Contents "The CANopen Coupler"

1 The CANopen coupler

1.1 Brief overview

1.1.1 Fundamental properties and fields of application

CANopen is a standardized layer 7 protocol used for decentralized industrial automation systems based on the Controller Area Network (CAN) and the CAN Application Layer (CAL). CANopen is based on a communication profile containing the determination of basic communication mechanisms and their descriptions, such as the mechanisms used for exchanging process data in real-time or for sending alarm telegrams. This common communication profile is the basis for the various CANopen device profiles. The device profiles describe the specific functionality and/or the parameters of a device class. Such device profiles are available for the most important device classes used in industrial automation, such as digital and analog I/O modules, sensors, drives, control units, regulators, programmable controllers or encoders. Further device profiles are projected.

The central element of the CANopen standard is the device functionality description in an object directory (OD). The object directory is divided into one general area containing information about the device (e.g. device identification, manufacturer's name, etc.) as well as communication parameters, and the device-specific area describing the particular functionality of the device. These properties of a CANopen module are documented in the form of a standardized "electronic data sheet" (EDS file).

A CANopen network can consist of a maximum of 128 modules, one NMT master and up to 127 NMT slaves. In contrast to the typical master-slave systems (e.g. PROFIBUS systems), the meanings of the terms master and slave are different for CANopen. In operational mode, all modules are independently able to send messages via the bus. Moreover, the master is able to change the operating mode of the slaves. The CANopen master is normally implemented by a PLC or a PC. The bus addresses of the CANopen slaves can be set in the range between 1 and 127. The device address results in a number of identifiers occupied by this module.

CANopen supports transmission rates of 10 kbit/s, 20 kbit/s, 50 kbit/s, 125 kbit/s, 250 kbit/s, 500 kbit/s, 800 kbit/s and 1 Mbit/s. Each CANopen device has at least to support a transmission rate of 20 kbit/s. Other transmission rates are optional.

1.1.2 Communication mechanisms

CANopen distinguishes two basic mechanisms for data transmission: The fast exchange of short process data via Process Data Objects (PDOs) and the access to entries of the Object Directory using Service Data Objects (SDOs). Service Data Objects are primarily used to transmit parameters during device configuration. The transmission of Process Data Objects is normally performed event oriented, cyclic or on request as broadcast objects.

Service Data Objects

Service Data Objects (SDOs) are used to modify Object Directory entries as well as for status requests. Transmission of SDOs is performed as a confirmed data transfer with two CAN objects in the form of a peer-to-peer connection between two network nodes. The corresponding Object Directory entry is addressed by specifying the index and the sub-index of the entry. It is possible to transmit messages of unlimited length. If necessary, the data are segmented into several CAN messages.

Process Data Objects

For the transmission of process data, the Process Data Object (PDO) mechanism is available. A PDO is transmitted unconfirmed because, in the end, the CAN link layer ensures the error-free transmission. According to the CAN specification, a maximum of 8 data bytes can be transmitted within one PDO. In conjunction with a synchronization message, the transmission as well as the take over of PDOs can be synchronized over the entire network (synchronous PDOs). The assignment of application objects to a PDO can be set using a structural description (PDO mapping) that is stored in the object directory. Thus, an adaptation according to the requirements of the individual applications is possible. The transmission of process data can be performed by various methods.

Event

The PDO transmission is controlled by an internal event, e.g. by a changing level of a digital input or by an expiring device-internal timer.

Request

In this case, another bus subscriber is requesting the process data by sending a remote transmission request (RTR) message.

Synchronous

In case of synchronous transmission, synchronization telegrams are sent by a bus subscriber. These telegrams are received by a PDO producer which in turn transmits the process data.

1.1.3 Network Management

Within a CANopen network, only one NMT master exists (NMT = Network management). All other modules are NMT slaves. The NMT master completely controls all modules and is able to change their states. The following states are distinguished:

Initialization

After switching-on, a node is first in the initialization state. During this phase, the device application and the device communication are initialized. Furthermore, a so-called boot-up message is transmitted by the node to signalize its basic readiness for operation. After this phase is finished, the node automatically changes to the pre-operational state.

Pre-operational

In this state, communication with the node is possible via Service Data Objects (SDOs). The node is not able to perform PDO communication and does not send any emergency messages.

Prepared

In the prepared state, a node is completely disconnected from the network. Neither SDO communication nor PDO communication is possible. A state change of the node can only be initiated by a corresponding network command (e.g. Start Node).

1.1.4 Emergency messages

Emergency messages are used to signalize device errors. An emergency message contains a code that clearly identifies the error (specified in the communication profile DS-301 and in the individual device profiles DS-40x). The following table shows some of the available error codes. Emergency messages are automatically sent by all CANopen modules.

Emergency error code (hex)	Meaning / error cause
00xx	Error on reset or no error
10xx	General error
20xx	Current error
21xx	- Error on device input side
22xx	- Error inside the device
23xx	- Error on device output side
30xx	Voltage error
31xx	- Supply voltage error
32xx	- Error inside the device
33xx	- Error on device output side
40xx	Temperature error
41xx	- Ambient temperature
42xx	- Temperature inside the device
50xx	Hardware error in the device
60xx	Software error in the device
61xx	- Device-internal software
62xx	- Application software
63xx	- Data
70xx	Error in additional modules
80xx	Monitoring
81xx	Communication
90xx	External error
F0xx	Error of additional functions
FFxx	Device-specific errors

Table: Error codes in emergency messages

1.1.5 Node guarding and heartbeat

Testing the functionality of a CAN node is particularly required if the node does not continuously send messages (cyclic PDOs). Two mechanisms can alternatively be used to monitor the CANopen nodes. When the node guarding protocol is used, the NMT master sends messages to the available CANopen slaves which have to respond to these messages within a certain time period. Therefore, the NMT master is able to detect if a node fails. Furthermore, the heartbeat protocol can be used with CANopen. In this case, each node automatically sends a periodic message. This message can be monitored by each other subscriber in the network.

1.1.6 Object directory

The object directory describes the entire functionality of a CANopen device. It is organized as a table. The object directory does not only contain the standardized data types and objects of the CANopen communication profile and the device profiles. If necessary, it also contains manufacturer-specific objects and data types. The entries are addressed by means of a 16 bit index (table row, 65536 entries max.) and an 8 bit sub-index (table column, 256 entries max.). Thus, objects belonging together can be easily grouped. The following table shows the structure of this CANopen object directory:

Index		Object
dec	hex	
0	0000	not used
1...31	0001...001F	Static data types
32...63	0020...003F	Complex data types
64...95	0040...005F	Manufacturer-specific data types
96...127	0060...007F	Profile-specific static data types
128...159	0080...009F	Profile-specific complex data types
160...4095	00A0...0FFF	Reserved
4096...8191	1000...1FFF	Communication profile (DS-301)
8192...24575	2000...5FFF	Manufacturer-specific parameters
24576...40959	6000...9FFF	Parameters of the standardized device profiles
40960...65535	A000...FFFF	Reserved

Table: Structure of the object directory

Several data types are defined for the objects themselves. If required, other structures (e.g. ARRAY, STRUCT) can be created from these standard types.

1.1.7 Identifiers

CANopen always uses identifiers with a length of 11 bits (standard frames). The number of available and allowed identifiers given by this is divided into several ranges by the pre-defined connection set. This structure is designed in a way that a maximum of 128 modules (1 NMT master and up to 127 slaves) can exist in a CANopen network. The list of identifiers is composed of some fix identifiers (e.g. network management identifier 0) and various functional groups where each existing node, that supports the corresponding function, is assigned to one unique identifier (e.g. Receive PDO 1 of node 3 = 512 + node number = 515). Using the pre-defined connection set therefore avoids double assignment of identifiers.

Identifier	Function	Calculation
0	Network management (NMT)	-
1...127	not used	-
128	Synchronization (SYNC)	-
129...255	Emergency message	128 + node ID
256	Timestamp message	-
257...384	not used	-
385...511	Transmit PDO 1	384 + node ID
512	not used	-
513...639	Receive PDO 1	512 + node ID
640	not used	-
641...767	Transmit PDO 2	640 + node ID
768	not used	-
769...895	Receive PDO 2	768 + node ID
896	not used	-
897...1023	Transmit PDO 3	896 + node ID
1024	not used	-
1025...1151	Receive PDO 3	1024 + node ID
1152	not used	-
1153...1279	Transmit PDO 4	1152 + node ID
1280	not used	-
1281...1407	Receive PDO 4	1280 + node ID
1408	not used	-
1409...1535	Transmit SDO	1408 + node ID
1536	not used	-
1537...1663	Receive SDO	1536 + node ID
1664...1792	not used	-
1793...1919	NMT error (node guarding, heartbeat, boot-up)	1792 + node ID
1920...2014	not used	-
2015...2031	NMT, LMT, DBT	-

Table: Assignment of identifiers

1.1.8 PDO mapping

As already explained, all 8 data bytes of a CAN message are available for the transmission of process data. As there is no additional protocol information, the data format has to be agreed between the sending (producer) and the receiving party (consumer). This is done by the so-called PDO mapping.

