SIEMENS

SIMATIC

ET 200SP Communication Module CAN (6ES7137-6EA00-0BA0)

Equipment Manual

01/2023

A5E48404255-AD

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

⚠ DANGER

indicates that death or severe personal injury **will** result if proper precautions are not taken.

⚠ WARNING

indicates that death or severe personal injury **may** result if proper precautions are not taken.

⚠ CAUTION

indicates that minor personal injury can result if proper precautions are not taken.

NOTICE

indicates that property damage can result if proper precautions are not taken.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

⚠ WARNING

Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens AG
Digital Industries
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E48404255-AD
Ⓟ 06/2023 Subject to change

Preface

Purpose of the documentation

This manual supplements the ET 200SP Distributed I/O System (https://support.industry.siemens.com/cs/ww/en/view/58649293) system manual. Functions that generally relate to the ET 200SP system are described in the system manual.

The information in this equipment manual and the system/function manuals enable you to commission the CAN communication module.

Conventions

SIMATIC S7 controller: When the term "SIMATIC S7" controller is used hereafter, it applies to the central modules of the S7-1200/1500 automation system, of the Open Controller and of the ET 200SP distributed I/O system.

STEP 7: In this documentation, "STEP 7" is used as a synonym for all versions of the configuration and programming software "STEP 7 (TIA Portal)".

Please also observe notes marked as follows:

Note

A note contains important information on the product described in the documentation, on the handling of the product or on the section of the documentation to which particular attention should be paid.

Recycling and disposal

For environmentally friendly recycling and disposal of your old equipment, contact a certified electronic waste disposal company and dispose of the equipment according to the applicable regulations in your country.

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions constitute one element of such a concept.

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place.

For additional information on industrial security measures that may be implemented, please visit (https://www.siemens.com/industrialsecurity).

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customers' exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed visit (https://www.siemens.com/cert).

Open Source Software

Open-source software is used in the firmware of the product described. Open Source Software is provided free of charge. We are liable for the product described, including the open-source software contained in it, pursuant to the conditions applicable to the product. Siemens accepts no liability for the use of the open source software over and above the intended program sequence, or for any faults caused by modifications to the software.

For legal reasons, we are obliged to publish the original text of the license conditions and copyright notices. Please read the information relating to this on the Internet (https://support.industry.siemens.com/cs/ww/en/view/109740777).

Table of contents

ET 200SP Documentation Guide

1

1.1 Information classes ET 200SP

The documentation for the SIMATIC ET 200SP distributed I/O system is arranged into three areas.

This arrangement enables you to access the specific content you require.

You can download the documentation free of charge from the Internet (https://support.industry.siemens.com/cs/ww/en/view/109742709).

Basic information

The System Manual describes in detail the configuration, installation, wiring and commissioning of the SIMATIC ET 200SP distributed I/O system.

The STEP 7 online help supports you in the configuration and programming.

Examples:

- ET 200SP System Manual
- System Manual ET 200SP HA/ET 200SP modules for devices used in a hazardous area
- Online help TIA Portal

Device information

Equipment manuals contain a compact description of the module-specific information, such as properties, wiring diagrams, characteristics and technical specifications.

Examples:

- Equipment Manuals CPUs
- Equipment Manuals Interface Modules
- Equipment Manuals Digital Modules
- Equipment Manuals Analog Modules
- Equipment Manuals Motor Starter
- BaseUnits Equipment Manuals
- Equipment Manual Server Module
- Equipment Manuals Communications Modules
- Equipment Manuals Technology Modules

General information

The function manuals contain detailed descriptions on general topics relating to the SIMATIC ET 200SP distributed I/O system.

Examples:

- Function Manual ET 200AL/ET 200SP Mixed Configuration
- Function Manual Diagnostics
- Function Manual Communication
- PROFINET Function Manual
- PROFIBUS Function Manual
- Function Manual Designing Interference-free Controllers
- MultiFieldbus Function Manual

Product Information

Changes and supplements to the manuals are documented in a Product Information. The Product Information takes precedence over the device and system manuals.

You can find the latest Product Information on the ET 200SP distributed I/O system on the Internet. (https://support.industry.siemens.com/cs/de/en/view/73021864)

Manual Collection ET 200SP

The Manual Collection contains the complete documentation on the SIMATIC ET 200SP distributed I/O system gathered together in one file.

You can find the Manual Collection on the Internet.
(https://support.industry.siemens.com/cs/cn/en/view/84133942)

Manual Collection fail-safe modules

The Manual Collection contains the complete documentation on the fail-safe SIMATIC modules, gathered together in one file.

You can find the Manual Collection on the Internet.
(https://support.industry.siemens.com/cs/ww/en/view/109806400)

1.2 Basic tools

The tools described below support you in all steps: from planning, over commissioning, all the way to analysis of your system.

TIA Selection Tool

The TIA Selection Tool tool supports you in the selection, configuration, and ordering of devices for Totally Integrated Automation (TIA).

As successor of the SIMATIC Selection Tools , the TIA Selection Tool assembles the already known configurators for automation technology into a single tool.

With the TIA Selection Tool , you can generate a complete order list from your product selection or product configuration.

You can find the TIA Selection Tool on the Internet.
(https://support.industry.siemens.com/cs/ww/en/view/109767888)

SIMATIC Automation Tool

You can use the SIMATIC Automation Tool to perform commissioning and maintenance activities on various SIMATIC S7 stations as bulk operations independent of TIA Portal.

The SIMATIC Automation Tool offers a wide range of functions:

- Scanning of a PROFINET/Ethernet system network and identification of all connected CPUs
- Assignment of addresses (IP, subnet, Gateway) and device name (PROFINET device) to a CPU
- Transfer of the date and the programming device/PC time converted to UTC time to the module
- Program download to CPU
- RUN/STOP mode switchover
- CPU localization through LED flashing
- Reading out of CPU error information
- Reading the CPU diagnostic buffer
- Reset to factory settings
- Firmware update of the CPU and connected modules

You can find the SIMATIC Automation Tool on the Internet.
(https://support.industry.siemens.com/cs/ww/en/view/98161300)

PRONETA

SIEMENS PRONETA (PROFINET network analysis) is a commissioning and diagnostic tool for PROFINET networks. PRONETA Basic has two core functions:

- In the network analysis, you get an overview of the PROFINET topology. Compare a real configuration with a reference installation or make simple parameter changes, e.g. to the names and IP addresses of the devices.
- The "IO test" is a simple and rapid test of the wiring and the module configuration of a plant, including documentation of the test results.

You can find SIEMENS PRONETA Basic on the Internet:
(https://support.industry.siemens.com/cs/ww/en/view/67460624)

SIEMENS PRONETA Professional is a licensed product that offers you additional functions. It offers you simple asset management in PROFINET networks and supports operators of automation systems in automatic data collection/acquisition of the components used through various functions:

- The user interface (API) offers an access point to the automation cell to automate the scan functions using MQTT or a command line.
- With PROFIenergy diagnostics, you can quickly detect the current pause mode or the readiness for operation of devices that support PROFIenergy and change these as needed.
- The data record wizard supports PROFINET developers in reading and writing acyclic PROFINET data records quickly and easily without PLC and engineering.

You can find SIEMENS PRONETA Professional on the Internet.
(https://www.siemens.com/proneta-professional)

SINETPLAN

SINETPLAN, the Siemens Network Planner, supports you in planning automation systems and networks based on PROFINET. The tool facilitates professional and predictive dimensioning of your PROFINET installation as early as in the planning stage. In addition, SINETPLAN supports you during network optimization and helps you to exploit network resources optimally and to plan reserves. This helps to prevent problems in commissioning or failures during productive operation even in advance of a planned operation. This increases the availability of the production plant and helps improve operational safety.

The advantages at a glance

- Network optimization thanks to port-specific calculation of the network load
- Increased production availability thanks to online scan and verification of existing systems
- Transparency before commissioning through importing and simulation of existing STEP 7 projects
- Efficiency through securing existing investments in the long term and the optimal use of resources

You can find SINETPLAN on the Internet
(https://new.siemens.com/global/en/products/automation/industrial-communication/profinet/sinetplan.html).

1.3 SIMATIC Technical Documentation

Additional SIMATIC documents will complete your information. You can find these documents and their use at the following links and QR codes.

The Industry Online Support gives you the option to get information on all topics. Application examples support you in solving your automation tasks.

Overview of the SIMATIC Technical Documentation

Here you will find an overview of the SIMATIC documentation available in Siemens Industry Online Support:

Industry Online Support International (https://support.industry.siemens.com/cs/ww/en/view/109742705)

Watch this short video to find out where you can find the overview directly in Siemens Industry Online Support and how to use Siemens Industry Online Support on your mobile device:

Quick introduction to the technical documentation of automation products per video (https://support.industry.siemens.com/cs/us/en/view/109780491)

YouTube video: Siemens Automation Products - Technical Documentation at a Glance (https://youtu.be/TwLSxxRQQsA)

mySupport

With "mySupport" you can get the most out of your Industry Online Support.

Registration	You must register once to use the full functionality of "mySupport". After registration, you can create filters, favorites and tabs in your personal workspace.
Support requests	Your data is already filled out in support requests, and you can get an overview of your current requests at any time.
Documentation	In the Documentation area you can build your personal library.
Favorites	You can use the "Add to mySupport favorites" to flag especially interesting or frequently needed content. Under "Favorites", you will find a list of your flagged entries.
Recently viewed articles	The most recently viewed pages in mySupport are available under "Recently viewed articles".
CAx data	The CAx data area gives you access to the latest product data for your CAx or CAe system. You configure your own download package with a few clicks: • Product images, 2D dimension drawings, 3D models, internal circuit diagrams, EPLAN macro files • Manuals, characteristics, operating manuals, certificates • Product master data

You can find "mySupport" on the Internet. (https://support.industry.siemens.com/My/ww/en)

Application examples

The application examples support you with various tools and examples for solving your automation tasks. Solutions are shown in interplay with multiple components in the system - separated from the focus on individual products.

You can find the application examples on the Internet.
(https://support.industry.siemens.com/cs/ww/en/ps/ae)

New properties/functions

2

2.1 Version overview

HSP/FW versions

TIA Portal version	HSP required	HSP version	FW version	Note
V15; V15.1	Yes	HSP_V15_1_0310_001_ET200SP_CM_CAN_1.0	V1.0	
V16	Yes	HSP_V16_0310_002_ET200SP_CM_CAN_1.0	V1.0; V1.1	"Simple slave functionality" included
V17	No	-	V1.0; V1.1	"Simple slave functionality" included
V17 Update 6	No	-	V1.0; V1.1; V1.2	OD index entry range extended, "Block Error passive alarm" function included
V18 Update 1	No	-	V1.0; V1.1; V1.2	OD index entry range extended, "Block Error passive alarm" function included
V19	No	-	V1.0; V1.1; V1.2	OD index entry range extended, "Block Error passive alarm" function included

2.2 Changes compared to previous version

What's new as of firmware version V1.2

There are now three entries in the HW catalog for the CM CAN module with firmware versions V1.0, V1.1 and V1.2. Version V1.2 includes all functions of Version V1.1 plus a new "Block Error passive alarm" function. The OD index entry range has been extended to 0x6FFF. These functions are available as of TIA Portal V17 Update 6.

Note

Backward compatibility

An existing configuration of CM CAN module V1.2 cannot be downgraded from V1.2 to V1.1 within a TIA Portal project. The same applies to a downgrade to Version V1.0 from a higher version (V1.1 and higher). If a downgrade to Version V1.0 is required, you must recreate the CM CAN module for Version V1.0 within a TIA Portal project. In this case, however, you do not have access to any of the functions that have been added in the higher versions.

An update of Version V1.0 to a higher version (V1.1 and higher) within a TIA Portal project is possible. In addition, you can upgrade Version V1.1 to V1.2.

You can perform a firmware update/firmware downgrade in the device without any restrictions.

What's new as of firmware version V1.1

There are now two entries in the HW catalog for the CM CAN module with firmware versions V1.0 and V1.1. The difference between the CM CAN module V1.0 and V1.1 is the "Simple Slave functionality" that is included in the CM CAN module V1.1.

This functionality is available as of TIA Portal V16 and enables the CM CAN module to work with CANopen devices that do not fully comply with the CANopen standards defined by the CiA group (https://www.can-cia.org/). You can find additional information below.

Note

Backward compatibility

An existing configuration of CM CAN module V1.1 cannot be downgraded from V1.1 to V1.0 within a TIA Portal project. If a downgrade is required, you must recreate the CM CAN module for V1.0 within a TIA Portal project. In this case, however, the "Simple Slave functionality" is not available.

You can perform a firmware update/firmware downgrade without any restrictions.

New parameters

Under "General > Module parameters > Diagnostics"

• Check box Block Error passive alarm

This option can be selected in any operating mode.

If you select this option and the CM CAN module receives an Error Passive alarm, the error passive state of the CAN bus is not reported as an alarm in the diagnostic buffer. The ERROR LED remains off.

If you deselect this option and the CM CAN module receives an error passive alarm, the Error Passive state of the CAN bus is reported as an alarm in the diagnostic buffer. The ERROR LED on the device flashes red.

This option is enabled by default.

Note

If the "Enable additional diagnostic alarms" check box is deselected, the "Block Error passive alarm" check box is disabled.

Under "General > Module parameters > Diagnostics"

• Check box Enable diagnostic alarms for not mandatory nodes

The option can only be selected in the CANopen Manager mode and applies only to nodes that are non-mandatory.

When you select this option, error messages from such nodes are also displayed in the diagnostic buffer and the ERROR LED lights up. All other nodes on the CAN bus are not affected by this option.

If you deselect this option, the disconnecting of nodes from the CAN bus and any pending error messages in the CANopen network are not sent to the diagnostic buffer, and the ERROR LED remains off.

Under "CANopen Manager > General > Communication"

• max. SDO Timeout

Some "simple slave" devices are slow to respond to commands of the CANopen Manager. Such devices work on other, more important tasks and CANopen communication has low priority. The reaction to the command from the manager is therefore delayed. In this case, the manager waits for the "max SDO timeout" (ms). The SDO request is aborted when the preset time has elapsed. There is also a 1 second delay before a new request is sent.

The SDO Timeout is only displayed to the user when an SDO with the data record number 0x200 - 0x20F is requested. The timeout is displayed instead of the 4-byte code "Additional error information code" as 0x05040000 (code number taken from CiA standard "CiA 301"). This code occupies the 6th to 9th byte of the data record read.

• max. BootUp Timeout

The max. bootup timeout gives the maximum response time in seconds for the startup process of a node. When a node does not respond to the requests of the manager within the specified time, the diagnostic message "BootUp, node not responding" is sent to the PLC. The diagnostic message is deleted as soon as the node responds correctly to the manager.

Under "CANopen Manager > Receive process data objects (PDOs) > Receive PDO definition"

• Disable checking of PDO length

When you select this option, the data length (number of received bytes) of the received PDOs is adjusted. Error, warning or diagnostic messages are not generated. The value is adjusted to the correct data length as follows:

- Data below the defined length are supplemented with the value 0.
- If data exceeds the defined length, the more significant bytes are deleted.

When you disable this option, all data from a received PDO with a length deviating from the standard is discarded and replaced with the value of the last valid PDO (last received PDO with the correct length). The CANopen Manager calls up an error message in the diagnostic buffer.

Under "CANopen Manager > CANopen node > Communication"

• No NMT / Layer 2 slave

When you select this option, the NMT status of the node is not evaluated. The CANopen configuration phase is completely skipped for this node including NMT commands. The configuration menus for EMCY, Heartbeat and Node Guarding as well as the PDOs are blocked (write-protected) and show the values from the object dictionary (EDS file).

Exception: A node can be monitored by the manager using the Heartbeat/Node Guarding functions. An error is only signaled to the CPU when the Heartbeat/Node Guarding message is not received in time. The NMT status of the message is not evaluated; any status is accepted.

Typical use case for this option are slave nodes that do not require NMT commands for their commissioning and use. After power-on, such devices immediately enter the operating mode and start sending the PDO messages configured by default.

If you have made parameter changes before setting the "No NMT / Layer 2 slave" option, these values are retained. However, they have no effect on the actual configuration of the node in the system.

• Loose configuration

The main purpose of this option is to ignore unexpected errors during the configuration phase. The configuration of the node is always completed.

This option only affects the configuration phase after startup or after resetting the bus or node. The status of a node is read by the master during the configuration phase, although it is ignored. In this way, the master can work with devices that are in operating mode immediately after power-on or with devices that communicate differently, contrary to the definition in their EDS file.

When a node of one of the mandatory objects 1000, 1018.01, 1018.02, 1018.03 is not supported, this check box is always selected and cannot be cleared.

Editable OD table (manually defined object dictionary)

You can find additional information in the section "Handling of the OD table" in the section Configuration in the TIA Portal (Page 79).

Disabling a configured node temporarily

You can find additional information in the section "Disabling a node temporarily**"** in the section Configuration in the TIA Portal (Page 79).

Implementation of the 0x1F82 object - NMT request

The object 0x1F82 NMT request is accessible from another CANopen device via the CANopen network upon RD-SDO request. Contrary to the definition in the CiA standard "CiA 302 Part 2", this object is implemented read-only. Write access is not permitted. When the manager itself wants to read data of this object, it must use the RDREC instruction with data record number 0x211 for this.

Note

This functionality is only available in the "CANopen Manager" operating mode.

It returns an array of 127 bytes for the status of the node ID from 1 to 127.

RET ARRAY[0] = Status of node 1

RET ARRAY[1] = Status of node 2

...

RET ARRAY[126] = Status of node 127

Note

Reading from 0x1F82 OD with the SDO command only returns a status byte of a node instead of an array of bytes. For each node, therefore, an SDO command must be sent with the ID of the requested node as an input parameter.

The status of a node can have the following values (according to CiA standard "CiA 302 Part 2"):

- 0 = unknown state
- 1 = node missing
- 4 = NMT/STOPPED
- 5 = NMT/OPERATIONAL
- 127 = NMT/PRE-OPERATIONAL

Product overview

3

3.1 Properties

Article number

6ES7 137-6EA00-0BA0

View of the module

Figure 3-1 SIMATIC ET 200SP CM CAN

The CAN communication module can be used in the ET 200SP distributed I/O system. You connect the module directly to the ET 200SP CPU or to the ET 200SP interface module.

You can find additional information on the configuration of the ET 200SP and the associated modules in the ET 200SP distributed I/O system (https://support.industry.siemens.com/cs/ww/en/view/58649293) system manual.

General properties

The CAN communication module has the following properties:

- CAN interface according to ISO 11898-2 (High Speed CAN)
- The CAN protocol and CANopen protocol are implemented in the module. The module assumes the function of a CANopen slave or CANopen manager in the CANopen network. The module can operate in the following three modes:
  - CANopen manager
  - CANopen slave
  - CAN transparent
- The operating mode for the module is set using the associated HSP in the TIA Portal.
- The module supports both the standard format (CAN 2.0A) and the extended CAN format (CAN 2.0B).
- The module can operate up to 60 nodes in the CAN network in CANopen manager mode.
- Data transmission rates of 10, 20, 50, 125, 250, 500, 800 and 1 000 kbit/s are supported on the CAN bus side. The data transmission rate 33.333 kbit/s is only supported in "CAN transparent" mode.
- The maximum allowable cable length of the CAN network is up to 1 000 m. The cable length is dependent on:
  - Data transmission rate
  - Cable cross-section
  - Number of nodes

CANopen manager

- CANopen implementation according to CANopen specification "CiA 301"
- The mode can operate up to 60 slaves as CANopen manager in the CANopen network.
- Network management functions according to CiA standard "CiA 302 Part 2".
  When the module is operated as CANopen manager, it takes on the following functions:
  - NMT Master: The module controls the NMT state of the other CAN devices and executes the boot-up procedure according to the CiA standard "CiA 302 Part 2".
  - Configuration Manager: The module configures the CANopen slaves during the boot-up procedure by means of SDO write access.
  - The CM CAN also supports the following CANopen standards:

    CiA 303 Part 1 - Cabling and pin assignment V1.8.0

    CiA 303 Part 3 - LED control - Display specification, V1.4.0

    CiA 309 Part 4 - Mapping of data types and their conversion (Amendment 7 to Fieldbus Integration into PROFINET IO, V1.0.0)
  - The CM CAN does not support the following standards:

    CiA 305 - regarding LSS functionality

    CiA 306 - regarding XDD files

    CiA 1301+CiA 601 - regarding CANopen FD (Flexible Data)
- I/O data is transmitted on the CANopen side with the help of process data objects (PDOs).
- All PDO transmission types specified in the CiA 301 standard are supported.
- Segmented SDO data transmission is supported.
- SYNC function as producer (transmitter) as well as consumer (receiver)
- "Heartbeat" function as producer (transmitter) as well as consumer (receiver)
- "Node Guarding" function as producer (transmitter) as well as consumer (receiver)
- "EMCY" (Emergency) function

CANopen slave

- The module can be used as an "NMT slave" according to "CiA 302 Part 2".
  When the module is an NMT slave, another CANopen manager performs the following processes:
  - Checks the communication and the state of the slaves via NMT protocol.
  - Creates and records PDO messages.
  - Initiates SDO transmission.
  - Determines what was exchanged on the CAN bus.

CAN transparent

- You can operate the module in "CAN transparent" mode.
- All CANopen functions are disabled.
- Control and status information are exchanged cyclically between the module and the SIMATIC S7 controller.
- Messages can be used in standard format and in extended CAN format.
- Configured CAN messages with fixed message ID and fixed length can be used.
- It is possible to use CAN messages in which the message ID and the length (max. 8 bytes) are changed at runtime.
- The user can send and receive CAN messages in the user program. For this purpose, "Transmit proxy modules" and "Receive proxy modules" can be configured in the TIA Portal.

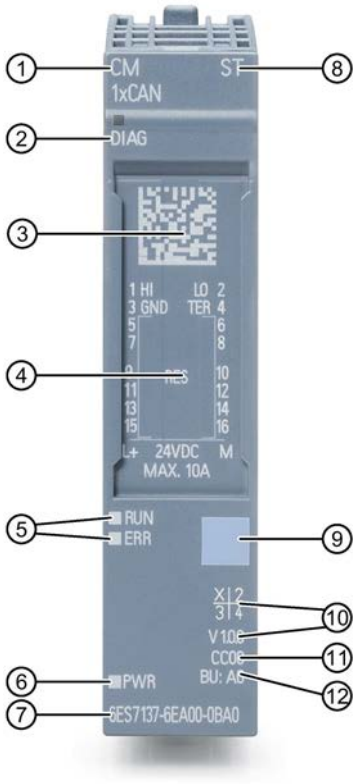
3.2 System requirements

System requirements

- ET 200SP CM CAN
- Controller: SIMATIC S7-1200, SIMATIC S7-1500, SIMATIC ET 200SP, SIMATIC Open Controller are supported.
- Interface module: ET 200SP, if you do not use a controller.
- Light-colored or dark-colored BaseUnit type A0
- 24 V voltage supply
- CAN bus
- Windows PC (for configuring, commissioning and diagnostics)
- TIA Portal V15.1, you can find the required HSP in the Internet (https://support.industry.siemens.com/cs/ww/en/view/72341852). In TIA Portal V17, the required HSP is already integrated. You can find the required library with the function block in Library (https://support.industry.siemens.com/cs/de/en/view/109775840).
- A LAN switch is recommended for configuration, commissioning and diagnostics.

3.3 Design

CM CAN design



① Module type
② LED for diagnostics
③ 2D matrix code
④ Wiring diagram
⑤ Status LEDs
⑥ LED for supply voltage
⑦ Article number
⑧ Function class
⑨ Color coding module type
⑩ Function version and firmware version of the module
⑪ Color code for selecting the color-coded labels
⑫ BU type

Figure 3-2 SIMATIC ET 200SP CM CAN

Functions

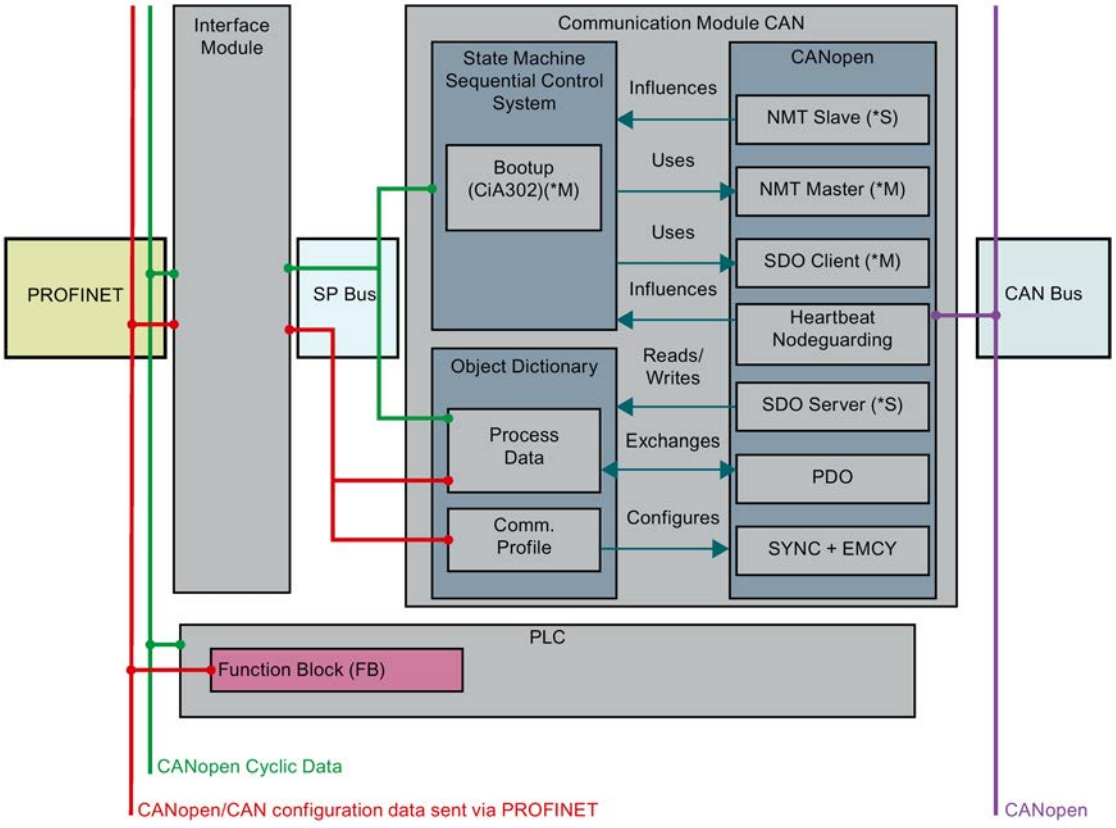
4

4.1 CANopen

4.1.1 Basic functions

Overview

The following overview diagram shows the relations in the communication between the SIMATIC S7 controller and the communication module.



