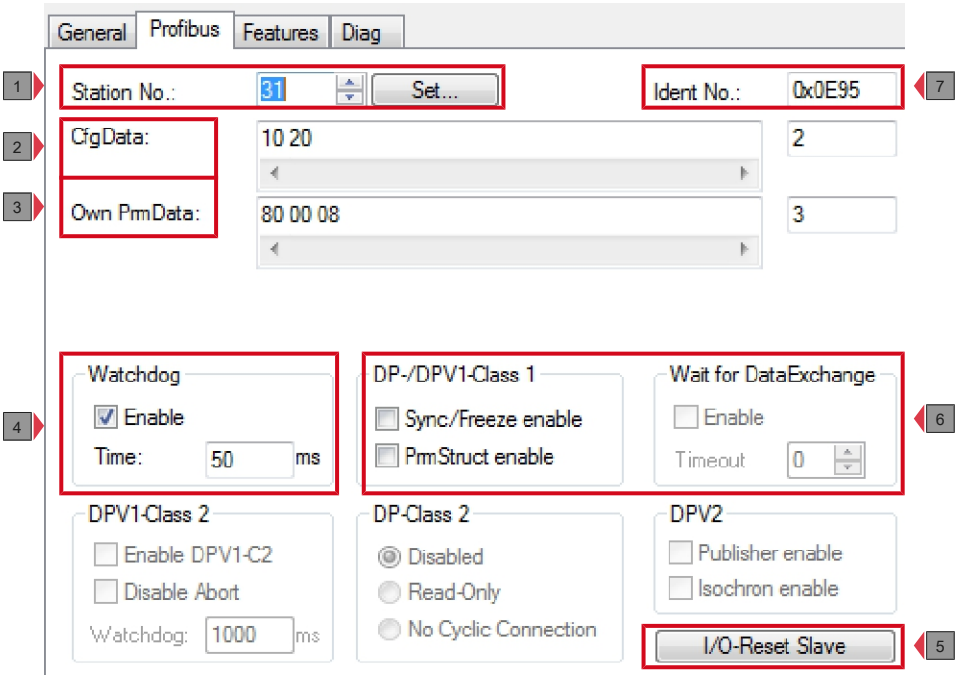
4.2.7 Box States



The connected Profibus slaves are listed in the Box States tab. This overview can be used to detect errors such as connection problems at the boxes.

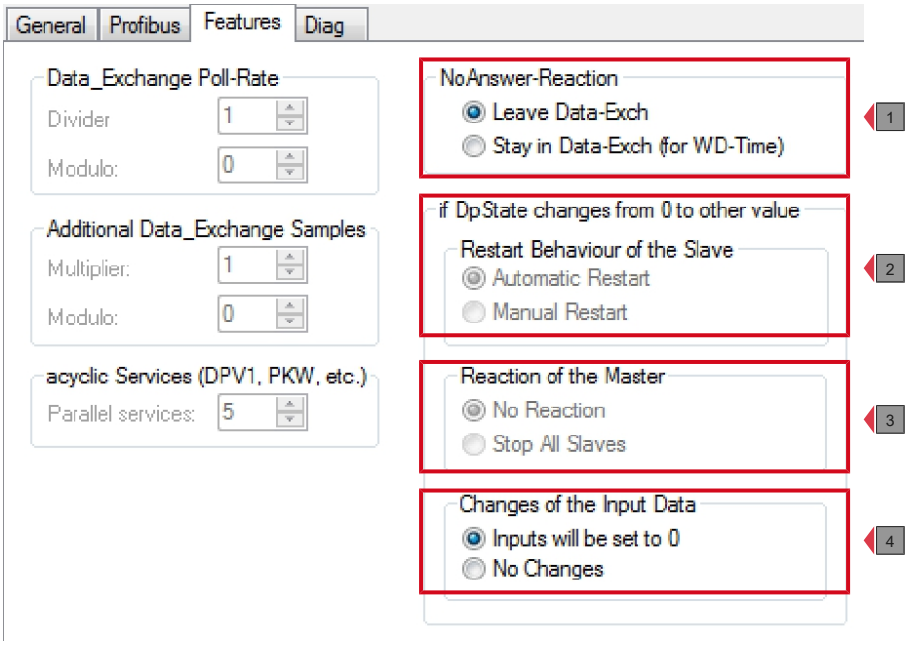
4.3 Profibus slave

4.3.1 Profibus



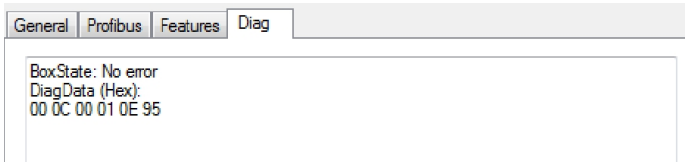
No.	Description
1	The hardware address set on the slave is entered here. There are slaves, for which the address cannot be set on the hardware, but only via the SetSlaveAddress service. In this case use the **Set** button to start a dialog for sending a SetSlaveAdress telegram and setting an address.
2	The current configuration data (resulting from the attached modules or terminals) as well as their length is displayed.
3	Enables editing of Profibus-specific parameter data. The size of the current parameter data is also displayed. The PrmData can usually be set as text (-> PrmData (text)) or for Beckhoff DP slaves partly via the “Beckhoff” tab.
4	Watchdog: Activates the DP watchdog. If the slave does not receive a DP telegram for the duration of the watchdog time with the watchdog switched on, it will automatically exit the data exchange. The minimum watchdog time depends on the DP cycle time and should be greater than the value calculated based on the following formula: **Estimated-Cycle-Time * 10.** For particularly critical outputs a DP watchdog of up to 2 ms can be set can for DP slaves, which support a watchdog base time of 1 ms. The DP watchdog time should be at least twice the maximum of **cycle time** and **estimated cycle time** (see: CCAT PBM [▶ 21]).
5	With this button, provided TwinCAT has been started, cyclic data exchange with the DP slave can be disabled and re-established immediately (corresponds to an IO reset but only for the one slave).
6	In operating mode DP/MC (equidistant) of the master, slaves can be operated with Sync and Freeze.
7	Here, the Ident number from the GSD file is displayed.

4.3.2 Features



No.	Description
1	Here you can set whether the slave should continue to send data (Stay in Data Exch), even if it does not respond or respond erratically. In this case the data exchange remains active until the slave fails to respond correctly within the watchdog time (watchdog function is enabled, see Profibus tab of the slave). Otherwise the data exchange is not interrupted until the slave has responded incorrectly 65535 times.
2	Here you can specify whether the slave should restart automatically or remain in WaitPrm state, once it has exited the data exchange.
3	For each slave you can specify whether exit from Data Exch should lead to a stop of the PROFIBUS cycle (all slaves exit the data exchange and enter the WaitPrm state. A restart is only possible through an IO reset or restart of the TwinCAT system).
4	For each slave it can be specified whether, on exiting of Data Exch (DpState not equal 0), its input data should be set to 0 or remain unchanged.

4.3.3 Diag



This tab offers an overview of the slave state and the Profibus connection. Here the diagnostic data of the preceding tabs can be displayed in consolidated form. The following and further important information can be viewed here:

BoxState: The current DpState is displayed here.

Receive-Error-Counter: Number of disturbed telegrams from the slave.

Repeat-Counter: Number of required repetitions due to missing or disturbed response from the slave.

NoAnswer-Counter: Number of telegrams that remained unanswered by the slave.

Last DPV1 error: Error-Decode, Error-Class, Error-Code and Error-Code 2 (see description of the DPV1 Error Codes).

5 Parameterization and commissioning

This documentation uses Profibus devices to illustrate the commissioning procedure. The configuration options shown in this section can be used for all Embedded PCs with Profibus interface.

The following devices are used in this documentation:

- CX2020-M310 (Embedded PC with optional Profibus master interface, D-sub socket, 9-pin)
- CX2500-B310 (Embedded PC with fieldbus module CX2500-B310 Profibus slave, D-sub socket, 9-pin)
- BK3100 (Profibus slave, D-sub socket, 9-pin)

The TwinCAT 2 or TwinCAT 3 software is used for configuring the devices.

For further information see the TwinCAT 2 and TwinCAT 3 documentation, which is available from the Beckhoff website:
www.beckhoff.de

5.1 Synchronizing Profibus

In TwinCAT Run mode the Profibus master is always synchronized with the highest priority task, with which variables are linked. Once the mapping was created, the cycle time of the corresponding task is displayed under **Cycle Time** on the CCAT PBM tab of the master. It is possible to set for the task whether the "I/O at task begin" should be updated or not.

Prerequisites for this step:

- A task created in TwinCAT, with which variables from the PLC project are linked.

The option "I/O at task begin" can be activated as follows:

1. Click on the corresponding task in the tree view on the left.

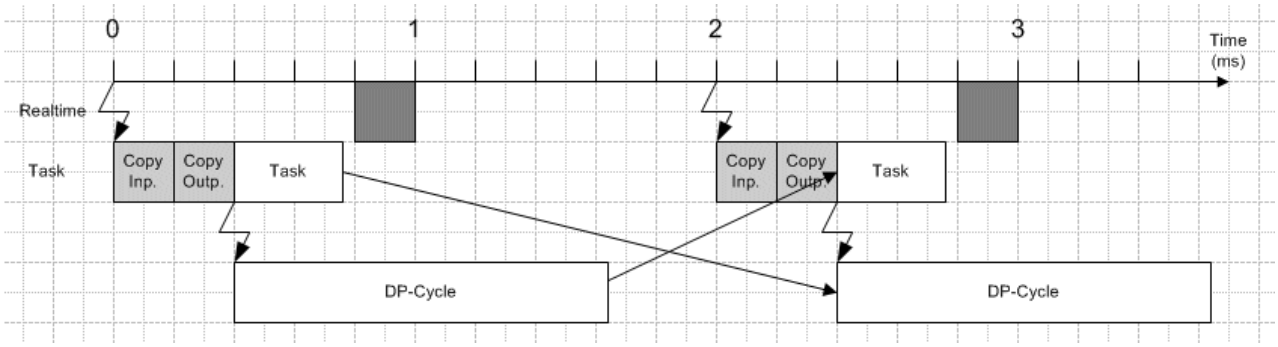
2. Click on the **Task** tab, then select the option "**I/O at task begin**".

⇨ You have successfully activated the option "I/O at task begin". The following section explains the effect of this setting on the task time and the DP cycle.

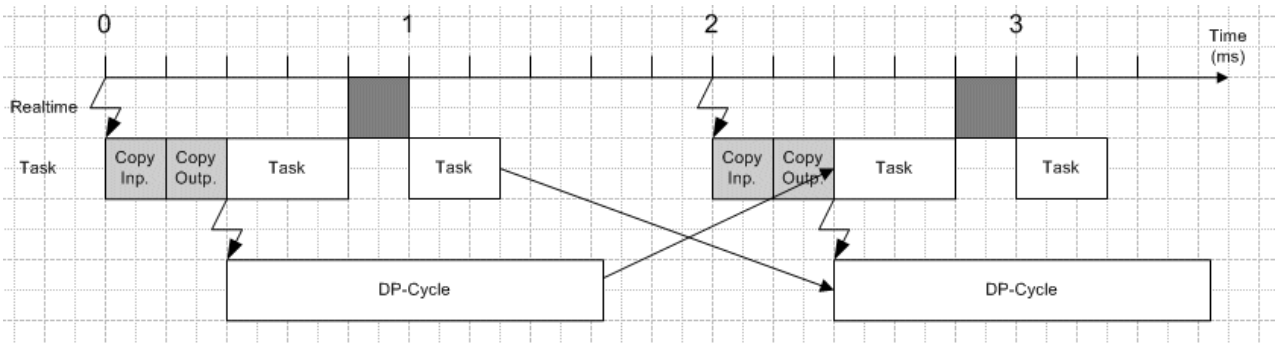
I/O at Task Start

If the setting **I/O at task begin** is activated (default for NC task), before the task starts the system checks whether the previous DP cycle was completed. The inputs and outputs are then copied (the outputs from the last task cycle are used), and the DP cycle is started.

Sample: Task cycle time 2 ms, real-time resources 80%.



If copying of the inputs and outputs and the task computing time of 0.8 ms is exceeded, task execution is interrupted, since the 80% real-time resources limit is reached:



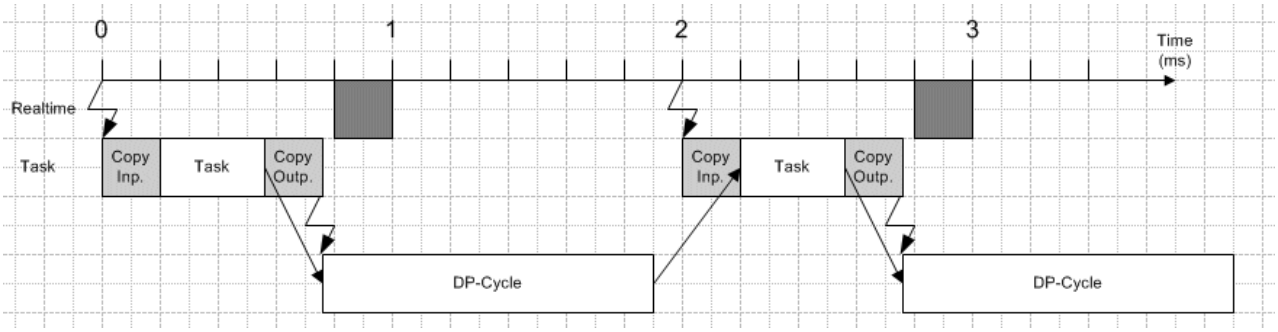
This case would still not be a problem, because the DP cycle was completed within the available time.

If the setting **I/O at task begin** is not activated, the sequence is somewhat more critical. The next example shows the effects.

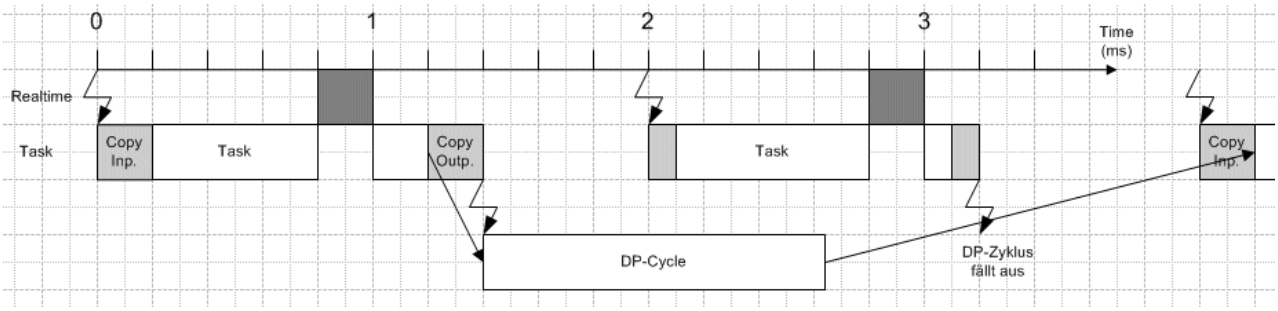
I/O not at Task Start

If the setting **I/O at task begin** is not selected for the task (default for PLC task), before the task starts the system checks whether the previous DP cycle was completed, and the inputs are copied. Task processing then commences. At the end of the task the outputs are copied, and the DP cycle is started.

Sample: Task cycle time 2 ms, real-time resources 80%.



Since in this case the task and the PROFIBUS have to share the bandwidth, exceedance of the real-time resources has a stronger effect than in the case "I/O at task begin":



The DP cycle starts later and is not completed in time before the next cycle. The system detects that the last DP cycle could not be completed. No inputs are copied before the start of the next task (the task uses the old inputs), and no outputs are copied after the task has been processed. The DP cycle is not restarted, which mean that a DP cycle is omitted.

The variable **CycleCounter** is used to indicate an omitted DP cycle (see: Master [▶ 66] diagnostics).

Comparison of I/O at task start and I/O not at task start

The advantage of the **I/O at task begin** setting is that the task and the DP cycle do not have to share the available bandwidth. The DP cycle starts very constant (the jitter corresponds to the TwinCAT jitter). If the setting **I/O at task begin** is not activated, it can easily happen that a DP cycle is omitted, and the temporal constancy of the DP cycle additionally depends on the jitter of the task processing.

The disadvantage of the **I/O at task begin** setting is that the dead time, i.e. the system reaction time, increases.