If a fixed mapping is used, the process data are arranged in a pre-defined order within the PDO message. This arrangement is predetermined by the device manufacturer and cannot be changed. If variable mapping is used, the process data can be arranged as desired within the PDO message. For this purpose, the address, consisting of index and sub-index, as well as the size (number of bytes) of an object directory entry are entered into the mapping object.

1.1.9 EDS files

The characteristic properties of a CANopen module are documented in the form of an electronic data sheet (EDS file, electronic data sheet). The file completely and clearly describes the characteristics (objects) of a module type in a standardized and manufacturer independent format. Programs for configuring a CANopen network use the module type descriptions available in the EDS files. This strongly simplifies the configuration of a CANopen system. Usually the EDS files are provided by the device manufacturer.

1.1.10 Features

CANopen:

- Operating mode CANopen-Master
- Process image with a maximum of 57344 I/O points
- Supports min. boot-up, emergency messages and life guarding
- Supported PDO modes: Event-controlled, synchronous, cyclic and remote PDO transmission
- Integrated device profiles: CiA DS-401, CiA DS-402 and CiA DS-406

CAN (additional functionality, not necessary for pure CANopen operation):

- Support of 11 bit identifiers according to CAN 2.0 A and 29 bit identifiers according to CAN 2.0 B
- Transmission and reception of any CAN telegrams via function blocks in the user program

Transmission technique:

- ISO 11898, potential separated
- Transfer rates of 20 kbit/s, 125 kbit/s, 250 kbit/s, 500 kbit/s and 1 Mbit/s
- Bus length up to 1000 m at 20 kbit/s and up to 40 m at 1 Mbit/s
- One bus can have up to 128 subscribers (master + 127 slaves)
- 5-pin COMBICON socket for bus connection

Communication:

- Message-oriented bus access, CSMA/CA
- Predefined master-slave connections
- 8 bytes of non-fragmented user data, for fragmentation any size is possible
- Synchronization of inputs and/or outputs via synchronous PDOs

Protection functions:

- Message transfer with Hamming distance HD = 6
- CAN fault recognition mechanisms via 15 bit CRC, frame check, acknowledge, bit monitoring and bit stuffing
- Incorrect parameter settings are avoided because bus subscribers with faulty parameters are not included in the user data operation.
- Adjustable behavior on subscriber failure. System continues normal operation and the error is indicated at the master or the entire system is stopped.
- Response monitoring of the subscribers (node guarding)

Diagnosis:

- Status indication with 5 LEDs
  - PWR (green): Supply voltage
  - RDY (yellow): Coupler ready for operation
  - RUN (green): Configuration and communication status
  - STA (yellow): Data transfer
  - ERR (red): CANopen error
- Extensive online diagnosis functions in SYCON.net
- Detailed diagnosis in the user program using function blocks

1.2 Technical data

1.2.1 Technical data of the coupler

Coupler CM578-CN	
Field bus	CANopen
Transmission rate	10 kbit/s up to 1 Mbit/s
Protocol	CANopen master
Attachment plug for field bus	Pluggable 5-pin COMBICON connector
Processor	EC1, 160 pins
Clock frequency	48 MHz
Possible CPUs	PM571-xxx, PM581-xxx, PM591-xxx
Possible terminal bases	All
Ambient temperature	0 °C...55 °C
Coupler interface	Dual-port memory, 8 kbytes
Current consumption over the coupler bus	typ. 290 mA
Internal RAM memory (EC1)	256 kbytes
External RAM memory	-
External Flash memory	512 kbytes (firmware)
Status display	PWR, RDY, RUN, STA, ERR
Weight	approx. 150 g

Table: Technical data of the CANopen coupler

1.2.2 Technical data of the interface

Interface socket	5-pin COMBICON
Transmission standard	ISO 11898, potential-free
Transmission protocol	CANopen (CAN), 1 Mbaud max.
Transfer rate (baud rate)	20 kbit/s, 125 kbit/s, 250 kbit/s, 500 kbit/s and 1 Mbit/s
Status indication	5 LEDs (see following figure)
Number of subscribers	127 slaves max.

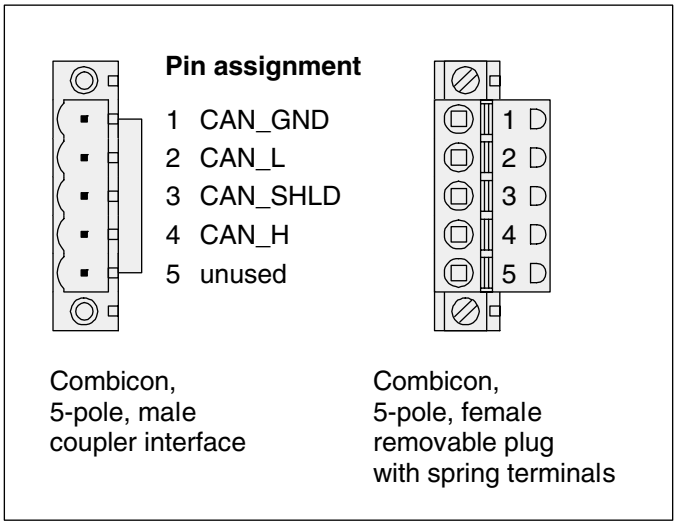


Figure: Pin assignment of the interface

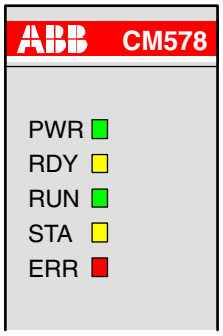


Figure: Position of the LEDs

LED	Color	Status	Meaning
PWR	green	ON	Voltage on
		OFF	Voltage off
RDY	yellow	ON	Coupler ready
		flashes cyclic	Bootstrap loader active
		flashes non-cyclic	Hardware or system error
		OFF	Defective hardware
RUN	green	ON	Communication in progress
		flashes cyclic	Ready for communication
		flashes non-cyclic	Parameterization error
		OFF	No communication or voltage off
STA	yellow	ON	CANopen master: Sends data
		OFF	CANopen master: No data
ERR	red	ON	CANopen error
		OFF	No error

Table: Meaning of the LED states

1.3 Connection and data transfer media

1.3.1 Attachment plug for the bus cable

Assignment:

5-pin COMBICON connector

Pin No.	Signal	Meaning
1	CAN_GND	CAN reference potential
2	CAN_L	Bus line, receive/transmit line, LOW
3	CAN_SHLD	Shield of the bus line
4	CAN_H	Bus line, receive/transmit line, HIGH
5	-	-

Table: Pin assignment of the attachment plug for the bus cable

Supplier:

e.g. COMBICON

Phoenix Contact GmbH & Co.
Flachsmarktstraße 8 - 28
D-32825 Blomberg
Germany
Phone: (+49) (0)52 35 / 3-00
Fax: (+49) (0)52 35 / 3-4 12 00
Internet: http://www.phoenixcontact.com

1.3.2 Bus terminating resistors

The ends of the data lines have to be terminated with a 120 Ω bus termination resistor. The bus termination resistor is usually installed directly at the bus connector.