*M Only available in Manager mode

*S Only available in slave mode

Figure 4-1 Communication principle of the CM CAN with CANopen

4.1.1.1 Object Dictionary

Structure of the Object Dictionary (OD)

All communication objects and all user objects are organized in the object dictionary (OD).

The following table describes the areas used by the module:

Table 4- 1 Ranges

OD entry	Meaning
0x1000 ... 0x1FFF	Communication profile
0x2000 ... 0x6FFF	Process data

Default settings and initialization

The default values specified for process data in the TIA Portal are used only for initialization of the values in the OD. The OD entries are created by HSP in the form of a data block. To transmit the configuration data of the data block, use a function block that transmits the configuration data to the module at runtime. You can find more information on the function block in the section Programming (Page 113).

In "CANopen manager" mode, the module remembers for all OD entries whether a PDO with the data has already been received since the last reset to the default values. This applies to the OD entries with content that is being transmitted as cyclic data to the SIMATIC S7 controller. Monitoring is performed in "CANopen slave" mode:

- Whether the entries were written at least once per SDO.
- Whether a matching PDO was received.

This information is compressed into one bit and transmitted cyclically as a group status of all data to the SIMATIC S7 controller.

According to the CiA standard "CiA 301", the default values are always applied to the OD in the NMT state "Initialization", sub-state "Reset Application".

This state is passed through in the following cases:

- After configuration/reconfiguration of the module by the SIMATIC S7 controller
- After reset of the module via a "reset bit" from the SIMATIC S7 controller
- After receiving the NMT command "Reset Node" (only in "CANopen slave" mode)
- After resetting the module and all nodes due to a Heartbeat error (only in "CANopen manager" mode and only when the error response is configured accordingly). You can find additional information on the monitoring functions in the section Heartbeat / Node Guarding (Page 28).

The following actions are being executed in the process:

- The default value is written to the OD.
- The information on whether the OD entry has already been written once by the CANopen side is reset.

Process data objects

Process data is transferred on the CANopen side by PDOs.

The module supports 128 PDOs for transmitting/receiving.

One entry for the "Communication parameters" and one entry for the "Mapping parameters" exists in the OD for each Receive PDO (RX PDO) and each Transmit PDO (TX PDO).

In the configuration you specify:

- The COB ID of the CAN message
- The format of the COB ID (11b or 29b)
- The transmission mode (synchronous, event-controlled with timer, RTR)
- Whether or not the PDO is active
- The OD entries onto which the data of the transmitted or received PDOs are mapped.

Transmission types

The module supports the following PDO transmission types:

Table 4- 2 PDO transmission types

Transmission type	Description
Acyclic synchronous	PDO is transmitted during the next "SYNC" after a value change.
Cyclic synchronous	Depending on the parameter assignment, PDO is transmitted during every 1st to 240th "SYNC", independent of value changes.
RTR only	Only on request (RTR message synchronous/asynchronous) PDO is only transmitted after an RTR message with the COB ID of the PDO has been received. The module supports the "RTR only" transmission mode only for Transmit PDOs and not for Receive PDOs.
Manufacturer-specific event control Application-specific event control	These transmission types are event-controlled (asynchronous). PDO is sent when an event occurs, independent of "SYNC": • If value has changed • After a configurable timeout has elapsed - event time and inhibit time Event-controlled Transmit PDOs with transmission type 254 (manufacturer-specific) and transmission type 255 (application-specific) are also transmitted immediately after the transition to NMT state "Operational".

4.1.1.2 Service Data Objects (SDO)

The module allows write/read access to the OD of other CANopen nodes by SDO from the user program.

The module provides 16 data records which can be accessed by the user program with the communication blocks RDREC (read data record) and WRREC (write data record).

Note

This functionality is only available in the "CANopen manager" operating mode.

For additional information, see section CANopen manager - Control and Status Information (Page 34).

EMCY messages

The module transmits the following EMCY messages:

Table 4- 3 EMCY messages

Error-Code	Meaning
8Fxxh	Heartbeat error or Node Guarding error for <xx> nodes. Transmitted when the module is in "CANopen slave" operating mode and the heartbeat for a monitored node fails.
8130h	General Heartbeat error or Node Guarding error. Sent if the module is in "CANopen slave" mode, Node Guarding is enabled and the monitoring by the master fails.
FF90h	Sent if the connection to the SIMATIC S7 controller fails in the following cases: • The connection (Application Relation) is terminated • The data from the SIMATIC S7 controller has the IO provider status "Bad" Transmitted when the module is being operated as a "CANopen slave" and the NMT Manager attempts to put the module into "Operational" state.
FF91h	Transmitted when the module is being operated as a "CANopen slave" and the desired state transmitted in the cyclic data from the SIMATIC S7 controller changes to "OFF". Transmitted when the NMT Manager tries to put the module into "Operational" state while the desired state transmitted in the cyclic data from the SIMATIC S7 controller is still "OFF".

The EMCY messages always use the default COB ID intended for this purpose.

4.1.1.3 Heartbeat / Node Guarding

The module supports the Heartbeat and Node Guarding monitoring functions both as producer (transmitter) and consumer (receiver).

- "CANopen manager" operating mode: The parameters for Heartbeat and Node Guarding are defined during configuration.
- "CANopen slave" operating mode: The settings for these two functions are written to the OD (100Ch, 100Dh, 1016h and 1017h) of the module via SDO access by the CANopen manager responsible for the configuration and management of the CANopen network.

The HSP ensures that either Heartbeat or Node Guarding are configured in "CANopen manager" mode.

Note

The setting of the monitoring function made in the HSP applies to the entire network with all CANopen nodes. Mixed operation with different monitoring functions is not possible. Nevertheless, slave nodes without the Heartbeat or Node Guarding functions can also be put into operation.

Note

If a node does not support monitoring functions (or if Heartbeat and Node Guarding are disabled), a failure or fault of the node is **not** recognized by the CM CAN as CANopen Manager. Even when you select the "Node is mandatory on the network" option for this node, you must ensure that the required monitoring is in place.

Fault reaction

When a monitored CANopen device fails, the module signals corresponding diagnostic information to the SIMATIC S7 controller. In addition, an EMCY message is sent on the CANopen side and the fault reaction that is configured or defined in the OD (1029h) is executed.

The possible error responses for manager/slave are available in the following sections:

- CANopen manager: CANopen manager (Page 31)
- CANopen slave: CANopen slave (Page 43)

4.1.1.4 SYNC messages

The module supports the SYNC protocol as transmitter (producer) and as receiver (consumer).

The SYNC receiver function is required for PDOs with "synchronous" transmission types. For additional information, see section Object Dictionary (Page 25).

In the configuration, you specify whether the module is a SYNC transmitter or SYNC receiver as well as the transmission interval. In "CANopen manager" mode, a different CANopen slave can be configured as SYNC transmitter. In "CANopen slave" mode, the COB ID to be used for this can be changed by the manager through SDO. In "CANopen manager" mode, the COB ID is preset to the fixed value 0x80.

Note

The SYNC function only has an effect on the CANopen side.

4.1.1.5 Data exchange between the SIMATIC S7 controller and the module

Cyclic data exchange: Process data, control and status information

The process data configured in the HSP is transmitted cyclically between the SIMATIC S7 controller and the module.

The exchanged process data is stored in the module as entries in the object dictionary (OD) (at index 2000h to 6FFFh). The module writes the values of the process data received from the SIMATIC S7 controller to the transmit data OD. The values of the process data to be transmitted to the SIMATIC S7 controller are read from the receive data OD.

Control information and status information is cyclically exchanged between the module and the SIMATIC S7 controller in "CANopen manager" and in "CANopen slave" mode.

The following control information is cyclically transmitted from the SIMATIC S7 controller to the module:

- **Control bit for the state of the module**
  The user can influence the data transmission in the module from the user program using a control bit.
- **Reset bit**
  The user can reset the CANopen side in the module from the user program using a reset bit.

Detailed information on the control and status information is available here:

CANopen manager (Page 31)

CANopen slave (Page 43)

Acyclic data exchange

The module provides 16 data records which can be accessed by the user program with the communication blocks RDREC (read data record) and WRREC (write data record).

For additional information, see section CANopen manager - Control and Status Information (Page 34).

Conversion of data formats between SIMATIC S7 controller and CANopen

The process data is automatically converted from SIMATIC S7 controller to CANopen data format and vice versa according to the configured data type.

The SIMATIC S7 controller interprets data in "Big Endian" format. CANopen and the processors used in the module use the "Little Endian" format. Therefore, the order of the Bytes is adjusted for all data types that occupy more than 1 Byte.

The conversion is performed in the following cases:

- Before the process data received from the SIMATIC S7 controller is written to the transmit data OD
- Before the process data read from the receive data OD is transmitted to the SIMATIC S7 controller

The following table shows the data types supported by the module and their conversion:

Table 4- 4 Conversion of the data types

PROFINET data type	CANopen data type	Conversion
Integer 8	INTEGER 8	None
Integer 16	INTEGER 16	Byte order is automatically converted.
Integer 32	INTEGER 32	
Integer 64	INTEGER 64	
Unsigned 8	UNSIGNED 8	None
Unsigned 16	UNSIGNED 16	Byte order is automatically converted.
Unsigned 32	UNSIGNED 32	
Unsigned 64	UNSIGNED 64	
Float 32 / float	REAL 32	
Float 64 / double	REAL 64	
Bool	Bool	None

4.1.2 CANopen manager

4.1.2.1 Overview

The module can be used as a CANopen manager according to CiA 302 Part 2.

If the module is operated as CANopen manager, the module in CANopen takes over the functions:

- NMT Master: The module controls the NMT state of the other CAN nodes and executes the Boot-Up procedure according to the CiA standard CiA 302 Part 2.
- Configuration Manager: The module configures the nodes during the Boot-Up procedure via SDO write access.

4.1.2.2 CANopen manager - State Model

State model

The state model in "CANopen manager" operating mode is based on the NMT state model described in the CiA standard "CiA 301", taking into account the network management function.

The following figure shows the states of the module in "CANopen manager" mode.

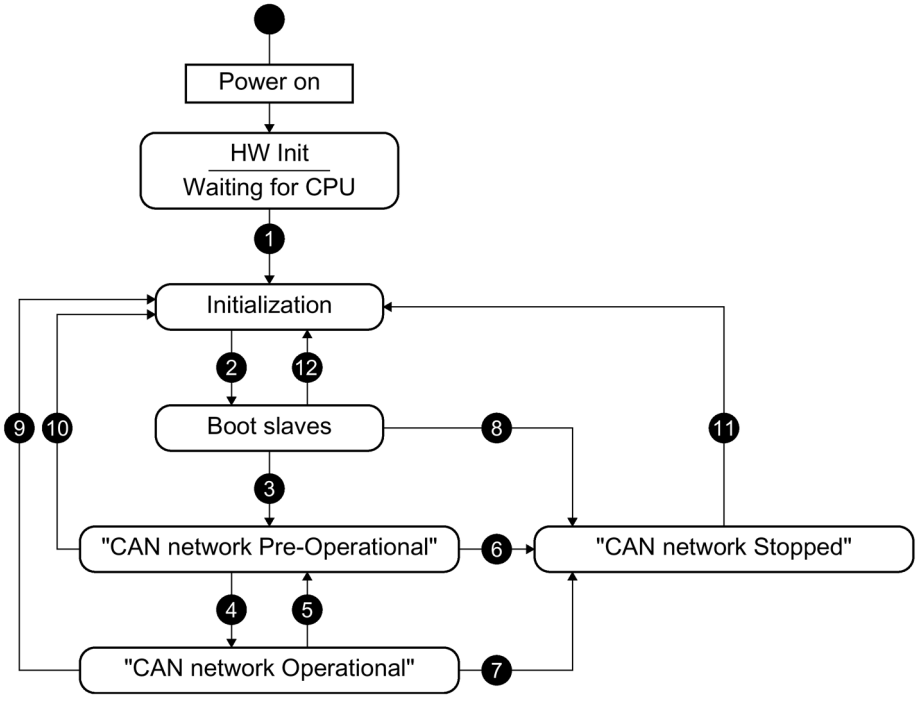


Figure 4-2 CANopen manager state model

Module states

The following table describes the states in the mode "CANopen manager".

Table 4- 5 "CANopen manager" module states

Module state	Meaning
HW Init / Waiting for SIMATIC S7 controller	The state of the module during booting and when the module is not yet fully configured by the SIMATIC S7 controller.
Initialization	Initialization state: Resetting the OD to the values configured in the HSP and resetting the communication. The state is not "visible" on the CANopen side because there is no CAN communication in this state. The NMT state of the slaves connected via CANopen does not change.
Boot slaves	The module performs the Boot Up procedure according to the CiA standard "CiA 302 Part 2" and configures the CANopen slaves via SDO. The module has the NMT state "Pre Operational". After booting, the NMT state of the slaves connected via CANopen is "Pre-Operational ".
CAN network Pre-Operational	The module as well as all slaves connected via CANopen are in the "Pre Operational" state.
CAN network Operational	The module as well as all slaves connected via CANopen are in the "Operational" state.
CAN network Stopped	The module as well as all slaves connected via CANopen are in the "Stopped" state.

State transitions

The following table describes the state transitions in "CANopen Manager" mode.

Table 4- 6 State transitions "CANopen manager":

Transition	Meaning / trigger
1	Takes place when the module is configured the first time by the SIMATIC S7 controller since booting.
2	Takes place automatically. A "Boot Up" message is sent because the module changes to the state NMT "Pre-Operational ".
3	This occurs when all configured "required" CANopen slaves are present and successfully booted.
4	Takes place when the control bit in the cyclic data from the SIMATIC S7 controller is set to "ON". During this transition, all fully booted CANopen slaves are set to "Operational" through NMT.
5	Takes place when the control bit in the cyclic data from the SIMATIC S7 controller is set to "OFF". During this transition, all CANopen slaves are set to "Pre-Operational" through NMT.
6, 7, 8	Triggered by Heartbeat / Node Guarding error of an appropriately configured "required" slave (NMT master error reaction is "Stop module and all nodes"). During this transition, all CANopen slaves are set to "Stopped" through NMT.
9, 10, 11, 12	Triggered by: • Heartbeat / Node Guarding error of an appropriately configured "required" slave, if (NMT master error reaction is "Restart module") • Reset bit contained in the cyclic data from the SIMATIC S7 controller • Reconfiguration by the SIMATIC S7 controller • Error during booting (except Heartbeat / Node Guarding error)

4.1.2.3 CANopen manager - Control and Status Information

Control information

The following control information is transmitted from the SIMATIC S7 controller to the ET 200SP CAN communication module:

Table 4- 7 Structure of control information in the PROFINET IO data (from the SIMATIC S7 controller to the module, 1 byte)

Bit		Value	Meaning
7 ... 3	Reserved	Must be "0"	--
2	Configuration bit	1	The manager configures nodes in the boot-up state. A successfully configured node can switch to the "Pre-Operational" state.
		0	The manager cannot configure nodes (the NMT status with the boot-up state has been received by the node).
1	Reset bit	1	The CANopen side is reset when the bit changes to "1". With the NMT command "Reset node", all nodes are reset to the initial values and the startup procedure is restarted. All OD entries are reset to initial values. The module acknowledges the receipt of the reset command by setting the reset acknowledgment bit in the data for the SIMATIC S7 controller to "1". The SIMATIC S7 controller can then set the reset bit to "0".
		0	When the reset has been performed and the reset bit is "0", the module sets the reset acknowledgment bit to "0". Only then can a reset be performed again.
0	Control bit	0	CANopen network is not in the state "Operational". If the current state of the CANopen network is "Operational", the CANopen network switches to the "CAN network Pre Operational" state. This means that the module and all nodes are set to "Pre Operational" by an NMT command.
		1	CANopen network is in the state "Operational". If the current state of the CANopen network is "Pre Operational", the CANopen network switches to the "CAN network Operational" state. This means that all nodes and the module are set to the "Operational" state with an NMT command. It is possible that not all slaves have finished booting at this state transition, if: • The reaction to the failure of a mandatory node is set to "Restart of triggering node" in TIA Portal or • Slaves are not configured as "mandatory". In this case, the slaves are not started with one NMT command "to all". An NMT command is sent instead to each fully booted slave individually. As soon as the other slaves are completely booted, they are also switched to "Operational" with an NMT command.

Status information

The following status information is transferred from the ET 200SP CAN communication module to the SIMATIC S7 controller:

Table 4- 8 Structure of status information in the PROFINET IO data (from the module to the SIMATIC S7 controller, 1 byte)

Bit		Values	Meaning	
7	NMT status feedback	0	One or more slaves are rebooted or the module state is not "CAN network Pre-Operational" or "CAN network Operational".	Bit indicates whether all slaves have already returned the NMT status matching the module.
		1	Module state is "CAN network Pre-Operational" or "CAN network Operational" and all nodes have the module state that is signaled in bit 5 ... 3.	--
6	Reset acknowledgment bit	0	The module is ready for reset.	In order to give feedback to the SIMATIC S7 controller about the reset of the CANopen side, a reset acknowledgment bit is used in the data cyclically transmitted from the module to the SIMATIC S7 controller. As soon as the command for resetting the CANopen side has been received, this bit is set to "1". Afterwards, the SIMATIC S7 controller must set the reset bit to "0". When the reset of the CANopen side is complete, the module sets the reset acknowledgment bit to "0"
		1	The module is currently performing or has performed a reset but the reset bit of the SIMATIC S7 controller is still set to "1".	--
5 ... 3	Module state	0	Boot slaves	--
		1	Status "Pre-Operational"	--
		2	Status "Operational"	--
		3	Status "Stopped"	--
		4	Waiting for boot-up enable	Currently not in use; if the status continuously signals bit 2, then it must be set to "1" in the control bit.
		5	State "Not configured"	Ready to receive the extended parameter assignment.
		6	Status "Bad parameterization"	Extended parameter assignment failed. Re-configuration required.
		7	Reserved	--

Bit		Val-ues	Meaning	
2	Group status of all data	0	OD entries with value that have not yet been updated by a corresponding PDO since the last reset.	--
		1	All OD entries have been updated at least once by a PDO or SDO since the last reset.	--
1 ... 0	State of the CAN controller	0	Off	When specified accordingly by the SIMATIC S7 controller
		1	Bus-Off	Error counter[1] has exceeded the specified second threshold. As a result, the module can no longer transmit/receive any packets. Note: "Bus Off" is also signaled when the module needs more than 2 seconds to synchronize with the CAN bus.
		2	Error Passive	Error counter[1] has exceeded the specified first threshold. The module transmits/receives packets nevertheless.
		3	Error Active	Error counter[1] is below a specified threshold. The module transmits/receives packets. This is the normal state. Everything is okay.

[1] The error counters customary in CAN and their associated thresholds are internal functions that are not visible to the user.

Note

For values made up of multiple bits, the first bit is the MSB and the last bit is the LSB.

Example: Bit 1 ... 0 = "2" means that bit 0 = "0" and bit 1 = "1".

See also

Configuration in the TIA Portal (Page 79)

Acyclic data exchange

The module provides several data records at the CANopen manager module, which can be accessed by the user program with the communication blocks RDREC (read data record) and WRREC (write data record). Such data records are not available at the configured CANopen slaves.

16 data records are offered for SDO reading and SDO writing.

The data record indexes used are 0x200 to 0x20F.

Note

A data record on a module is uniquely defined by an index that specifies the data record number. The index is a 32-bit integer. Different data records differ in their index.

Each data record represents an independent communication channel that can be used independently of the other channels. Via the corresponding data record, SDO communication is initiated with the communication blocks RDREC (read data record) and WRREC (write data record). The RDREC may have to be repeated multiple times if the module returns information that the SDO access has not yet finished (status code = "Busy").

A maximum of 128 bytes of data are transmitted in one SDO access.

Write data record

In this case, there is always first a RDREC on the same index to fetch the result and then a WRREC to transfer the desired SDO command to the module.

Table 4- 9 Structure of the configuration data of the data to be written

Byte	Read value for "SDO" (RDREC)	Write value for "SDO" (WRREC)
0	Command = 0x52 = 82 ('R')	Command = 0x57 = 87 ('W')
1	Node ID	
2 ... 3	OD index (Big Endian, i.e. MSB of value in byte 2 and LSB in byte 3)	
4	OD subindex	
5 ... 6	Number of bytes to be read by SDO (Big Endian), permitted: 1 ... 128	Number of bytes to be written by SDO (Big Endian)
7 ... <END>	<missing>	Bytes to be written

The WRREC access can be successful or return one of the following error codes.

The error codes standardized in PROFINET are used adapted for the transmission of SDOs.

Table 4- 10 Error codes

Error code (Hex)	Meaning
DF80B200	Module is incorrect (data record must be called on "CANopen manager" submodule)
DF80B000	Data record index is incorrect (not 0x200 ... 0x20F)
DF80B100	Written size of data record from SIMATIC S7 controller is too small
DF80B800	"Code" field not "R" or "W"
DF80B800	"NodeID" field invalid (node not configured)
DF80B800	"Number of bytes" field invalid (permitted: 1 ... 128)
DF80B500	No completely and validly configured PROFINET connection Application Relation is established
DF80A900	The module is not "CANopen manager".
DF80C300	Node not completely booted or Heartbeat / Node Guarding error after booting
DF80C200	Communication channel (data record index) is already "busy". The module first expects a RDREC before it permits a new WRREC.

If an error is detected, the SDO access will not be initiated.

Read data record

Data record size: 12 + <data length> bytes for SDO read access, 10 bytes for SDO write access

A WRREC always takes place first in order to transfer the desired SDO command to the module. Then, a RDREC takes place on the same index in order to retrieve the result.

The values of the SDO command, i.e. values of the previous WRREC, are returned in the first 5 bytes. This ensures on the SIMATIC S7 controller side that a data record is not used concurrently at several places in the SIMATIC S7 controller application.

Byte	Value for "Read SDO"	Value for "Write SDO"
0	Command = 0x52 = 82 ('R')	Command = 0x57 = 87 ('W')
1	Node ID	
2 ... 3	OD index (Big Endian, i.e. MSB of value in byte 2 and LSB in byte 3)	
4	OD subindex	
5	Status code	
6 ... 9	Additional error information code (Big Endian, i.e. MSB of value in byte 6 and LSB in byte 9)	
10 ... 11	Number of read bytes Big (Endian) (only available when status code = OK)	<missing>
12 ... <END>	Data (only available when status code = OK)	<missing>

The RDREC access itself can be successful or return one of the following error codes.

Table 4- 11 Error codes

Error code (Hex)	Meaning
DE80B200	Module is incorrect (data record must be called on the CANopen manager sub-module)
DE80B000	Data record index is incorrect (not 0x200 to 0x20F)
DE80B500	No completely and validly configured PROFINET connection (Application Relation) is established
DE80A900	The module is not CANopen manager
DE80C300	No previous successful WRREC
DE80B700	Read buffer too small

If RDREC returns no error, the user program can evaluate the data transmitted by the module. The "Status-Code" field can have the following values:

Table 4- 12 Values of the "Status code" field

Status code	Meaning
0	OK, SDO successfully completed.
1	SDO aborted or not started because a reset was triggered from PROFINET or the boot operation for the node was restarted.
2	SDO aborted or not yet started because once the node was successfully booted it had a Heartbeat or Node Guarding error.
3	SDO started but aborted by the module (cause according to CiA 301 in additional error information code).
4	SDO started but SDO abort received (received cause according to CiA 301 in additional error information code).
255	Busy = SDO access not complete yet. RDREC must be repeated.

Following an abort of the SDO access (triggered by the module itself or by the slave addressed by the SDO), the "Additional error information code" field is filled with the SDO abort code according to CiA 301.

When the RDREC is finished, the SDO access for the module is complete. The device is then ready for a new SDO command. An additional RDREC without a preceding successful WRREC returns an error code.

See also

Service Data Objects (SDO) (Page 27)

Data exchange between the SIMATIC S7 controller and the module (Page 29)

Changes compared to previous version (Page 14)

4.1.2.4 CANopen manager - Monitoring Functions

The module supports the "Heartbeat" and "Node Guarding" functions both as transmitter (producer) and as receiver (consumer). One of these two monitoring functions must be activated in the HSP.

Heartbeat

The module can be operated as a heartbeat producer and a heartbeat consumer simultaneously.

The required monitoring times (Transmit interval, Monitoring time) can be configured in the HSP.

Note

As of firmware version V1.1

The "Heartbeat" function can be disabled by setting the Transmit interval and the Monitoring time to "0".

Node Guarding

Node Guarding monitoring can be configured. You can set the needed parameters "Monitoring time" and "Repetition factor". The module can be operated as a transmitter and a receiver simultaneously.

Note

As of firmware version V1.1

The "Node Guarding" function can be disabled by setting the Repetition factor or the Monitoring time to "0".

Error responses

The module saves the current state of the slaves configured in the HSP. If a monitored CANopen slave fails or changes to an unexpected state, the module signals corresponding diagnostic information to the SIMATIC S7 controller. If the slave is configured as "required in the network", the defined error reaction is performed.

The following reactions can be set for the failure of a mandatory node:

Reaction	Description
Restart of triggering node	The CANopen slave is reset by NMT and the boot-up procedure including configuration is repeated for this slave. The state of the module and the state of the other CANopen slaves does not change.
Restart of the device (CM CAN) and all nodes	All CANopen slaves are reset via NMT and the boot-up procedure including configuration is repeated for all slaves.
Stop the device (CM CAN) and all nodes	The state of the module switches to "CAN network Stopped" and all CANopen slaves are put into "Stopped" state via NMT. The state can only be exited by a reset as follows: • By the reset bit in the cyclic data of the SIMATIC S7 controller • By re-configuration of the module by the SIMATIC S7 controller

Note

- CANopen slaves that are not configured by the user as "mandatory" do not trigger the set error response. When such a slave fails or signals an unexpected state, the slave is reset by NMT and the boot-up procedure including configuration is repeated for this slave. The module state and the state of the other CANopen slaves does not change.
- If the error response is "Restart of the device (CM CAN) and all nodes" or "Stop the device (CM CAN) and all nodes", the following applies: The triggered error response affects both the "mandatory" slaves and the slaves that are not configured as "mandatory" by the user.

As of firmware version V1.1:

- When the "Enable diagnostic alarms for not mandatory nodes" check box is selected, the master works exactly as the original device version 1.0. This means all error messages are sent to the diagnostics buffer and the ERROR LED lights up.
- When the "Enable diagnostic alarms for not mandatory nodes" check box is cleared, no error messages are sent to the diagnostics buffer and the ERROR LED remains off.