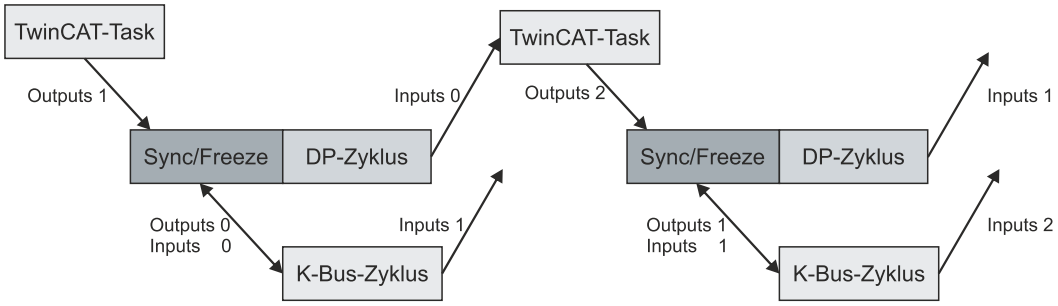
Sync/Freeze functionality

Sync is used for the simultaneous outputting of outputs for several slaves, Freeze is used for reading in inputs from several slaves simultaneously.

The TwinCAT process is as follows:

- The outputs are written at the beginning (I/O at the start of the task) or the end (I/O not at the start of the task) of the task cycle
- This will start the PROFIBUS cycle
- A Sync/Freeze telegram is sent at the start of the PROFIBUS cycle
- This will cause the Bus Couplers to start a K-Bus cycle with the outputs from the last task cycle and transfer the inputs from the last K-Bus cycle
- The master will then send the current outputs to each slave and pick up the transferred inputs
- The inputs are read at the start of the next task cycle
- etc.

Outputs and inputs are therefore always one cycle old.



For a master that is to be operated in Sync/Freeze mode, the option **DP/MC (Equidistant)** must be set in the **CCAT PBM** tab under **Operation Mode** (see: CCAT PBM [▶ 21]).

For a slave that is to be operated in Sync/Freeze mode, the option **Sync/Freeze enable** must be selected on the **Profibus** tab (see: Profibus [▶ 27]). The master always uses group 1 for the Sync/Freeze synchronization.

5.2 Parameterization with TwinCAT 2

This section illustrates how Profibus devices can be parameterized with the aid of TwinCAT 2. A total of three devices are used for the example, including a Profibus master, to which two Profibus slaves are connected.

First, the process of finding and selecting a target system in TwinCAT is illustrated. Next, a Profibus slave is added and parameterized in TwinCAT, and the Profibus address of the slave is set. Then a PLC project is created and added in TwinCAT. Then, the variables from the PLC project are linked with the hardware, and the finished configuration is loaded on the Profibus slave.

In the last step, the Profibus master is added in TwinCAT, and the two Profibus slaves are located via the master. The process of testing the Profibus networking is then illustrated.

5.2.1 Searching for target systems

Before you can work with the devices, you must connect your local computer to the target device. Then you can search for the devices with the help of the IP address or the host name.

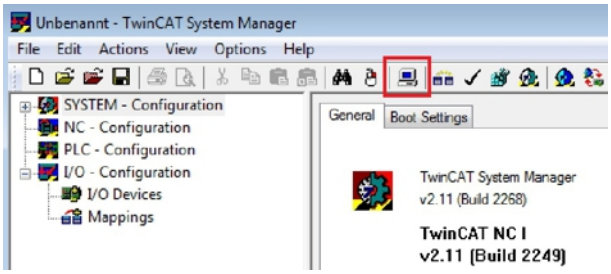
The local PC and the target devices must be connected to the same network or directly to each other via an Ethernet cable. In TwinCAT a search can be performed for all devices in this way and project planning subsequently carried out.

Prerequisites for this step:

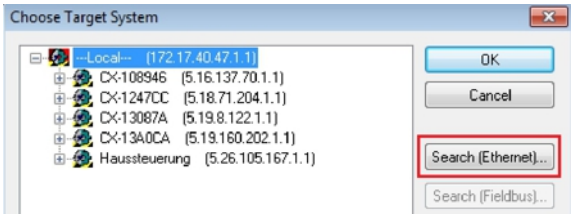
- TwinCAT 2 must be in Config mode.
- IP address or host name of the device. The host name is composed of CX- and the last 3 bytes of the MAC address. The MAC address is located on the side of the device.

Search for the devices as follows:

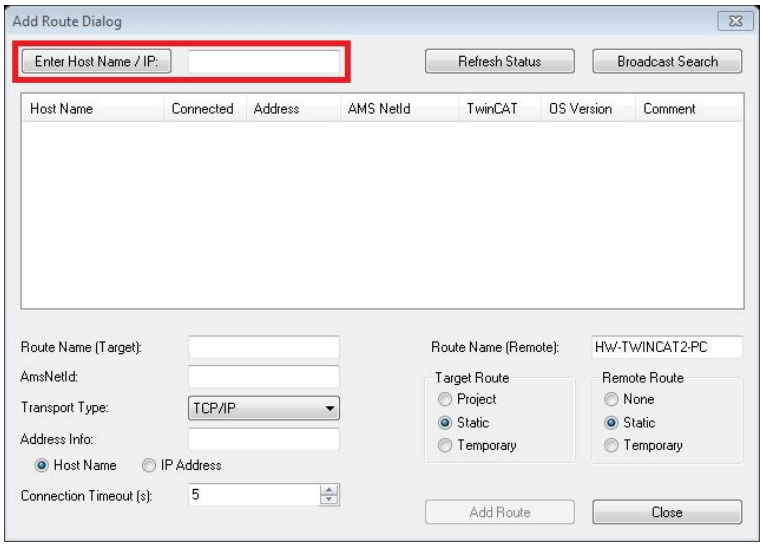
1. Click on **File > New** in the menu at the top.
2. Click on **Choose Target System** in the toolbar at the top.



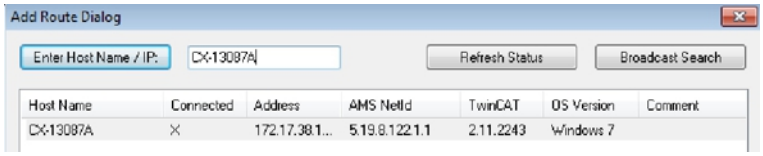
3. Click on **Search (Ethernet)**.



4. Type the host name or the IP address of the device into the **Enter Host Name / IP** box and press **[Enter]**.



5. Mark the device found and click on **Add Route**.

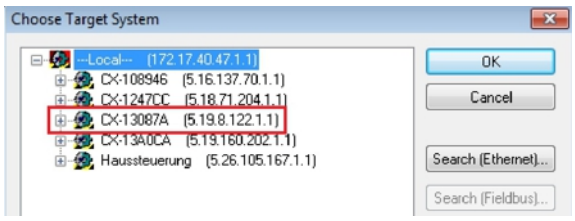


The Logon Information window appears.

6. Enter the user name and password for the CX in the **User Name** and **Password** fields and click **OK**.

The following information is set as standard in CX devices:
User name: Administrator **Password:** 1

7. If you do not wish to search for any further devices, click on **Close** to close the Add Route Dialog. The new device is displayed in the Choose Target System window.
8. Mark the device that you wish to set as the target system and click on **OK**.



⇨ You have successfully searched for a device in TwinCAT and inserted the device as the target system. The new target system is displayed in the bottom right-hand corner together with the host name and IP address (AMS Net ID).

Using this procedure you can search for all available devices and also switch between the target systems at any time. Next, you can append the device to the tree view in TwinCAT.

5.2.2 Adding a Profibus slave

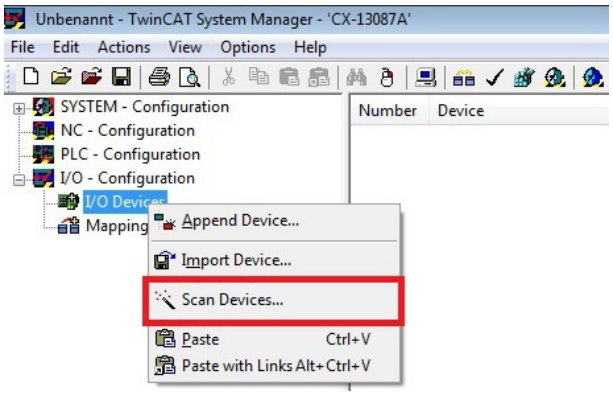
The example shows a CX2020 Profibus slave with CX2500-B310 fieldbus module, connected to the Profibus master. In order to ensure that the Profibus slave is configured and subsequently detected by the Profibus master with all inputs and outputs, the Profibus slave first must be added in TwinCAT.

Prerequisites for this step:

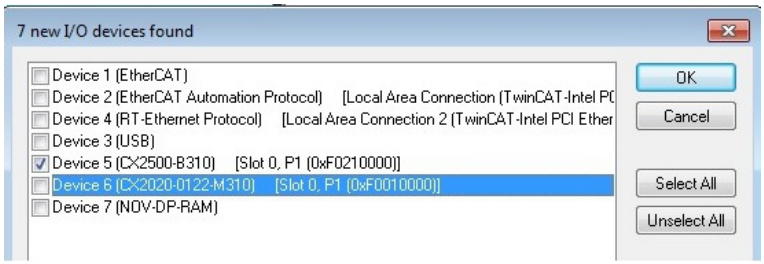
- A scanned and selected target device with Profibus slave. This example uses a CX2020 with CX2500-B310 fieldbus module.

Add the Profibus slave as follows:

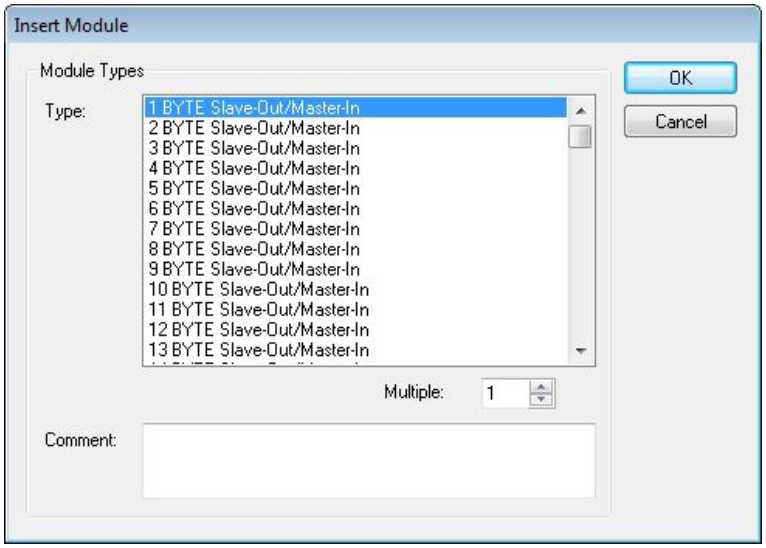
1. Start the System Manager.
2. In the tree view on the left, right-click on **I/O Devices**.
3. In the context menu click on **Scan Devices**.



4. Select the devices you want to use and confirm the selection with **OK**.



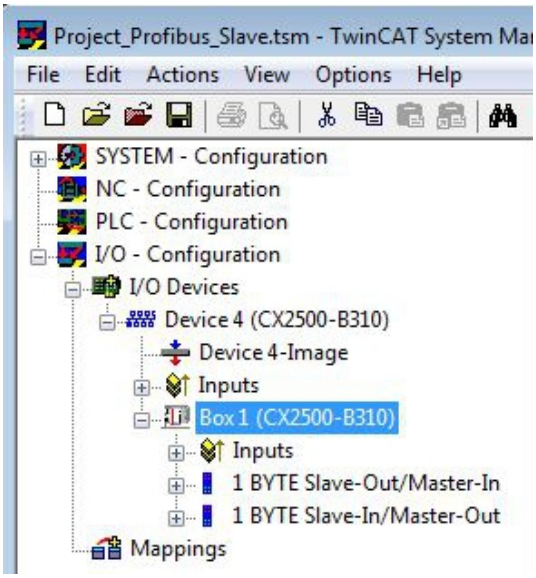
5. Confirm the request with **Yes**, in order to look for boxes.
   Box 1 (CX2500-B310) is integrated. The **Insert Module** window appears.
6. Add modules such as **1 BYTE Slave-Out/Master-In** and **1 BYTE Slave-In/Master-Out** for your process image.



7. Click on **Cancel** to close the **Insert Module** window.
8. Confirm the request whether to enable FreeRun with **Yes**.

⇨ The Profibus slave was successfully added in TwinCAT 2 and is displayed in the tree view with the inputs and outputs.

You can add further variables by right-clicking on the box and then clicking on Append Module in the context menu.



In the next step you can extend the process image by creating additional virtual slaves. Or you can set the address, once the slave configuration is complete.

5.2.3 Creating a virtual slave

Additional virtual slaves can be created on the same hardware interface. This enables more data to be exchanged with a Profibus master, or a connection with a second Profibus master can be established.

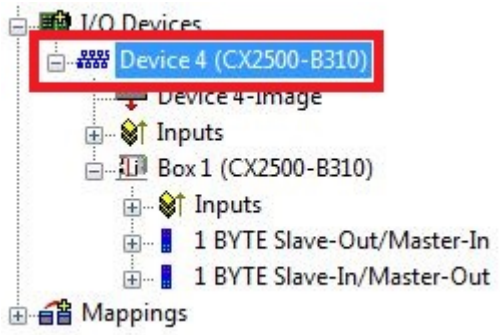
Each virtual slave is assigned a dedicated address via TwinCAT and is configured like an independent device for the Profibus master.

Prerequisites for this step:

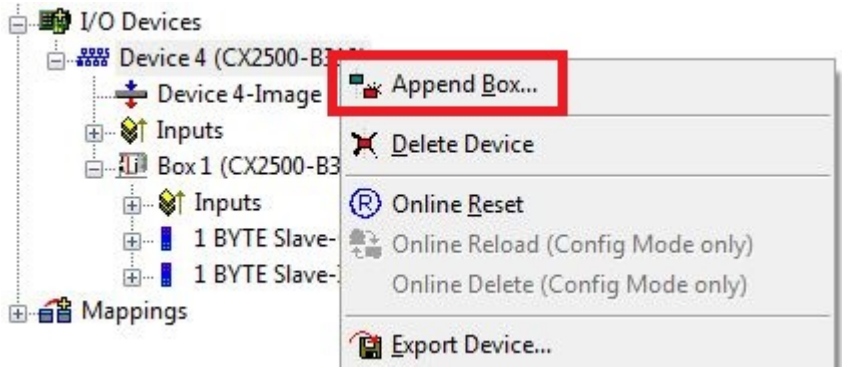
- A Profibus slave, created in TwinCAT.

Create a virtual slave as follows:

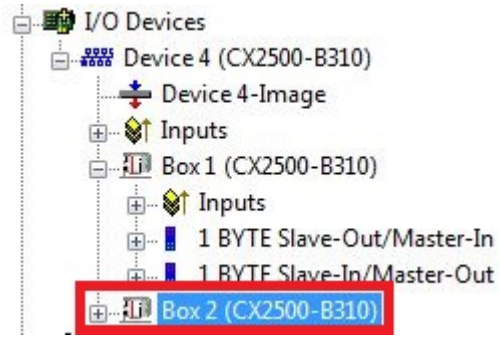
1. Right-click on a Profibus slave in the tree view on the left.



2. Click on **Append Box** in the context menu.



⇨ A further box (virtual slave) is created.



Variables for the virtual slave can now be created. In the next step you can set the address for the slave.

5.2.4 Setting the address

Once the Profibus slave was successfully added in TwinCAT, the address of the Profibus slave can be set. Devices with a DIP switch have a preset address. The address on the DIP switch must match the address set in TwinCAT.

For devices without DIP switch the address is only set in TwinCAT.

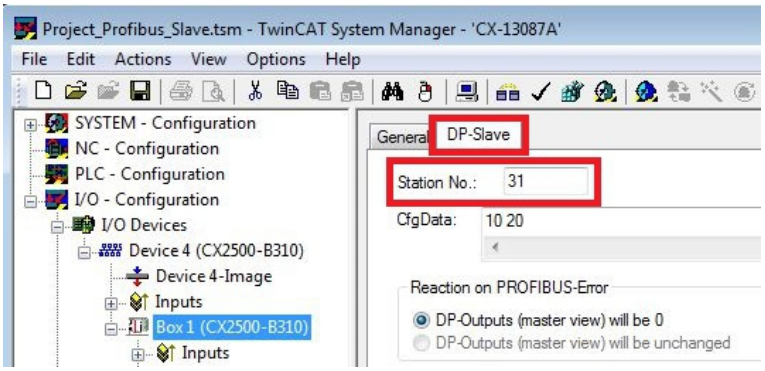
In this step the address is set in TwinCAT, so that the Profibus slave can be reached by the Profibus master via this address.

Prerequisites for this step:

- An added Profibus slave in TwinCAT.

Parameterize the Profibus slave as follows:

1. Click on a slave box.
2. Click on the **DP Slave** tab.
3. Enter a value for the Profibus address in the **Station No** field, e.g. „31“.



⇨ You have set the address successfully. The Profibus master can reach the Profibus slave with the set address.
Next, you can create a PLC project for the Profibus slave.