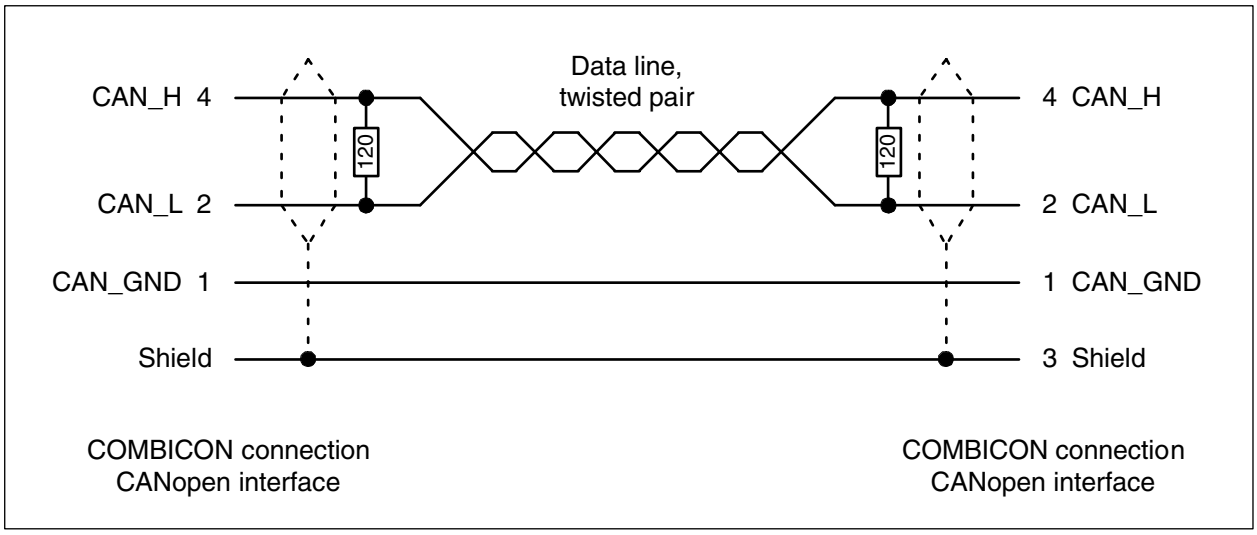


Figure: CANopen interface, bus terminating resistors connected to the line ends

1.3.3 Bus cables

For CANopen, only bus cables with characteristics as recommended in ISO 11898 have to be used. The requirements to the bus cables depend on the length of the bus segment. Regarding this, the following recommendations are given by ISO 11898:

Length of segment [m]	**Bus cable**			**Max. baud rate [kbit/s]**
	Conductor cross section [mm²]	Line resistance [Ω/km]	Wave impedance [Ω]	
0...40	0.25...0.34 / AWG23, AWG22	70	120	1000 at 40 m
40...300	0.34...0.60 / AWG22, AWG20	< 60	120	< 500 at 100 m
300...600	0.50...0.60 / AWG20	< 40	120	< 100 at 500 m
600...1000	0.75...0.80 / AWG18	< 26	120	< 50 at 1000 m

Table: Recommendations for bus cables

Supplier:

e.g. UNITRONIC® BUS CAN

U.I. LAPP GmbH
Schulze-Delitzsch-Straße 25
D-70565 Stuttgart
Germany
Phone: (+49) (0)711 7838 01
Fax: (+49) (0)711 7838 264
Internet: http://www.lappkabel.de

1.4 Possibilities for networking

The CANopen coupler is connected to the bus using the 5-pin COMBICON socket. For EMC suppression and protection against dangerous contact voltages, the shield of the bus line has to be connected to protective earth outside the housing. The line ends of the bus cable have to be terminated using bus terminating resistors.

Within a CANopen network, the controller with the CANopen coupler is the NMT master. No other NMT master is allowed in this network. The NMT master completely controls all modules and their operational states. Up to 127 NMT slaves can be connected to an NMT master.

The CANopen master is able to:

- Change operational states of the slaves
- Parameterize the slaves (e.g. communication connections, time supervision, bus traffic)
- Configure slaves (e.g. type, number and channel operating mode)
- Read input data of the slaves
- Write output data of the slaves
- Read diagnostic data of the slaves
- Monitor the availability of the slaves
- Transmit control commands to synchronize the inputs or outputs of the slaves
- Read and write slave objects even during running operation

The CANopen coupler is as well able to:

- Transmit and receive CAN telegrams according to CAN 2.0 A (11 bit identifier) and CAN 2.0 B (29 bit identifier). (This additional functionality is not required for pure CANopen operation.)

1.5 CANopen implementation

1.5.1 Configuration

The integration of the coupler into the PLC configuration of AC500 is an assumption for the correct function of the CANopen coupler CM578. The configuration of the coupler and the connected CANopen subscribers is done using the tool SYCON.net which is part of the Control Builder programming software.

In the following configuration example, the coupler CM578 is configured as CANopen master device. An I/O device is used as CANopen slave.

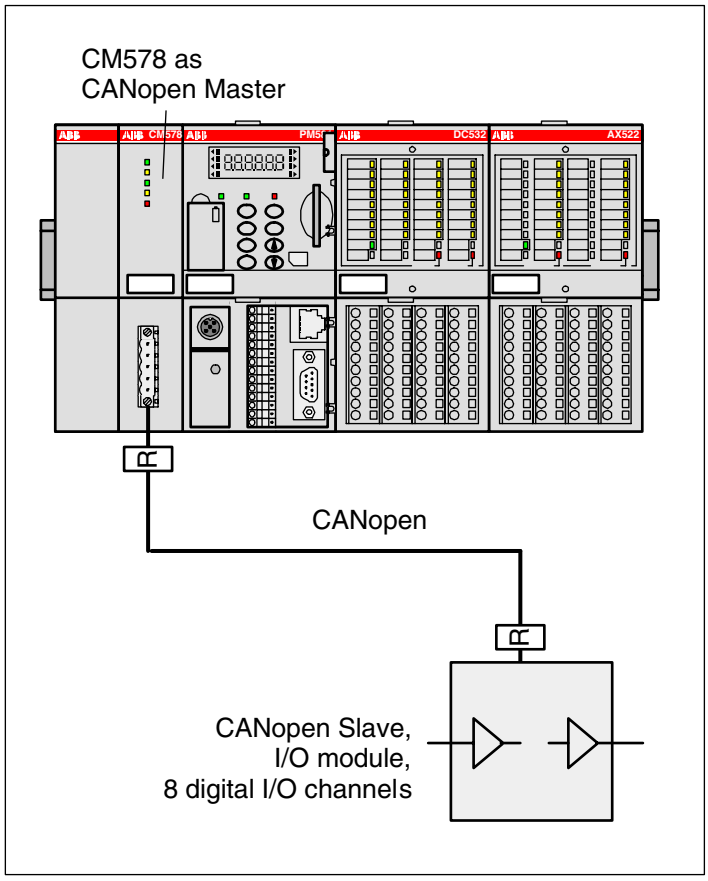
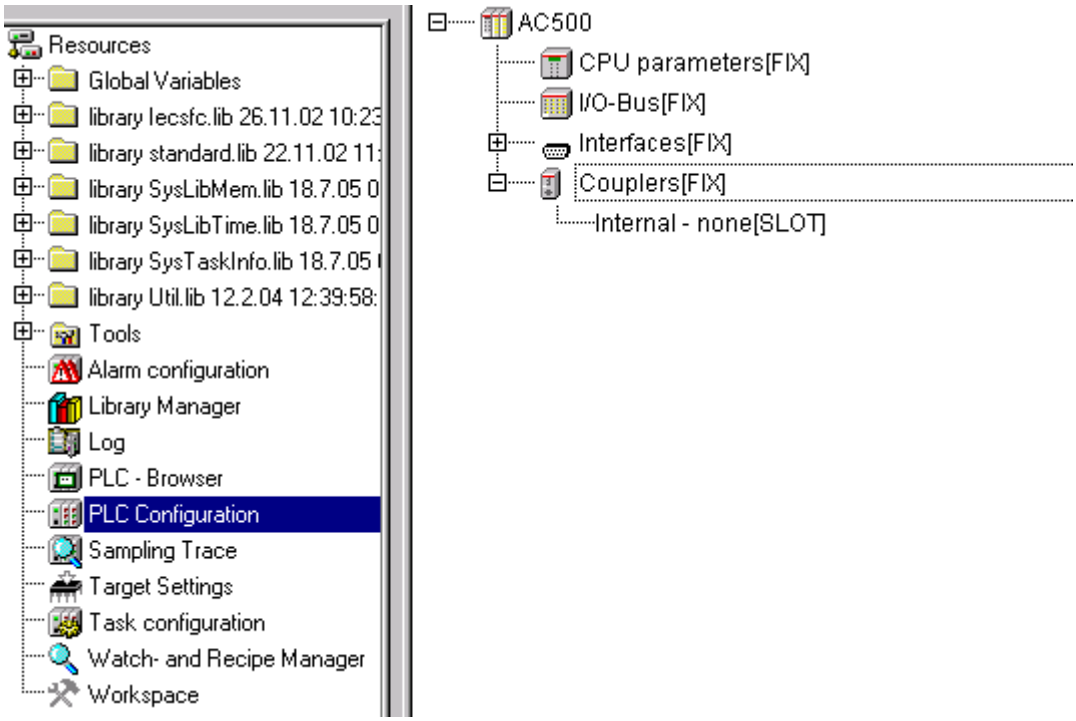