As of firmware version V1.2:

The following behavior supplements the error response behavior of V1.1:

- If the "Enable diagnostic interrupts for not mandatory nodes" check box and "Block Error passive alarm" check box are both selected at once and a node that you have selected as not mandatory triggers an Error Passive alarm, this alarm is not sent to the diagnostic buffer.

  If the "Enable diagnostic alarms for not mandatory nodes" check box is selected and the "Block Error passive alarm" check box is deselected and a passive error occurs in the device, this error is sent to the diagnostic buffer. The ERROR LED flashes red.
- If the "Enable diagnostic alarms for not mandatory nodes" check box is deselected, the "Error Block passive alarm" check box has no effect: No error messages are set to the diagnostic buffer, and the ERROR LED remains off for nodes that you have selected as not mandatory.

4.1.3 CANopen slave

4.1.3.1 Overview

When the module is operated in "CANopen slave" mode, the module does not take over the following functions in the CANopen network:

- Function of the NMT master
- Function of the Configuration Manager

If you operate the module in "CANopen slave" mode, most of the configuration settings are not defined in the TIA Portal, but instead during startup, e.g. configuration settings for Heartbeat or for the PDOs. During startup, the configuration settings are written to the OD of the module by the responsible CANopen manager using SDO commands.

Note

The module in the "CANopen slave" mode obtains a configuration from a "CANopen manager", whose remanent storage in the module is not supported.

4.1.3.2 CANopen slave - Object Dictionary

In "CANopen slave" mode, the module offers the following OD entries. The "Access type" column specifies whether the entry can be read (R) or written (W) via SDO.

OD entry	Access type	Meaning
1000h	R	Device type (always "0")
1001h	R	Current error state.
1003h	R	List of errors that occurred (max. 4 entries).
1005h	RW	COB-ID for SYNC message together with the setting of whether the module is a SYNC transmitter.
1006h	RW	Interval for SYNC messages.
1008h	R	Device name as character string. Default: "ET 200SP CM CAN Slave"; the entry can be edited, you can find more information in section Configuration in the TIA Portal (Page 79) under Setting communication parameters > Vendor device name
1009h	R	Hardware version as character string. Corresponds to the HW version that can also be read out via PROFINET by IM0.
100Ah	R	Software version as character string. Corresponds to the SW version that can also be read out via PROFINET by IM0, e.g. "V 1.0.0 [Build xxx]".
100Ch	RW	"Monitoring time" for the Node Guarding protocol
100Dh	RW	"Repetition factor" for the Node Guarding protocol
1014h	RW	"COB ID" for the EMCY messages transmitted by the module
1016h	RW	Field for "Receiver Heartbeat Time", i.e. the setting that the module monitors the CANopen nodes via a Heartbeat .

1017h	RW	"Sender Heartbeat Time". Interval at which the module transmits heartbeat messages.
1018h	R	Identity Object, identifies the device
1029h	RW	Determines the error response of the module. Only the entry for "Communication errors" is implemented.
1200h	R	SDO Server Channel
1400h-15FFh	RW	RX-PDO Communication parameters
1600h-17FFh	RW	RX-PDO Mapping parameters
1800h-19FFh	RW	TX-PDO Communication parameters
1A00h-1BFFh	RW	TX-PDO Mapping parameters
2000h-6FFFh	RW	Entries for the process data to be exchanged with the SIMATIC S7 controller. Entries for input data that comes from the SIMATIC S7 controller are in CANopen "Read Only".

4.1.3.3 CANopen slave - State Model

State model of the "CANopen slave" mode

The state model in "CANopen slave" mode is based on the NMT state model described in the CiA standard "CiA 301".

The following figure shows the states of the module in "CANopen slave" mode.

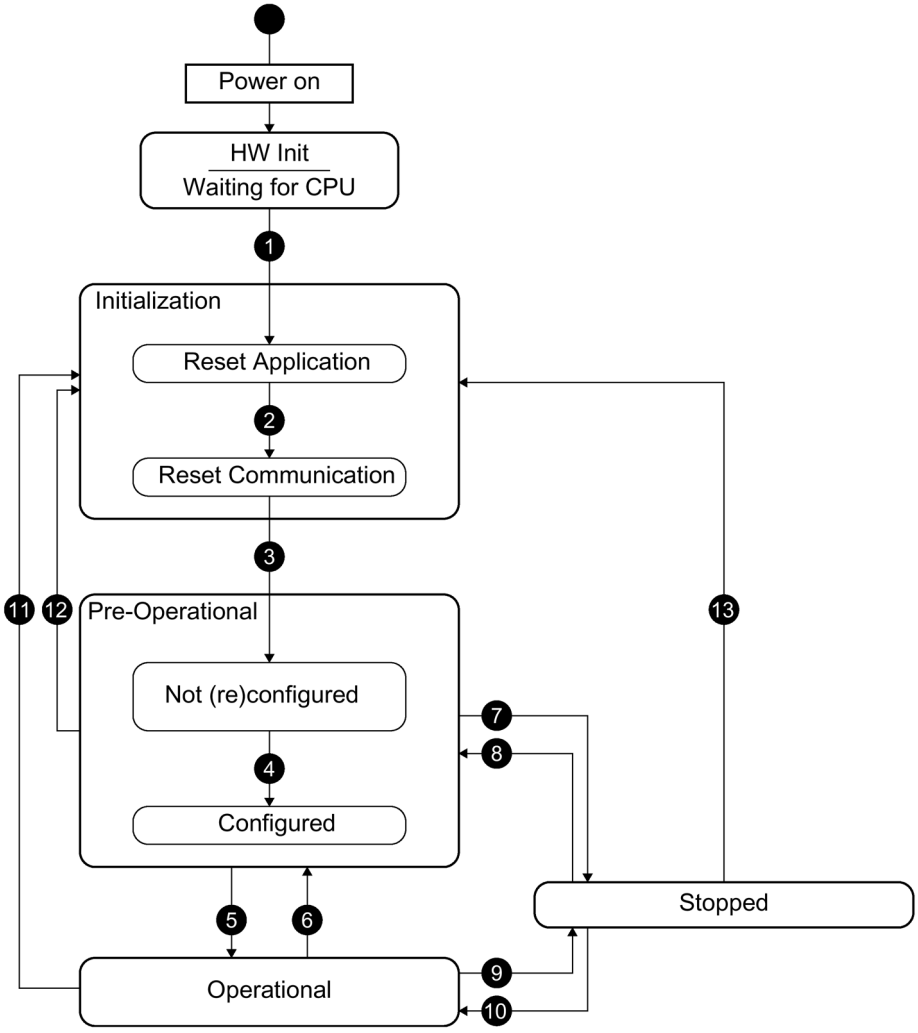


Figure 4-3 State model CANopen slave

Module states

The following table describes the states in "CANopen slave" mode.

Table 4- 13 "CANopen slave" module states

Module state	Meaning
HW Init / Waiting for SIMATIC S7 controller.	The state of the module during booting and when the module is not yet fully configured by the SIMATIC S7 controller. This is not a defined NMT state in CANopen standard.
Initialization	Initialization state. The entries in the "Communication parameters" are reset to factory settings. All parameters configured in the HSP are reset to the values set in the HSP. Communication is reset as well. CANopen distinguishes between the following sub-states: • "Reset Application": The OD entries not equal to 1xxxh are reset. For the module, this is the process data that is exchanged with the SIMATIC S7 controller. • "Reset Communication": The OD entries of the "Communication Profile Area" (OD 1xxxh) and thus the CANopen communication are reset.
Pre-Operational	In this state, the CANopen manager can configure the Slaves and synchronize them via SYNC. Except for PDOs, all other types of communication are active, e.g. SYNC, SDO, Heartbeat.
Operational	Normal operation, all communication types are active.
Stopped	Only Heartbeat / Node-Guarding are active but device can still be controlled via NMT, "applicative OFF state".

State transitions

The following table describes the state transitions in "CANopen slave" mode.

Table 4- 14 State transitions "CANopen slave":

Transition	Meaning / trigger
1	Takes place automatically if the module has been configured for the first time since boot-up of the SIMATIC S7 controller.
2	Takes place automatically.
3	Takes place automatically. A "Boot-up" message is transmitted.
4	No "real" transition, does not know "configured" / "not configured" state. But: During configuration by the CANopen manager, the communication behavior of the module changes, for example, when the manager has configured "Heartbeat".
5	Triggered by an NMT command from the CANopen manager.
6	Triggered by an NMT command from the CANopen manager, by a "Communication Error" if OD 1029h was configured accordingly, or by a change of the control bit transmitted by the SIMATIC S7 controller in the cyclic data to "OFF".
7	Triggered by an NMT command from the CANopen manager or by a "Communication Error" if OD 1029h was configured accordingly.
8	Triggered by an NMT command from the CANopen manager.
9	Triggered by an NMT command from the CANopen manager or by a "Communication Error" if OD 1029h was configured accordingly.
10	Triggered by an NMT command from the CANopen manager.
11, 12, 13	Can be triggered by: • An NMT command from the CANopen Manager (transition to sub-state "Reset Application" or "Reset Communication", depending on the command) • The Reset Bit from the SIMATIC S7 controller (transition to "Reset Application" sub-state) • Reconfiguration by the SIMATIC S7 controller (transition to "Reset Application" sub-state)

4.1.3.4 CANopen slave - Control and Status Information

Control information

The following control information is transmitted from the SIMATIC S7 controller to the ET 200SP CM CAN communication module:

Table 4- 15 Structure of control information in the PROFINET IO data (from the SIMATIC S7 controller to the module, 1 byte)

Bit		Value	Meaning
7 ... 2	Reserved	Must be "0"	--
1	Reset bit	1	The CANopen side is reset when the bit changes to "1". All OD entries are reset to initial values and the startup message is sent. In this case, the CANopen manager must configure and start the CM CAN again. The module acknowledges the receipt of the reset command by setting the reset acknowledgment bit in the data for the SIMATIC S7 controller to "1". The SIMATIC S7 controller can then set the reset bit to "0" .
		0	When the reset has been performed and the reset bit is "0", the module sets the reset acknowledgment bit to "0".
0	Control bit	0	The module must not be in "Operational" NMT state. If the current state is "Operational", the module switches to the "Pre-Operational" state. Commands from the NMT master to switch to the "Operational" state are ignored.
		1	The NMT master can set the module to the "Operational" state with the NMT command.

Status information

The following status information is transferred from the ET 200SP CM CAN to the SIMATIC S7-controller:

Table 4- 16 Structure of status information in the PROFINET IO data (from the module to the SIMATIC S7 controller, 1 byte)

Bit	Meaning	Possible values		Note
7	Reserved	--		--
6	Reset acknowledgment bit	0	The module is ready for reset.	In order to give feedback to the SIMATIC S7 controller about the reset of the CANopen side, the reset acknowledgment bit is used in the data cyclically transmitted from the module to the SIMATIC S7 controller. As soon as the command for resetting the CANopen side has been received, this bit is set to "1". Afterwards, the SIMATIC S7 controller must set the reset bit to "0". When the reset of the CANopen side is complete, the module sets the reset acknowledgment bit to "0".
		1	The module is currently performing or has performed a reset but the reset bit of the SIMATIC S7 controller is still set to "1".	--

Bit	Meaning	Possible values		Note
5 ... 3	Module state	0	Reserved	--
		1	Status "Pre-Operational"	--
		2	Status "Operational"	--
		3	Status "Stopped"	--
		4	Reserved	--
		5	Status "Not configured"	Ready to receive the extended parameter assignment
		6	Status "Bad parameterization"	The extended parameter assignment failed. Re-configuration required.
		7	Reserved	--
2	Group status of all data	0	OD entries whose values have not yet been updated by a corresponding PDO since the last reset.	--
		1	All OD entries have been updated at least once since the last reset via PDO or SDO.	--

Bit	Meaning	Possible values		Note
1 ... 0	State of the CAN controller	0	Off	When specified accordingly by the SIMATIC S7 controller.
		1	Bus-Off	Error counter[1] has exceeded the specified threshold. The module can then no longer transmit/receive packets. Note: "Bus Off" is also signaled when the module needs more than 2 seconds to synchronize with the CAN bus.
		2	Error Passive	Error counter[1] has reached the specified threshold. The module continues to transmit/receive packets.
		3	Error Active	Error counter[1] is below a specified threshold. The module transmits/receives packets. This is the normal state. Everything is okay.

[1] The error counters customary in CAN and their associated thresholds are internal functions that are not visible to the user.

Note

For values made up of multiple bits, the first bit is the MSB and the last bit is the LSB.

Example: Bit 1 ... 0 = "2" means that bit 0 = "0" and bit 1 = "1".

4.1.3.5 CANopen slave - Monitoring Functions

The device supports Heartbeat and Node Guarding as transmitter (producer) as well as receiver (consumer). The settings for these two functions are written to the OD of the module via SDO access by the "CANopen manager" responsible for the CANopen network.

When a monitored CANopen device fails, the module signals corresponding diagnostic information to the SIMATIC S7 controller. In addition, an EMCY message is transmitted on the CANopen side and the configured error response specified in the OD entry "ErrorBehaviour" at address 1029h, subindex 01h is executed.

Error responses

The following error responses are supported:

Subindex value	Description
00h	Change to the state "Pre-Operational" only when the current state is "Operational".
01h	Current state is retained, i.e. no response
02h	Change to the state "Stopped"

4.1.4 Response to error

Diagnostic information

When an error occurs, the corresponding error LED on the CAN side (ERR LED) is activated. The diagnostic message can be read out with the TIA Portal.

Information on the events that trigger a diagnostics alarm is available in the section Service Data Objects (SDO) (Page 27). Section Diagnostics alarms (Page 124) includes a detailed description of the error that triggers the diagnostic message and possible measures. This information is also stored in the TIA Portal .

As of firmware version V1.2

If the "Block Error passive alarm" check box is selected and a node triggers an Error Passive alarm, this alarm is not sent to the diagnostic buffer. The ERROR LED remains off.

If the "Block Error passive alarm" check box is deselected and a passive error occurs in the device, this error is sent to the diagnostic buffer. The ERROR LED flashes red.

Failure of communication via PDOs

When communication via PDOs fails, the values in the corresponding OD entries remain valid. Data contents remain valid until a new value is present.

Failure of a CAN node

The failure of a fieldbus node as well as "bus problems" on the CAN side are signaled to the SIMATIC S7 controller using diagnostic information.

Error Register

The Error Register entry (OD 1001h) indicates the current error state of the device.

The module uses the following bits from this entry:

Table 4- 17 Bits in the Error Register entry

Bit		Meaning
7	Manufacturer-specific error	Bit is set when the module is configured as "CANopen slave" and the control bit transmitted by the SIMATIC S7 controller in the cyclic data is set to "OFF".
4	Communication error	Bit is set when: • SIMATIC S7 controller connection fails. • The data from the SIMATIC S7 controller has the IO provider status "Bad".
0	Generic error	Bit is set in "any error situation", according to CiA standard "CiA 301", which means when a specific error bit is set as well as for errors that are not mapped to specific error bits.

EMCY messages

The EMCY messages to be sent when errors occur are permanently specified and cannot be configured by the user.

In "CANopen manager" mode, users can configure whether diagnostic information is sent to the SIMATIC S7 controller by the module when an error occurs.

Received EMCY messages are signaled to the IO controller in advanced PROFINET diagnostics. This function can be activated or deactivated through configuration. You can find more information in section Service Data Objects (SDO) (Page 27).

LED displays

The module signals its state in both "CANopen manager" mode and "CANopen slave" mode with the LEDs on the front of the enclosure. The meaning of the LEDs is described in section Status and error display (Page 118).

4.2 CAN transparent

4.2.1 Overview

In "CAN transparent" mode, all CANopen functions are disabled. Over the bus, the module only communicates with the CAN2.0A(B) protocol according to ISO standard 11898 (ISO 11898-2). The operating mode for the module is set using the associated HSP in the TIA Portal.

4.2.2 CAN transparent - state model

In "CAN transparent" mode, the module knows only the two states "OFF" and "ON".

Module state "OFF"

No CAN frames are transmitted or received in this state. The module does not participate in the CAN Bus communication and does not transmit or receive messages or any error frames.

The module assumes the "OFF" state when

- The module is not connected to a SIMATIC S7 controller.
- The data from the SIMATIC S7 controller has the IO Provider status "Bad".

Module state "ON"

In this state, the module participates in the CAN Bus communication. The module transmits and receives CAN frames except when the CAN controller is in Bus-Off state.

Dependency of the module state on the status of the SIMATIC S7 controller

Event	Reaction
The module has never been connected to the SIMATIC S7 controller or has not been configured since boot-up.	Module state is "OFF".
SIMATIC S7 controller connected / not yet completely configured	The module receives the necessary configuration data from the SIMATIC S7 controller. For the "receive message modules" the input data is preassigned according to the parameter assignment. Module state is "OFF".
SIMATIC S7 controller connected / completely configured / data from the SIMATIC S7 controller has the IO Provider Status "Bad".	Module state is "OFF".
Data from the SIMATIC S7 controller has the IO Provider Status "Good"	The module takes on its state based on the control bits ("OFF" or "ON"). The control bits are contained in the cyclic data that comes from the SIMATIC S7 controller.
SIMATIC S7 controller goes to STOP	Module state is "OFF".
SIMATIC S7 controller goes to RUN	The module takes on its state based on the control bits ("OFF" or "ON"). The control bits are contained in the cyclic data that comes from the SIMATIC S7 controller.
Termination of the SIMATIC S7 controller connection (after the module has already been configured once by the SIMATIC S7 controller)	Module state is "OFF".
Reconfiguration by the SIMATIC S7 controller	The module receives the necessary configuration data from the SIMATIC S7 controller. For the "receive message modules" the input data is preassigned according to the parameter assignment. Module state is "OFF".

4.2.3 CAN transparent - CAN messages

In "CAN transparent" mode, data can be exchanged using configurable or programmable messages, both of which can be used simultaneously.

Configurable CAN messages

The configurable CAN messages have a fixed message ID and a fixed length. The "Transmit message" and "Receive message" submodules are available in the TIA Portal for configuring. For more information, see section Configuring CAN transparent (Page 107).

You can use message IDs in the following formats:

- Standard format CAN 2.0A (11-bit message ID)
- Extended CAN format CAN 2.0B (29-bit message ID)

Note

Flow control is not provided. Intermediate values may be overwritten in the following cases:

- When data arrives via CAN messages faster than it can be sent via PROFINET.
- When data is changed by PROFINET faster than it can be sent via CAN.

Changes on the CAN bus may not be visible in this process.

The operating principle of the fixed CAN messages is based on a data image. This means a loss of intermediate values is acceptable.

Fixed messages can be transmitted as follows:

- When the data to be transmitted changes
- Time-controlled, i.e. as soon as the configured cycle time has elapsed

With this transmission mode, the user data of the CAN messages is transmitted between the SIMATIC S7 controller and the module in the cyclic IO image.

Programmable CAN messages (proxy)

In "CAN transparent" mode, the use of CAN messages for which the message ID and the length (max. 8 bytes) is only defined by the user program during runtime is possible.

With this transmission mode, protocol information in addition to the actual user data is transmitted between the SIMATIC S7 controller and the module in the cyclic IO image.

The message ID format (11 bits or 29 bits) is determined by bit 29 in the message ID. If this bit is set, use the message ID format with 29 bits.

The transmitting and receiving of CAN messages from the user program is possible using transmit and receive proxy modules. For additional information, see section Cyclical data exchange between the SIMATIC S7 controller and the module with programmed CAN messages (proxy) (Page 60)

4.2.4 CAN transparent - control and status information

CAN transparent

The following control information is transmitted from the SIMATIC S7 controller to the ET 200SP CM CAN communication module:

Table 4- 18 Control information in the PROFINET IO data (from the SIMATIC S7 controller to the module, 1 byte)

Bit	Meaning	Possible values		Note
7 ... 1	Reserved	0	Values must be "0"	---
0	Control bit	0	No CAN communication	The CAN controller should be "OFF".
		1	CAN communication active	The CAN controller should be "ON".

The following status information is transferred from the ET 200SP CM CAN communication module to the SIMATIC S7 controller:

Table 4- 19 Status information in the PROFINET IO data (from the module to the SIMATIC S7 controller, 1 byte)

Bit	Meaning	Possible values		Note
7 ... 6	Reserved	--	--	--
5 ... 3	Module state	0 ... 4	Reserved	--
		5	Status "Not configured"	Ready to receive the extended parameter assignment.
		6	Status "Bad parameterization"	Extended parameter assignment failed. Re-configuration required.
		7	Reserved	--
2	Group status of all data	0	Not all configured receive messages have been received at least once since the last transition to "ON".	--
		1	All configured receive messages have been received at least once since the last transition to "ON".	--

Bit	Meaning	Possible values		Note
1 ... 0	State of the CAN controller	0	Off	As specified by the SIMATIC S7 controller.
		1	Bus-off	Error counter[1)] has exceeded the specified threshold or the module needs more than 2 seconds to synchronize with the CAN Bus. The module no longer transmits/receives any packets.
		2	Error Passive	Error counter[1)] has reached the specified threshold. The module continues to transmit/receive packets.
		3	Error Active	Error counter[1)] is below a specified threshold. This is the normal state. Everything is okay.

1) The error counters customary in CAN and their associated thresholds are internal functions that are not visible to the user.

Note

For values made up of multiple bits, the first bit is the MSB and the last bit is the LSB.

Example: Bit 1 ... 0 = 2 means: Bit 0 = "0" and Bit 1 = "1".

4.2.5 Cyclical data exchange between the SIMATIC S7 controller and the module with configured CAN messages

CAN messages with output data (transmit message)

The "transmit messages" can be configured in the TIA Portal. Each of these messages has its own message ID, which is configured in the TIA Portal, and its own data, which is transmitted cyclically to the module in the output process image of the SIMATIC S7 controller.

The module sends a CAN message with the message ID and the current data:

- When the data changes
- When a CAN RTR frame (Remote Transmit Request) with the matching message ID has been received.
- During transition of the module to "ON" state
- Cyclically, if configured accordingly.

The triggers are independent of each other. That is, the configured transmit cycle is retained even if a message based on a data change or an RTR has been sent in the interim.

If configured, the module changes the order of specific data bytes for fixed CAN messages prior to sending to compensate for the different "endianness" of PROFINET and CAN. PROFINET uses "Big Endian", CAN usually uses "Little Endian". For programmed CAN messages, however, the order of the data bytes is not changed.

CAN messages with input data (receive messages)

The "receive messages" can be configured in the TIA Portal. Each of these messages has its own message ID, which is configured in the TIA Portal, and its own data, which is transmitted cyclically from the module to the SIMATIC S7 controller as part of the input process image.

The values of the input data are assigned during configuration of the module by the SIMATIC S7 controller according to the configuration.

If a CAN frame with the configured message ID has been received, the order of the data bytes is first changed, if necessary, to correct the "endianness" before the data is transferred to the input data for the SIMATIC S7 controller.

A setting can be made during configuration so that the module transmits an RTR frame with the configured message ID on the CAN bus during transition to module state "ON". This frame requests that the associated counterpart sends the corresponding data message.

As soon as a module changes to "ON" state, it monitors the receipt of CAN messages of all configured message IDs and transmits information about them, compressed into one status bit, to the SIMATIC S7 controller. As long as there is at least one configured message ID from which a CAN message has not yet been received, a zero is transmitted to the SIMATIC S7 controller in the status bit.

4.2.6 Cyclical data exchange between the SIMATIC S7 controller and the module with programmed CAN messages (proxy)

"Transmit proxies" and "Receive proxies" can be set up in the TIA Portal.

These proxies serve as a "communication channel" that enables the user program to send and receive any CAN messages not configured beforehand.

The use of multiple transmit / receive proxies facilitates programming of the user program because they are independent of the communication channels.

Example of a transmit proxy:

Two code locations want to transmit CAN messages. When you set up two transmit proxies, these messages do not have to be synchronized or matched. Each code location uses its own proxy.

Example of a receive proxy:

Since each receive proxy has its own "receive filter list", you can use several proxies to pre-filter messages according to the CAN ID, e.g. a receive proxy only for "commands" and a receive proxy for "status messages".

Max. number of send/receive proxies

The boundary condition for defining the proxy messages is the size of the IO data.

A transmit proxy message (Tx) has the size of 1 input byte and 13 output bytes.

A receive proxy message (Rx) has the size of 13 input bytes and 1 output byte.

The maximum number of free IO data for the CM CAN module is 255 (256 - 1) bytes per direction.