5.2.5 Creating a PLC project

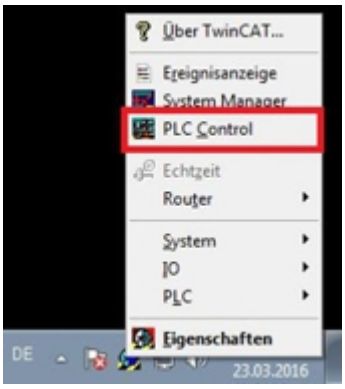
Use PLC Control to create a PLC project. The next steps describe how to create a PLC project in TwinCAT and add it in the tree view.

Prerequisites for this step:

- An Embedded PC, added in TwinCAT.

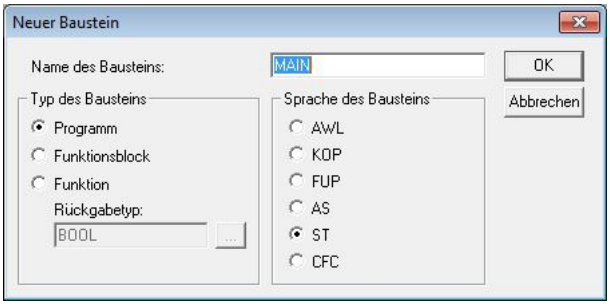
Create a PLC project as follows:

1. In the Start menu, right-click on the TwinCAT symbol.
2. In the context menu click on **PLC Control**.

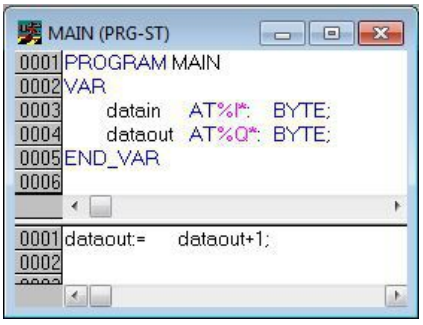


The TwinCAT PLC Control window appears.

3. In the menu click on **File > New** and select the option **PC or CX (x86)**.
4. Under **Block type** select the option **Program**, and under block language select the option **ST (Structured Text)**.



5. Write a small program.



6. Save the PLC project and click on **Project > Compile** in the menu.

⇨ Once the project has been compiled, a file with the extension .tpy is created in the same location as the project file. The file name of the new file is the same as the file name of the PLC project.

In the next step you can add the compiled PLC project in the TwinCAT System Manager.

Adding a PLC project

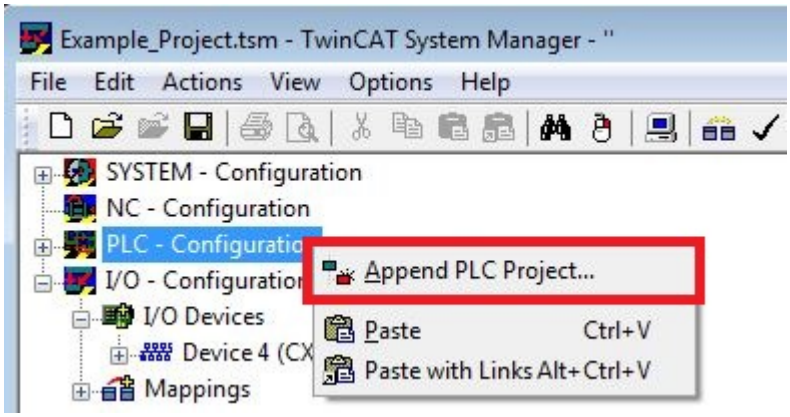
The PLC project can be added in the System Manager. The newly created variables from a PLC project are integrated in the System Manager and can be linked with the inputs and outputs of the hardware.

Prerequisites for this step:

- An Embedded PC, added in TwinCAT.
- A correctly compiled PLC project and a .tpy file.

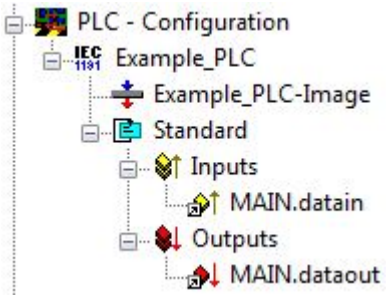
Proceed as follows:

1. Switch back to the System Manager window.
2. Right-click on **PLC – Configuration** in the tree view on the left.
3. In the context menu click on **Append PLC Project**.



4. Select a file with the extension .tpy in your system directory and confirm with **OK**.

   The PLC project is added in the tree view under PLC – Configuration. The variables defined in the project are shown under the inputs and outputs.



   In the next step you can link the variables with the hardware.

5.2.6 Linking variables

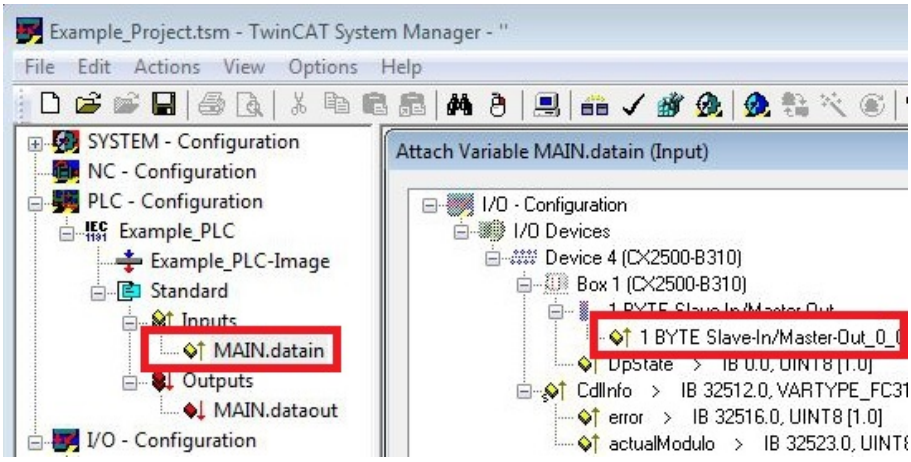
Once the PLC project was successfully added in the System Manager, you can link the newly created input and output variables from the PLC project with the inputs and outputs of your devices.

Prerequisites for this step:

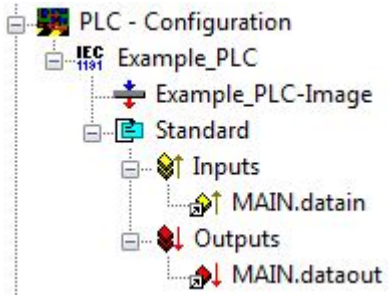
- An added PLC project in the System Manager.

Link the variables as follows:

1. Double-click on the input or output variables in the tree view under **PLC - Configuration**.
   The **Attach Variable** window appears and shows which inputs or outputs can be linked with variables.

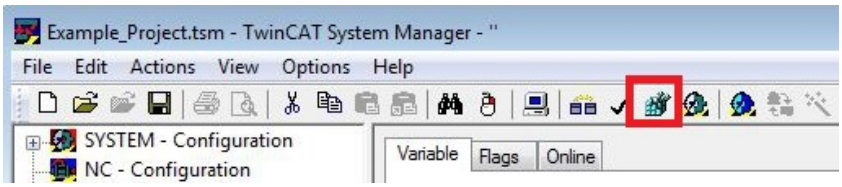


2. Double-click on the inputs or outputs in the Attach Variable window.
   The input variables are linked with the inputs of your hardware, and the output variables with the outputs.



   Variables that are already linked are indicated with a small arrow icon in TwinCAT.

3. In the toolbar click on **Activate Configuration**.



4. Confirm the request whether TwinCAT is to start in Free Run mode with **Yes**.

⇨ You have successfully linked variables with the hardware. Use Activate Configuration to save and activate the current configuration.

The configuration can now be loaded on the CX, in order to automatically start TwinCAT in Run mode, followed by the PLC project.

5.2.7 Load configuration to CX

Once all variables are linked, the configuration can be saved and loaded on the CX. This has the advantage that the PLC project is loaded and started automatically when the CX is switched on. The start of the previously created PLC project can thus be automated.

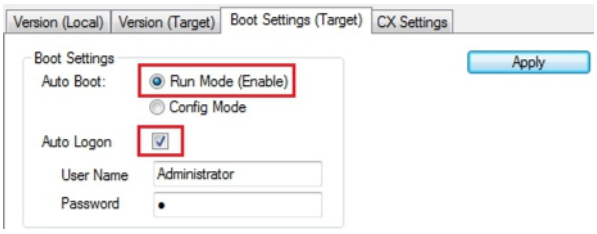
Prerequisites for this step:

- A completed PLC project, added in the System Manager.
- Variables from the PLC project, linked with the hardware in the System Manager.
- A CX selected as target system.

Load the configuration on the CX as follows:

1. In the tree view on the left click on **SYSTEM – Configuration**.
2. Click on the **Boot Settings (Target)** tab.

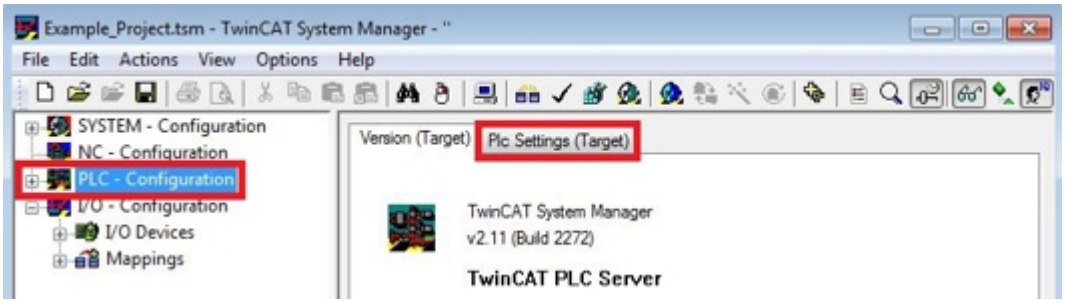
3. Under Boot Settings select the option **Run Mode (Enable)** and tick the **Auto Logon** checkbox.



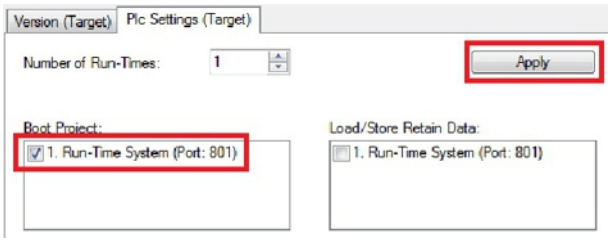
4. Enter the user name and password for the CX in the **User Name** and **Password** fields.
5. Click on **Apply**.
   The Logon Information window appears.

6. Re-enter the user name and the password and click **OK**.

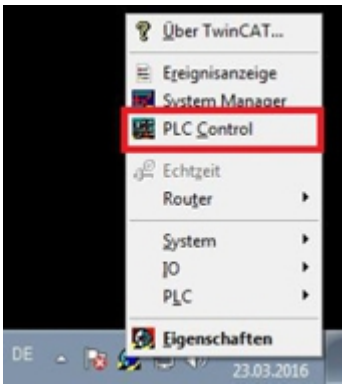
7. In the tree view on the left click on **PLC – Configuration**, then on the **PLC Settings (Target)** tab.



8. Select the Start PLC under Boot Project and click on **Apply**.

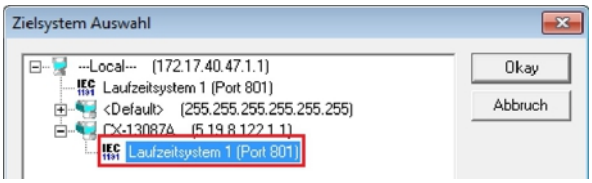


9. Start PLC Control and open the PLC project.



10. In the menu bar at the top click on **Online**, and then on **Choose Runtime System**.

11. Select the runtime system from the CX and click on **OK**.



12. In the menu bar at the top click on **Online**, then **Login**.
    The PLC project is logged in.

13. In the menu bar at the top click on **Online**, then **Create Boot Project**.

⇨ You have successfully loaded the CX configuration. From now on, TwinCAT will start in Run mode and the PLC project will start automatically.

Next, the master can be added in a new project in the System Manager and can then be used to find slaves that have already been set up.

5.2.8 Adding a Profibus master

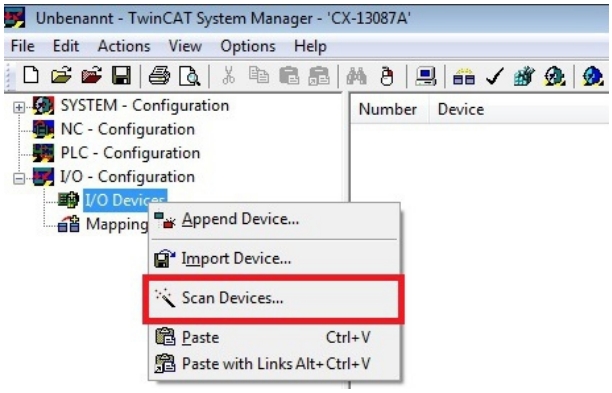
The Profibus master is added with the TwinCAT System Manager, like the other devices. The attached master can then be used to find all connected slaves. The following section illustrates how to add a Profibus master in TwinCAT.

Prerequisites for this step:

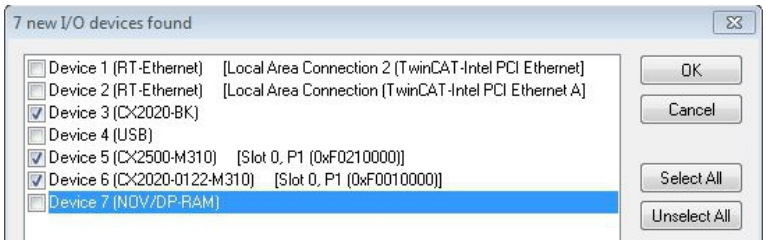
- TwinCAT must be in Config mode.
- A selected target system (in this example it is the Embedded PC CX2020-M310)

Add a Profibus master as follows:

1. Start the System Manager.
2. In the tree view on the left, right-click on **I/O Devices**.
3. In the context menu click on **Scan Devices**.

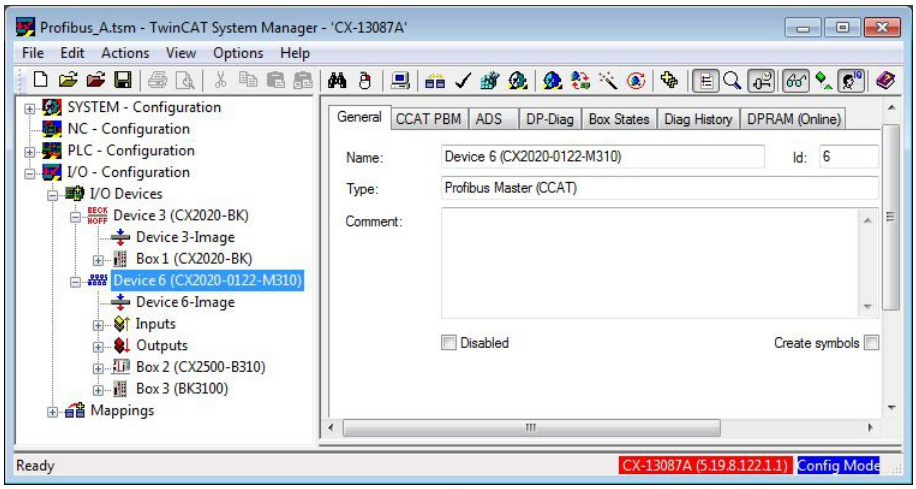


4. Select the devices you want to use and confirm the selection with **OK**.



5. Confirm the request whether to search for boxes with **Yes**.

⇨ All devices and slave boxes that are found are displayed in the tree view on the left, including Bus Terminals connected to the devices or slave boxes.



Repeat the steps if not all devices are displayed. If not all devices and slave boxes are found despite the repeat operation, check the cabling of the devices and slave boxes.
In the next step you can test the connection between Profibus master and Profibus slave.

5.2.9 Testing Profibus networking

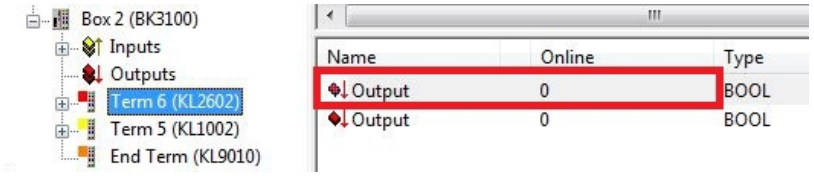
The Profibus networking between a Profibus master and a Profibus slave can be tested with TwinCAT. Select the Bus Coupler (BK3100) as Profibus slave for the test. During the test an LED is linked to the Bus Terminal, which is connected to the Profibus slave.