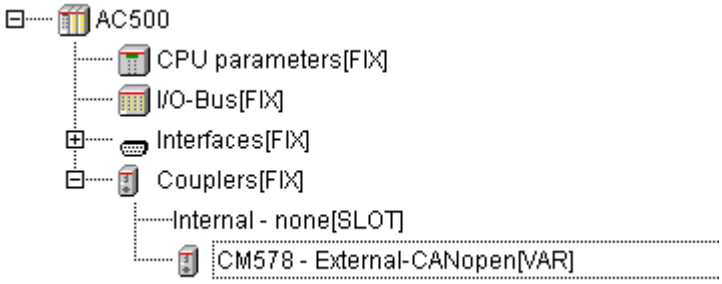
Figure: Example configuration consisting of an AC500 with a CANopen coupler and a CANopen slave

PLC configuration

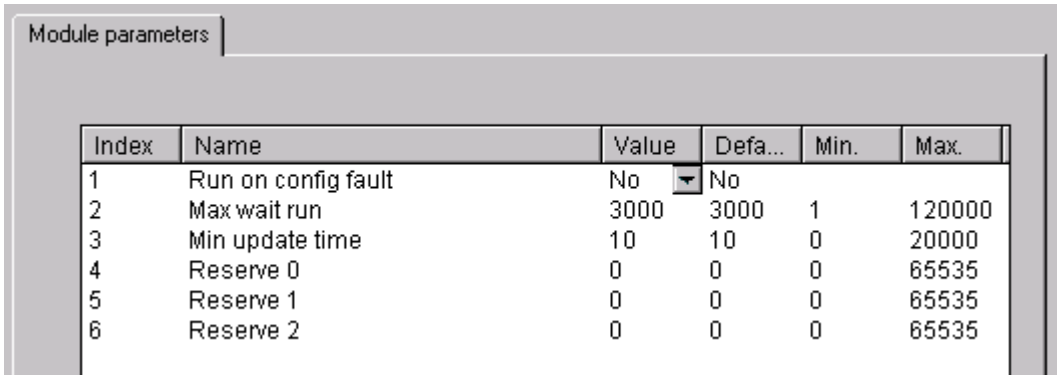
The configuration of the CANopen coupler starts with the integration of the coupler into the PLC configuration.



To insert the coupler into the configuration, select "Couplers", press the right mouse button and then select "Append Subelement" -> "CM578".



Do not change the default values for the coupler parameters.

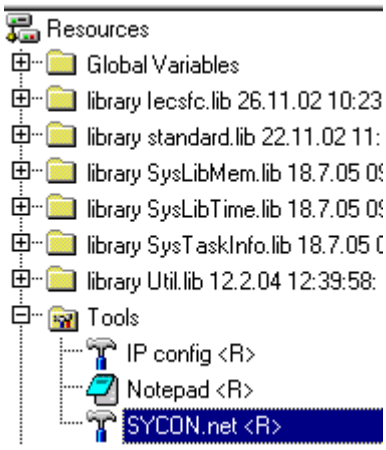


Index	Name	Value	Defa...	Min.	Max.
1	Run on config fault	No	No		
2	Max wait run	3000	3000	1	120000
3	Min update time	10	10	0	20000
4	Reserve 0	0	0	0	65535
5	Reserve 1	0	0	0	65535
6	Reserve 2	0	0	0	65535

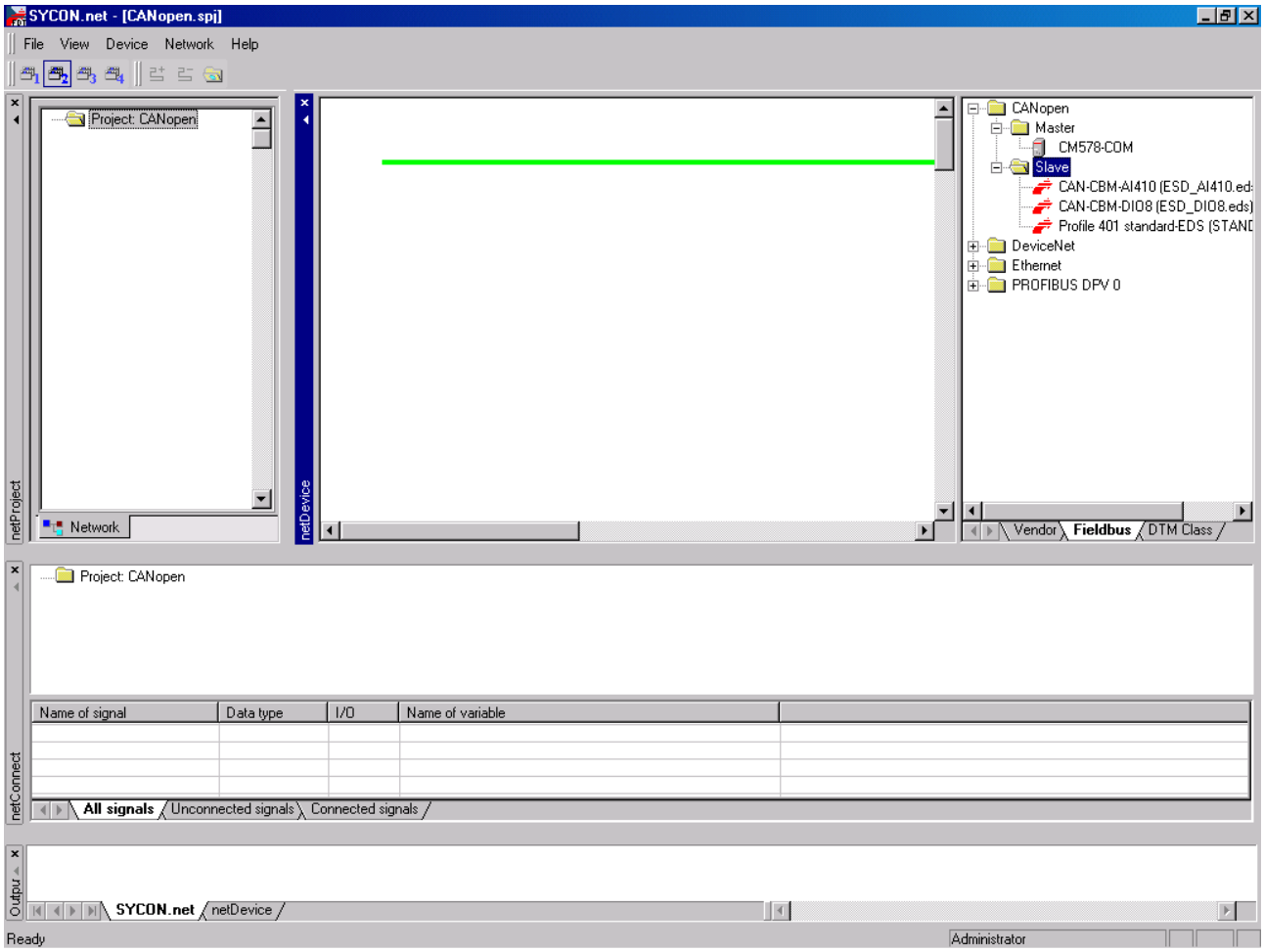
The coupler is now integrated in the PLC configuration.

Configuration using SYCON.net

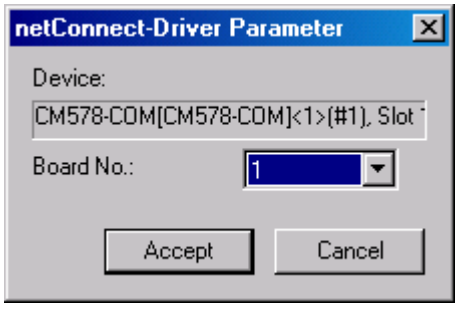
When configuring the CANopen coupler, the configuration data are a definite element of a project. They are specified using the tool SYCON.net (Resources tab -> Tools -> SYCON.net) which is part of the Control Builder. The configuration data are transferred to the coupler with the SYCON.net tool.