Thus you can configure the following proxies:

- 19 Tx (or Rx): 255 / 13 = 19 (8 bytes left)
- And 8 Rx (or Tx): the rest of 8 bytes

The maximum number of proxies in a mixed configuration of Tx and Rx proxies is 18 (255 / 14) in one direction. This configuration contains a total of 36 proxies:

- 18 Tx: 255 / 14
- And 18 Rx: 255 / 14

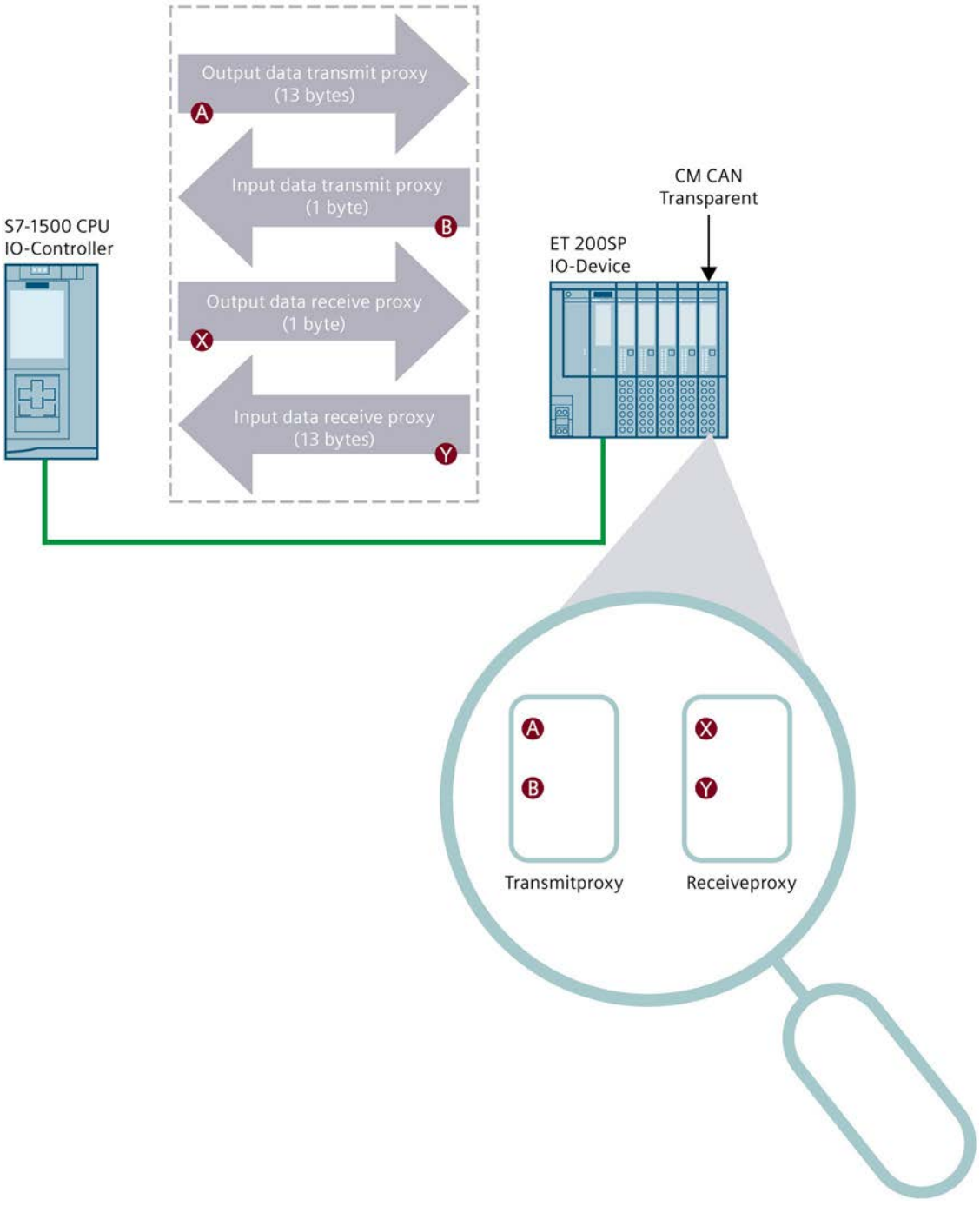


Figure 4-4 Cyclic data exchange

Transmit proxy

CAN messages can be transmitted from the user program with the transmit proxy. The transmit proxy contains cyclic PROFINET input and output data.

In the output data from the SIMATIC S7 controller to the module, the following information is transmitted for the frame to be sent to the module:

- Message ID
- Length
- User data
- Flag, whether RTR or data CAN message

In the input data from the module to the SIMATIC S7 controller, the module acknowledges the entry of the message in the CAN transmit buffer in the module. This does not always mean, however, that the message has been transmitted on the CAN bus.

Note

The order of the data bytes in the CAN messages to be sent is not changed in the module. Any necessary adaptation to compensate for "endianness" must be made in the user program.

Note

For values made up of multiple bits, the first bit is the MSB and the last bit is the LSB.

Example: Byte.Bit 3.0 ... 4.6 = 2 means: Byte 3 Bit 0 = "0", Byte 4 Bit 7 = "1" and Byte 4 Bit 6 = "0"

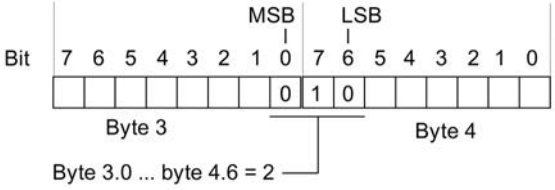


Table 4- 20 Structure of the output data for send proxy (from the SIMATIC S7 controller to the module, 13 bytes)

Byte.Bit	Meaning
0.7 ... 0.5	Reserved, must be "0"
0.4 ... 0.1	**Number of data bytes** Permitted values are 0 to 8. Values 9 to 15 cause the CAN message to be discarded and the "Last data faulty bit" to be set.
0.0	**Send** When the bit changes to 1, the other data is applied and a corresponding CAN message is entered in the transmit buffer. The module acknowledges the receipt of the transmit command by setting the transmit acknowledgment bit in the data for the SIMATIC S7 controller to "1". The SIMATIC S7 controller can then set the transmit bit to "0" and must wait until the module has also set the transmit acknowledgment bit to "0". Only then can a new transmit command be issued. Note: If the control bit is "0", the CAN messages are not entered in the transmit buffer but are discarded instead.
1.7	Reserved, must be "0"
1.6	**RTR-Flag** 0 = Normal data frame transmission 1 = RTR frame transmission
1.5	**11/29-bit message ID** 0 = 11 Message ID bit 1 = 29 Message ID bit
1.4 ... 4.0	**Message ID (in Big Endian)** For 11-bit message IDs, byte 3 bit 2 is the most significant bit and byte 4 bit 0 the least significant bit. The remaining bits are ignored. For 29 bit message IDs, byte 1 bit 4 is the most significant bit and byte 4 bit 0 the least significant bit.
5-12	**Data bytes** In the order in which they are going to be sent

Table 4- 21 Structure of the input data for send proxy (from the SIMATIC S7 controller to the module, 1 bytes)

Byte.Bit	Meaning
0.7 ... 0.3	Reserved, must not be evaluated
0.2	**Transmit buffer overflow** Set to "1" by the module when a CAN message could not be entered in the transmit buffer because it is full. A corresponding diagnostic information is also sent in this case. The bit is reset to "0" when the transmit acknowledgment bit is set to "0".
0.1	**Last data faulty** Set to "1" by the module when "Number of data bytes" > 8. The bit is reset to "0" when the transmit acknowledgment bit is set to "0".
0.0	**Transmit acknowledgment** Set to "1" by the module as soon as the transmit bit is "1". When the transmit bit changes to "0", the module sets the transmit acknowledgment bit to "0" as soon as it has transmitted the message to be sent to the transmit buffer of the CAN controller.

Receive proxy

CAN messages can be received from the user program with the receive proxy. The module manages a list of message IDs ("filters") on the module side that it receives on the CAN bus.

The list can be preassigned during configuration and changed at runtime from the user program by writing a corresponding data record.

If a corresponding CAN message is received, it is entered in a receive buffer. The message is transmitted from this buffer to the SIMATIC S7 controller in the cyclic input data. The size of the receive buffer is specified in the configuration.

Using the output data (from the SIMATIC S7 controller to the module), the user program can:

- Acknowledge the successful transmission of a CAN message from the module to the SIMATIC S7 controller
- Instruct the module to ignore the receive filter. Then all CAN messages are copied into the receive buffer.
- Instruct the module to delete all CAN messages in the receive buffer.

In the input data from the module to the SIMATIC S7 controller , the module transmits received CAN messages and information on whether there was a receive buffer overflow. Such an overflow occurs when too many CAN messages with matching message IDs were received before they could be transmitted to the SIMATIC S7 controller.

Note

The order of the data bytes in the received CAN messages is not changed in the module. Any necessary adaptation to compensate for "endianness" must be made in the user program.

Note

For values made up of multiple bits, the first bit is the MSB and the last bit is the LSB.

Example: Byte.Bit 3.0 ... 4.6 = 2 means that Byte 3 Bit 0 = "0", Byte 4 Bit 7 = "1" and Byte 4 Bit 6 = "0".

Table 4- 22 Structure of the output data for receive proxy (from the SIMATIC S7 controller to the module, 1 byte)

Byte.Bit	Meaning
0.7	**Receive buffer** 0 = No action. 1 = All CAN messages in the receive buffer are deleted.
0.6	**Receive filter** 0 = Only received CAN messages that match the receive filter are entered in the receive buffer and forwarded to the SIMATIC S7 controller later. 1 = All received CAN messages are entered into the receive buffer and later forwarded to the SIMATIC S7 controller.
0.5 ... 0.1	Reserved, must be 0
0.0	**Receive enable** If the bit has the value 1 and the module has a CAN message in the receive buffer of the receive proxy, the module enters the message in the data for the SIMATIC S7 controller and sets the "Message exists" bit to "1". The SIMATIC S7 controller must then set the receive enable bit to "0" and must wait until the module has also set the "Message exists" bit to "0". Only then may the SIMATIC S7 controller set the receive enable bit to "1" again.

Table 4- 23 Structure of the input data for receive proxy (from the SIMATIC S7 controller to the module, 13 bytes)

Byte.Bit	Meaning
0.7	**Message exists** 0 = No message in the receive buffer 1 = Message(s) in the receive buffer Bit is always "1" when the transfer bit = 1 because the message stays "in the buffer" until the transfer is "completed".
0.6	**Receive filter** 0 = A valid receive filter is set. 1 = No valid receive filter was set. If the receive filter bit is set to "0", no messages at all are forwarded to the SIMATIC S7 controller.
0.5	**Receive buffer overflow** 1 = The receive buffer is full. At least one message was discarded. Note: May only be evaluated when the transfer bit is "1".
0.4 ... 0.1	**Number of data bytes in the received CAN message** Value range is 0 to 8. Note: May only be evaluated when the transfer bit is "1".
0.0	**Transfer** If the bit is "1", a received message is currently being transferred. The SIMATIC S7 controller should then set the receive enable bit to "0".
1.7	Reserved, must be "0"
1.6	**RTR-Flag of the received CAN message** 0 = Normal data message 1 = RTR message Note: May only be evaluated when the transfer bit is "1".

Byte.Bit	Meaning
1.5	**11/29-bit message ID of the received CAN message** 0 = 11 Message ID bit 1 = 29 Message ID bit Note: May only be evaluated when the transfer bit is "1".
1.4 ... 4.0	**Message ID of the received CAN message (in Big Endian)** For 11-bit message IDs, byte 3 bit 2 is the most significant bit and byte 4 bit 0 the least significant bit (the other bits are ignored) For 29 bit message IDs, byte 1 bit 4 is the most significant bit and byte 4 bit 0 the least significant bit. Note: May only be evaluated when the transfer bit is "1".
5-12	**Data bytes of the received CAN message** In the order in which they were received. Only the corresponding "Number of data bytes in the received CAN message" may be evaluated. Note: May only be evaluated when the transfer bit is "1".

Note on the use of a receive proxy

Representation of bits

1. "Receive enable" is a bit that can be used to copy data to the SIMATIC S7 controller (you can control this bit).
2. "Message exists" is a bit that signals that there is a message in the input buffer.
3. "Transfer" signals that the current position from the input buffer is actively transferred to the SIMATIC S7 controller.

Procedure

1. Set "Receive enable" to "1" and wait until "Message exists" is also "1".
2. When this happens, the first message is copied to the SIMATIC S7 controller and "Transfer" is set to "1".
3. Set "Receive enable" to "0" and wait until "Transfer" is set to "0".
4. Repeat the operation starting at step 1.

Changing the receive filters during runtime

By writing a record set (with WRREC) with the index 0x210 into the module, the receive filters may be changed by the user program during runtime.

The module makes the corresponding data record available on the respective receive proxy.

The data record consists of:

- 1 byte long index of receive proxy

- n*8 byte long filters

The data record overwrites all filters of the receive proxy. After the first index byte, filters from 1 to 16 are sent to the data record.

Each individual filter consists of a 32-bit value "Criterion" and a 32-bit value "Mask".

If a bit in "Mask" is "0", the received CAN messages are **not** checked against the corresponding bit in the "Criterion".

To ensure that a bit set in "Criterion" acts as a filter, the corresponding bit in the "Mask" must be set to "1".

Structure of the data to be written with WRREC

Data record size: 1+8*n bytes; n = Number of filters = 1 to 16.

```
Index of the receive proxy (8-bit)
Criterion filter 1 (32-bit, Big Endian)
Mask filter 1 (32-bit, Big Endian)
Criterion filter 2 (32-bit, Big Endian)
Mask filter 2 (32-bit, Big Endian)
...
Criterion filter 16 (32-bit, Big Endian)
Mask filter 16 (32-bit, Big Endian)
```

Note

If the number of written bytes is not a multiple of "8", the extra bytes are ignored. A maximum of 16 filters with 4 bytes each for Criterion and Mask are permitted. If the number of written bytes is > 128, write access (WRREC) is rejected with error code 0xDF80B100 and the filters remain unchanged.

Coding of the bits in "Criterion" and "Mask"

The bits in "Criterion" and "Mask" are coded as follows.

Note

For values made up of multiple bits, the first bit is the MSB and the last bit is the LSB.

Example: Bit 28 ... 27 = "2" means that bit 28 = "1" and bit 27 = "0"

Bit	Meaning
31	**Filter valid (only with "Criterion")** 0 = Filter is valid 1 = Filter is invalid and is ignored by the module For "Mask" the bit is reserved and must always be 0.
30	**RTR** 0 = Only data messages are entered in the receive buffer 1 = Only RTR messages are entered in the receive buffer
29	**11/29-bit message ID** 0 = Only CAN messages with 11-bit message ID are entered in the receive buffer. 1 = Only CAN messages with 29-bit message ID are entered in the receive buffer.
28 ... 0	**Message ID** Only CAN messages with the message ID are entered in the receive buffer. Note: For received CAN messages with 11-bit message ID, bits 28 ...11 in the message are accepted as "0". If one of the bits 28 ... 11 is set in "Criterion" as well as "Filter", the filter will not match any of the received 11-bit messages.

Example:

If a filter is to be set so that only 29-bit data messages with message ID "0x?????9" are received, the following conditions must be met:

- RTR for "Criterion" = 0 and for "Mask" = 1
- 29/11-bit message ID for "Criterion" = 1 and for "Mask" = 1
- Message ID for "Criterion" = 0x9 and for "Mask" = 0xF

This means "Criterion" = 0x20000009 and "Mask" = 0x6000000F

The bytes to be written as data record are therefore "0x20 0x00 0x00 0x09 0x60 0x00 0x00 0x0F"

Note

If less than 8 bytes are written or if bit 31 = "1" is true for "Criterion" for all filters, the receive proxy has "no valid receive filter". This is reported to the SIMATIC S7 controller in the corresponding bit in the input data.

4.2.7 CAN transparent - Reaction to fault

Configured CAN messages

If communication fails when communicating via configured CAN messages, the last transmitted values continue to be valid. Data contents remain valid until a new value is present.

If the length of a received CAN package does not match the configured length, the data are not applied and a corresponding diagnostics alarm is sent to the SIMATIC S7 controller.

Diagnostic information

The module sends diagnostics to the SIMATIC S7 controller.

When an error occurs, the corresponding error LED on the CAN side (ERR-LED) is triggered. The diagnostic message can be read out with the TIA Portal.

Information on the events that trigger a diagnostics alarm, the cause of the error and a possible solution is available in the section Diagnostics alarms (Page 124). This information is also stored in the TIA Portal.

As of firmware version V1.2

If the "Block Error passive alarm" check box is selected and a node triggers an Error Passive alarm, this alarm is not sent to the diagnostic buffer. The ERROR LED remains off.

If the "Block Error passive alarm" check box is deselected and a passive error occurs in the device, this error is sent to the diagnostic buffer. The ERROR LED flashes red.

LED displays

In "CAN transparent" mode, the module signals its status via the fieldbus LEDs on the front of the enclosure. The meaning of the LEDs is described in section Status and error display (Page 118).

Connecting

5

5.1 Wiring

Requirements

For the connections, you need a BaseUnit:

- Type A0, BaseUnit light-colored
- Type A0, BaseUnit dark-colored

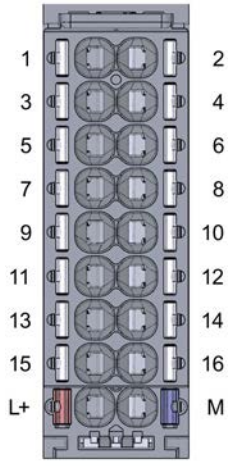
The BaseUnit is not included in the scope of delivery of the module. Order the BaseUnit separately.

You can find more information on connecting the module in the system manual ET 200SP Distributed I/O System (https://support.industry.siemens.com/cs/ww/en/view/58649293).

Terminal assignment on the BaseUnit

The designations of the individual terminals in the following table correspond to the labelling on the front of the module.

Table 5- 1 CAN connection

Terminal assignment of the BaseUnits of the module	Terminal	Designation	Terminal function	Meaning
	1	HI	CAN_H	CAN_H bus cable (dominant high)
	2	LO	CAN_L	CAN_L bus cable (dominant low)
	3	GND	CAN_GND	CAN Ground
	4	TER	Terminating	Terminating resistor 120 Ω (install jumper from pin 2 to pin 4 for use)
	5-16	RES	Reserved	Reserve remains unconnected for future expansions
Front view				

NOTICE
The electromagnetic compatibility (EMC) is only maintained with shielded CAN cables.

Address space

6

6.1 Address space

Address space

The input and output addresses of the communication module use an address space of 1-256 bytes. The input and output addresses are automatically assigned for each communication module when the device configuration is specified in TIA Portal.

Hardware identifier (not freely configurable)

The hardware identifier (HW identifier) is automatically assigned when the device configuration is defined in the TIA Portal.

The HW identifier is included in diagnostics alarms so that the module involved can be located. The SIMATIC S7 controller requires the HW identifier in communication instructions in order to identify the module involved.

See also

Interface module (https://support.industry.siemens.com/cs/ww/en/ps/14034/man)

CPU (https://support.industry.siemens.com/cs/ww/en/ps/13889/man)

Parameter assignment

7

7.1 Overview

The following section contains compact descriptions of the configuration steps for the 3 operating modes of the CAN communication module:

- CANopen manager
- CANopen slave
- CAN transparent

The descriptions are not based on a concrete example configuration. They are intended to illustrate the basic method for configuring the module and provide a quick introduction to configuring the module.

Firmware version V1.2 supports new functions, which are available as of the following TIA Portal versions:

- TIA Portal V17 Update 6
- TIA Portal V18 Update 1
- TIA Portal V19

Firmware version V1.1 supports new functions that are made available with the following Hardware Support Package (HSP): HSP_V16_0310_003_ET200SP_CM_CAN_1.0 (HSP_V16_0310_002_ET200SP_CM_CAN_1.0.isp16)

In order to work with firmware version V1.1, you must use HSP_V16_0310_003_ET200SP_CM_CAN_1.0. You can configure a module with firmware version V1.0 using HSP 0310 ET200SP CM CAN V2.0.

HSP_V16_0310_003_ET200SP_CM_CAN_1.0 is available for TIA Portal version V16 only.

You must use the latest HSP_V16_0310_003_ET200SP_CM_CAN_1.0.

The new functions mainly refer to using the CANopen Manager mode.

Note

TIA Portal Openness

The ET 200SP CAN communications module only supports mandatory attributes in TIA Portal Openness.

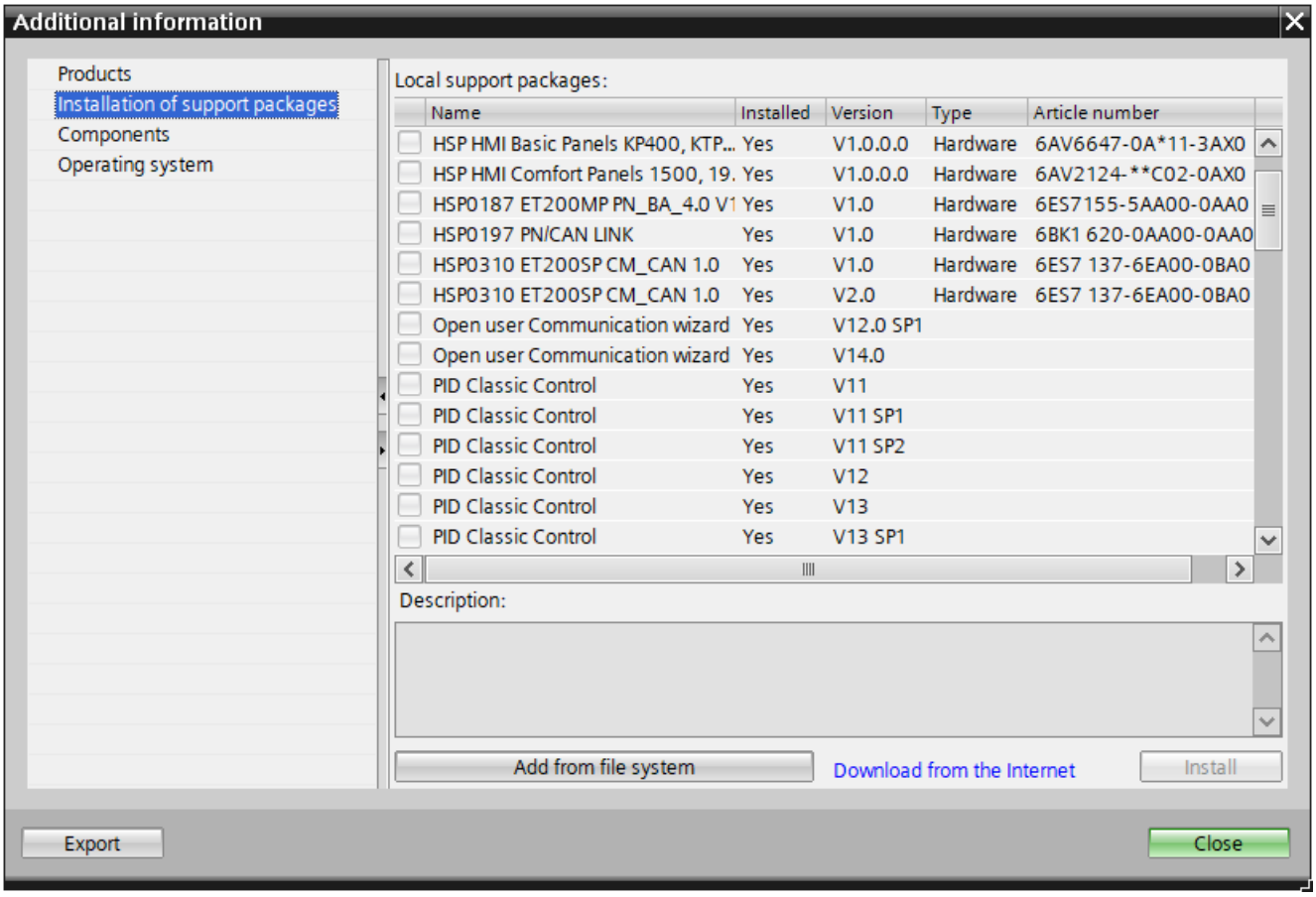


Figure 7-1 Selecting an HSP

When selecting the module, you can choose the desired firmware version in the HW catalog.

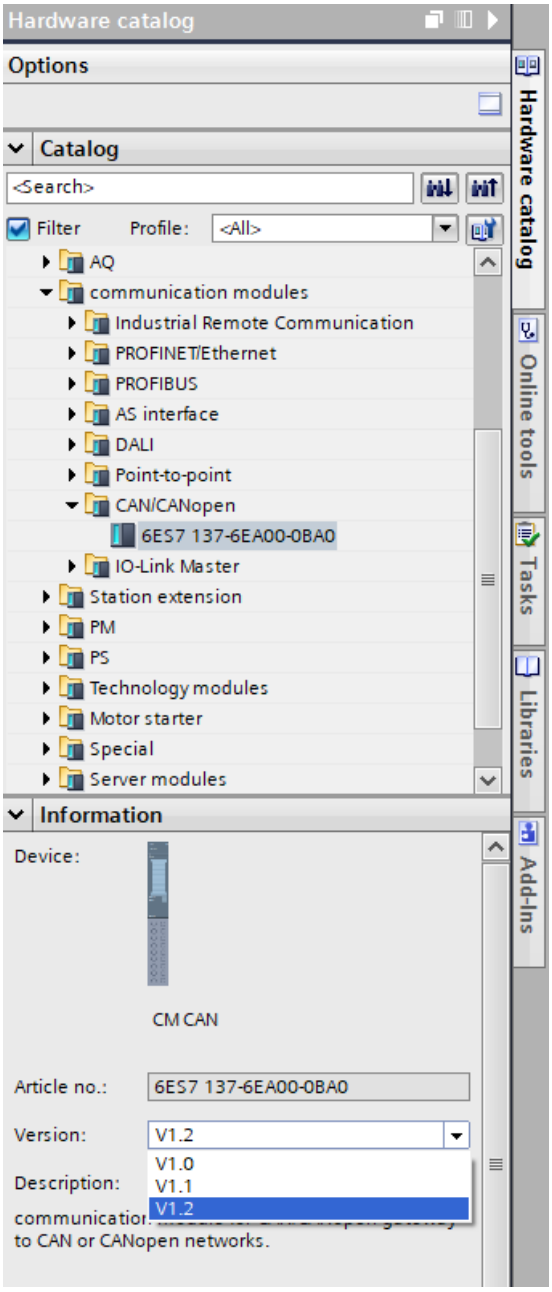


Figure 7-2 Selecting a firmware version in the HW catalog

Or you can select the "Change firmware version" option for a module that has already been configured.