Prerequisites for this step:

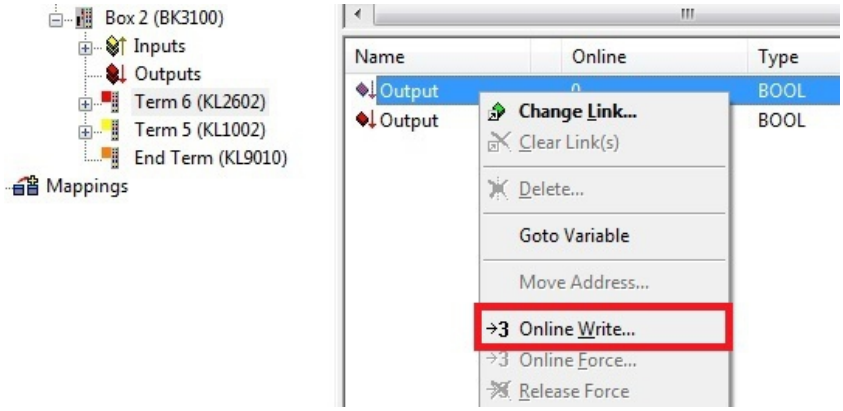
- TwinCAT must be in Free Run mode.

Test the link as follows:

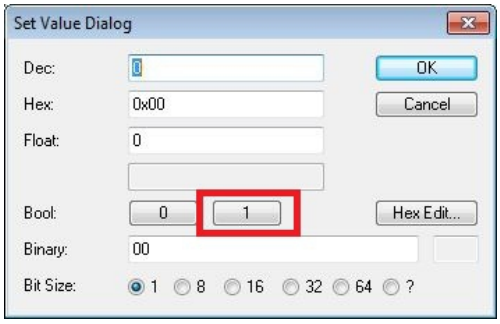
1. Find the Profibus slave in the tree view on the left.
2. Click on the Bus Terminal for which you want to activate the LED.
3. Right-click on the input or output.



4. In the context menu click on **Online Write.**



5. Set the **Bool** value to „1“.



⇨ In TwinCAT the value „1“ is shown under Online for the corresponding input or output. A signal was successfully transferred to the Profibus slave, if the LED also lights up for the Bus Terminal that is connected to the Bus Coupler. The Profibus master and Profibus slave are then successfully linked with each other.

5.2.10 'Turning' process data

The process data are transferred in Intel format as standard. If the data are required in Motorola format, they have to be 'turned' accordingly. This step illustrates how to 'turn' the data in TwinCAT.

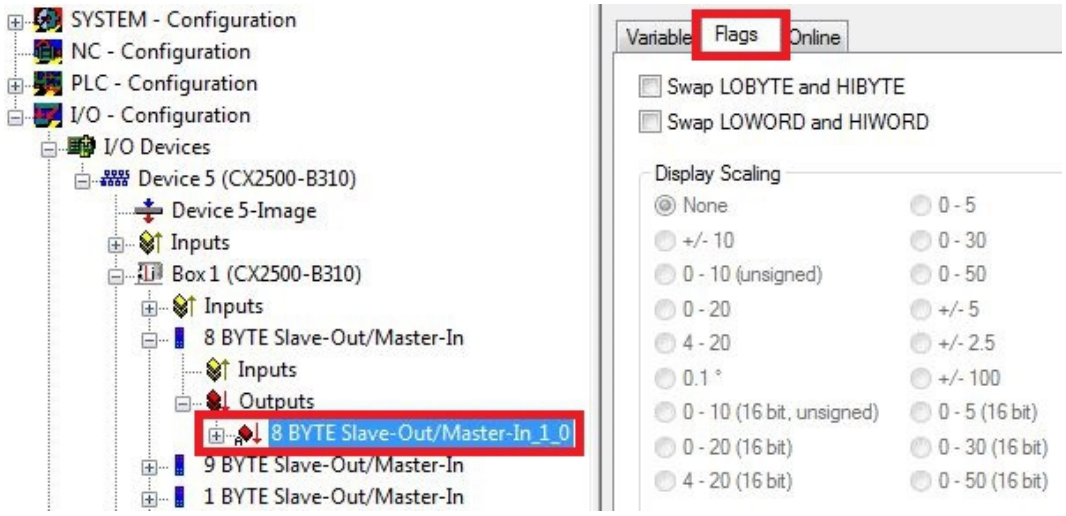
If the standard format is required, you can skip this step.

Prerequisites for this step:

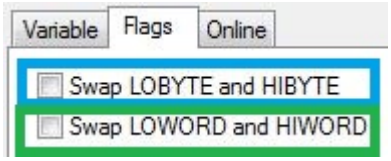
- A parameterized slave.
- A slave connected to a master. The master is added in TwinCAT and then scanned for the slave via the master.

'Turn' the process data as follows:

1. In the tree view, right-click on a variable containing data to be 'turned'.
2. Click on the **Flags** tab.



3. Click on the required option. For WORD variables, only LOBYTE and HIBYTE can be swapped. For DWORDs, the WORD can be swapped in addition.



⇨ In this way you can 'turn' process data. Use the following example to see how the data change for the individual options.
Example for DWORD.

Data of the slave	Data which the master receives			
Original data	No option selected	Swap Byte (blue)	Swap Word (green)	Swap both (blue and green)
0x01020304	0x01020304	0x02010403	0x03040102	0x04030201

The data can also be 'turned' in the PLC project, using the command ROR.
Example for ST: VarProfibus:=ROR(VarAnalog,8); (*Both variables of type WORD*)

5.3 Parameterization with TwinCAT 3

This section illustrates how Profibus devices can be parameterized with the aid of TwinCAT 3. A total of three devices are used for the example, including a Profibus master, to which two Profibus slaves are connected.

First, the process of finding and selecting a target system in TwinCAT is illustrated. Next, a Profibus slave is added and parameterized in TwinCAT, and the Profibus address of the slave is set. Then a PLC project is created and added in TwinCAT. Then, the variables from the PLC project are linked with the hardware, and the finished configuration is loaded on the Profibus slave.

In the last step, the Profibus master is added in TwinCAT, and the two Profibus slaves are located via the master. The process of testing the Profibus networking is then illustrated.

5.3.1 Searching for target systems

Before you can work with the devices, you must connect your local computer to the target device. Then you can search for devices with the help of the IP address or the host name.

The local PC and the target devices must be connected to the same network or directly to each other via an Ethernet cable. In TwinCAT a search can be performed for all devices in this way and project planning subsequently carried out.

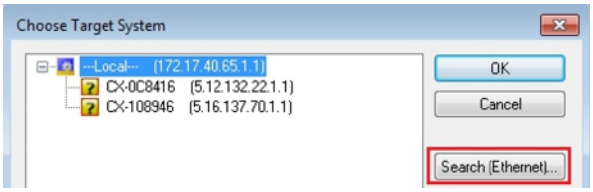
Prerequisites for this step:

- TwinCAT 3 must be in Config mode.
- IP address or host name of the device.

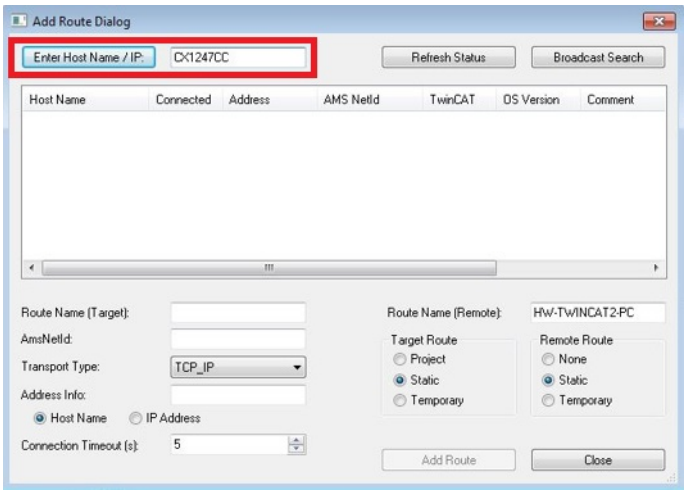
Search for the devices as follows:

1. In the menu at the top click on **File > New > Project** and create a new TwinCAT XAE project.
2. In the tree view on the left click on **SYSTEM**, and then **Choose Target**.

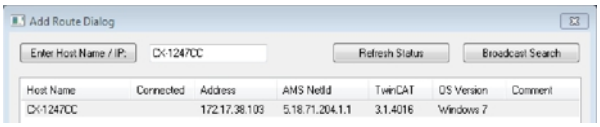
3. Click on **Search (Ethernet)**.



4. Type the host name or the IP address of the device into the **Enter Host Name / IP** box and press **[Enter]**.



5. Mark the device found and click on **Add Route**.

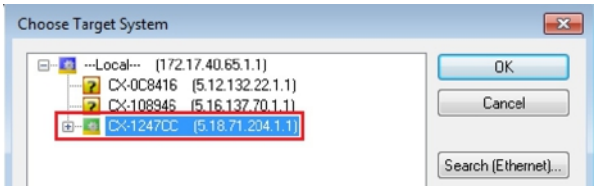


The Logon Information window appears.
Enter the user name and password for the CX in the **User Name** and **Password** fields and click **OK**.

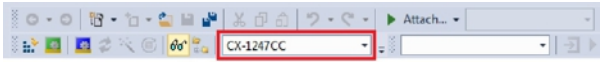
The following information is set as standard in CX devices:
User name: Administrator **Password:** 1

6. If you do not wish to search for any further devices, click on **Close** to close the Add Route Dialog.
The new device is displayed in the Choose Target System window.

7. Select the device you want to specify as target system and click **OK**.



⇨ You have successfully searched for a device in TwinCAT and inserted the device as the target system.
The new target system and the host name are displayed in the menu bar.



Using this procedure you can search for all available devices and also switch between the target systems at any time. Next, you can append the device to the tree view in TwinCAT.

5.3.2 Adding a Profibus slave

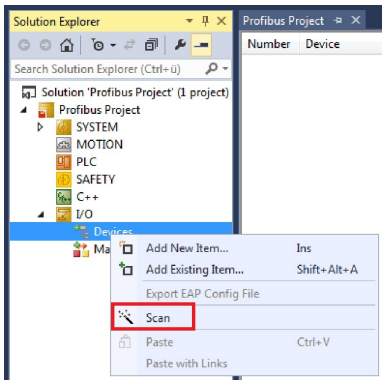
The example shows a CX2020 Profibus slave with CX2500-B310 fieldbus module, connected to the Profibus master. In order to ensure that the Profibus slave is configured and subsequently detected by the Profibus master with all inputs and outputs, the Profibus slave first must be added in TwinCAT.

Prerequisites for this step:

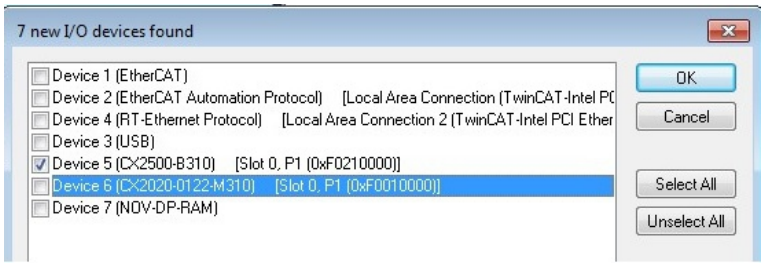
- A scanned and selected target device with Profibus slave. This example uses a CX2020 with CX2500-B310 fieldbus module.

Add the Profibus slave as follows:

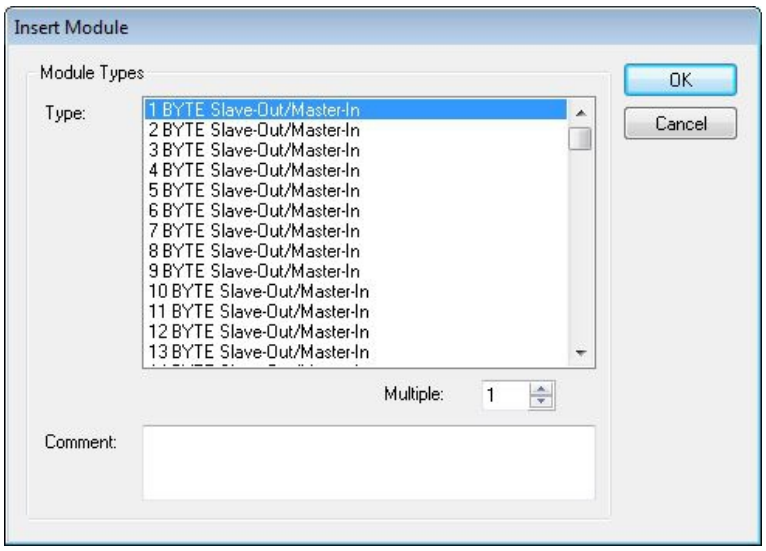
1. Start TwinCAT and open an empty project.
2. In the tree view on the left, right-click on **I/O Devices**.
3. In the context menu click on **Scan**.



4. Select the devices you want to use and confirm the selection with **OK**.



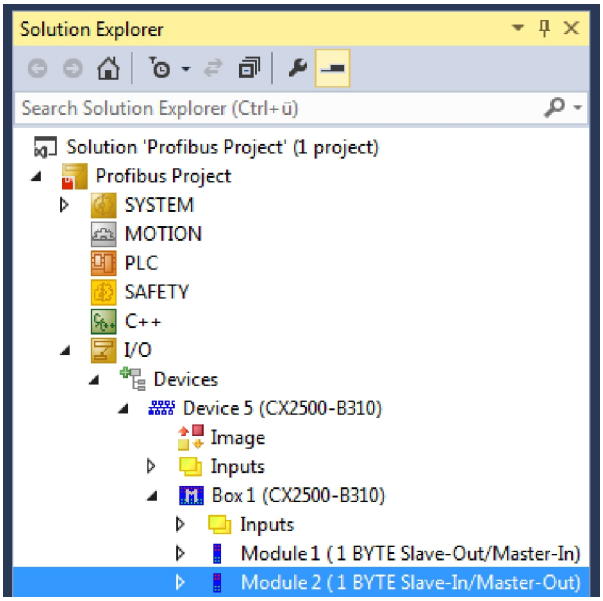
5. Confirm the request with **Yes**, in order to look for boxes.
   Box 1 (CX2500-B310) is integrated. The **Insert Module** window appears.
6. Add modules such as **1 BYTE Slave-Out/Master-In** and **1 BYTE Slave-In/Master-Out** for your process image.



7. Click on **Cancel** to close the **Insert Module** window.
8. Confirm the request whether to enable FreeRun with **Yes**.

⇨ The Profibus slave was successfully added in TwinCAT 3 and is displayed in the tree view with the inputs and outputs.

You can add further variables by right-clicking on the box and then clicking on **Add New Item** in the context menu.



In the next step you can extend the process image by creating additional virtual slaves. Or you can set the address, once the slave configuration is complete.

5.3.3 Creating a virtual slave

Additional virtual slaves can be created on the same hardware interface. This enables more data to be exchanged with a Profibus master, or a connection with a second Profibus master can be established.

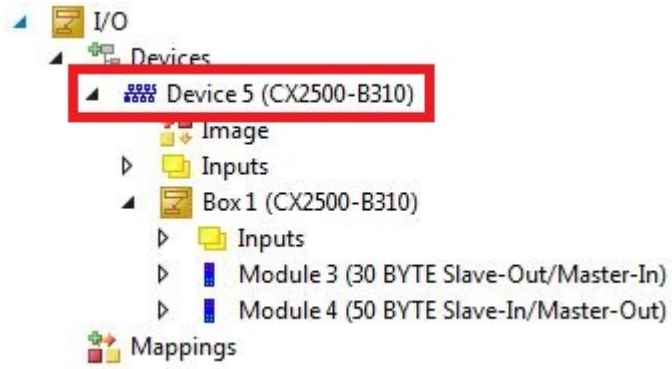
Each virtual slave is assigned a dedicated address via TwinCAT and is configured like an independent device for the Profibus master.

Prerequisites for this step:

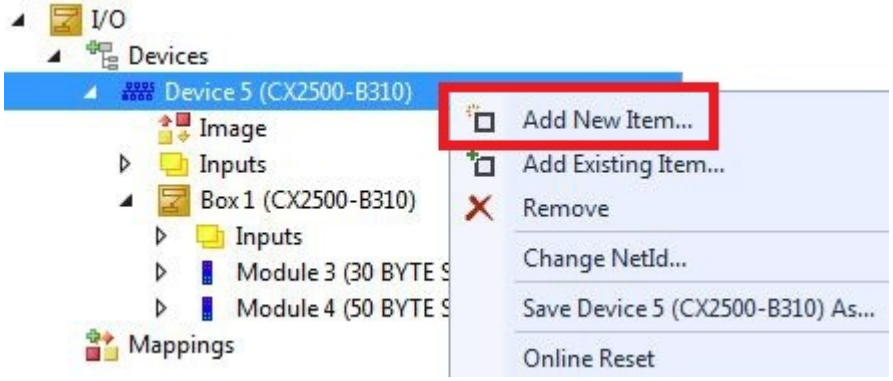
- A Profibus slave, created in TwinCAT.