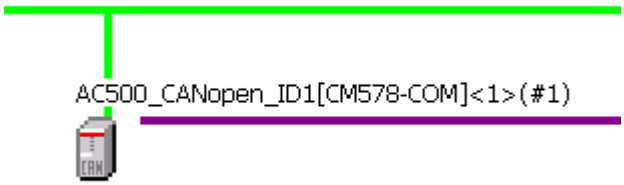
The following view appears when starting the configuration tool SYCON.net.



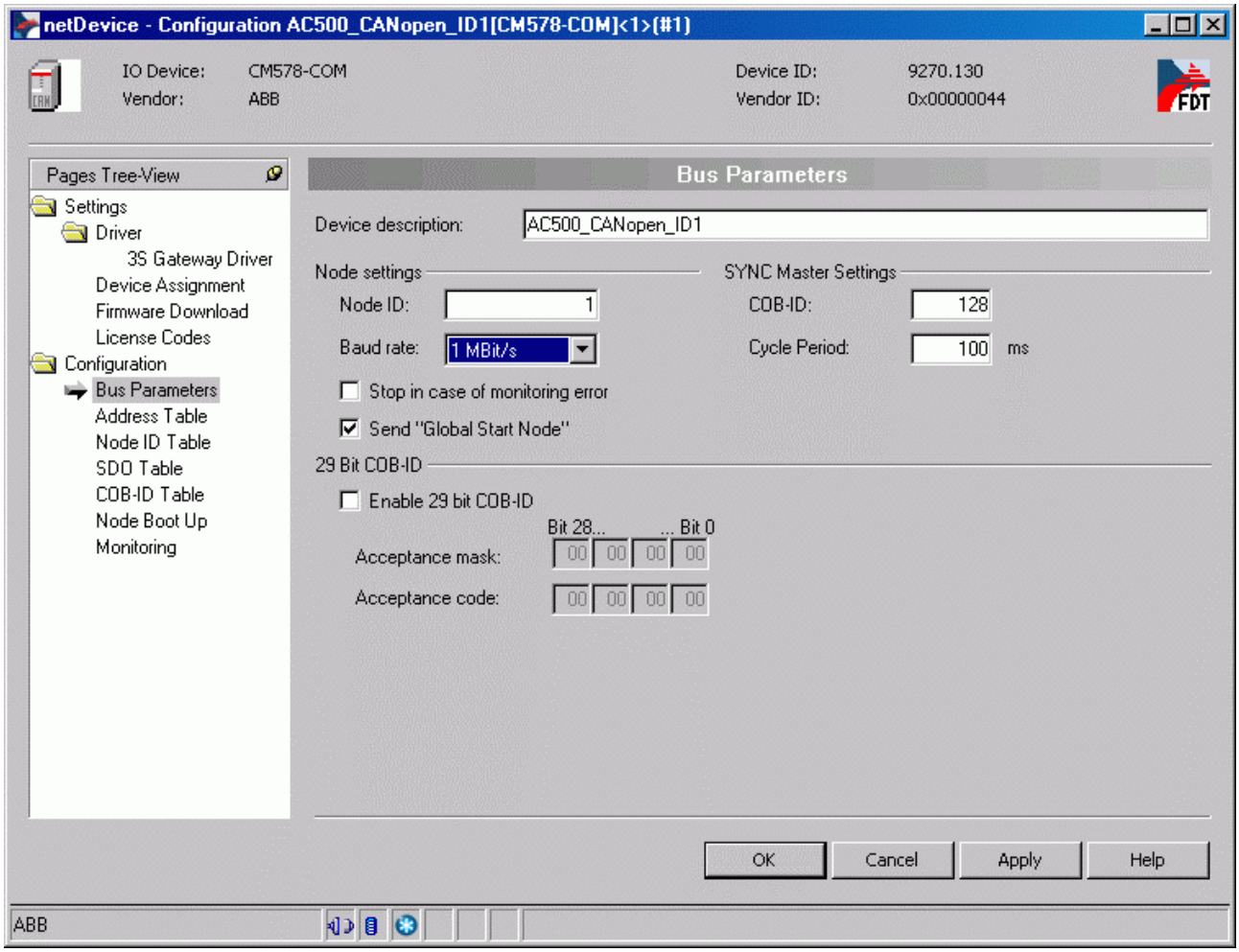
In the top right window, click on the entry "CM578-COM" in the folder "CANopen/Master" and drag it onto the green line displayed in the middle window. Correct insertion positions are displayed by a "+".



A dialog appears where you have to select the board number according to the coupler slot. The first slot left of the CPU is slot 1 (or board number 1).

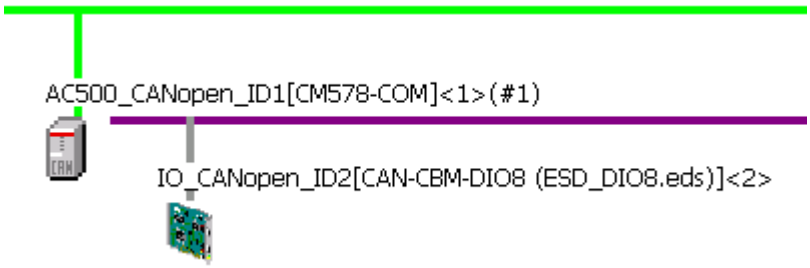


To configure the CANopen master, place the cursor on the "CM578" icon and then press the right mouse button and select "Configuration".



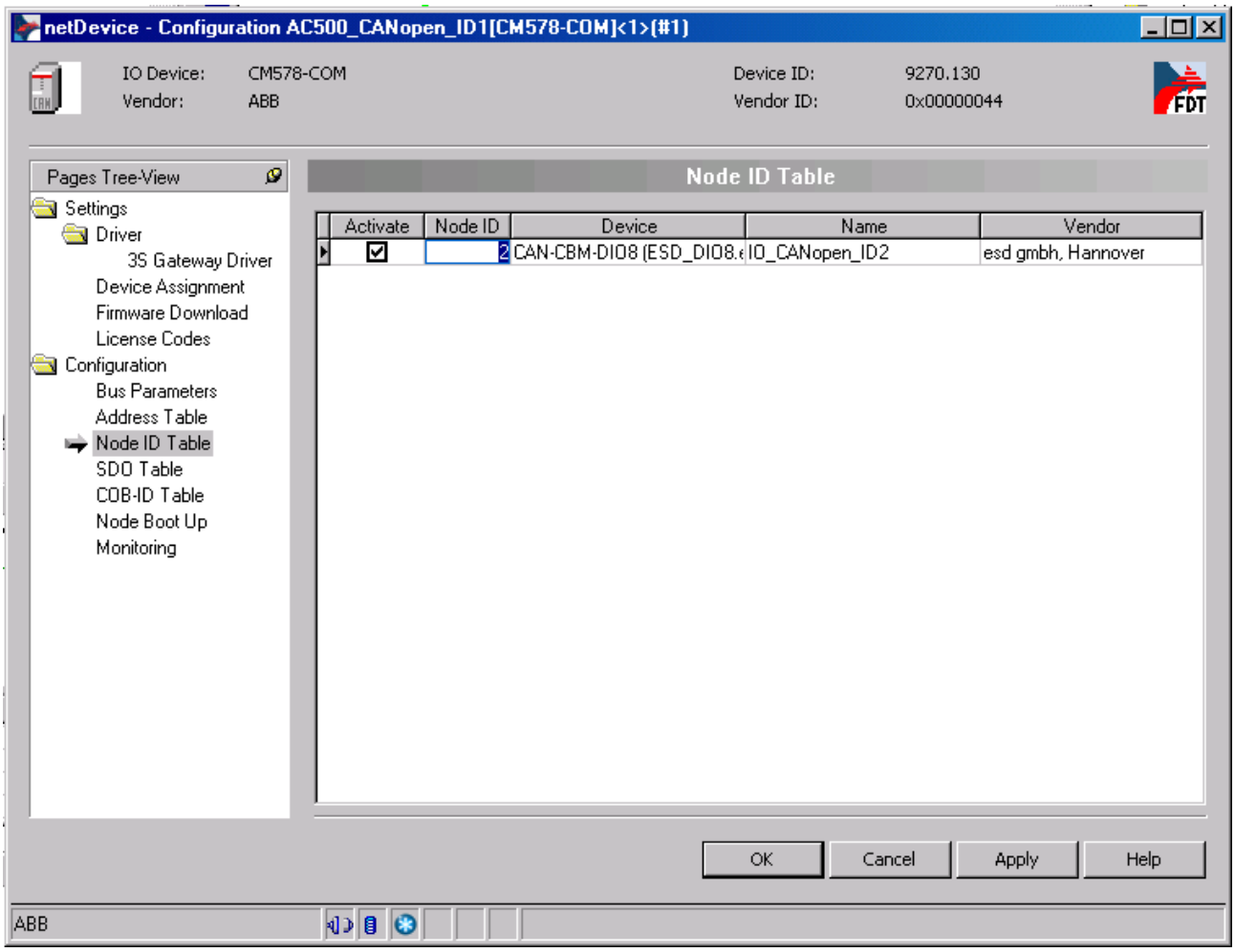
Enter the node ID, select the baud rate of the CANopen coupler and apply the parameters by pressing "OK".