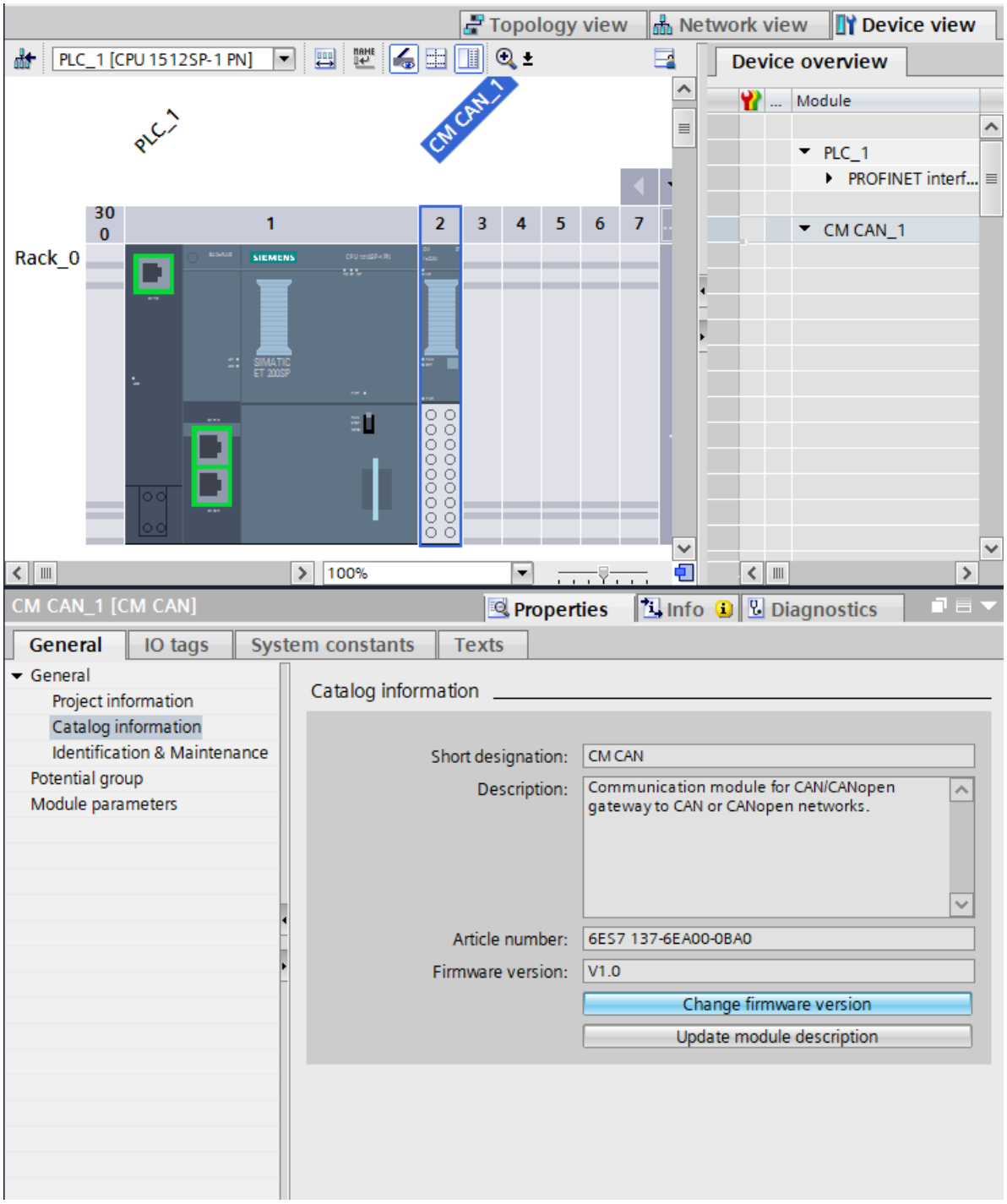


Figure 7-3 Changing the firmware version

Version dependencies of the firmware update/firmware downgrade

A firmware update is possible in the following cases:

- Update of firmware version V1.0 to higher versions (V1.1, V1.2)
- Update of firmware version V1.1 to version V1.2

A firmware downgrade to an earlier version is not possible.

If a firmware update/firmware downgrade is not possible, the "Change firmware version" option is disabled automatically.

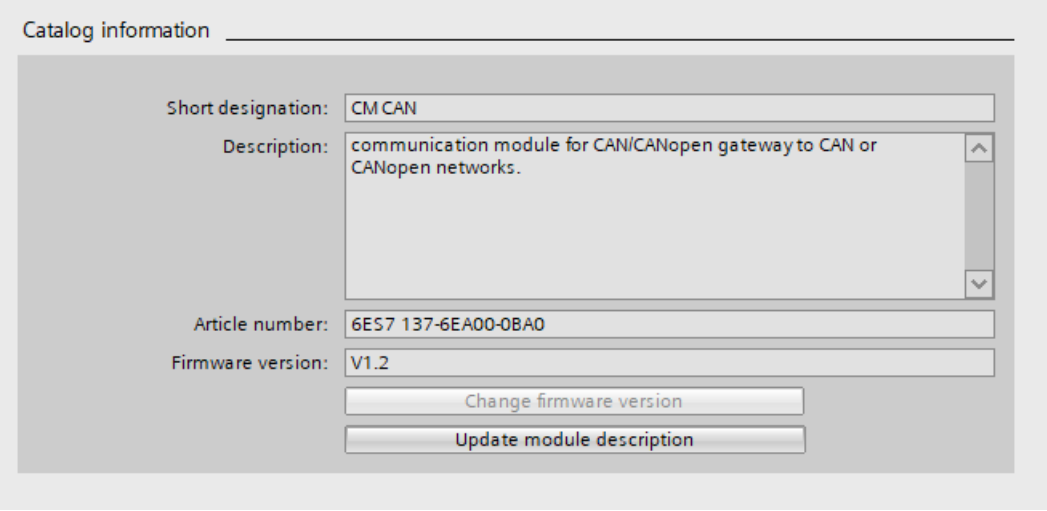


Figure 7-4 "Change firmware version" option disabled

7.2 Configuring CANopen manager

7.2.1 Overview

The module is configured in the TIA Portal. For "CANopen manager" operating mode, configuring consists mainly of the following steps:

1. Drag the communication module CAN from the hardware catalog onto an IM or SIMATIC ET 200SP.
2. Define the mode of the module ("CANopen manager" here).
3. Set the bus-specific parameters, e.g. node ID and transmission rate.
4. Integrate additional CAN nodes; import the EDS files or define the Object Dictionary (OD) manually.
5. Define transmit and receive PDOs of the node.
6. Create OD entries for the transmit and receive data to be exchanged in the Manager module.
7. Define corresponding receive and transmit PDOs in the Manager module.
8. If required, make further settings, e.g. Heartbeat, Node Guarding and SYNC.
9. Check and compile the configuration.

7.2.2 Configuration in the TIA Portal

TIA Portal: Devices & networks

Proceed as follows:

1. Select the CAN/CANopen module with its specific article number from the HW catalog. ①
2. Select the firmware version (V1.0, V1.1 or V1.2) of your module. ② This step is only possible if
   - you have installed HSP_V16_0310_003_ET200SP_CM_CAN_1.0, or
   - you are using the corresponding TIA Portal version (into which the device has been integrated)
3. Drag the communication module to a free slot in the ET 200SP system. ③ The module is a part of the modular IO system connected to the fieldbus.

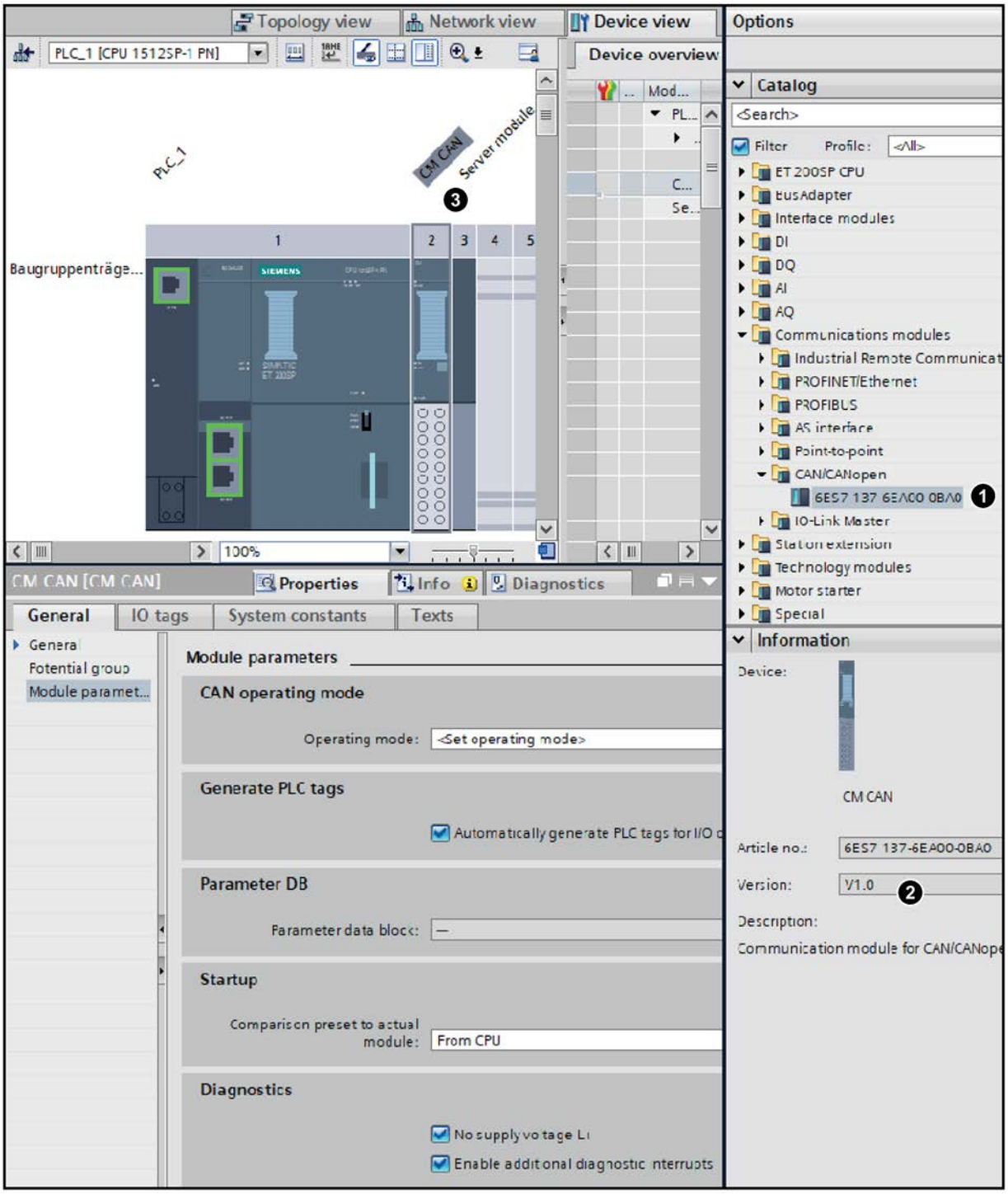


Figure 7-5 TIA Portal Devices&Networks

Selecting CAN operating mode for the module

The CAN operating mode is selected using the "Set operating mode" drop-down list, which contains the following selection options:

- CANopen manager
- CANopen slave
- CAN transparent

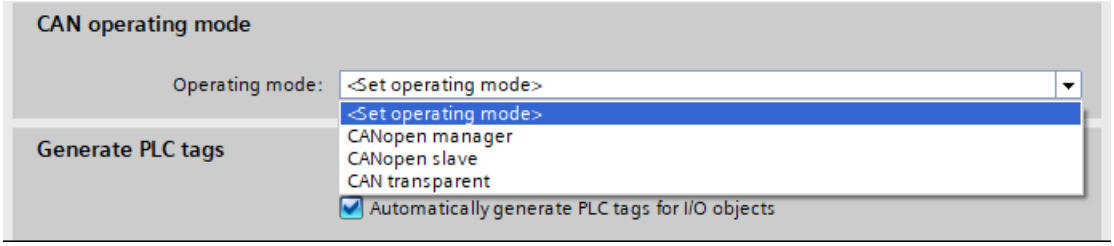


Figure 7-6 Selecting an operating mode

Select the "CANopen manager" operating mode.
Set the appropriate parameters for the selected operating mode. All settings at the CANopen manager are done in the "CANopen manager Menu" of the module.

Setting the communication parameters

Set the bus-specific parameters, for example, node ID and transmission rate. ①

As of firmware version 1.1, you can set the parameters "max. SDO Timeout" and "max. BootUp Timeout" ②.

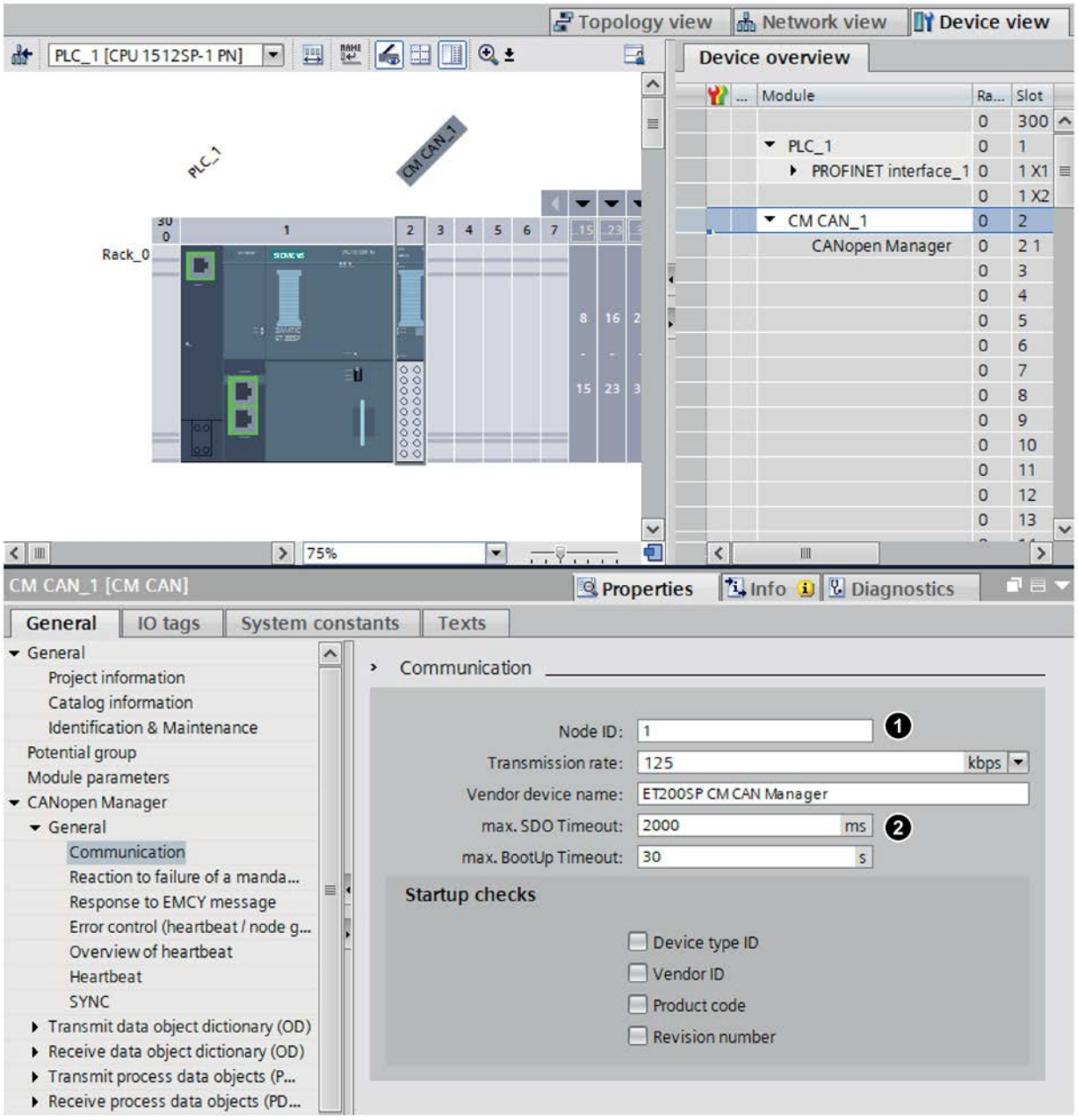


Figure 7-7 Setting the communication parameters

Adding CAN nodes

CAN nodes are placed in the "CANopen nodes" (Node submenu) under the CANopen manager in the CM CAN module.

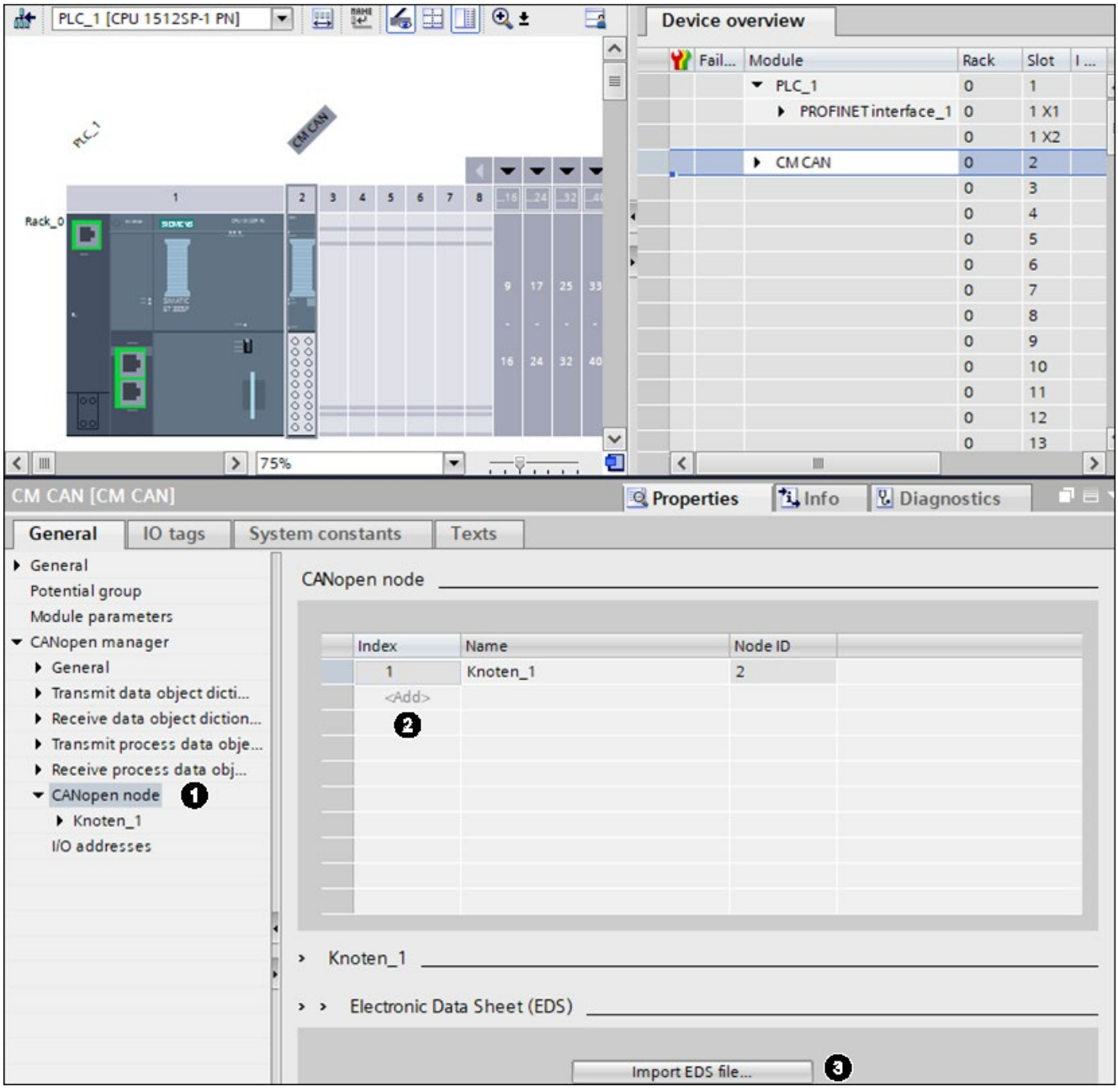


Figure 7-8 Adding CAN nodes

1. Add a "CAN node" to the list of nodes in the "CANopen manager" table. ① Create a new node by double-clicking the "Add" button in the index column. ②
2. Import the EDS (Electronic Data Sheet) file belonging to the CAN device. Select the EDS file in the menu "CANopen Manager CANopen node". Click the "Import EDS file" button ③ to open the corresponding file selection dialog.
3. As of firmware version 1.1 you can manually define the object dictionary (OD). For additional information about this function, refer to section "Handling of the OD table" below.

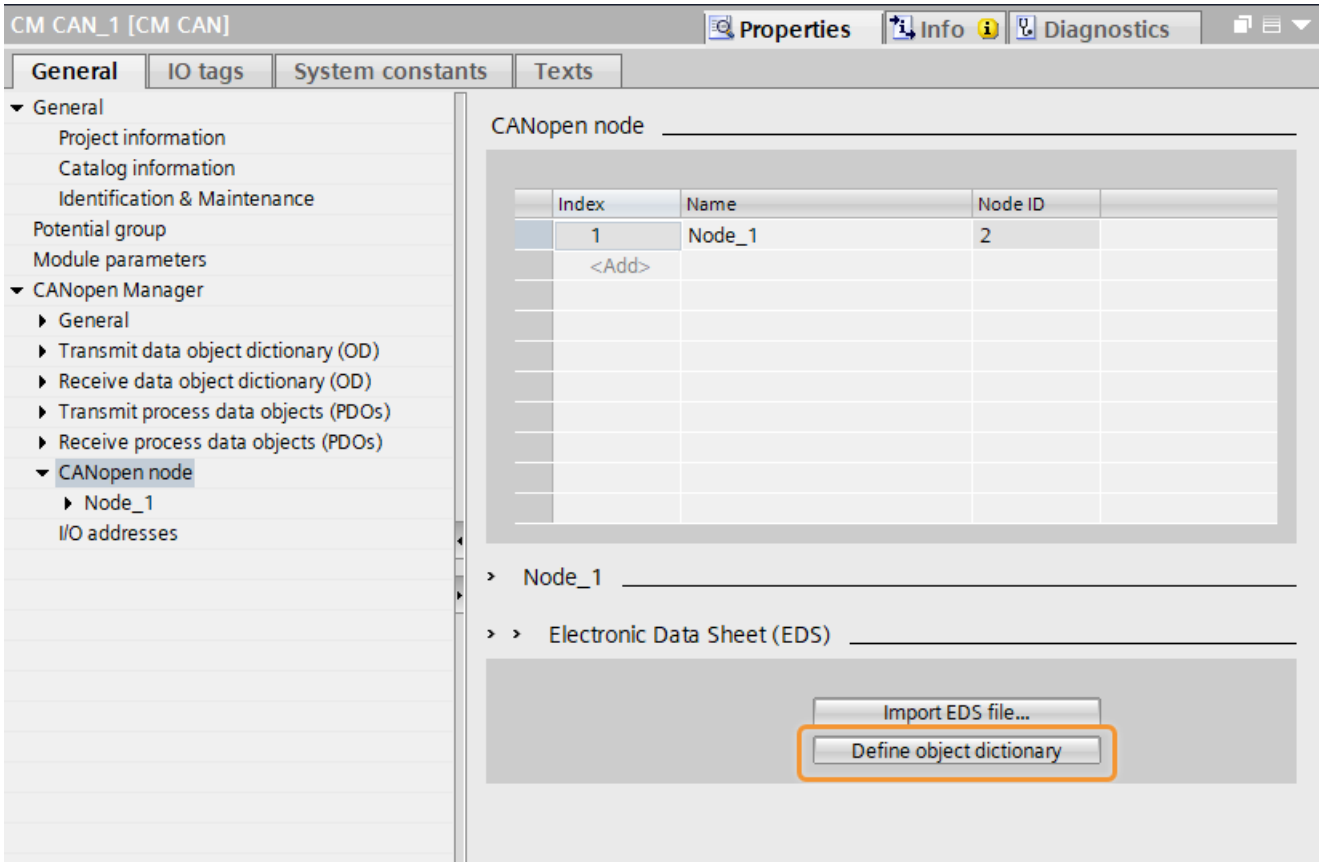


Figure 7-9 Define object dictionary (as of firmware version 1.1)

Note

General information

- The basis for working with the CANopen node is the object dictionary (OD) of a node that is usually made available by the manufacturer as an EDS file.
- You can import an existing EDS file by clicking on the "Import EDS file..." button. If no EDS file is available yet, create the object dictionary (OD) manually. To do this, use the "Define object dictionary" button. Both steps will result in an object dictionary (OD).
- The import of an EDS file can take several minutes depending on the size and complexity of the file. You can significantly shorten the time required for the import by reducing the EDS file to just the entries that are actually needed.

For users of the firmware version V1.0

- Once the EDS file is imported, all necessary information is part of the TIA Portal project so that the EDS file is no longer needed.
- To edit an EDS file properly, you need to know the structure of the EDS file well or use a specific EDS editor.
- Data from the EDS file cannot be updated at a later time. If the EDS file needs to be imported again due to changes (e.g. corrections), you must delete the node and create it again.
- Check the object dictionary (OD) of the CAN device for completeness and correctness. We assume hereafter that the OD can be used without changes.

For users of the firmware version V1.1

- The object dictionary (OD) is no longer write-protected so that you can edit, add and delete entries.
- An imported EDS-file creates an object dictionary (OD) that can be used immediately. However, you must confirm a manually created object dictionary (OD) by pressing the "Apply and check data" button.
- As of V1.1, you can correct any incorrect OD entry directly without deleting the node.
- Relationship between an OD and configuration menus:

You perform the entire configuration in the corresponding menus and **not** in the OD table. Settings from the menus are **not** visible in the OD table. You can find additional information in the section "Handling of the OD table" below.

For users of firmware version V1.2

- The OD index entry range has been extended to 2000h-6FFFh.

Handling of the OD table

After you have added a CAN node, you have two options in the "Electronic Data Sheet (EDS)" menu:

1. Using an existing EDS file via the "Import EDS file..." button

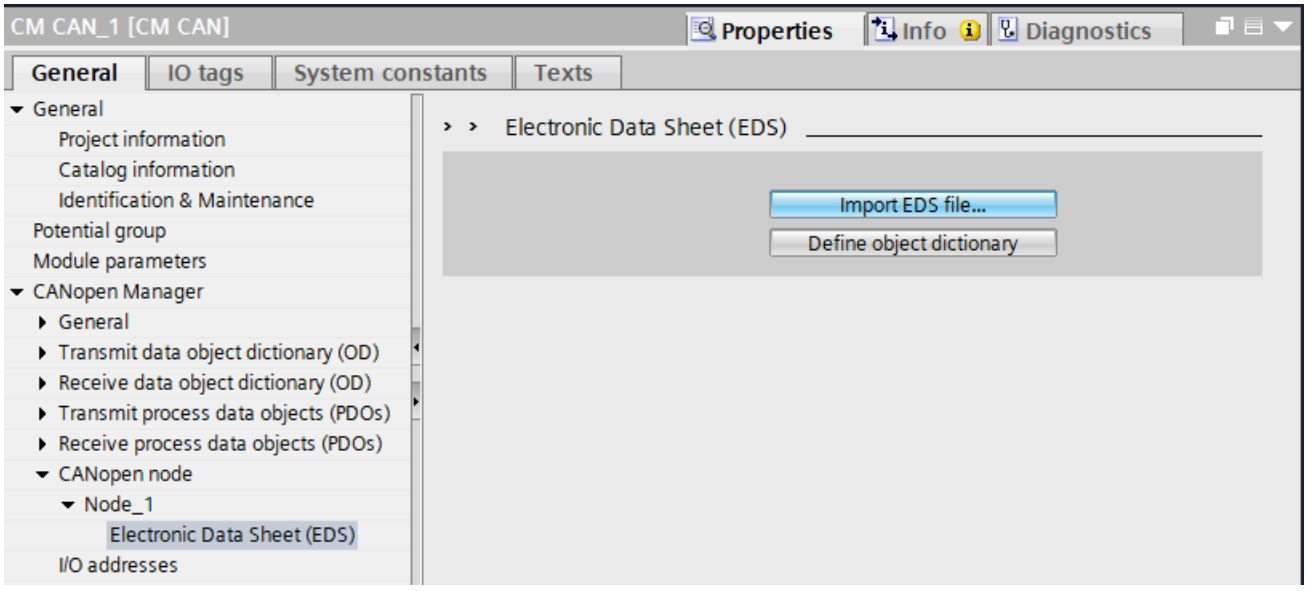


Figure 7-10 Importing the EDS file

2. Manual creation of an object dictionary (OD) using the "Define object dictionary" button:

A template is displayed in the "Object dictionary (OD)" menu.