Create a virtual slave as follows:

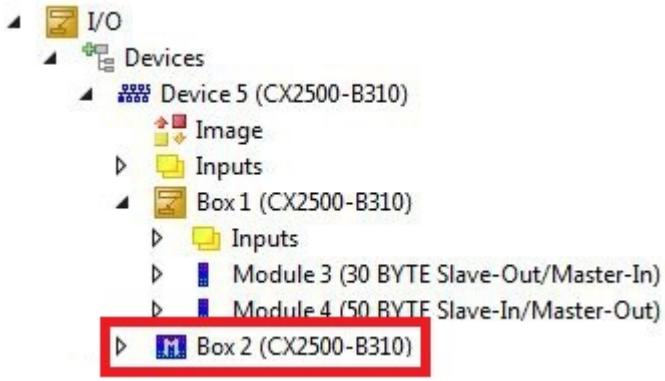
1. Right-click on a Profibus slave in the tree view on the left.



2. Click on **Add New Item** in the context menu.



⇨ A further box (virtual slave) is created.



Variables for the virtual slave can now be created. In the next step you can set the address for the slave.

5.3.4 Setting the address

Once the Profibus slave was successfully added in TwinCAT, the address of the Profibus slave can be set. Devices with a DIP switch have a preset address. The address on the DIP switch must match the address set in TwinCAT.

For devices without DIP switch the address is only set in TwinCAT.

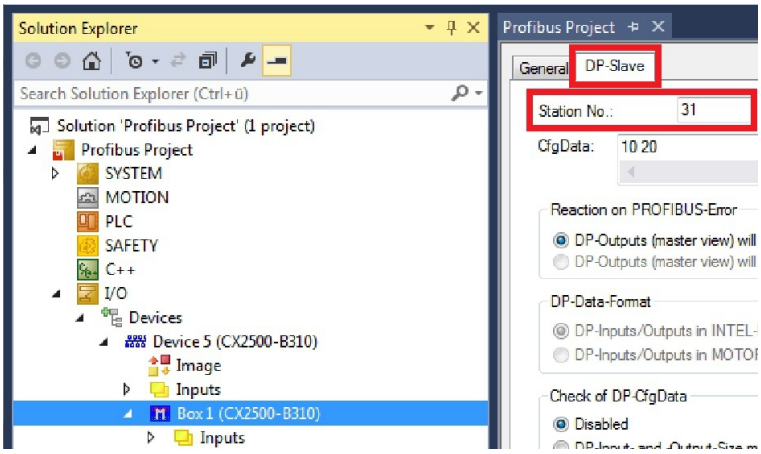
In this step the address is set in TwinCAT, so that the Profibus slave can be reached by the Profibus master via this address.

Prerequisites for this step:

- An added Profibus slave in TwinCAT.

Parameterize the Profibus slave as follows:

1. Click on a slave box.
2. Click on the **DP Slave** tab.
3. Enter a value for the Profibus address in the **Station No** field, e.g. „31“.



⇨ You have set the address successfully. The Profibus master can reach the Profibus slave with the set address.
Next, you can create a PLC project for the Profibus slave.

5.3.5 Creating a PLC project

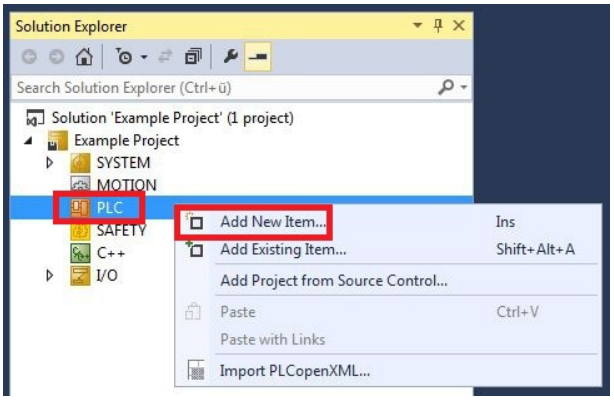
The next steps describe how to create a PLC project in TwinCAT and add it in the tree view.

Prerequisites for this step:

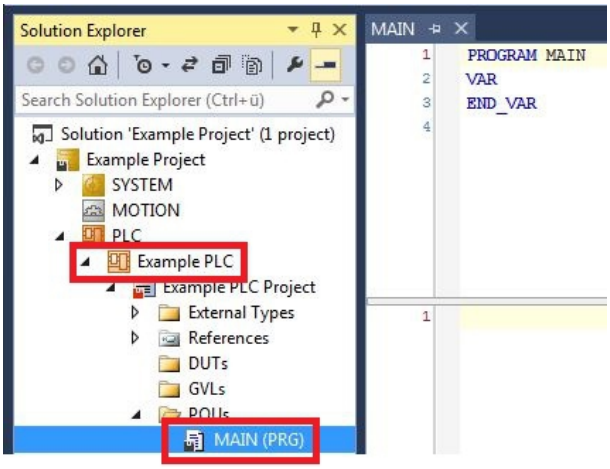
- A newly created TwinCAT XAE project.

Create a PLC project as follows:

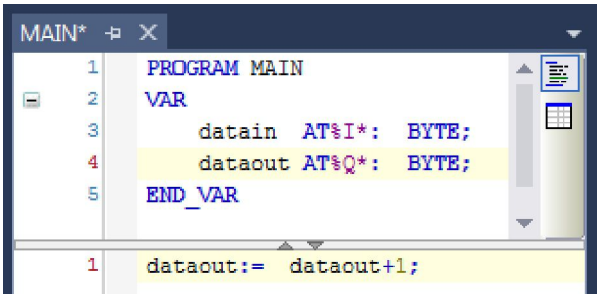
1. Right-click on **PLC** in the tree view.
2. In the context menu click on **Add New Item** and select the **Standard PLC Project**.



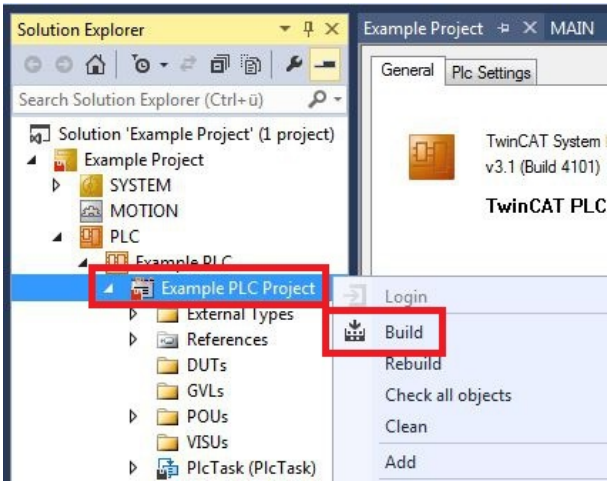
3. In the tree view click on the newly created PLC project, then double-click on **MAIN (PRG)** under **POUs**.



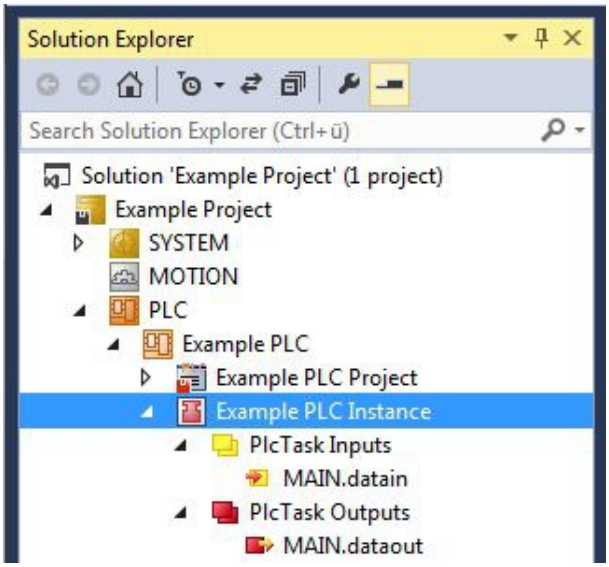
4. Write a small program, as shown in the diagram below.



5. In the tree view right-click on the PLC project, then click on **Build** in the context menu.



⇨ You have successfully created a PLC project and added the project in TwinCAT. A PLC instance with the variables for the inputs and outputs is created from the PLC project.



In the next step you can link the variables with the hardware.

5.3.6 Linking variables

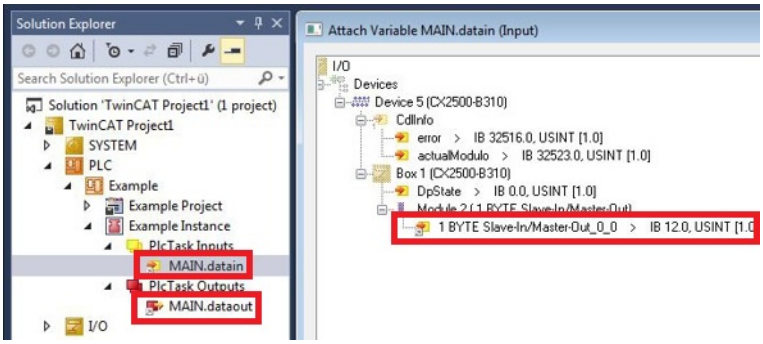
Once the PLC project was successfully added in the System Manager, you can link the newly created input and output variables from the PLC project with the inputs and outputs of your hardware.

Prerequisites for this step:

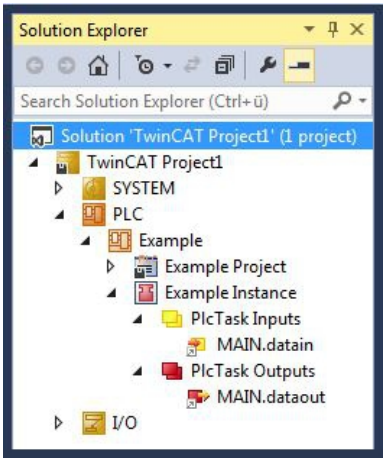
- A PLC program attached in TwinCAT.

Link the variables as follows:

1. Double-click on the input or output variables in the tree view under **PLC**.
   The **Attach Variable** window appears and shows which inputs or outputs can be linked with the variables from the PLC project.



2. Double-click on the inputs or outputs of the hardware in the **Attach Variable** window.
   Link the input variables with the inputs and the output variables with the outputs of the hardware.



   Variables that are already linked are indicated with a small arrow icon in TwinCAT.

3. In the toolbar click on **Activate Configuration**.

4. Confirm the request whether TwinCAT is to start in Free Run mode with **Yes**.

⇨ You have successfully linked variables with the hardware. Use Activate Configuration to save and activate the current configuration.

The configuration can now be loaded on the CX, in order to automatically start TwinCAT in Run mode, followed by the PLC project.

5.3.7 Load configuration to CX

Once variables are linked, the configuration can be saved and loaded on the CX. This has the advantage that the PLC project is loaded and started automatically when the CX is switched on. The start of the previously created PLC project can thus be automated.

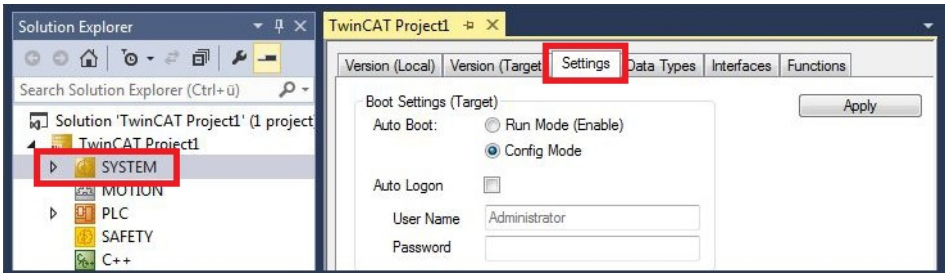
Prerequisites for this step:

- A completed PLC project, added in the System Manager.
- Variables from the PLC project, linked with the hardware in the System Manager.
- A CX selected as target system.

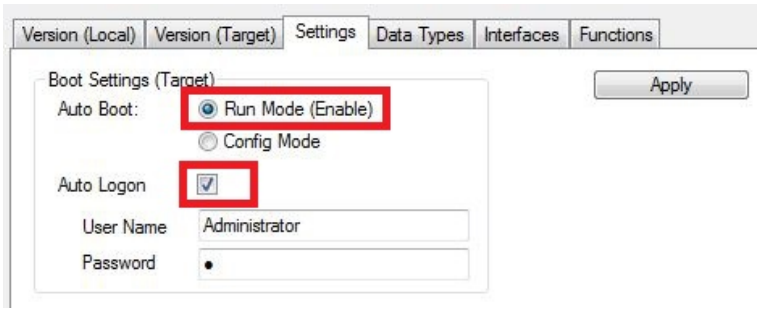
Load the configuration from the System Manager to the CX as follows:

1. In the tree view on the left click on **SYSTEM**.

2. Click on the **Settings** tab.



3. Under Boot Settings select the option **Run Mode (Enable)** and tick the **Auto Logon** checkbox.

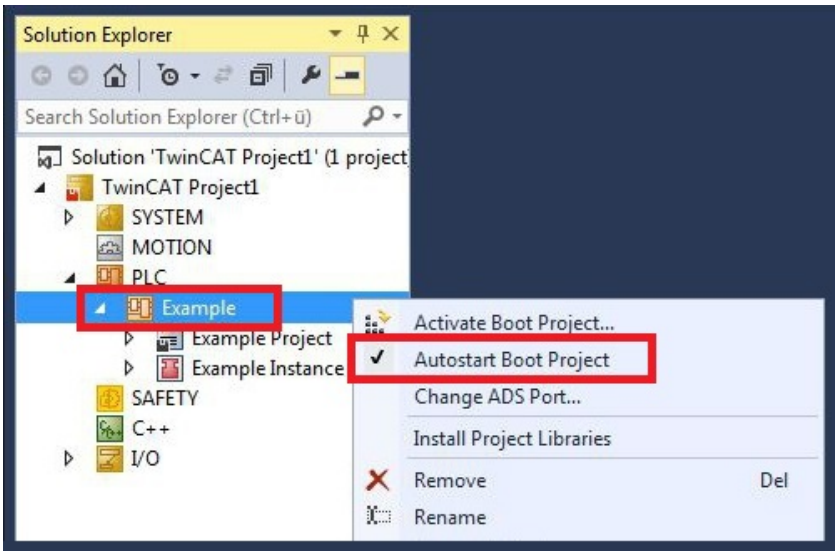


4. Enter the user name and password for the CX in the **User Name** and **Password** fields.

5. Click on **Apply**.

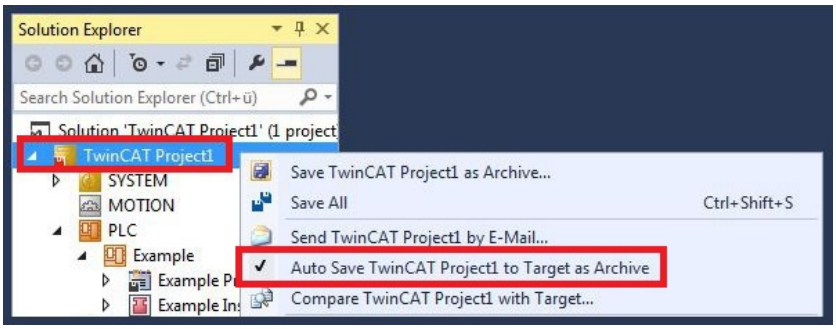
6. In the tree view on the left right-click on the PLC project under **PLC**.

7. In the context menu click on **Autostart Boot Project**.
   The setting is selected



8. Right-click on the project folder in the tree view.

9. In the context menu click on **Auto Save to Target as Archive**.
   The setting is selected.



⇨ You have successfully loaded the CX configuration. From now on, TwinCAT will start in Run mode and the PLC project will start automatically.

Next, the master can be added in a new project in the System Manager and can then be used to find slaves that have already been set up.

5.3.8 Adding a Profibus master

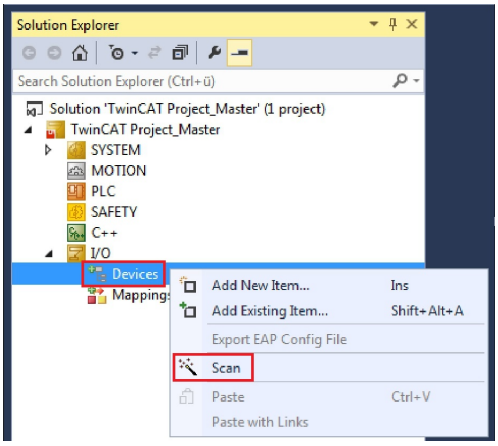
The Profibus master is added with the TwinCAT System Manager, like the other devices. The attached master can then be used to find all connected slaves. The following section illustrates how to add a Profibus master.