In the top right window, click on the entry "CAN-CBM-DIO8" in the folder "CANopen/Slave" and drag it onto the purple line (CANopen line) displayed in the middle window. Correct insertion positions are displayed by a "+".



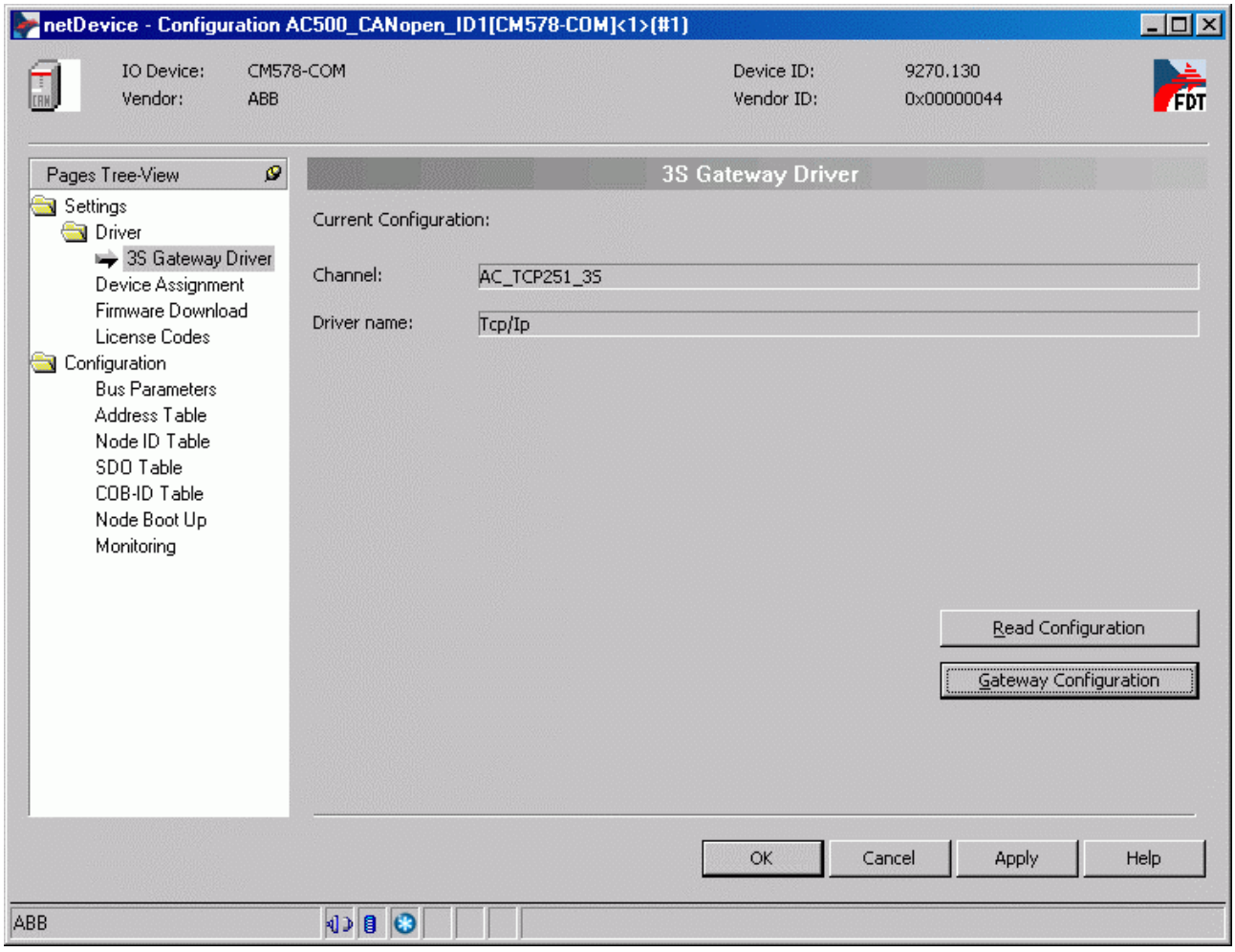
To configure the CANopen slave, place the cursor on the slave icon and then press the right mouse button and select "Configuration".

In this example, we do not change the configuration of the CANopen slave. Only the node ID of the slave is changed according to the HW setting. This is done in the master configuration dialog. Open the master configuration and select the subitem "Node ID table" in the "Configuration" folder.

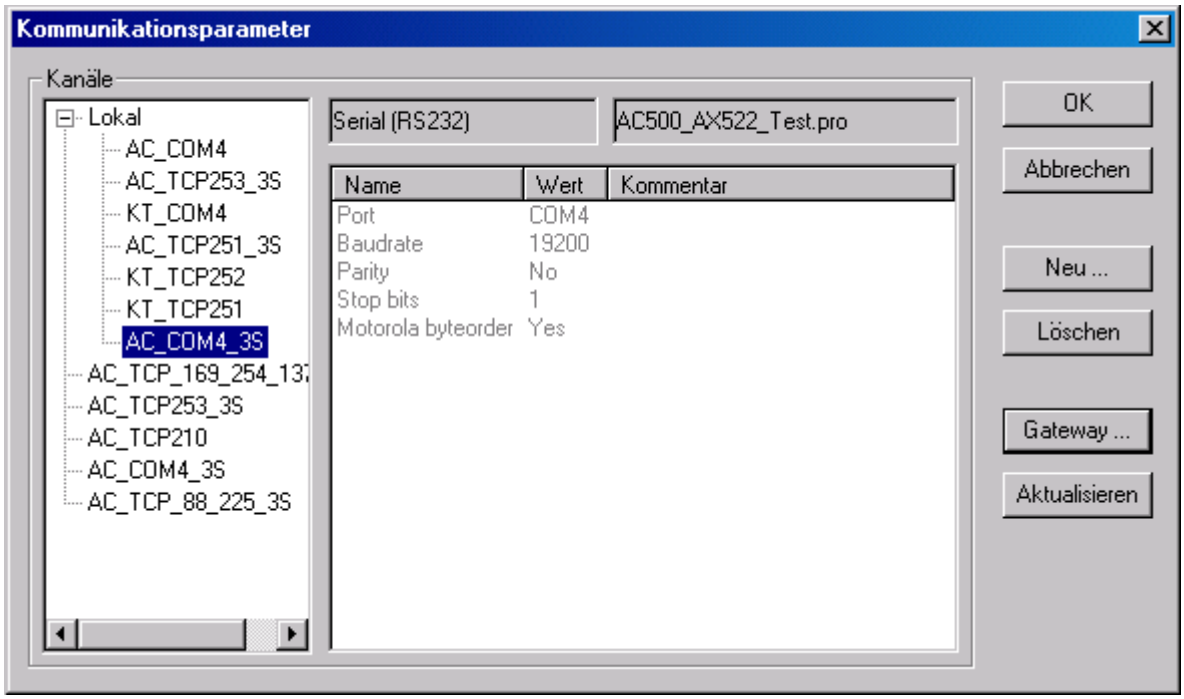


Set the node ID of the slave. Leave all other settings unchanged.

Now the configuration can be downloaded to the CANopen master. To do this, first the interface used to download the data to the coupler has to be specified. The interface to be used is configured in the tab "Settings -> Driver -> 3S Gateway Driver" in the CANopen master configuration.