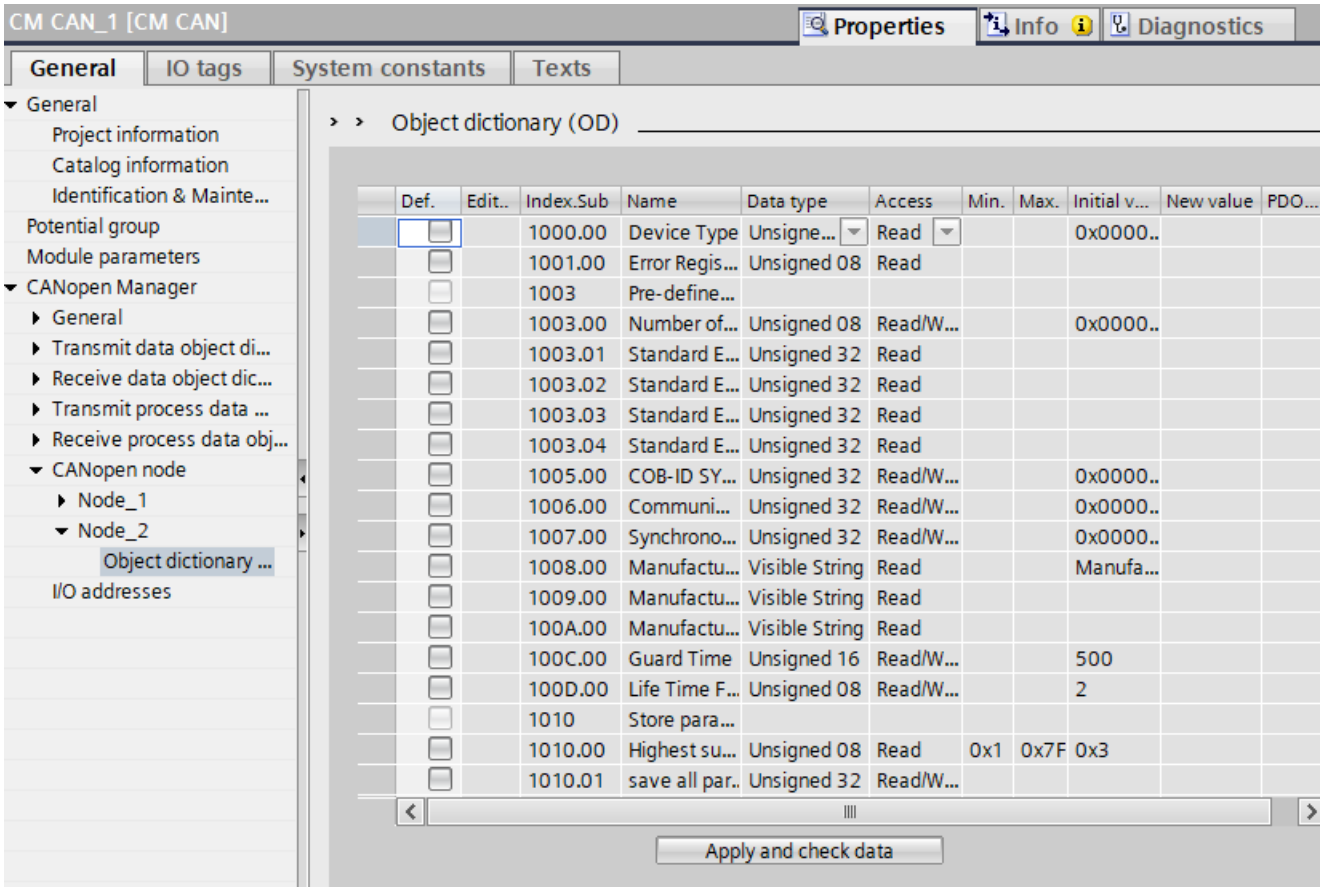


Figure 7-11 Defining the object dictionary

Here you can describe all objects that are available in your device. To do this, select the check box in the "Def." column or add a new entry at the end of the table.

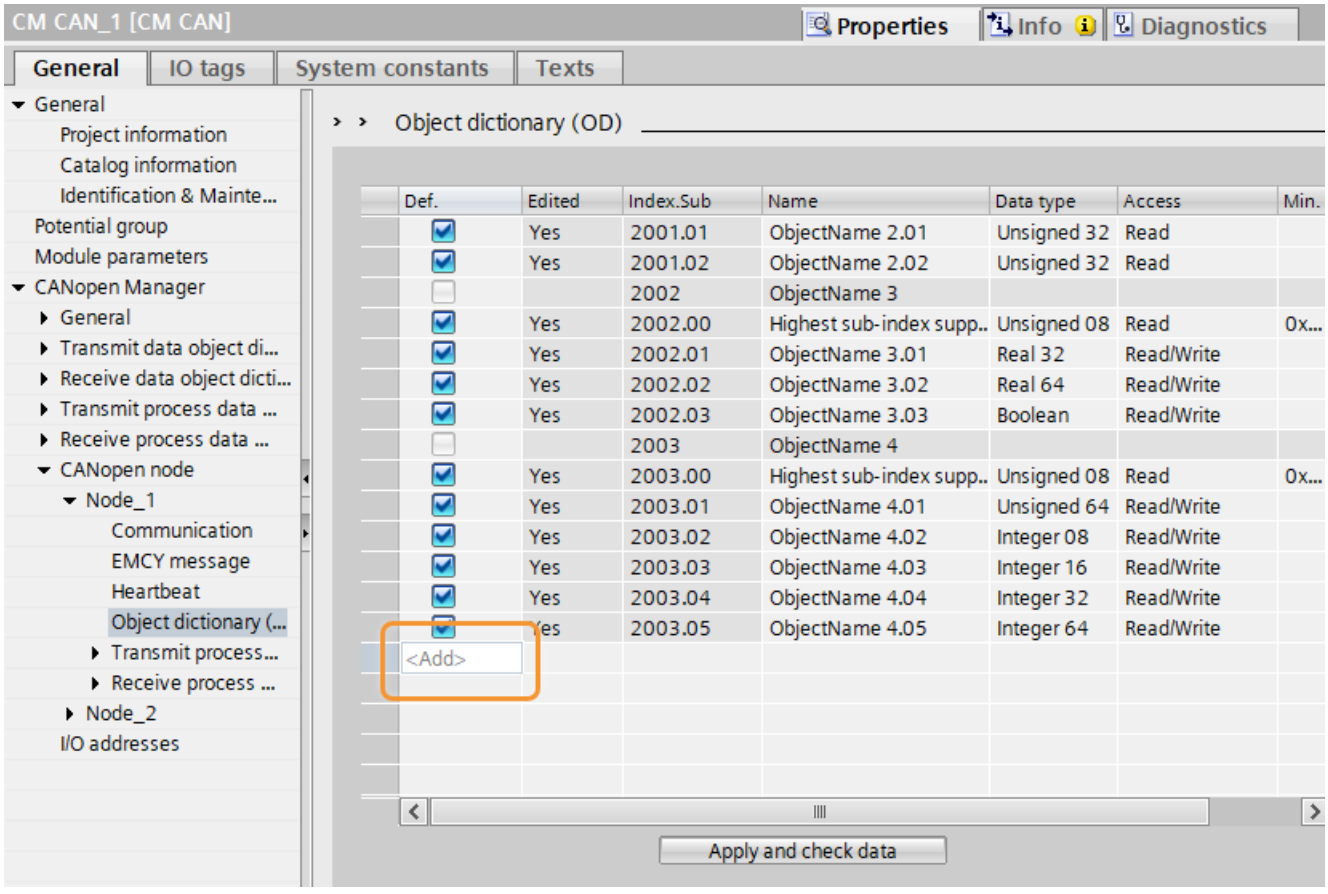


Figure 7-12 Add new entry

To add new rows, double-click the selected cell.

You can delete existing rows by using the shortcut menu of a table row. If you are not sure whether you want to delete an entry, disable it. The entry will be ignored during the next configuration.

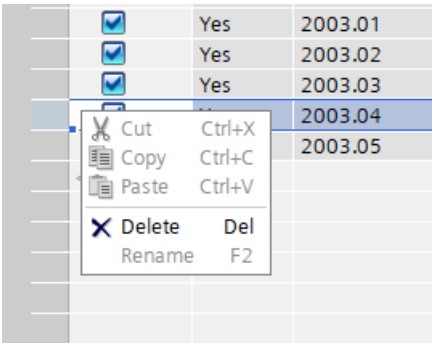


Figure 7-13 Shortcut menu of a table row

You must complete the finished OD table by clicking on the "Apply and check data" button. Once the data are applied, the other menus will become active and visible.

As soon as the OD table is completed by clicking on the "Apply and check data" button, you use it as you would an imported and manually defined EDS.

The following rules apply:

- The OD can be changed at any time but this may invalidate settings that were valid before the change.
- Changes in the OD table must be confirmed with the "Apply and check data" button to have an effect on the menus.
- Settings in the menus are **not** visible in the corresponding OD entries of the OD table (changes in the menus are not transferred to the OD table).
- You cannot bypass the menus and make corresponding configuration settings directly in the OD. This does **not** work.
- Only new object values that are not subject to a configuration menu can be set directly in the OD (these entries can be edited).
- Only entries with a selected check box in the "Def." column are considered to be available in the node device.
- You can add any object that is missing in the template or your EDS file.
- The "Apply and check data" button is hidden by default. It appears only when you make a change in the OD table.

Details on the individual columns

"**Def.**" column (Define):

- Entries can be edited depending on the "Def." column.
- When an EDS file is read in (similar to firmware version V 1.0), the default setting is Def.=True (enabled) for all entries. When "Define object dictionary" is selected, the default value is Def.=False (disabled) for all entries.
- When the "Def." option is selected, the CANopen node supports this object.

"**Edited**" column:

- "Yes" in the "Edited" column indicates that you have made a change in this row since the last check ("Apply and check data" button).
- The column is always write-protected.
- After you have pressed the "Apply and check data" button, the marking for a "valid" row is removed.
- After you have successfully checked everything, "Yes" is no longer displayed in the column and the "Apply and check data" button disappears again.

Editing an entry

You can edit any entry with a fully defined field in the Index.Subindex column. There is also a type of row where only one index is defined. Such a row can **never** be edited. It only involves a header with a name of OD.

You can only edit entries when the check box in the "Def." column is selected.

Editable columns:

- Name
- Data type
- Access
- Min.
- Max.
- Initial value
- New value
- PDO assignable
- Index.Subindex - You can only edit this column when you have added a new entry. It can no longer be edited once you press the "Apply and check data" button.

The following consistency checks are performed between the columns during editing:

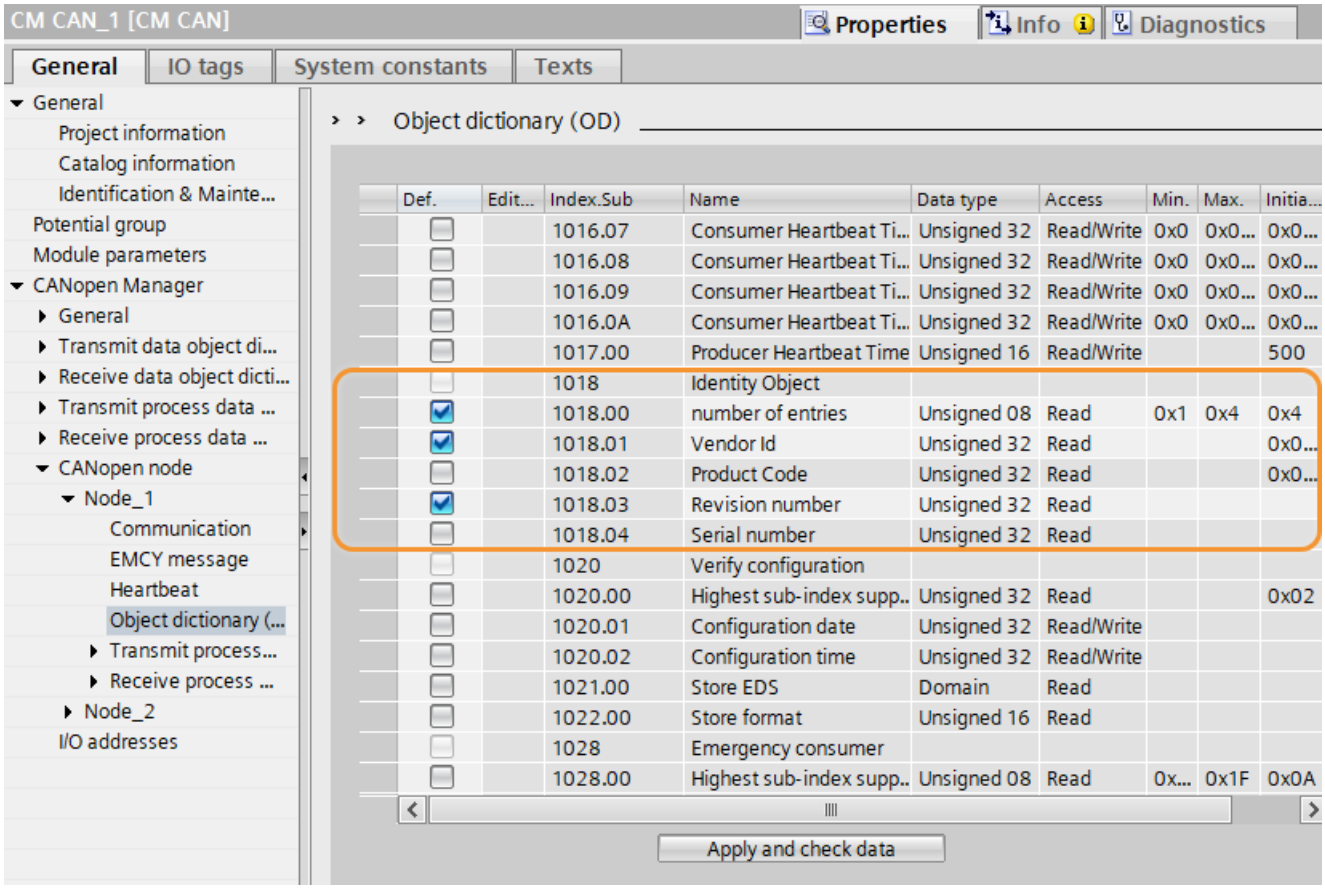
- Data type ↔ Min./Max. ↔ Initial value/New value
- Access ↔ PDO assignable

These changes are only local for the OD table until you press the "Apply and check data" button. No other menu is affected.

Pressing the "Apply and check data" button initiates the following checks and updates:

1. A test is performed to see whether entries were added. These entries are then placed at the correct location in the list. Only the newly added entries are taken into account during sorting. You can only edit the "Index.Sub" field when you have added an entry. The field remains editable until the entry is sorted in the table. After that, you can no longer change the "Index.Sub" field. If you want to change it, you have to delete the entry and add it again.
2. A check is performed to determine whether the object can be read or written to so that data exchange is possible.

3. Checking the mandatory ODs:
   - The OD entries (OD 1000, OD 1018.01, OD 1018.02, OD 1018.03) are checked regardless of whether an EDS file is imported or the object dictionary (OD table) is edited.
   - When one of these ODs is not available in the EDS file or is not selected in the OD table, the "Loose configuration" check box is selected and disabled in the "Node > Communication" menu.



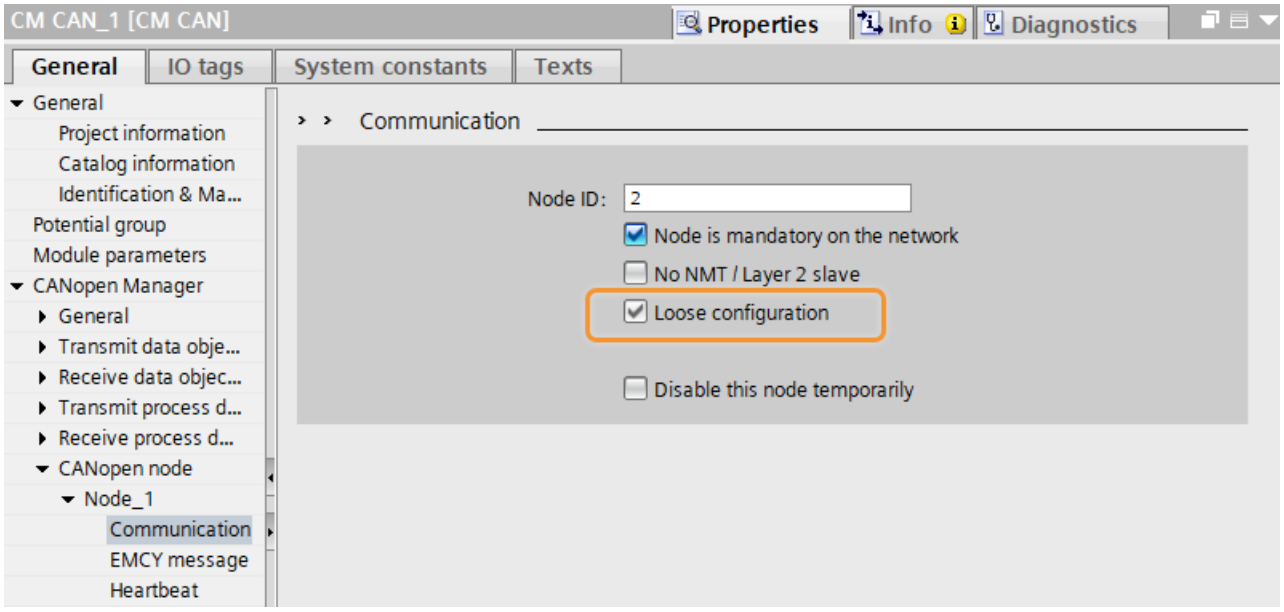


Figure 7-14 Loose configuration

4. Creating new temporary lists for RX-PDOs and TX PDOs according to the entries in the OD. When entries are incomplete (missing subindex or mapping entry), the PDO is rejected (this can have an effect on the previously correctly defined PDO).
5. A check is performed to determine whether changes were made that require an update of the list of mappable objects or objects actually mapped. All affected lists are updated.

Note

When you make changes to OD entries that have an effect on a previously configured PDO, these configurations are lost. You must make this configuration again.

Creating an OD index entry for transmit data in the manager

1. Open "Transmit Data Object Dictionary (OD)" ① in "CANopen manager". Select "Transmit ODs".
2. Double-click the "Add" button to create a new table entry. ②

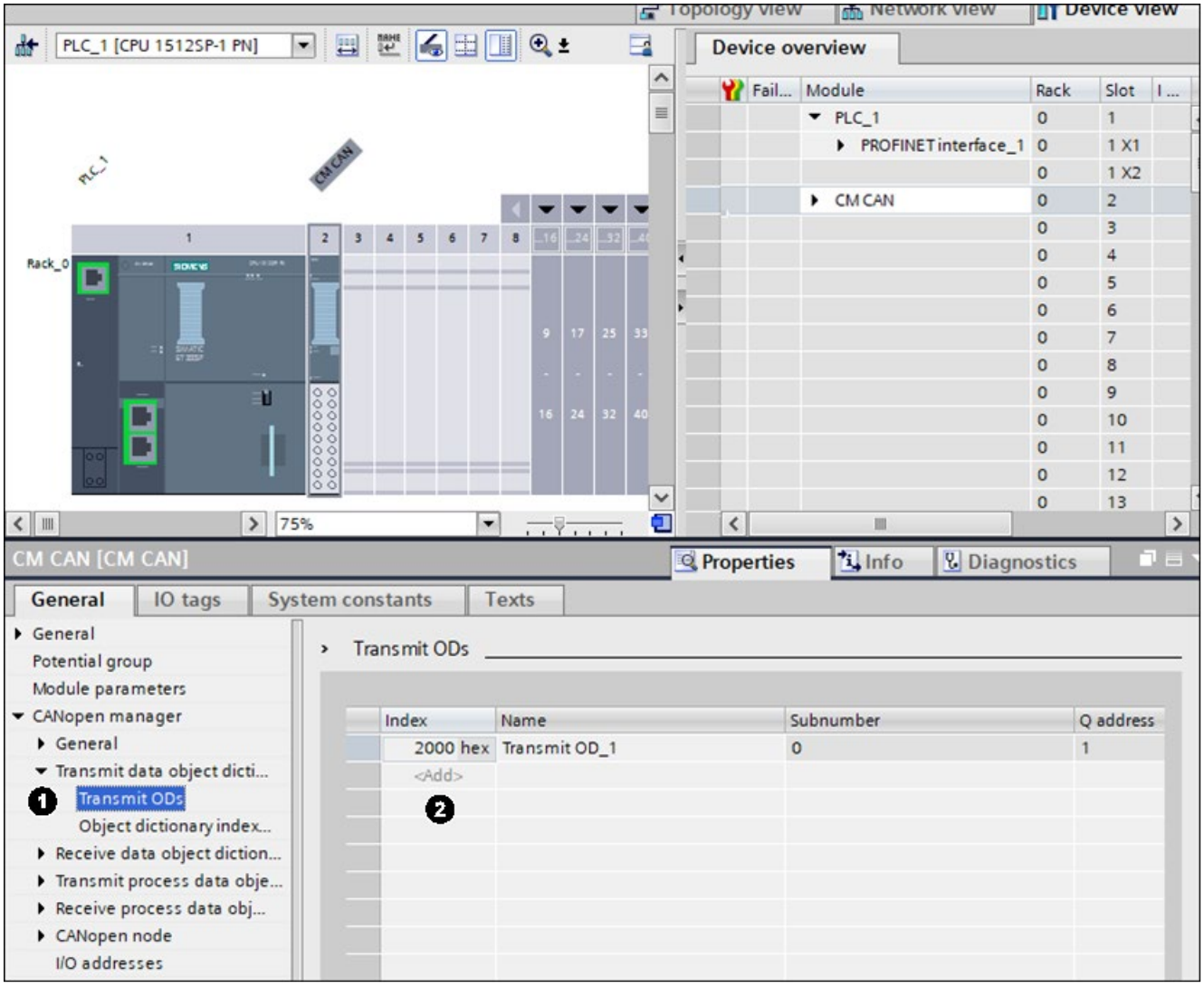


Figure 7-15 Create OD index entry

Define transmit subindex

1. Select "Transmit Data Object Dictionary (OD)" and then "Object Dictionary Index Definition" in the CANopen manager menu.①
2. Create the subindexes belonging to the OD index.
   To create additional subindex entries, double-click "Add".②

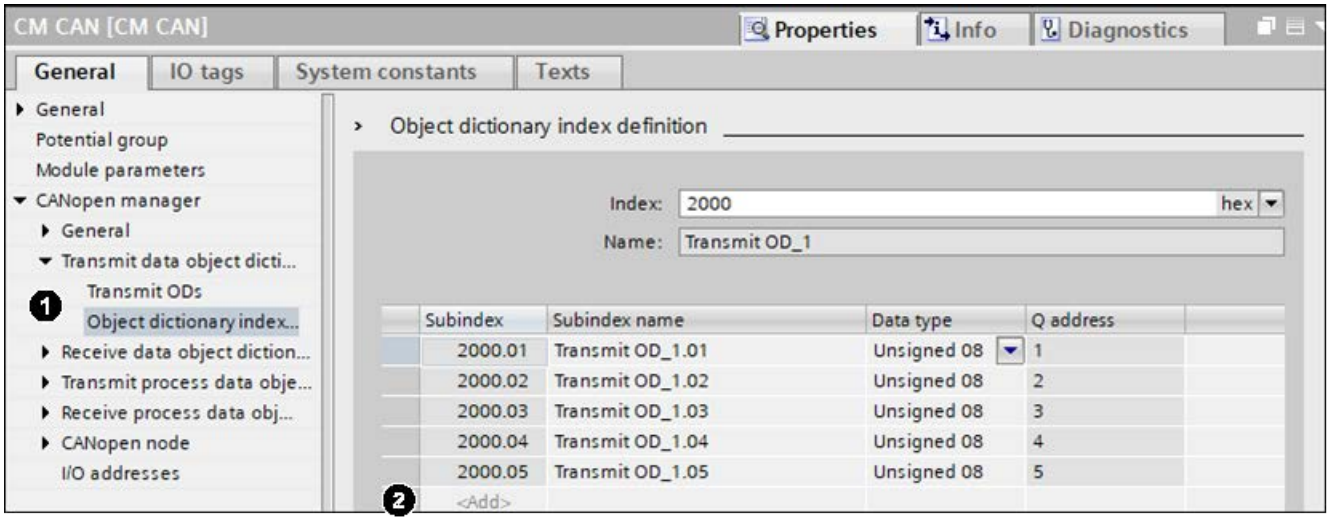


Figure 7-16 Object dictionary index definition

Creating additional OD entries for transmit data in the Manager

Repeat the above steps until all desired transmit data has been created in the OD of the Manager.

Note

Limited number of OD and subindex entries

You can create a maximum of 100 OD entries (indexes). This maximum number is independent of the distribution of entries between the transmit data OD and receive data OD.

The number of subindex entries is also limited. You can create up to 256 subindexes for all indexes together if the subindexes are 1 byte long (256 byte memory limit).

At least 1 subindex must be defined for each index. This means when 100 indexes are defined, one index can have 156 subindexes with a length of 1 byte and the remaining indexes can have only one subindex with a length of 1 byte.