Prerequisites for this step:

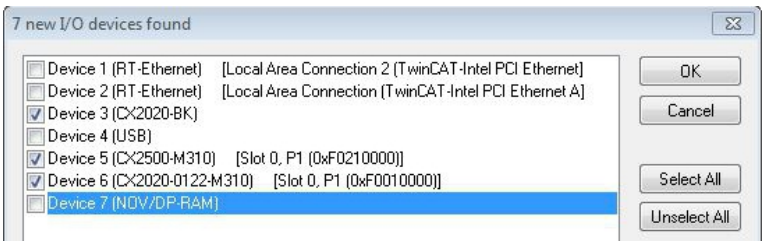
- TwinCAT must be in Config mode.
- A selected target system (in this example it is the Embedded PC CX2020-M310)

Add a Profibus master as follows:

1. Start TwinCAT.
2. In the tree view on the left, right-click on **Devices**.
3. In the context menu click on **Scan**.

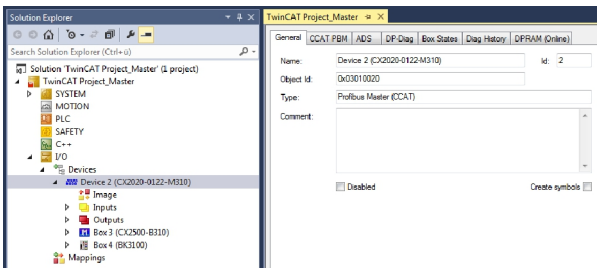


4. Select the devices you want to use and confirm the selection with **OK**.



5. Confirm the request whether to search for boxes with **Yes**.

⇨ All devices and slave boxes that are found are displayed in the tree view on the left, including Bus Terminals connected to the devices or slave boxes.



Repeat the steps if not all devices are displayed. If not all devices and slave boxes are found despite the repeat operation, check the cabling of the devices and slave boxes.
In the next step you can test the connection between Profibus master and Profibus slave.

5.3.9 Testing Profibus networking

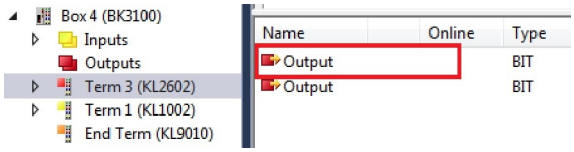
The Profibus networking between a Profibus master and a Profibus slave can be tested with TwinCAT. Select the Bus Coupler (BK3100) as Profibus slave for the test. During the test an LED is linked to the Bus Terminal, which is connected to the Profibus slave.

Prerequisites for this step:

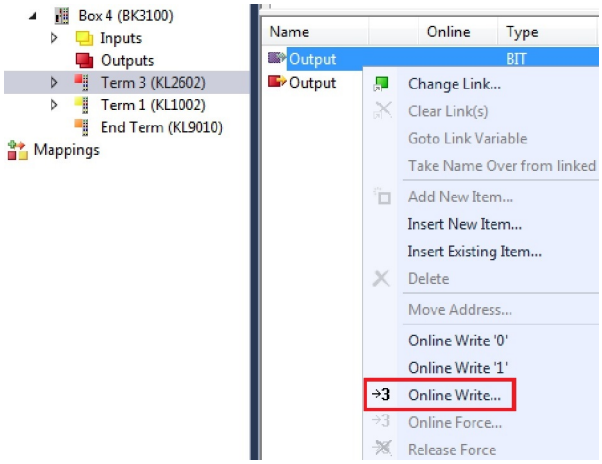
- TwinCAT must be in Free Run mode.

Test the link as follows:

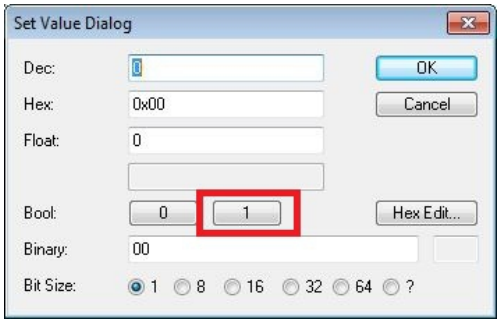
1. Find the Profibus slave in the tree view on the left.
2. Click on the Bus Terminal for which you want to activate the LED.
3. Right-click on the input or output.



4. In the context menu click on **Online Write.**



5. Set the **Bool** value to „1“.



⇨ In TwinCAT the value „1“ is shown under Online for the corresponding input or output. A signal was successfully transferred to the Profibus slave, if the LED also lights up for the Bus Terminal that is connected to the Bus Coupler. The Profibus master and Profibus slave are then successfully linked with each other.

5.3.10 'Turning' process data

The process data are transferred in Intel format as standard. If the data are required in Motorola format, they have to be 'turned' accordingly. This step illustrates how to 'turn' the data in TwinCAT.

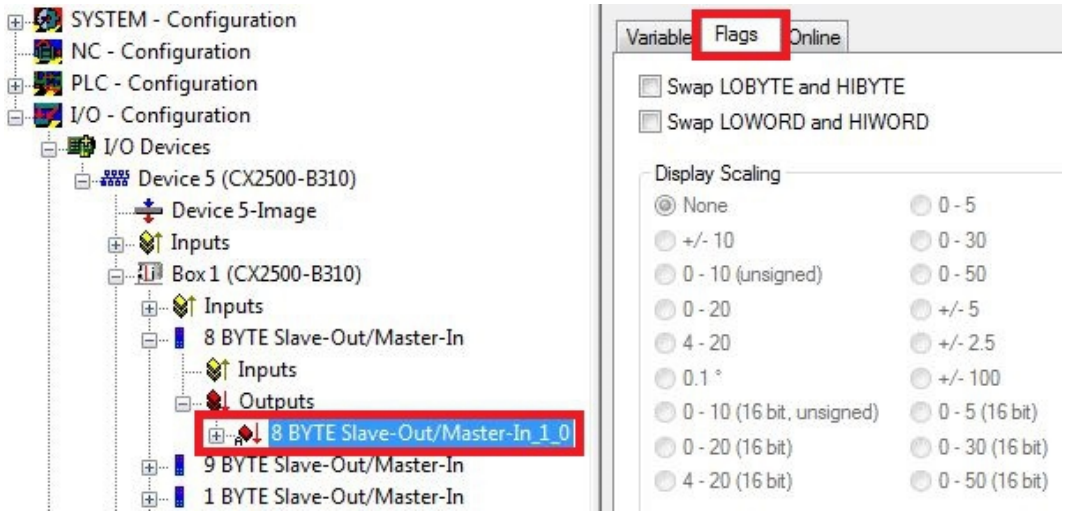
If the standard format is required, you can skip this step.

Prerequisites for this step:

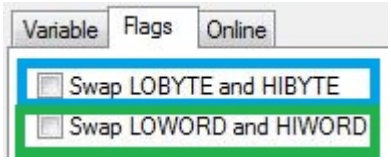
- A parameterized slave.
- A slave connected to a master. The master is added in TwinCAT and then scanned for the slave via the master.

'Turn' the process data as follows:

1. In the tree view, right-click on a variable containing data to be 'turned'.
2. Click on the **Flags** tab.



3. Click on the required option. For WORD variables, only LOBYTE and HIBYTE can be swapped. For DWORDs, the WORD can be swapped in addition.



⇨ In this way you can 'turn' process data. Use the following example to see how the data change for the individual options.
Example for DWORD.

Data of the slave	Data which the master receives			
Original data	No option selected	Swap Byte (blue)	Swap Word (green)	Swap both (blue and green)
0x01020304	0x01020304	0x02010403	0x03040102	0x04030201

The data can also be 'turned' in the PLC project, using the command ROR.
Example for ST: VarProfibus:=ROR(VarAnalog,8); (*Both variables of type WORD*)

6 Error handling and diagnostics

The Error reactions section describes the reactions that will be given to slaves that do not answer or that answer incorrectly, to a PLC stop or at start-up.

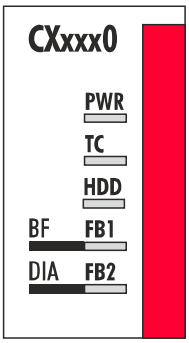
The Master section describes the different diagnostic variables and how the diagnostic data can be read. The Slave section describes the diagnostic information, which indicated the state of the Profibus slave. The section also shows how the diagnostic data can be read. In addition, the error codes of the Profibus slave are listed.

6.1 Diagnostic LEDs

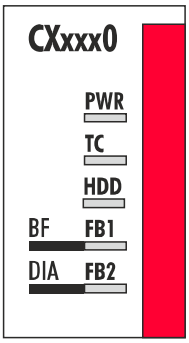
The diagnostic LEDs of a PROFIBUS master and PROFIBUS slaves are described here. The labelling of the diagnostic LEDs on a CX2500 fieldbus module and an Embedded PC with optional interface is identical.

The LED description therefore only distinguishes between PROFIBUS master and PROFIBUS slave.

M310 (Master)

Display	LED	Colours	Meaning
	BF Shows the PROFIBUS status	Green, on	PROFIBUS error-free
		Green, flashing	PROFIBUS, at least one slave in error
	DIA Indicates PROFIBUS error(s)	Red, lit	No PROFIBUS configured

B310 (Slave)

Display	LED	Colours	Meaning
	BF Shows the PROFIBUS status	Green, on	PROFIBUS in data exchange
		Green, flashing	PROFIBUS, waiting for Cfg data
	DIA Indicates PROFIBUS error(s)	Red, lit	No PROFIBUS configured

6.2 Error Reactions

Failure of a slave

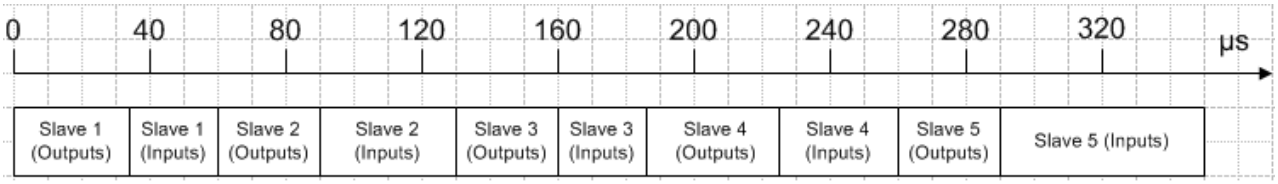
If a slave does not respond or the response is faulty, the master repeats the telegram several times, up to the **Max Retry limit** (see: Bus Parameters (DP) [▶ 22]). If a faulty telegram is received, the master repeats the telegram immediately. The **slot time**, which can be set on the same tab, indicates how long the master waits for a response from the slave.

For a Data_Exchange telegram with 12 Mbaud, a slot time of 1000 tbits and a Max Retry_Limit of 4 (defaults), sending of the next telegram is delayed by the value:

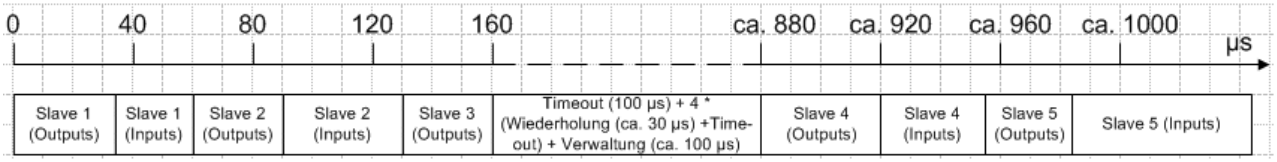
TDelay = (4 x ((15 + number of outputs) x 11 + 1000) - (15 + number of inputs) x 11)/12 µs

The DpState of the slave is set to a value other than 0. The effect on the DP connection can be set (see below: response in the master).

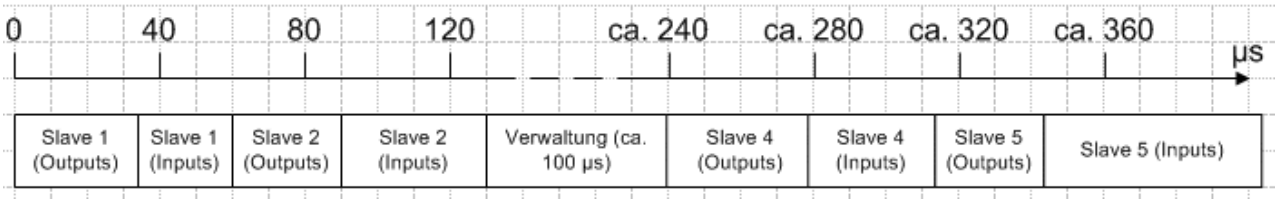
Normal DP cycle (12 Mbaud, 5 slaves, 20 bytes in, 20 bytes out per slave on average)



First occurrence of a faulty DP cycle (slave 3 does not answer)



Subsequent DP cycles (slave 3 no longer in the polling list)



It can happen that the slave answers incorrectly, e.g. because, as a result of a local event on the slave, the DP connection has been removed. In this case the telegram is not repeated, but the next telegram is sent. The DpState is set to a value other than 0. The slave is removed from the poll list and no longer addressed in the next DP cycle. The send time of the next telegram changes, until the DP connection has been re-established.

Reactions in the master

The response in the master can be set for each slave (see: Features [▶ 28] tab). The response in the master can be set with the option **NoAnswer Reaction**.

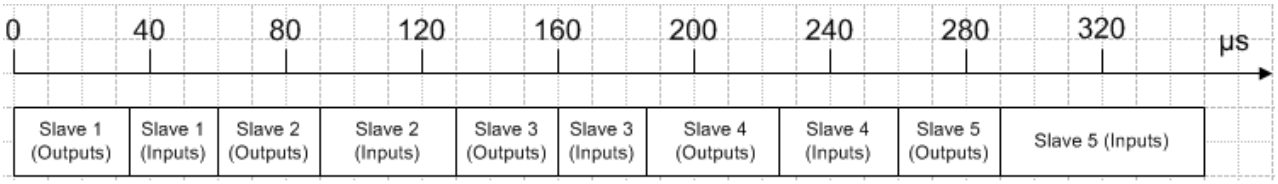
Here you can specify whether the DP connection to the slave should be interrupted immediately or not until the DP watchdog time has elapsed without a correct receipt telegram (Data_Exchange telegram).

If the DP connection is to be interrupted immediately (Leave Data Exch, Features tab of the slave), the slave is removed from the poll list discharged and no longer addressed in next DP cycle, until the DP connection has been re-established. At least seven telegrams are sent, in order to re-establish the DP connection. The process usually takes at least 10-20 ms.

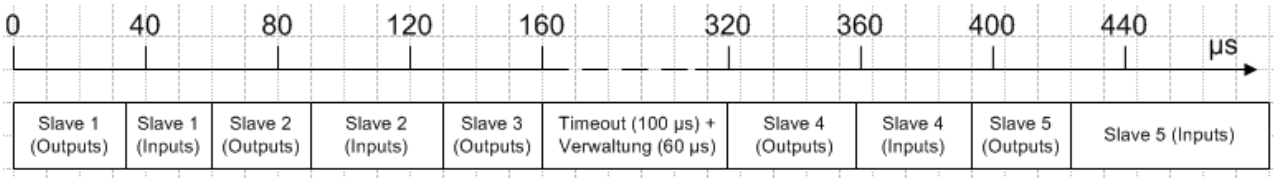
If the DP connection is not to be interrupted until the watchdog time has elapsed (Star in Data Exch for WD time, Features tab of the slave), the system waits for a correct response for the watchdog time set for the slave. The slave is addressed once more in the next poll cycle. If the slave again fails to respond, no further repeat telegram is sent.

It makes sense to wait for the watchdog time to elapse in situations where the PROFIBUS cycle should be as constant as possible in the event of a slave failure, and the failure of a slave can be tolerated for one or several cycles (e.g. in DP/MC mode (equidistant)). In this case the DP watchdog time for the slave should be set according to the tolerable outage time of the slave, and the Max Retry_Limit (DX) (Bus Parameter (DP) tab of master) should be set to 0.

Normal DP cycle (12 Mbaud, 5 slaves, an average of 20 bytes In, 20 bytes Out for each slave) in the "Stay in Data-Exch (for WD-Time)" mode



First faulty and subsequent DP cycles in the "Stay in Data-Exch (for WD-Time)" mode (slave 3 does not respond)



- **Changes of the slave's input data if the slave does not respond correctly**
  Here you can specify whether the slave input data are set to 0 (inputs will be set to 0, Features [▶ 28] tab of the slave) or the old value is retained (No changes), if the slave fails. In either case the DpState of the slave is set to value other than 0, so that the task can always recognize whether or not the data is valid. If a slave gives a faulty answer, the input data is always set to 0, independently of the setting of **Changes of the Input Data**.
- **Setting the slave's restart behaviour if the DP connection to the slave is removed**
  Here you can specify whether the DP connection to a slave, whose DP connection was interrupted, should automatically be re-established, or whether this should be done manually through an ADS WriteControl call.
  This setting is available if the slave is not connected to a CX master.
- **The reaction of the master if the DP connection to the slave is removed**
  This specifies whether removing the DP connection to a slave has no other effects (No Reaction, the default setting), or whether the master should enter the STOP state, thus removing the DP connections to all the slaves.
  This setting is available if the slave is not connected to a CX master.