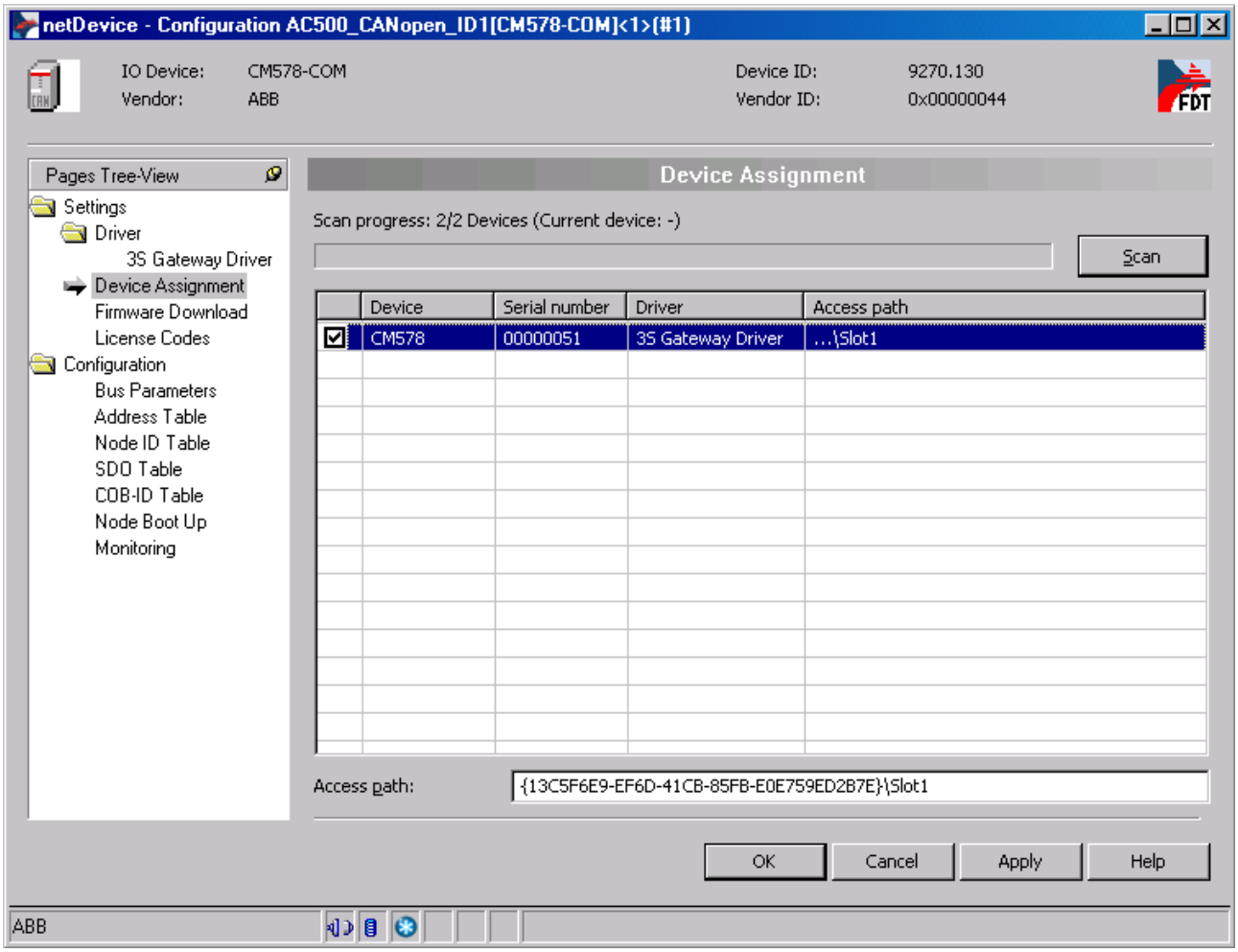


Press the button "Gateway Configuration".



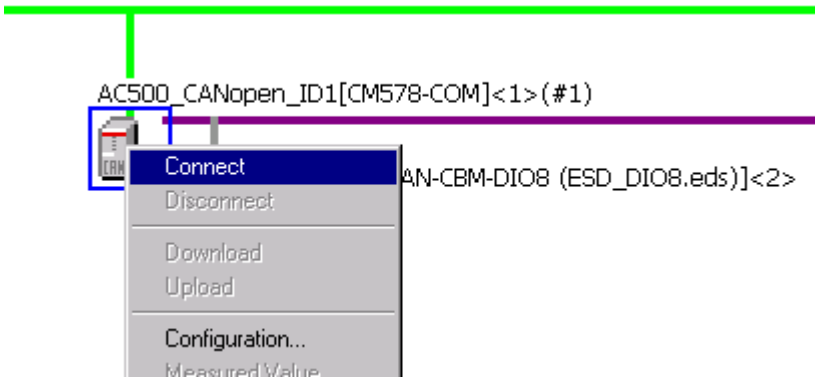
Select the gateway and click on "OK".

Now the configuration tool SYCON.net searches for CANopen couplers that are connected to the selected interface. The tab "Device Assignment" in the CANopen master configuration shows the detected CANopen couplers.



Select the desired coupler by marking the relevant checkbox and confirm your selection with "OK".

Position the cursor on the "CM578" icon, press the right mouse button and select "Connect".

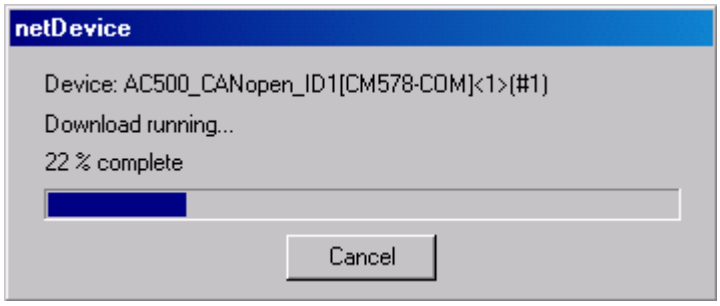


The CM578 icon is then highlighted by green background.

Set the controller to stop state.

Position the cursor on the "CM578" icon, press the right mouse button and select "Download".

Confirm the appearing dialog with "Yes".



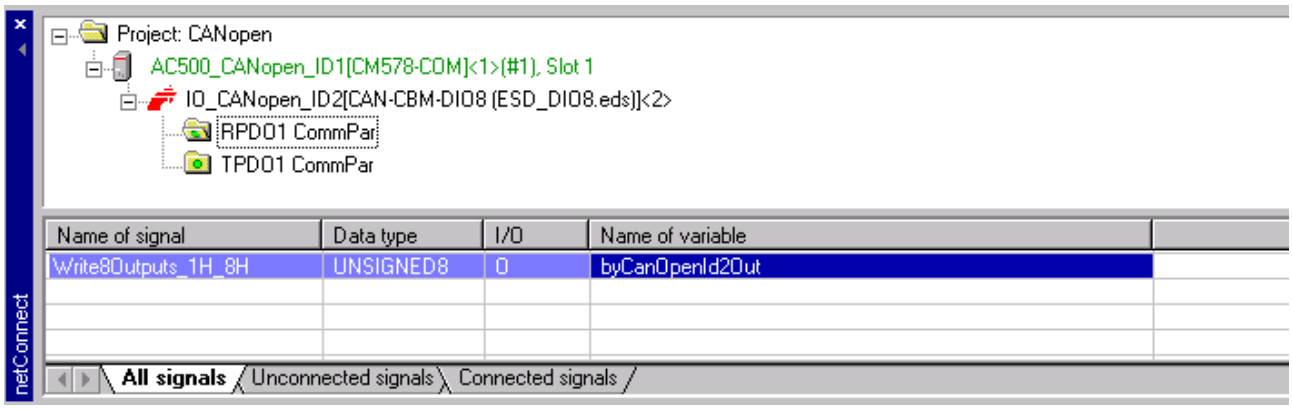
After successful completion of the download process, the "PWR" LED at the CM578 is on and the "RUN" LED flashes.

Position the cursor on the "CM578" icon, press the right mouse button and select "Disconnect".

The coupler configuration is now completed.