Defining transmit PDO in the Manager

1. Select the "Transmit PDO definition" window ①.
2. Select a free PDO number to which a COB ID has not yet been assigned in the "PDO" line ② and click on the "Define PDO" button.③
   Set the COB ID to the same value that is used by the corresponding Receive PDO in the slave. ④

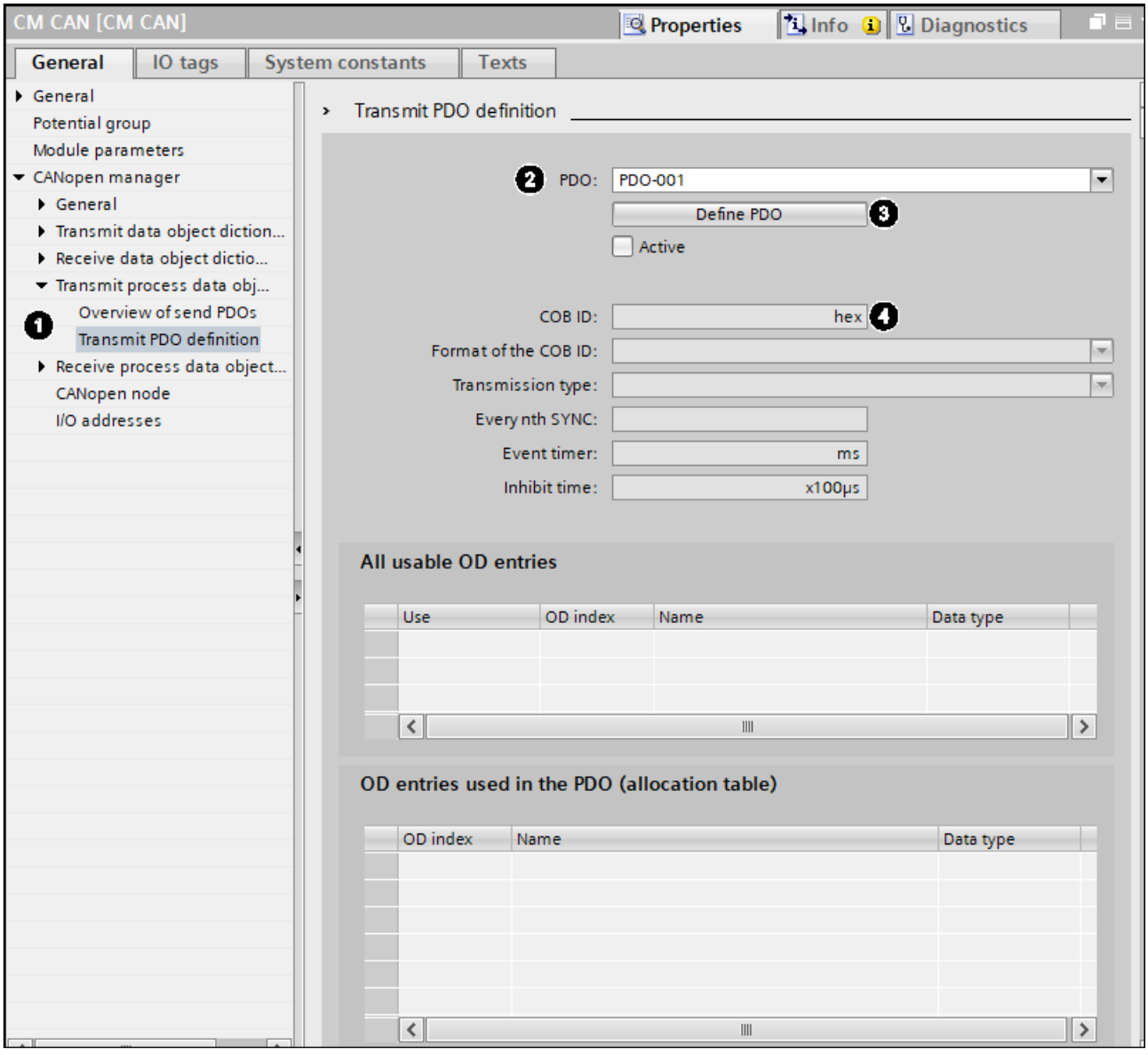


Figure 7-17 Transmit PDO definition_1Part

3. Assign transmit data to the transmit PDO.
   To assign the transmit data, select the data that is to be transmitted in the transmit PDO from the "All usable OD entries" table with the buttons in the "Use" column ⑤.
4. The selected OD entries now appear in the table "OD entries used in the PDO". ⑥
   Two additional tables below allow easy reference to see how any previously defined receive PDOs with the same COB-ID are structured.

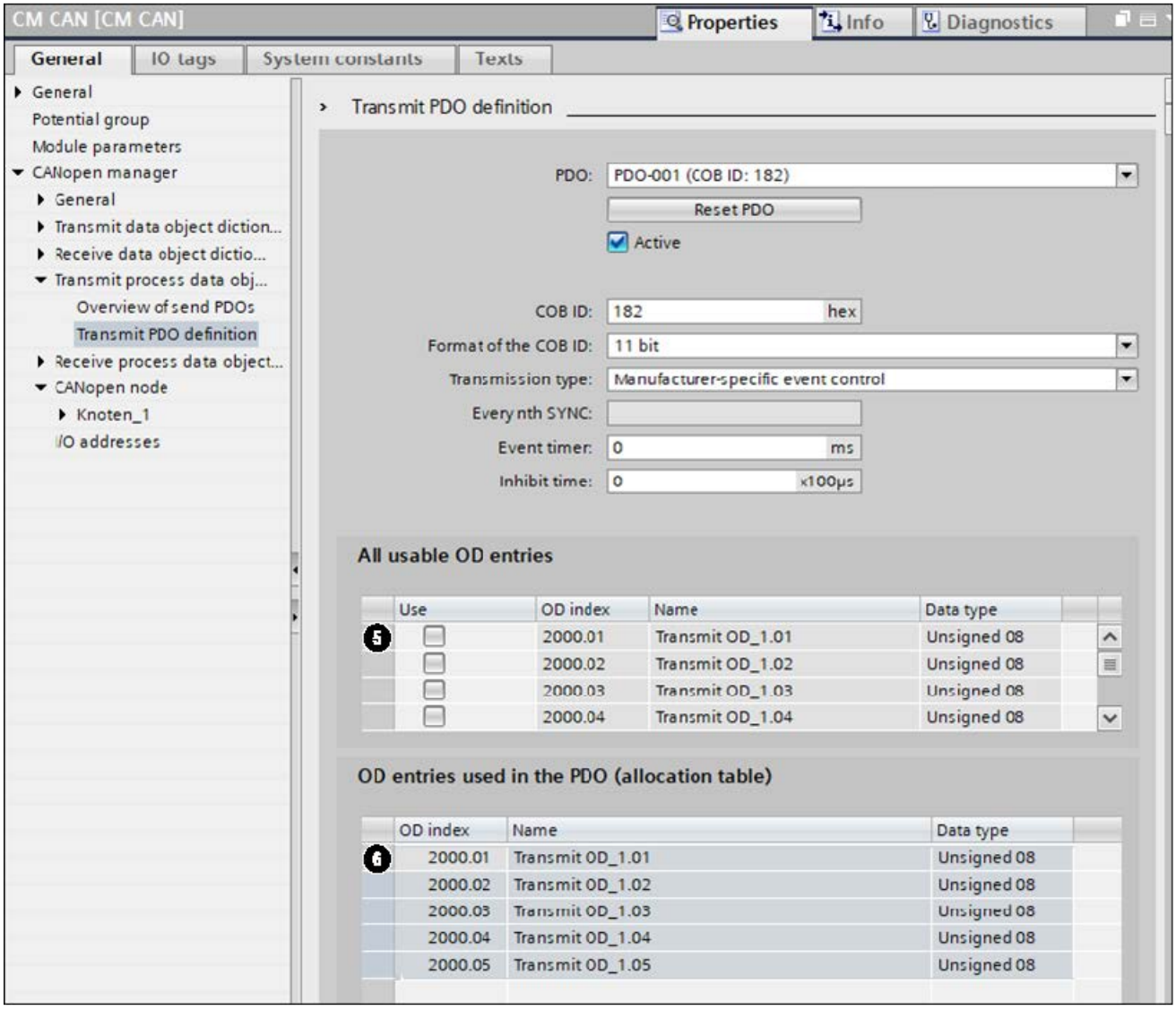


Figure 7-18 Assignment of transmit data

Note

The length of the transmit data assigned to a Transmit PDO must match the length of the receive data in the corresponding Receive PDO in each case at the end of configuring.

Defining a receive data OD and assigning Receive PDOs

The steps for defining and assigning the receive data are analogous to those for transmit data:

1. Create receive data object directory
2. Define a receive OD index and subindex
3. Define Receive PDOs
4. Assign the OD entries used in the PDO

Note

The length of the receive data assigned to a Receive PDO must match the length of the transmit data in the corresponding Transmit PDO in each case at the end of configuring. If necessary, the length of the Receive PDO must be adapted by using predefined dummy OD entries.

Disable checking of PDO length (for configuration with HSP_V16_0310_003_ET200SP_CM_CAN_1.0)

As of firmware version 1.1, you can select the "Disable checking of PDO length" option (see figure below).

Option enabled

When you select this option, the data length (number of received bytes) of the received PDOs is adapted. Error, warning or diagnostic messages are not generated.

The value is adjusted to the correct data length as follows:

- Data below the defined length are supplemented with the value 0.
- If data exceeds the defined length, the more significant bytes are deleted.

Option disabled

If you disable this option, all data from a received PDO with a length deviating from the standard is discarded and replaced with the value of the last valid PDO (last received PDO with the correct length). The CANopen Manager calls up an error message in the diagnostic buffer.

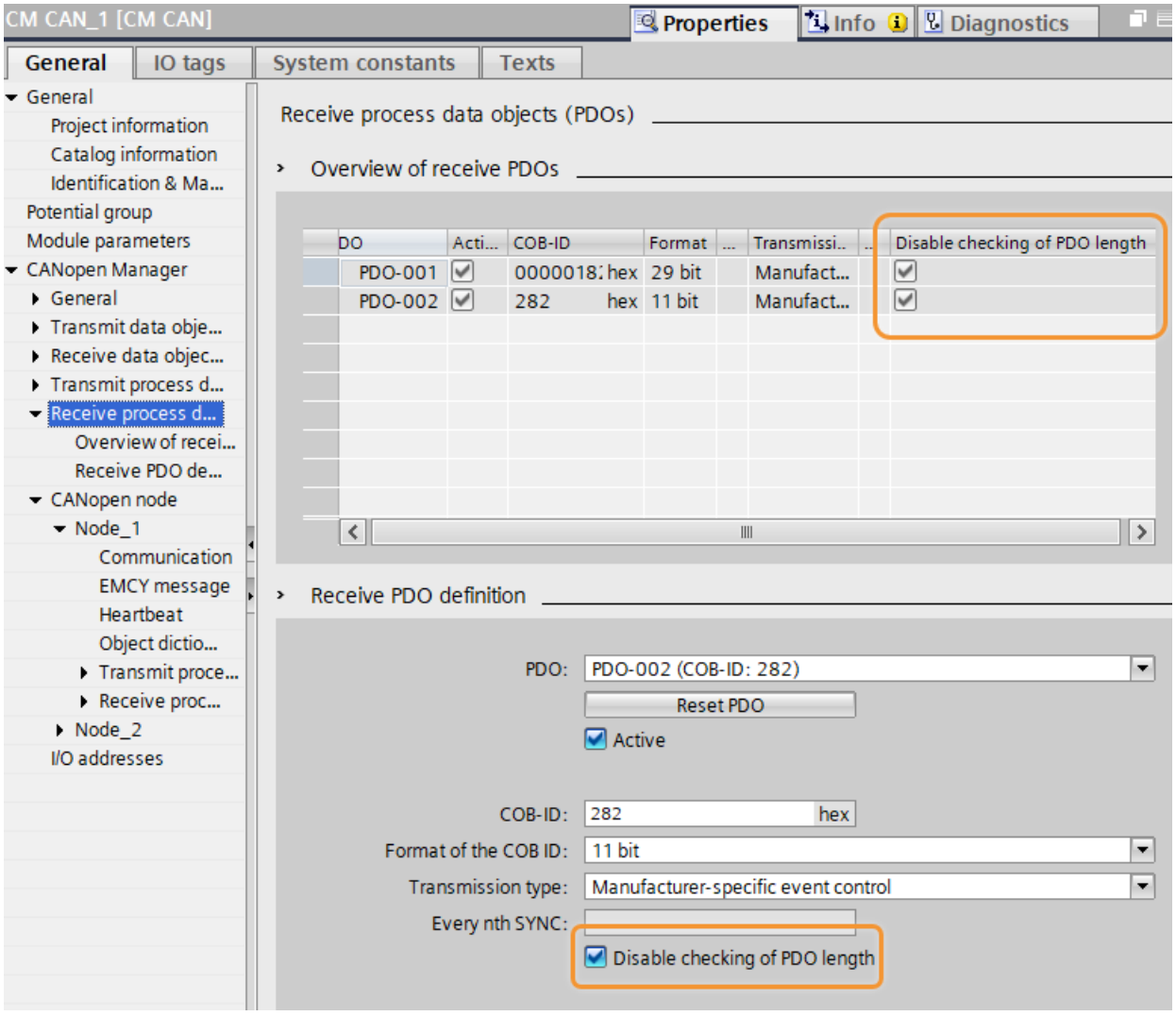


Figure 7-19 Disable checking of PDO length

Adapting I/O addresses

Except for the start of the I/O ranges, you cannot adapt the I/O addresses automatically assigned by the TIA Portal.
The first I/O addresses assigned to the module are control and status information. The control and status information is exchanged with the user program via these addresses. It is important for startup of the module that the control information transferred here from the user program is set correctly.

Disabling a node temporarily (for configuration with HSP_V16_0310_003_ET200SP_CM_CAN_1.0)

You have configured a series of nodes in the TIA Portal, but one node does not physically exist in the CAN network. To work with the "reduced" network, you can temporarily disable one or more nodes. Such a CANopen node with its entire configuration remains in the project but is **not** downloaded to the CM CAN module. The node **does not exist** in the configuration that is used by the CM CAN module. No errors or warnings are generated by the module.

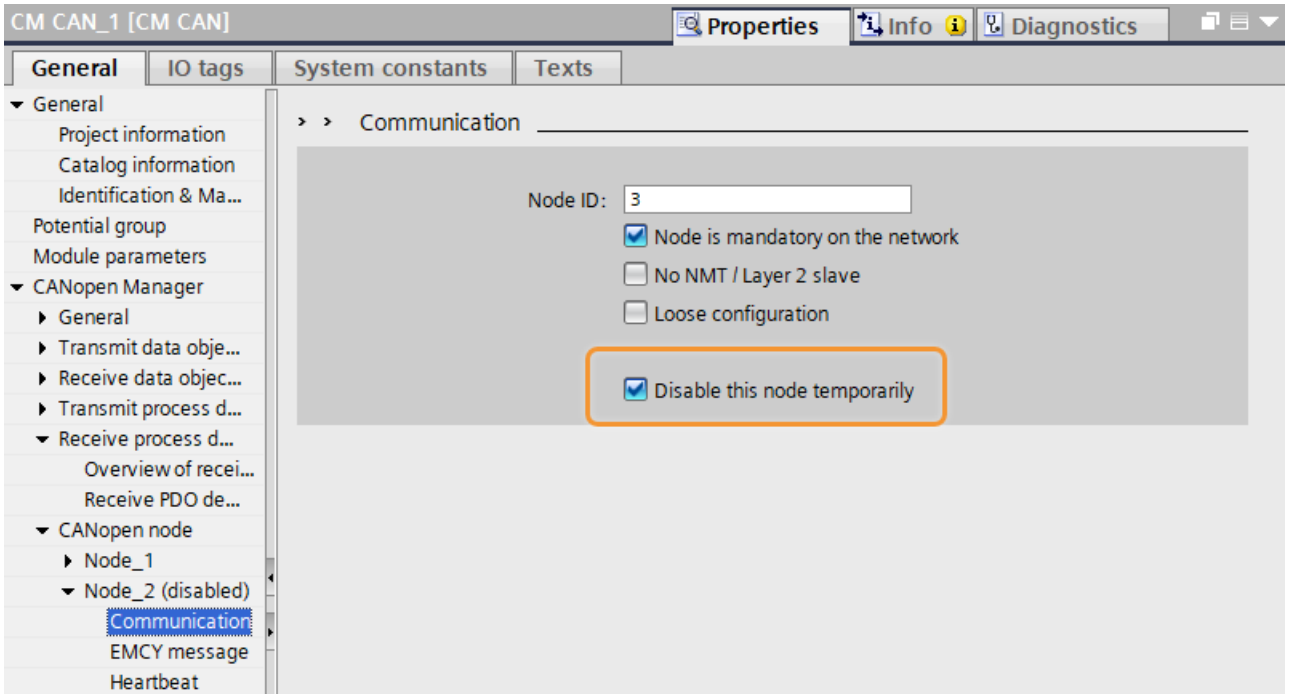


Figure 7-20 Disabling a node temporarily

When such a node appears in the menu, it is marked as "(disabled)".

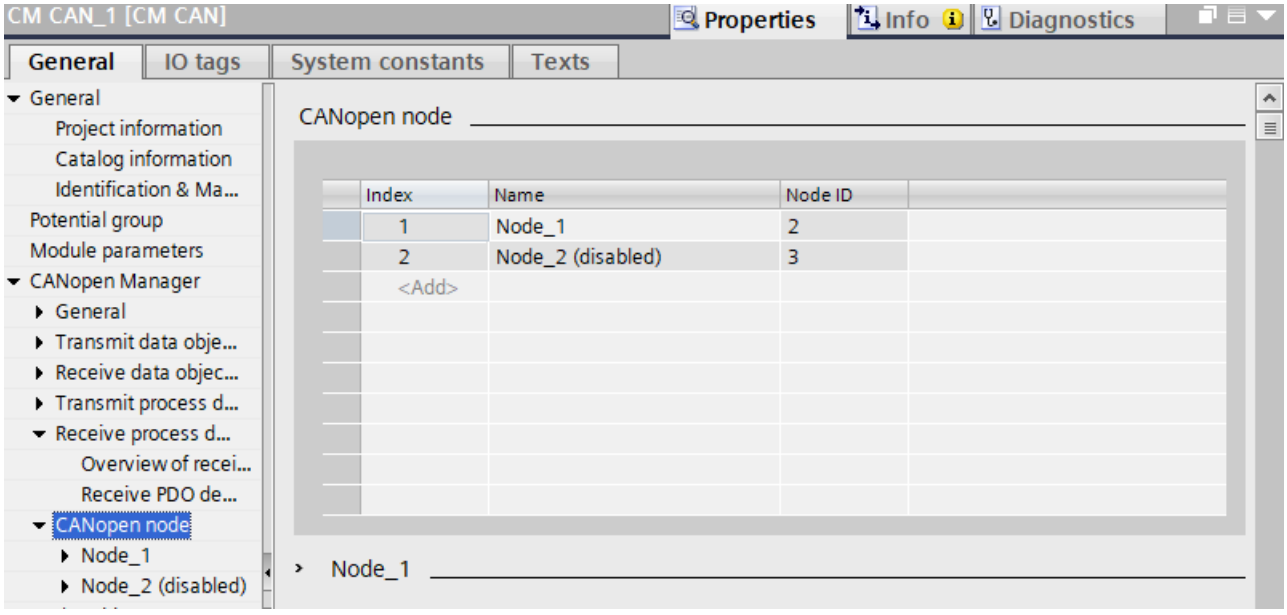


Figure 7-21 Node disabled

Checking data consistency

You can check the consistency of the assignments for the receive data and transmit data as well as the data types used with a compilation.

Setting the module to "Operational" with the user program

To enable data to be transferred between the SIMATIC S7 controller and the module, you must set the control bit to "1" with the user program. This control bit is transmitted cyclically from the SIMATIC S7 controller to the module as part of the IO data.

Note

In "CANopen manager" mode, bit 2 must also be set to "1" in the control information transmitted from the SIMATIC S7 controller to the module. You can find additional information on the effect of the control bits under Control and Status Information (Page 34).

7.3 Configuring CANopen slave

7.3.1 Overview

Configuration with the HSP in the TIA Portal (CM CAN is NMT slave).

The module is configured in the TIA Portal. The configuration for "CANopen slave" operating mode mainly consists of the following steps:

- Import the HSP of the CM CAN module to the TIA Portal.
- Drag the module from the HW catalog to the project.
- Set the "CANopen slave" operating mode.
- Set bus-specific parameters for the "CANopen slave" module:
  - Node ID
  - Transmission rate
- Create OD entries for the process data to be exchanged between the SIMATIC S7 controller and the CANopen network.
- Check and compile the configuration.
- Export the EDS file.

7.3.2 Configuration in the TIA Portal

TIA Portal: Devices & networks

Follow these steps:

1. Select the module with its specific article number from the HW catalog. ①
2. Drag the communication module to a free slot in the ET 200SP system. ② The module is a part of the modular IO system. The IO system is connected to the fieldbus.

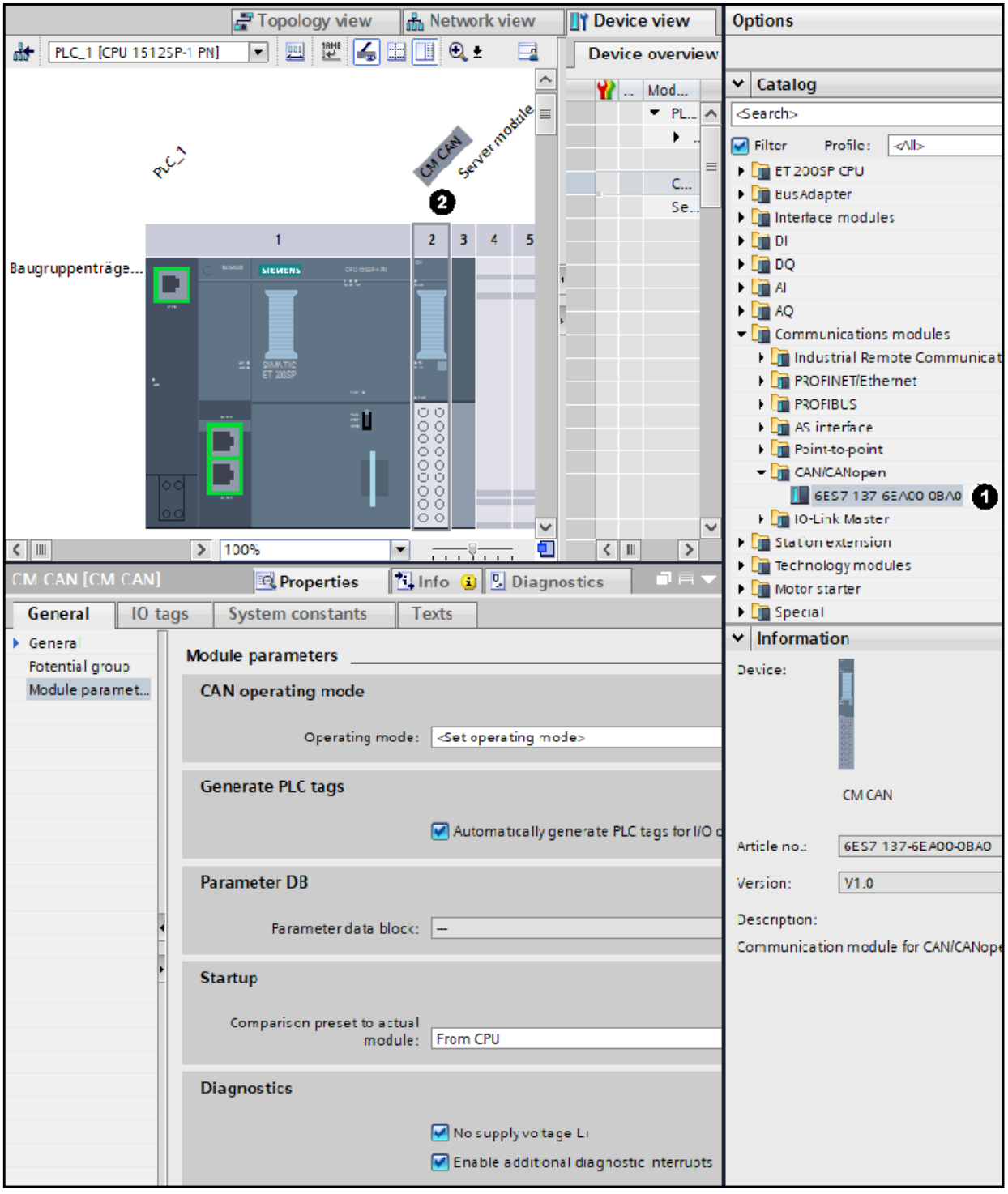


Figure 7-22 TIA Portal: Devices & networks

Selecting CAN operating mode of the module

The CAN operating mode is selected using the "Set operating mode" drop-down list with the following selection options:

- CANopen manager
- CANopen slave
- CAN transparent

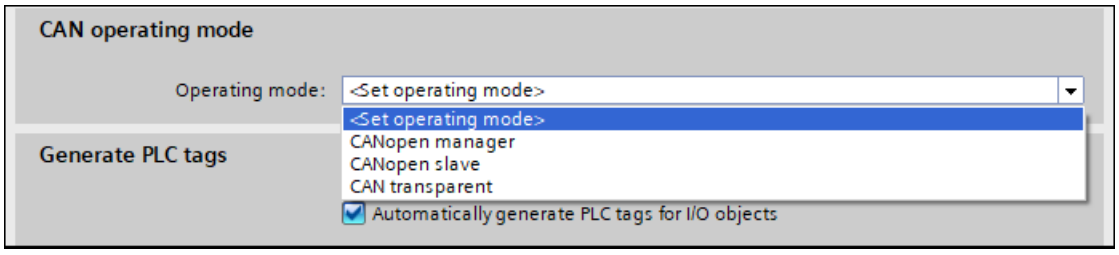


Figure 7-23 Selecting an operating mode

Select the "CANopen slave" operating mode.

After selection of the operating mode the parameters associated with the selected mode can be set.