Failure of the master

- **Monitoring in the PLC/IO task**
  In the presence of a persistent bus fault, the DP cycle can be extended up to 100 ms, even at 12 Mbaud. CycleCounter status variable, which can be linked in the PLC, is available for monitoring the DP master. This variable is incremented by master after each DP cycle. The variable can be monitored in the PLC monitored, in order to detect a failure of the master.
- **Monitoring in the slave**
  A watchdog time can be enabled in the slave, in order to monitor the failure of the master or the transfer to the PROFIBUS (see: Profibus [▶ 27] tab). The watchdog time must be set to at least twice the maximum of the estimated DP cycle and the cycle time (see: CCAT PBM [▶ 21] tab).

Failure of the PLC/IO task

A distinction is made between a PLC stop, reaching a break point and a task stop (the I/O task and NC task are only stopped when the entire system stops). In the case of a PLC stop, the output data is set to 0 by the PLC, whereas when a breakpoint is reached the data initially remains unchanged.

In the master the task is monitored with a task watchdog time (see: Startup/Fault settings [▶ 23] tab). If no new data transfer takes place within this task watchdog time, the master switches to Clear state, according to the setting Reaction on Task Stop (outputs are set to 0 or to the safe state Fail_Safe = 1 in the GSD file). With the “Operate” setting, the outputs retain the last value.

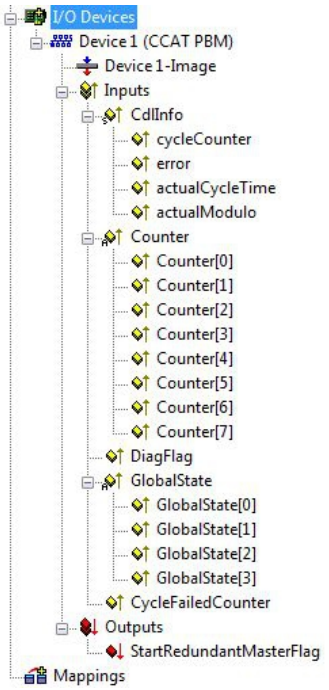
Start-up behaviour

The DP connections to all the slaves are established when the TwinCAT system starts up. Until the highest priority task that is involved has not been started, the master still does not send any Data_Exchange telegrams even after the DP connection has been established, and sends only diagnostic telegrams. As soon as the highest priority task has transferred data once, and the DP connection for the corresponding DP slave has been established, the master cyclically (with the highest priority assigned task) sends one Data_Exchange telegram to each of the corresponding slaves.

6.3 Master

Diagnostic variables

The master possesses a variety of diagnostic variables that describe the state of the master and of the Profibus. They can be linked in the PLC:



CdlInfo:

Variable	Description
cycleCounter	Is incremented at the end of each PROFIBUS cycle in order that this variable can indicate whether the last cycle was completed before the task was started.
error	Indicates the number of slaves, with which no data exchange could take place during the last cycle. The variable BoxState of the slaves only has to be checked, if this value is not 0.
actualCycleTime	Shows the current cycle time in 4/25 µs. This variable is only updated, if all slaves are in data exchange (i.e. CdlInfo.error is 0).
actualModulo	Shows the current modulo. This variable is only relevant, if the slaves are prioritized.

Counter: These variables are only used for the redundancy mode.

Variable	Description
Counter[0-1]	These variables are reserved for extensions.
Counter[2]	This variable counts received telegrams. The counter is incremented by 1 whenever an error-free PROFIBUS telegram is received.
Counter[3]	This variable counts the telegrams received by the primary master. The counter is incremented by 1, whenever the primary master receives an error-free telegram (The primary master has the same address as the redundancy master).
Counter[4]	This variable is a timeout counter. The counter is incremented by 1 whenever a timeout occurs on the PROFIBUS.
Counter[5-7]	These variables are reserved for extensions.

DiagFlag:

Variable	Description
DiagFlag	Indicates whether the master diagnostic information has changed. In this case the control program reads the information via ADS, and the variable "DiagFlag" is then reset. 0 = diagnostic data unchanged. 1 = diagnostic data have changed. Use ADS Read to read the data.

GlobalState:

Variable	Description
GlobalState[0]	Indicates the state of the master. 0 = RUN 1 = RESET 2 = OFFLINE 3 = STOP
GlobalState[1]	Indicates the number of detected bus errors.
GlobalState[2]	Reserved for extensions.
GlobalState[3]	Reserved for extensions.

CycleFailedCounter:

Variable	Description
CycleFailedCounter	This counter indicates how often the PROFIBUS cycle was not complete on startup of the TwinCAT task.

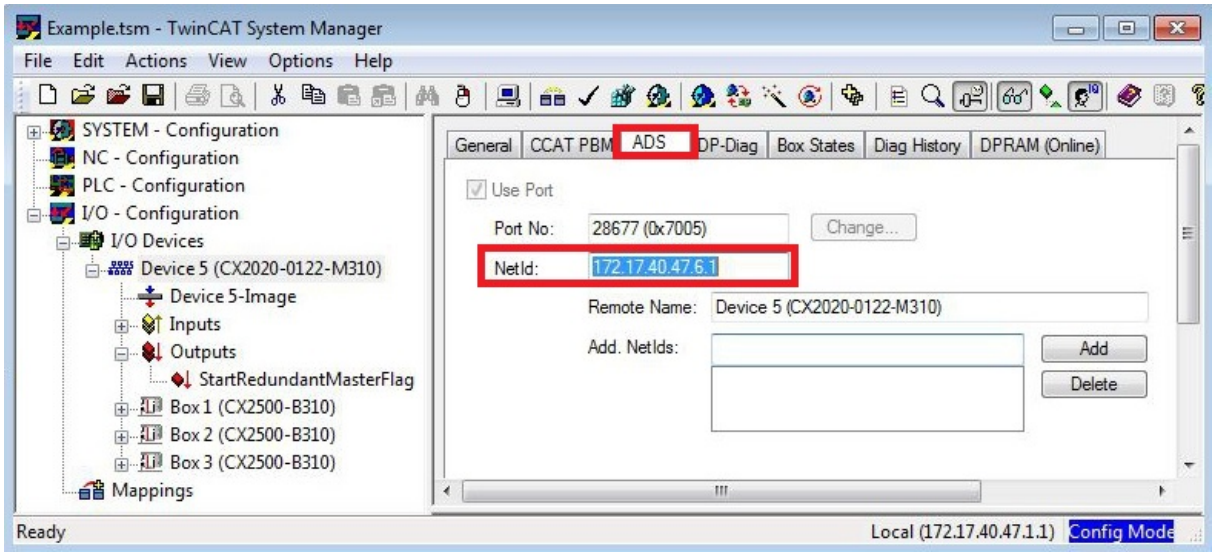
StartRedundantMasterFlag: Is for the redundancy mode.

Variable	Description
StartRedundantMasterFlag	If the value is set to 1 (True), the redundancy master on the PROFIBUS becomes active. If the value is 0 (False), the redundancy master is offline.

6.3.1 ADS-Interface

ADS NetId of the PROFIBUS interface

A NetId is required for the ADS communication. This can be found in the ADS tab of the PROFIBUS interface in the TwinCAT System Manager.



ADS-Interface

All acyclic data are transferred to or from the Embedded PCs with ADS Read, ADS Write or ADS Write Control. The Profibus interface of the Embedded PCs has a dedicated Net ID and supports the following ports:

Port	Description
200	Addresses the PROFIBUS interface itself, i.e. data stored locally on the PROFIBUS interface, for which usually no additional bus access is required.
0x1000 - 0x107E	Addresses a connected PROFIBUS device, whereby the address is calculated from port 0x1000 (or 0x1000 + slave address); always includes bus access.

ADS-Read

An overview of the IndexGroup/IndexOffset supported by the Embedded PCs for ADS Read is provided below.

Table 1: IndexGroup for local addressing of the Embedded PC (port 200)

IndexGroup (Lo-Word)	IndexGroup (Hi-Word)	IndexOffset	Description
0xF181	0x0000-0x007E	BYTE offset within the data	This will read the diagnostic data of a configured DP slave. The station address is calculated from the IndexGroup(Hi-Word). If the ADS Read is answered without error (error code = 0), the data will contain the diagnostic data of a configured DP slave described in section Slave-Diagnose [▶ 74].
0xF830	0x0000-0x007E	always 0	Can be used to determine which DP slaves are present on the PROFIBUS, regardless of whether they were configured or not; the station address is entered in the IndexGroup (high word). If the ADS-Read is answered without error (error code = 0), the corresponding DP slave has answered correctly. The data contain the Ident no. of the slave (BYTE offset 0-1) and the read CfgData (from BYTE offset 2) (see: Uploading the Configuration [▶ 71]).

Table 2: IndexGroup for addressing of a configured PROFIBUS device (port 0x1000-0x107E)

IndexGroup (Lo-Word)	IndexGroup (Hi-Word)	IndexOffset	Description
0x00-0xFF	0x00	0x00-0xFF	This will send a DPV1-Read to the appropriate, configured DPV1 slave via a Class -1 connection; the DPV1 slot number corresponds to the IndexGroup, the DPV1 index corresponds to the IndexOffset. If the ADS Read response is error-free (error code = 0), the data contain the read DPV1 data (see: DPV1 communication [▶ 70])

ADS-Write

An overview of the IndexGroup/IndexOffset supported by the Embedded PCs for ADS Write is provided below.

Table 3: IndexGroup for addressing of a configured PROFIBUS device (port 0x1000-0x107E)

IndexGroup (Lo-Word)	IndexGroup (Hi-Word)	IndexOffset	Description
0x00-0xFF	0x00	0x00-0xFF	This will send a DPV1-Write to the appropriate, configured DPV1 slave via a Class -1 connection; the DPV1 slot number corresponds to the IndexGroup, the DPV1 index corresponds to the IndexOffset (see: DPV1 communication [▶ 70]).

ADS-WriteControl

An overview of services supported by the Embedded PCs for ADS WriteControl is provided below.

Table 4: ADS-WriteControl for addressing a configured PROFIBUS device (port 0x1000-0x107E)

AdsState	DeviceState	State of the slave	Description
STOP (6)	0x00	RUN (5)	This will stop the slave, i.e. the process data connection to the relevant DP slave (Data_Exchange) is removed (with SetPrm,Unlock).
RUN (5)	0x00	STOP (6)	This will restart the slave after a stop, i.e. the process data connection to the relevant DP slave (Data_Exchange) is re-established (normal DP start-up).

6.3.2 DPV1 communication

On a C1 connection, the master supports the Read and Write services, and on the C2 connection it supports the Read, Write, Data_Transport, Initiate and Abort services.

C1 Connection (MSAC-C1)

The C1 connection is reserved for the master that cyclically exchanges data with the slave (C1 master). In order for a slave to be able to use the C1 connection, the slave must support DPV1 (this means that the line "DPV1_Slave = 1" and the key word "C1_Max_Data_Len" with an appropriate length must be in the GSD file). If it is also generally necessary to activate the C1 functionality by setting bit 7 in the **PrmData** byte 0 for the corresponding slave (this is done automatically for those Beckhoff devices that support DPV1).

MSAC-C1-Read is shown in ADS-Read, and MSAC-C1-Write is represented in ADS-Write:

MSAC-C1 Read

ADS-Read parameters	Meaning
Net-ID	Net-ID of the master (see the device's ADS tab)
Port	0x1000 + station address of the slave
IndexGroup	Slot number (DPV1 parameter)
IndexOffset	Index (DPV1 parameter)
Length	Length of the data that is to be read
Data	In response: data that has been read

MSAC-C1 Write

ADS-Write parameters	Meaning
Net-ID	Net-ID of the master (see the device's ADS tab)
Port	0x1000 + station address of the slave
IndexGroup	Slot number (DPV1 parameter)
IndexOffset	Index (DPV1 parameter)
Length	Length of the data that is to be written
Data	In request: data that is to be written

6.3.3 Uploading the Configuration

During operation the PROFIBUS can be scanned for new devices via ADS Read:

ADS-Read parameters	Meaning
Net-ID	Net-ID of the master (see the device's ADS tab)
Port	200
IndexGroup	0x00yyF830 (yy = station address, Beckhoff devices provide the same information as other devices)
IndexOffset	0
Length	1538
Data	Configuration data of the slave

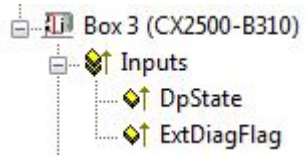
If the device is from another manufacturer, or if the IndexGroup indicates that Beckhoff devices are to behave in exactly the same way as devices from other manufacturers, then the following information is returned in the ADS read response:

Offset	Description
0 -1	1
2 - 7	DP diagnostic data bytes 0-5 (see Slave-Diagnose [▶ 74])
8 - 251	DP configuration data (**CfgData**)

6.4 Slave

Diagnostic variables

The slave features various diagnostic variables, which describe the state of the slave and the Profibus and can be linked in the PLC:



DpState:

Value	Description
0	No Error: Station is in data exchange.
1	Station deactivated: The slave was deactivated.
2	Station not exists: Slave does not respond. Check whether the slave is switched on, the cabling is OK and the station address is correct.
3	Master lock: The slave is in data exchange with another master.
4	Invalid slave response: Incorrect response from the slave. Occurs temporarily if the slave has terminated the data exchange due to a local event.
5	Parameter fault: Indicates a paramétrisation error.
6	Not supported: A DP function is not supported. Check whether the GSD file and the station address is correct.
7	Config fault: Indicates a configuration error. Check whether the added terminals or modules are OK.
8	Station not ready: The device is starting up. This is displayed temporarily during startup.
9	Static diagnosis: The slave indicates a static diagnosis and is currently unable to provide valid data. Check the operating state at the slave.
10	Diagnosis overflow: The slave indicates a diagnostics overflow. Read the diagnostic data with ADS Read and check the operating state of the slave.
11	Physical fault: Indicates a physical error in the response from a slave. Check the cabling.
12	Data-Exchange left: The data transfer was not completed.
13	Severe bus fault: Indicates a serious bus error. Please check the cabling.
14	Telegram fault: The slave responds with an invalid telegram.
15	Station has no resources: The slave has insufficient resources for the telegram. Check whether the GSD file is correct.
16	Service not activated: Occurs if the slave terminates the data exchange due to a local event. Check whether DP functionality is disabled for the slave.
17	Unexpected telegram: Unexpected telegram; can occur temporarily, if two PROFIBUS networks are linked, or check whether bus times in the second master are set the identically
18	Station ready: Occurs temporarily during startup, as long as the task has not yet started.
31	only for EtherCAT gateways: WC state of the cyclic EtherCAT frame is 1.
128	Slave waiting for data transfer: The slave was parameterized and configured, but has not yet received a data exchange telegram.
129	Slave waiting for configuration: The slave was parameterized, but has not yet received a Chk_Cfg telegram.
130	Slave waiting for parameter: The slave was not yet parameterized and is waiting for a Set_Prm (Lock) telegram.
131	Slave waiting for baud rate: The baud rate was not set.

Value	Description
132	Slave waiting for station no. from PLC: The Profibus address was not set.

ExtDiagFlag:

Value	Description
0	Diagnostic data have not changed.
1	Diagnostic data have changed. Use ADS Read to read the data.

6.4.1 Slave-Diagnose

DP-State

Each DP slave has a status variable that indicates the current state of that DP slave. This status is maintained in real time, so that it is always adapted to the current DP slave data, and can be linked to a PLC variable (-> **DpState** of the slave):

Diagnostic data

Any DP slave can acyclically report DP diagnostic data during data exchange operation. The slave here sets the Diag_Flag in the response to the cyclic Data_Exchange telegram, as a result of which the DP master automatically reads the DP diagnostic data from the slave. This does not affect the Data-Exchange cycle in the Beckhoff DP master, because the DP diagnostic telegram is sent at the end of the cyclic Data-Exchange cycle, and before the beginning of the next cycle. If the DP diagnostic data read from the slave has changed from its previous state, the DP master sets the ExtDiagFlag variable, which can be linked to a variable in the controller program.