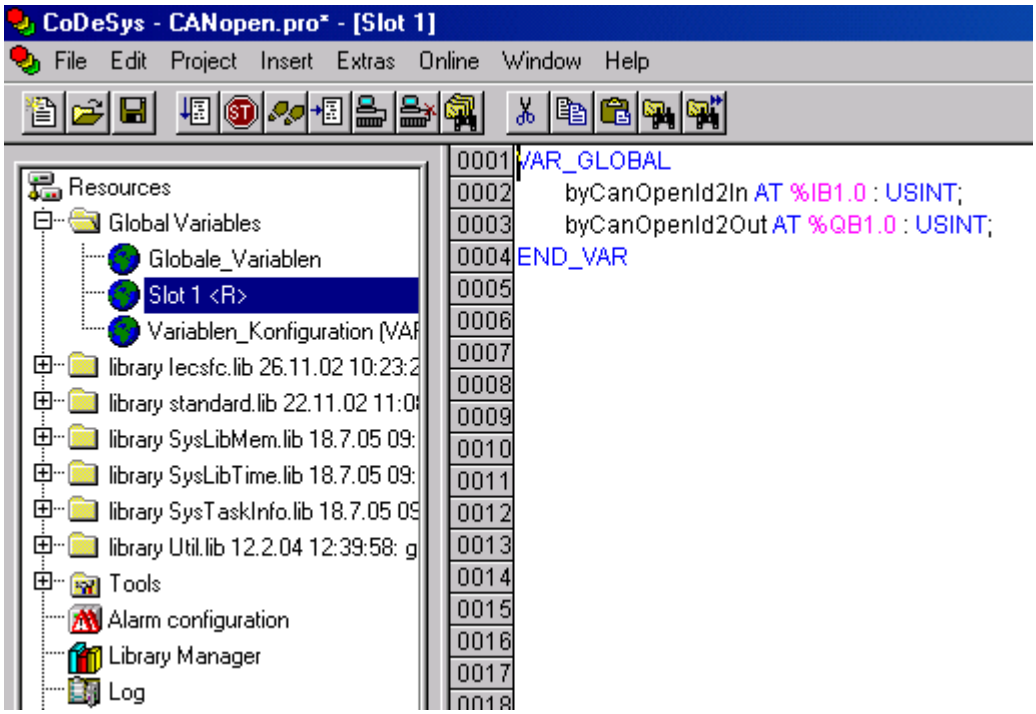
To be able to use the CANopen data in the PLC program, you should assign corresponding variable names to the physical addresses using SYCON.net. These variables will then be available in the Control Builder and can be used directly there.

The assignment of the variable names is done in the netConnect window in SYCON.net.



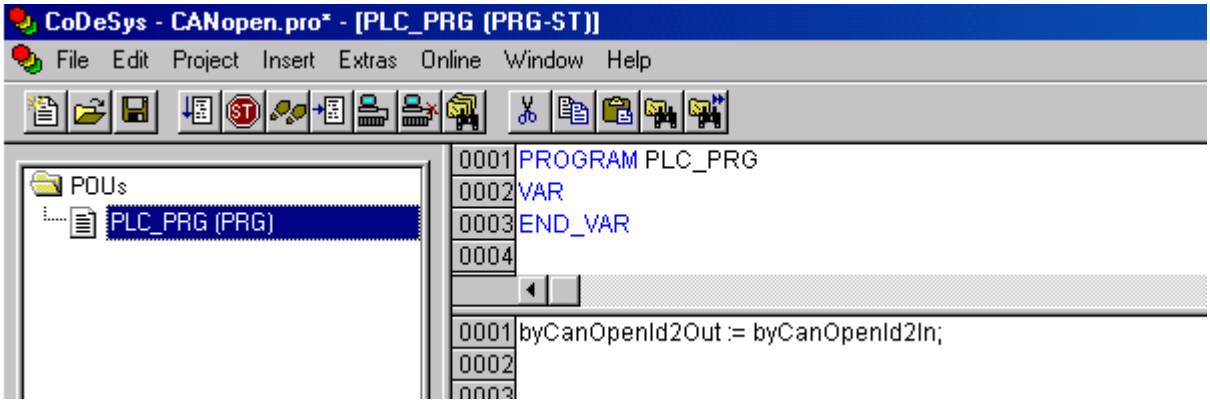
The variable name has to be entered in the field "Variable Name". Double click to open the corresponding input field.

All variables declared here are automatically added to the "Global variables" folder when switching from SYCON.net to the Control Builder.



The declaration of the variables is now completed. The coupler variables can now be used in the user program.

The CANopen configuration is now tested by copying the CANopen slave ID2 inputs to the CANopen slave ID2 outputs in the user program.



Now the user program can be downloaded to the controller and started.

During exchange of CANopen data, the "RUN" LED at the CM578 continuously lights up.

1.5.2 Running operation

The CANopen protocol is automatically processed by the coupler and the operating system of the controller. The coupler is only active on the bus if it has been initialized correctly before and if the user program is running. No function blocks are necessary for exchanging process data via CANopen. Special CAN functions can be realized using the function blocks of the CANopen library.

The coupler starts communication via CANopen after the user program is started and then attempts to initialize the configured slaves. After a successful initialization, the slave exchanges the process data. The exchange of I/O data with the slaves is done automatically.

If the user program is stopped, the coupler shuts down the CANopen communication in a controlled manner.

1.5.3 Error diagnosis

CANopen communication errors are indicated by the coupler LEDs. Malfunctions of the CANopen driver or the coupler itself are indicated by the corresponding error class in the PLC (refer to System Technology of the CPU / The diagnosis system in AC500). Furthermore, the CANopen library provides different function blocks which allow detailed error diagnosis (refer to "The CANopen Library").

1.5.4 Function blocks

Libraries:

CANopen_AC500_V11.lib

Group: CAN 2.0A	
CAN2A_INFO	Reading information about CAN 2.0A communication
CAN2A_REC	Reading CAN 2.0A telegrams (with 11 bit identifier) from a receive buffer
CAN2A_SEND	Transmitting CAN 2.0A telegrams (with 11 bit identifier)

Group: CAN 2.0B	
CAN2B_INFO	Reading information about CAN 2.0B communication
CAN2B_REC	Reading CAN 2.0B telegrams (with 29 bit identifier) from a receive buffer
CAN2B_SEND	Transmitting CAN 2.0B telegrams (with 29 bit identifier)

Group: CANopen master / NMT controller	
CANOM_NMT	Controlling NMT node states via network management

Group: CANopen master / Status / Diagnosis	
CANOM_NODE_DIAG	Polling diagnosis data from a slave
CANOM_RES_ERR	Resetting the coupler's error indications
CANOM_STATE	Reading the CANopen coupler status
CANOM_SYS_DIAG	Displaying status surveys of all slaves

Group: SDO parameters	
CANOM_SDO_READ	Reading the value of a slave object
CANOM_SDO_WRITE	Writing the value of a slave object

1.6 Diagnosis

1.6.1 Status LEDs

LED	Color	Status	Meaning
PWR	green	ON	Voltage on
		OFF	Voltage off
RDY	yellow	ON	Coupler ready
		flashes cyclic	Bootstrap loader active
		flashes non-cyclic	Hardware or system error
		OFF	Defective hardware
RUN	green	ON	Communication in progress
		flashes cyclic	Ready for communication
		flashes non-cyclic	Parameterization error
		OFF	No communication or voltage off
STA	yellow	ON	CANopen master: Sends data
		OFF	CANopen master: No data
ERR	red	ON	CANopen error
		OFF	no error

1.6.2 CANopen error messages

The CANopen error messages are listed in the section 'Error messages for the block libraries'.

1.6.3 Function blocks

CANopen master:

Refer to 1.5.4 Function blocks

Online diagnosis:

Refer to the documentation for the field bus configuration tool SYCON.net

1.7 Further information

1.7.1 Standardization

BOSCH CAN specification - version 2.0, part A & part B

ISO 11898

CiA DS 201 V1.1 - CAN Application Layer

CiA DS 301 V3.0 - CAL based Communication Profile for Industrial Systems

CiA DS 301 V4.02 - CANopen Application Layer and Communication Profile

CiA DS 401 V2.1 - CANopen Device Profile Generic I/O modules

CiA DS 402 V2.0 - CANopen Device Profile Driver and Motion Control

CiA DS 406 V3.0 - CANopen Device Profile Encoder

1.7.2 Important address

CAN in Automation (CiA)
Am Weichselgarten 26
D-91058 Erlangen
Germany

Phone: (+49) 9131 69086-0
Fax: (+49) 9131 69086-79
Internet: http://www.can-cia.de

1.7.3 Terms, definitions and abbreviations

CAL	CAN Application Layer
CAN	Controller Area Network
CiA	CAN in Automation international users and manufacturers group e.V.
DLC	Data Length Code
EDS	Electronic Data Sheet
ISO	International Standardization Organization
NMT	Network Management
OD	Object Directory
PDO	Process Data Object
RTR	Remote Transmission Request
SDO	Service Data Object

Index - System Technology of the CANopen Couplers

O

P

R

S

T