The DP slave's current diagnostic data is displayed in the system manager on the slave's **Diag** tab. It can also be read by the controller program via ADS, which will cause the "ExtDiagFlag" flag to be reset once more:

ADS-Read parameters	Meaning
Net-ID	Net-ID of the master (see the device's ADS tab)
Port	200
IndexGroup	0x00yyF181 (yy = station address of the slave)
IndexOffset	Offset within the diagnosis data
Length	Length of the diagnostic data that is to be read
Data	Diagnostic data

The diagnostic data contains the slave statistics (32 bytes) and the DP diagnostic data sent by the slave (up to 244 bytes), and is constructed as follows:

Offset	Meaning
Slave statistics	
0	Receive Error Counter (WORD): The number of faulty telegrams occurring while communicating with this slave.
2	Repeat counter[8] (WORD): The repeat counters indicate how many repeats have had to be made, and how often. Repeat Counter[0] indicates how often it has been necessary to repeat a telegram for this slave once, Repeat Counter[1] shows how often a telegram for this slave has had to be repeated twice, and so on. The maximum number of retries is set with the parameter **Max Retry Limit** (see **Bus parameters** dialog). The value range is from 0 to 8, therefore there are 8 repeat counters (for 1 to 8 retries)
18	reserved for extensions
20	NoAnswer Counter (DWORD): The number of telegrams in communication with this slave that have not received an answer. The first time that a slave fails to answer, the telegram is repeated up to **MaxRetryLimit** times, but if it does not answer even then, further telegrams are not repeated.
24-27	Last-DPV1-Error[4] (BYTE): The most recent faulty DPV1 response is entered here (byte 0: DPV1 service (bit 7 is set, thus indicating an error), byte 1: Error_Decode, byte 2: Error_Code_1 (Error_Class/Error_Code), byte 3: Error_Code_2), see description of the DPV1 error codes [▶ 81]
27-31	reserved for extensions
from 32	**DP diagnostic data**

There follows a description of the DP diagnostic data

Offset	Meaning
0x00.0	StationNonExistent: slave did not reply to the last telegram
0x00.1	StationNotReady: slave still processing the Set_Prm / Chk_Cfg telegram
0x00.2	CfgFault: slave signaling a configuration error
0x00.3	ExtDiag: extended DiagData available and valid
0x00.4	NotSupported: slave does not support a feature requested via Set_Prm or Global_Control
0x00.5	InvalidSlaveResponse: slave response not DP-compatible
0x00.6	PrmFault: slave reports a paramétrisation error
0x00.7	MasterLock: slave currently exchanging data with another master
0x01.0	PrmReq: re-parameterize and reconfigure slave
0x01.1	StatDiag: slave signaling static diagnosis / DPV1 slave application not yet ready for data exchange
0x01.2	PROFIBUS DP slave
0x01.3	WdOn: DP watchdog on
0x01.4	FreezeMode: DP slave in freeze mode
0x01.5	SyncMode: DP slave in sync mode
0x01.6	reserved
0x01.7	Deactivated: DP slave has been deactivated
0x02.0	reserved
0x02.1	reserved
0x02.2	reserved
0x02.3	reserved
0x02.4	reserved
0x02.5	reserved
0x02.6	reserved
0x02.7	ExtDiagOverflow: too much extended data present
0x03	MasterAdd: station address of master with which slave is exchanging data
0x04,0x05	IdentNumber
ab 0x06	Extended DiagData

Extended DiagData

A distinction is made in the Extended DiagData between identification diagnosis, channel diagnosis and manufacturer-specific diagnosis. The first byte indicates the type of the diagnosis and the length of the associated data. Several diagnostic types can also follow one another in the Extended DiagData.

Header-Byte

Bit	Meaning
0-5	Length of the associated diagnostic data, including header byte
6-7	0 = manufacturer-specific diagnosis (DPV1 is not supported) or DPV1 diagnosis (DPV1 is supported (DPV1_Enable = 1) in associated GSD file)
	Module diagnosis
	Channel diagnosis
	Revision number

Manufacturer-specific diagnosis

The structure of the manufacturer-specific diagnosis may be found in the documentation for the DP slave.

6.4.2 Individual diagnostic data

The controller enables sending of diagnostic data from the PLC. You can write your own diagnostic message for the master and fill it individually with data (see Device-specific diagnostic data below).

DP Diagnostic Data (DiagData)

Transmission of the diagnostic telegram

A diagnostic telegram is only transferred to the controller when the diagnostic data have changed.

The DP diagnostic data consists of six bytes of DP standard diagnosis, along with up to 238 bytes of device-specific diagnostic data.

When the DP diagnostic data changes, the slave reports this fact to the master, and the master will automatically fetch the changed diagnostic data. This means that DP diagnostic data is not included in the DP process data in real-time, but is always sent to the controller a few cycles later.

In TwinCAT the DP diagnostic data are read by the DP master connection via ADS.

DP standard diagnostic data

Offset	Meaning
0x00.0	StationNonExistent: slave did not reply to the last telegram
0x00.1	StationNotReady: slave still processing the Set_Prm / Chk_Cfg telegram
0x00.2	CfgFault: slave signaling a configuration error
0x00.3	ExtDiag: extended DiagData available and valid
0x00.4	NotSupported: slave does not support a feature requested via Set_Prm or Global_Control
0x00.5	InvalidSlaveResponse: slave response not DP-compatible
0x00.6	PrmFault: slave reports a paramétrisation error
0x00.7	MasterLock: slave currently exchanging data with another master
0x01.0	PrmReq: re-parameterize and reconfigure slave
0x01.1	StatDiag: slave signaling static diagnosis / DPV1 slave application not yet ready for data exchange
0x01.2	PROFIBUS DP slave
0x01.3	WdOn: DP watchdog on
0x01.4	FreezeMode: DP slave in freeze mode
0x01.5	SyncMode: DP slave in sync mode
0x01.6	reserved
0x01.7	Deactivated: DP slave has been deactivated
0x02.0	reserved
0x02.1	reserved
0x02.2	reserved
0x02.3	reserved
0x02.4	reserved
0x02.5	reserved
0x02.6	reserved
0x02.7	ExtDiagOverflow: too much extended data present
0x03	MasterAdd: station address of master with which slave is exchanging data
0x04,0x05	IdentNumber
ab 0x06	Device-specific diagnostic data (extended DiagData)

Device-specific diagnostic data

Transmission of user-specific diagnostic data

Byte 0 of the data must contain a 0x08. Bytes 1 to 5 are overwritten by the CX. Byte 6 and higher can contain your own diagnostic data. Make sure that your own diagnostic data conforms to the PROFIBUS standard for user-specific diagnosis.

The ADSWRITE block is used for sending diagnostic data. The current DP diagnosis sent to the bus can be read via ADSREAD. Please note that an additional 6 bytes (PROFIBUS standard DP diagnosis) are required for reading, i.e. the number of bytes that are read exceeds the number of bytes that were written by 6. The ADS parameters for the read process are identical.

Input parameters	Description
NETID	local NetId of the Profibus device
PORT number	0x1000+slave address
IDXGRP	16#F481
IDXOFFS	0
LEN	max. 244
SRCADDR	Pointer to diagnostic data

6.4.3 DP-V1 communication

C1 Connection (MSAC-C1)

The C1 connection is reserved for the master that cyclically exchanges data with the slave (C1 master). In order for a slave to be able to use the C1 connection, the slave must support DPV1 (this means that the line "DPV1_Slave = 1" and the key word "C1_Max_Data_Len" with an appropriate length must be in the GSD file). If it is also generally necessary to activate the C1 functionality by setting bit 7 in the **PrmData** byte 0 (see the slave's **PROFIBUS** tab) for the corresponding slave (this is done automatically for those Beckhoff devices that support DPV1).

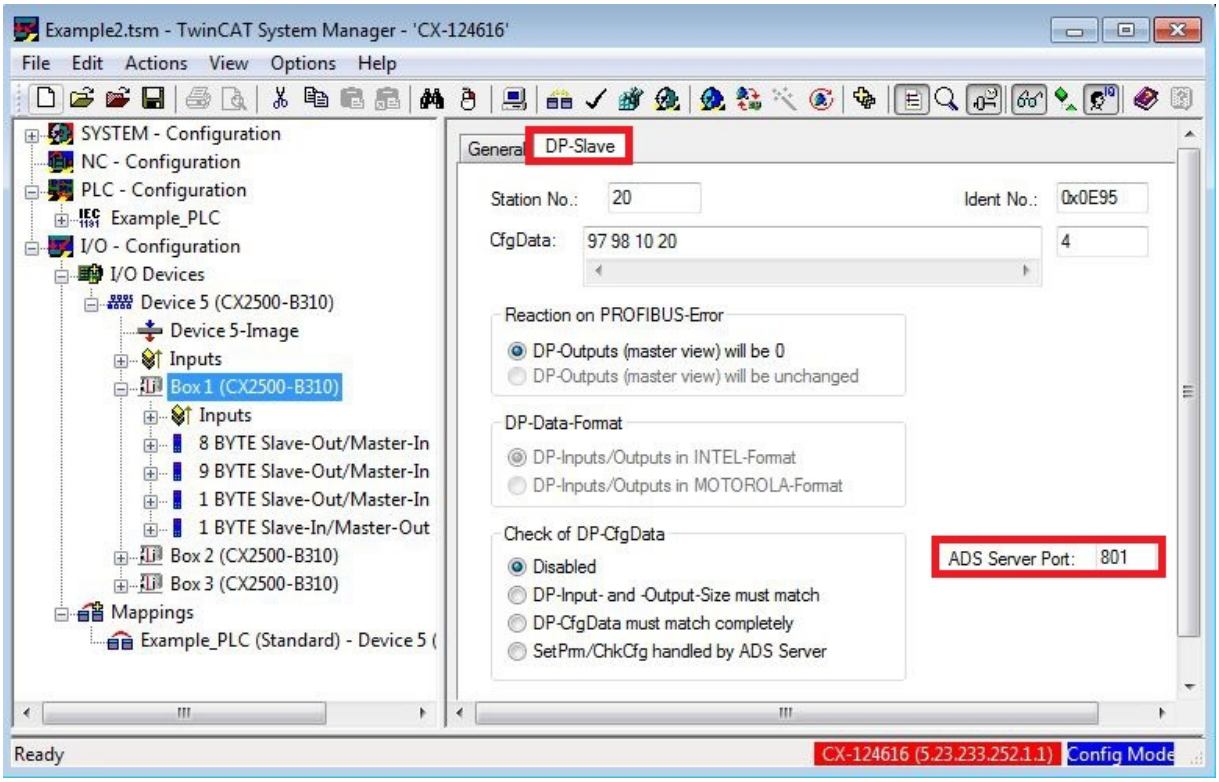
MSAC-C1-Read is mapped on ADS-Readind, MSAC-C1-Write on ADS-Writeinc:

PROFIBUS DPV1

The DP slave supports a DPV1-MSAC_C1 server connection that is established along with the cyclic connection. This can be used so that larger quantities of acyclic data can be transferred alongside the cyclic data. A DPV1 read telegram received by the master is reported to the PLC as an ADS read indication, while a DPV1 write telegram is reported to the PLC as an ADS write indication. The PLC program is then responsible for the read or write response. To do this, the ADS read response or ADS write response functions are to be called.

Settings in the System Manager

In order to transport the DP-V1 services into the PLC this function must be activated by specifying the port.



Port 0 deactivated, 802 PLC runtime 1, task 1

MSAC-C1 Read

A DPV1-MSAC_C1 read indication is represented in an ADS read indication as follows:

ADS read indication parameter	Meaning
Source-Net-ID (NETID)	Net-ID of the slave (see the device's **ADS** tab)
Source-Port (PORT)	0x1000+slave address
Invoke-ID (INVOKEID)	A unique number that must reappear in the response
IndexGroup (IDXGRP)	Slot number (DPV1 parameter)
IndexOffset (IDXOFFS)	Index (DPV1 parameter)
Length (LENGTH)	Length of the data that is to be read

An ADS read response is represented in a DPV1-MSAC_C1 read response as follows:

ADS read response parameter	Meaning
Destination-Net-ID (NETID)	Net-ID of the slave (see the device's **ADS** tab)
Destination-Port (PORT)	0x1000+slave address
Invoke-ID (INVOKEID)	A unique number, as under indication
Result (RESULT)	Result of the read: 0 = no error, otherwise: bits 0-15 = standard ADS error codes, bits 16-23 = Error_Code_1, bits 24-31 = Error_Code_2, See: DP-V1 error codes [▶ 81].
Length (LENGTH)	Length of the data that has been read
Data (DATAADDR)	read data

MSAC-C1 Write

A DPV1-MSAC_C1 write indication is represented in an ADS write indication as follows:

ADS write indication parameter	Meaning
Source-Net-ID (NETID)	Net-ID of the slave (see the device's **ADS** tab)
Source-Port (PORT)	0x1000+slave address
Invoke-ID (INVOKEID)	A unique number that must reappear in the response
IndexGroup (IDXGRP)	Slot number (DPV1 parameter)
IndexOffset (IDXOFFS)	Index (DPV1 parameter)
Length (LENGTH)	Length of the data that is to be written
Data (DATAADDR)	data that is to be written

An ADS read response is represented in a DPV1-MSAC_C1 read response as follows:

ADS read response parameter	Meaning
Destination-Net-ID (NETID)	Net-ID of the slave (see the device's **ADS** tab)
Destination-Port (PORT)	0x1000+slave address
Invoke-ID (INVOKEID)	A unique number, as under indication
Result (RESULT)	Result of the read: 0 = no error, otherwise: bits 0-15 = standard ADS error codes, bits 16-23 = Error_Code_1, bits 24-31 = Error_Code_2, See: DP-V1 error codes [▶ 81].
Length (LENGTH)	Length of the data that has been read

6.4.4 DP-V1 error codes

In the event of an incorrect DPV1 access, the slave replies with 4 bytes of data (any values that are not described here are not defined in the DPV1 standard, and are therefore to be found in the slave's manual).

byte 0	DPV1 service
0xD1	Data_Transport
0xD7	Initiate
0xDE	Read
0xDF	Write

byte 1	Error_Decode
0x80	DPV1
0xFE	FMS
0xFF	HART

byte 2		Error_Code_1
Error-Class (bits 4-7)	Error-Code (bits 0-3)	
0x0A	0x00	Application, Read Error
	0x01	Application, Write Error
	0x02	Application, Module Failure
	0x08	Application, Version Conflict
	0x09	Application, Feature Not Supported
0x0B	0x00	Access, Invalid Index
	0x01	Access, Write Length Error
	0x02	Access, Invalid Slot
	0x03	Access, Type Conflict
	0x04	Access, Invalid Area
	0x05	Access, State Conflict
	0x06	Access, Access Denied
	0x07	Access, Invalid Range
	0x08	Access, Invalid Parameter
	0x09	Access, Invalid Type
0x0C	0x00	Resource, Read Constrain Conflict
	0x01	Resource, Write Constrain Conflict
	0x02	Resource, Busy
	0x03	Resource, Unavailable

7 Appendix

7.1 Accessories

Cables and connectors for the connection of the Profibus components.

Connector

Item number	Description
ZB3100	9-pin D sub connector up to 12 Mbaud, with switchable termination resistor
ZB3101	9-pin D sub connector up to 12 Mbaud, with programming interface
ZB3102	9-pin D sub connector up to 12 Mbaud, with switchable termination resistor, inverse PROFIBUS plug connector
ZS1031-3000	9-pin D sub connector up to 12 Mbaud, with switchable termination resistor (other design)
ZS1031-3500	Fiber-optic connector for BK35xx (included with the Bus Coupler)

Cable

Item number	Description
ZB3200	PROFIBUS cable, 12 Mbaud, 1 x 2 x 0.64 mm² for fixed installation
ZB3300	PROFIBUS cable, 12 Mbaud, 2 x 0.25; 3 x 0.75 mm², 5-core, suitable for drag chains

Fiber-optic cable for Bus Coupler BK3500 (PROFIBUS)

Item number	Description
Z1100	1-core, 100 µm synthetic fiber, diameter 2.2 mm
Z1101	Synthetic fiber-optic cable, core ø 1000 µm, PU coating ø 5.5 mm, Kevlar strain relief
Z1102	as Z1101, on drum, length = 500 m
Z1121	Synthetic fiber optic cable, 1000 µm core, 2-core, PU coating, diameter 5.5 mm

7.2 Certifications

All products of the Embedded PC family are CE, UL and EAC certified. Since the product family is continuously developed further, we are unable to provide a full listing here. The current list of certified products can be found at www.beckhoff.com.

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.

7.3 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.

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

Phone:	+49(0)5246/963-0
Fax:	+49(0)5246/963-198
e-mail:	info@beckhoff.com

Beckhoff Support

Support offers you comprehensive technical assistance, helping you not only with the application of individual Beckhoff products, but also with other, wide-ranging services:

- support
- design, programming and commissioning of complex automation systems
- and extensive training program for Beckhoff system components

Hotline:	+49(0)5246/963-157
Fax:	+49(0)5246/963-9157
e-mail:	support@beckhoff.com

Beckhoff Service

The Beckhoff Service Center supports you in all matters of after-sales service:

- on-site service
- repair service
- spare parts service
- hotline service

Hotline:	+49(0)5246/963-460
Fax:	+49(0)5246/963-479
e-mail:	service@beckhoff.com

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