Setting the communication parameters

Set the bus-specific parameters for the CANopen slave module: Node ID, transmission rate and the number of desired process data objects. ①

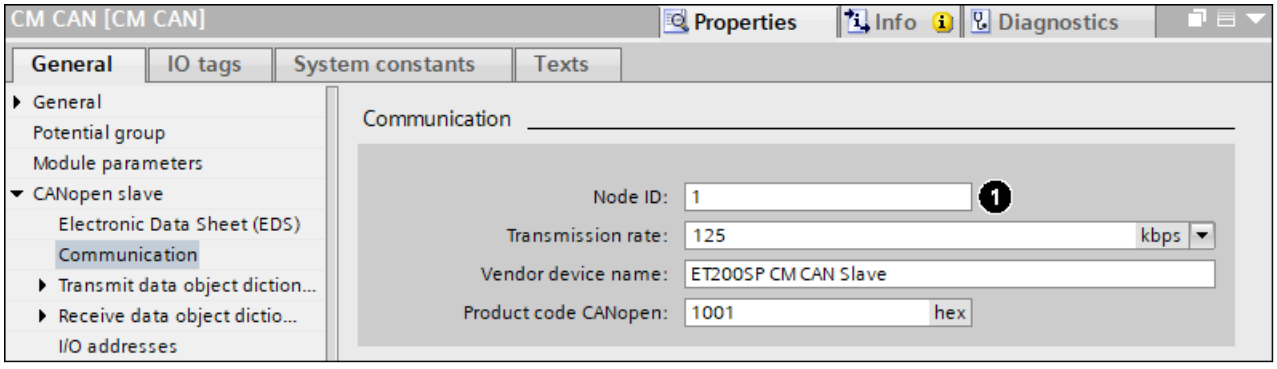


Figure 7-24 Setting the communication parameters

Creating an OD index entry for transmit data in the slave

1. Open "Transmit Data Object Dictionary (OD)" in the "CANopen slave" module and then select "Transmit ODs". ①
2. Double-click the "Add" button to create a new table entry. ②

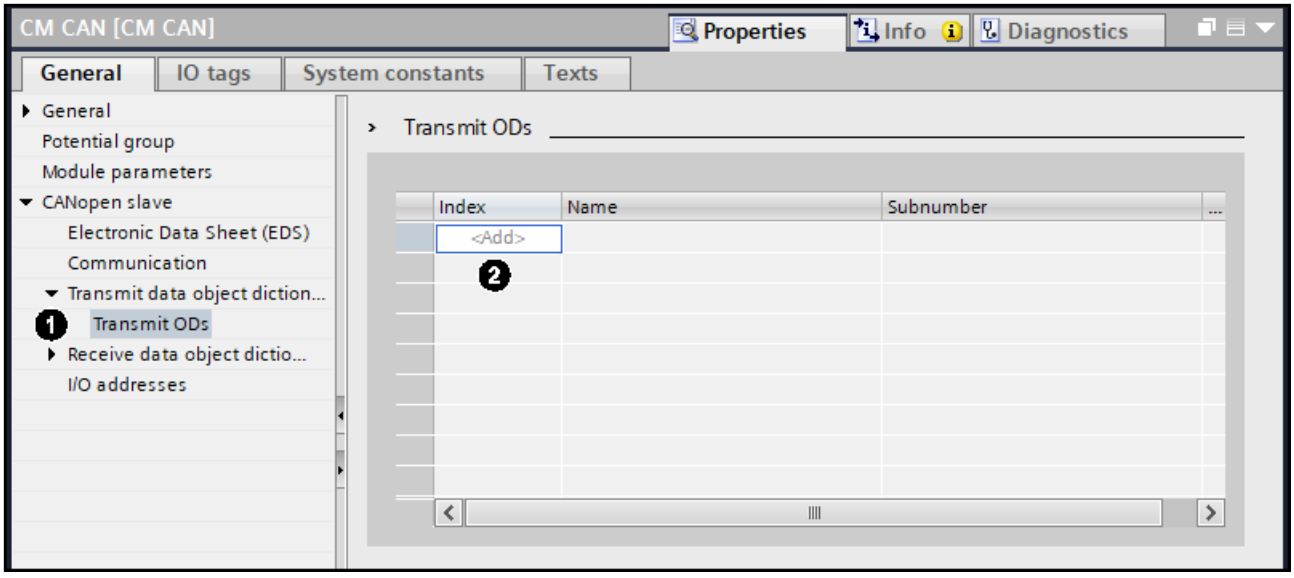


Figure 7-25 Creating a transmit data object dictionary

Define transmit data subindex

1. Select "Transmit Data Object Dictionary (OD) - "Object Dictionary Index Definition" in "CANopen slave". ①
2. Create the subindexes belonging to the OD index.
   To create additional subindex entries, double-click "Add" ②.

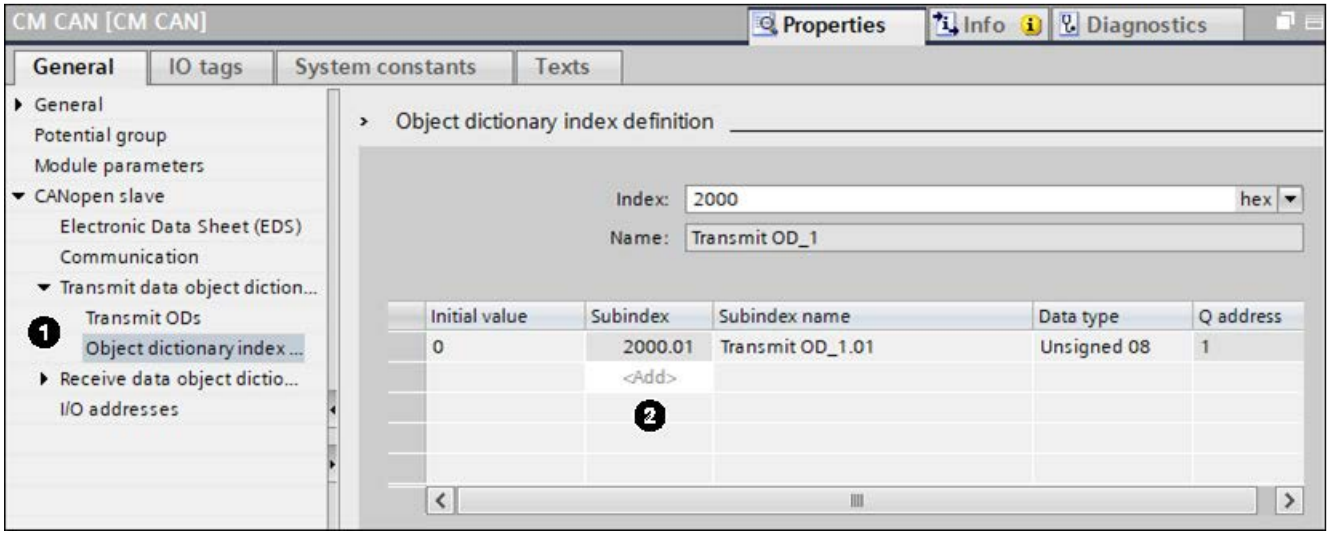


Figure 7-26 Object dictionary index definition

Creating additional OD entries for transmit data in the Manager

Repeat the above steps until all desired transmit data has been created in the OD of the Manager.

Note

Limited number of OD and subindex entries

You can create a maximum of 100 OD entries (indexes). This maximum number is independent of the distribution of entries between the transmit data OD and receive data OD.

The number of subindex entries is also limited. You can create up to 256 subindexes for all indexes together if the subindexes are 1 byte long (256 byte memory limit).

At least 1 subindex must be defined for each index. This means when 100 indexes are defined, one index can have 156 subindexes with a length of 1 byte and the remaining indexes can have only one subindex with a length of 1 byte.

Defining a receive data OD

The steps for defining and assigning the receive data ODs are similar to those for transmit data ODs:

1. Create receive data object dictionary
2. Define receive OD index and subindex

Adapting I/O addresses

The input and output addresses of the communication module use an address space of 1-256 bytes. The I/O addresses are automatically assigned for each communication module when the device configuration is specified in TIA Portal.

Checking data consistency

Subsequent compilation allows you to check the consistency of the assignments and the data types used.

Exporting the EDS file

Export the EDS file of the module.

1. Select electronic data sheet (EDS). ①
2. By clicking the "Export EDS file" button, the corresponding file selection dialog opens. ②

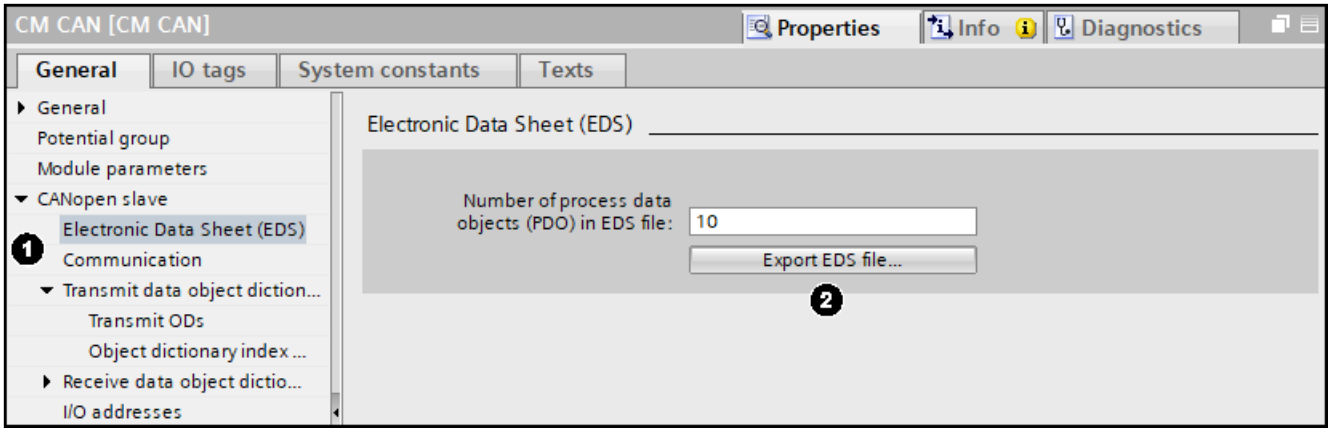


Figure 7-27 Exporting the EDS file

Allow module to transition to "Operational" via the S7 user program

To allow the transition of the module to "Operational" state and thus the transmission of data between the SIMATIC S7 controller and the module, you must set the control bit to "1" via the S7 user program. This control bit is transmitted cyclically from the SIMATIC S7 controller to the module as part of the IO data. However, the actual state transition must be triggered by the current NMT master.

7.4 Configuring CAN transparent

7.4.1 Overview

Configuration with the HSP in the TIA Portal

The module is configured in the TIA Portal. The configuration for "CAN transparent" operating mode mainly consists of the following steps:

- Import the HSP for the module into the TIA Portal.
- Drag the module from the HW catalog to the project.
- Set specific parameters.
- Set the "CAN transparent" operating mode.
- Set the transmission rate.
- Set the transmit/receive messages and transmit/receive proxies.
- Check and compile the configuration.

7.4.2 Configuration in the TIA Portal

TIA Portal: Devices & networks

Proceed as follows:

1. Drag the communication module CAN from the HW Catalog ① to the ET 200SP CPU or to the ET 200SP interface module. ②
2. Specify the "CAN transparent" operating mode.
3. Set the bus-specific parameters: The transmission rate.

4. Define the transmit and receive messages / proxy messages.
5. Check and compile the configuration.

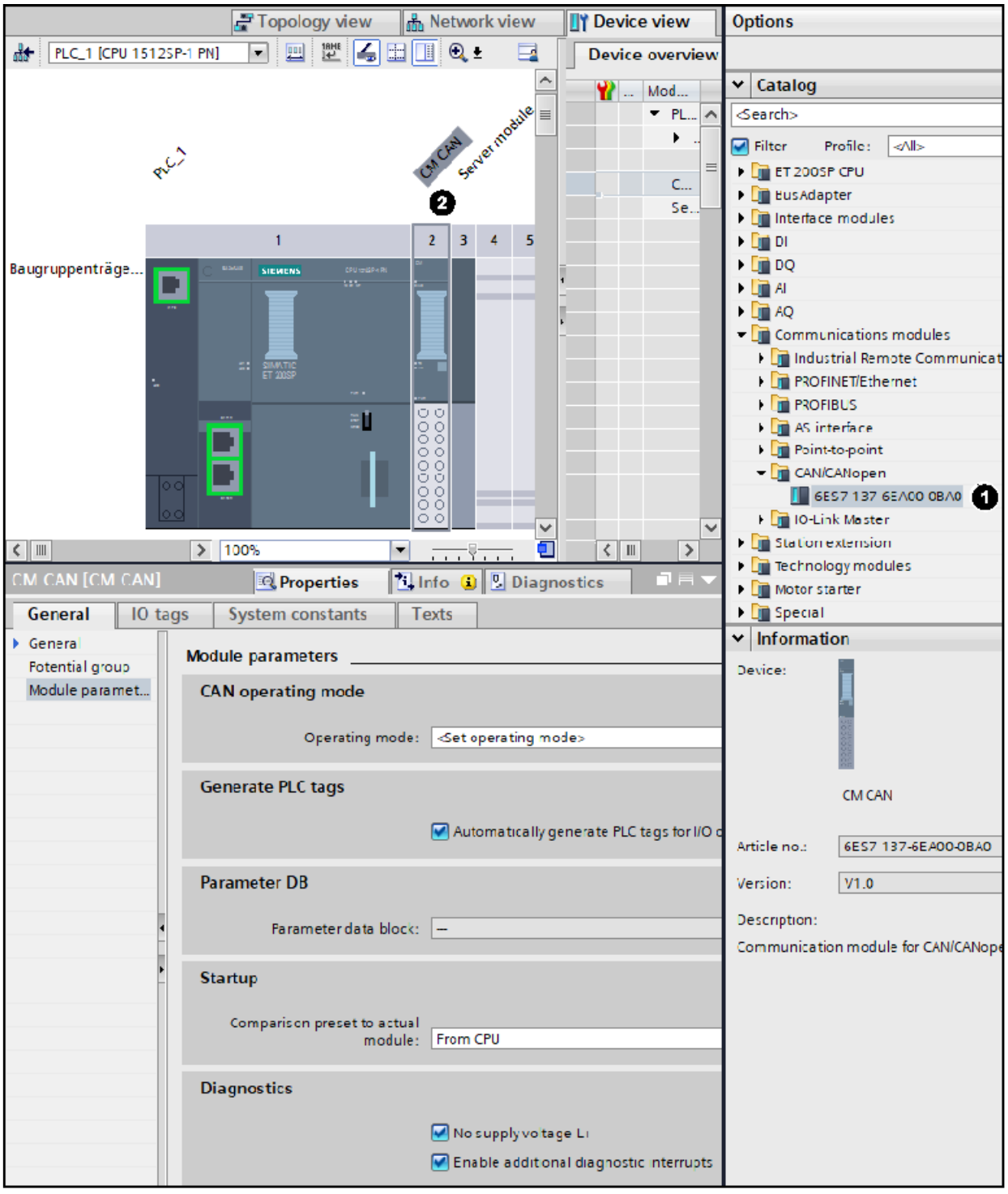


Figure 7-28 TIA Portal: Devices & networks

Selecting CAN operating mode of the module

The CAN operating mode is selected using the "Set operating mode" drop-down list with the following selection options:

- CANopen manager
- CANopen slave
- CAN transparent

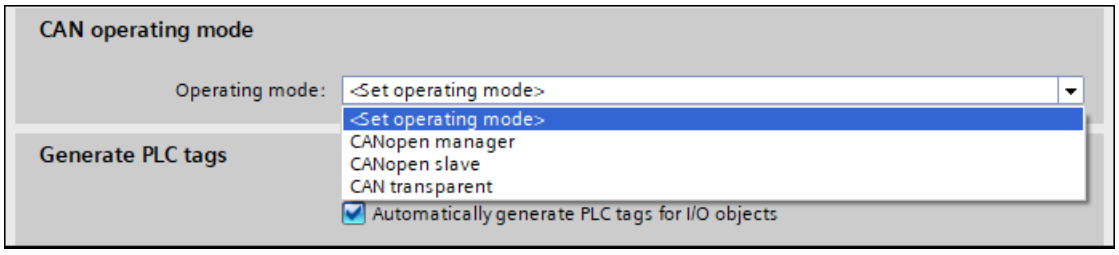


Figure 7-29 Selecting the CAN operating mode

1. Select the "CAN transparent" operating mode.
   After selection of the operating mode the parameters associated with the selected mode can be set. All settings for the "CAN transparent" module of the module are made in slot 1.

Setting the communication parameters

Select the transmission rate for the communication. ①

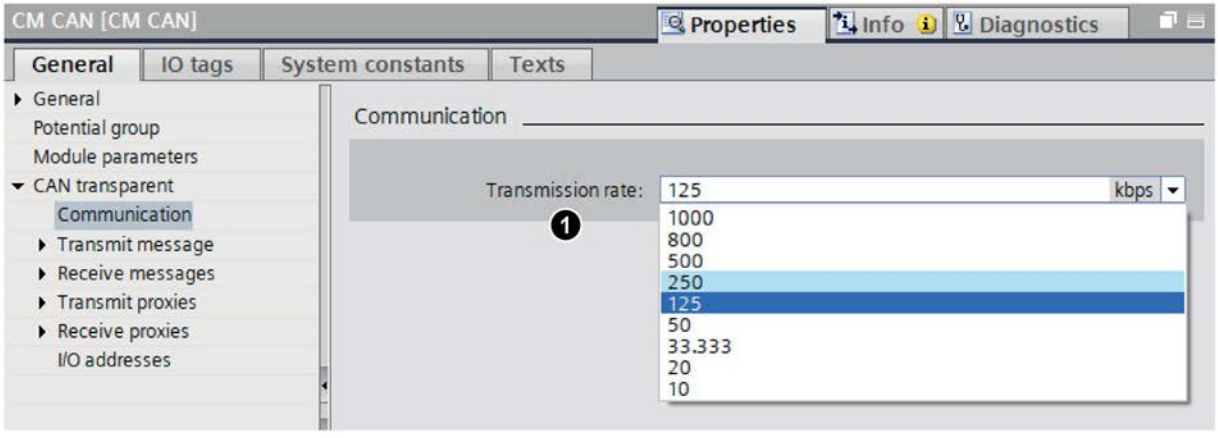


Figure 7-30 Selecting the transmission rate

Configuring transmit/receive messages

1. Open "Transmit messages" in the CAN transparent menu. Select the "Message definition" ① and create a transmit message by double-clicking the "Add" button. ②
2. Select "Message index definition" ③ and create the necessary message definitions. ④

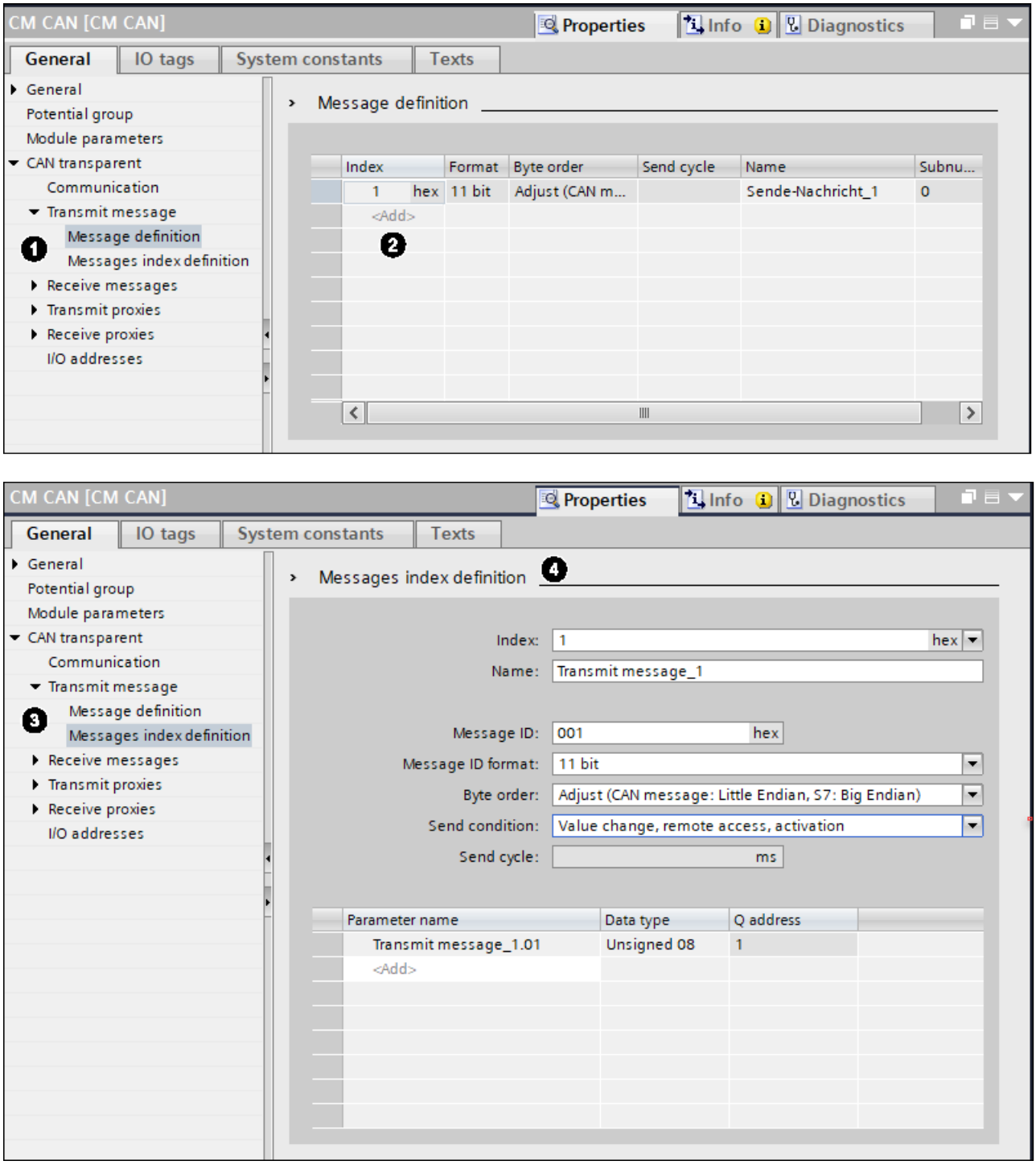


Figure 7-31 Transmit and receive messages

Creating and setting transmit proxies and receive proxies

1. In the "CAN transparent menu", open the "Transmit proxies" and select the "Proxy definition". ①.
2. Double-click on the "Add" button to create a new transmit proxy. ②

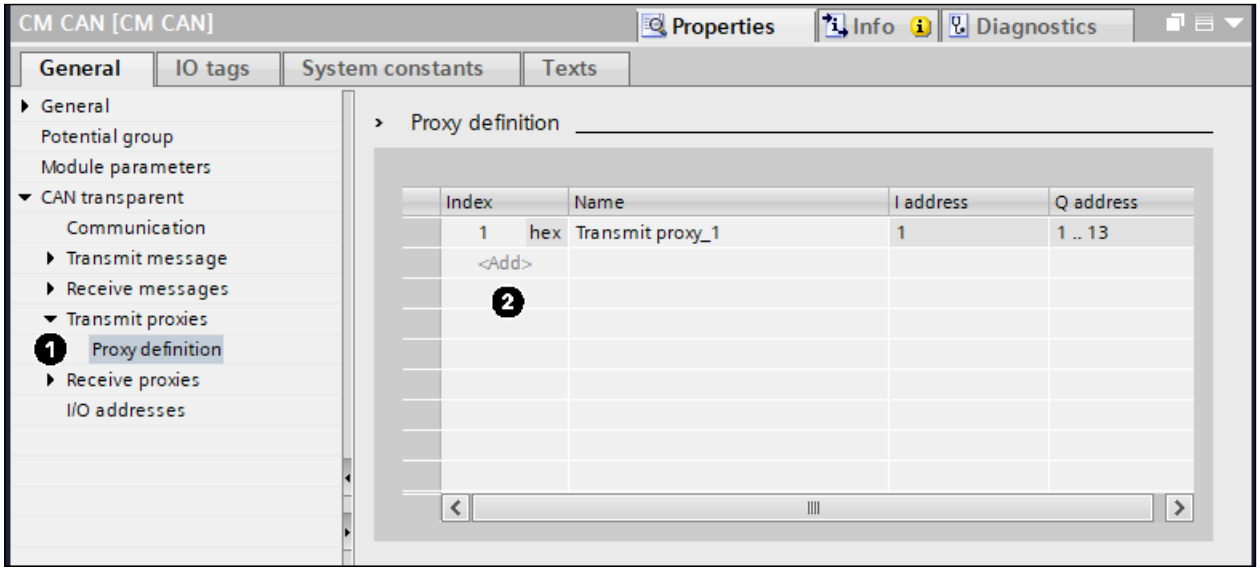


Figure 7-32 Creating and setting up transmit proxies

1. In the "CAN transparent menu" open the "Receive proxies" and select the "Proxy definition". ①.
2. Double-click on the "Add" button to create a new receive proxy. ②

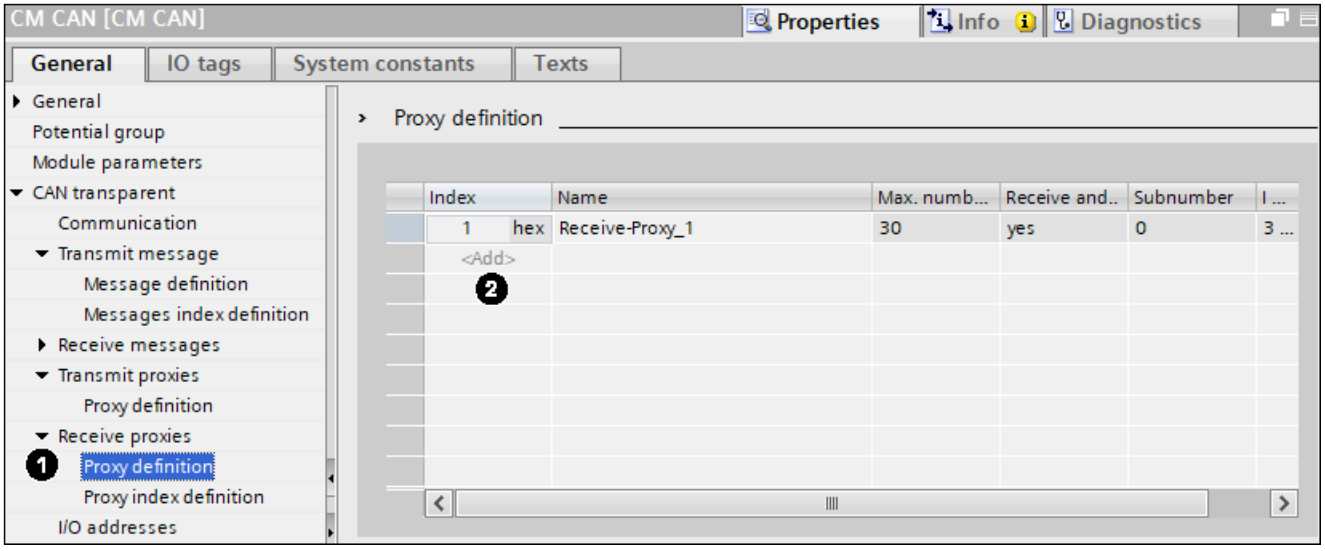


Figure 7-33 Creating and setting receive proxies

For receive proxies, you can specify the maximum number of messages the receive buffer can accommodate. You can use a message filter to define which messages are to be processed.

1. In the "CAN transparent menu", open the "Receive proxies" and select the "Proxy index definition" ①.
2. Define the maximum number of messages the receive buffer should hold. ②

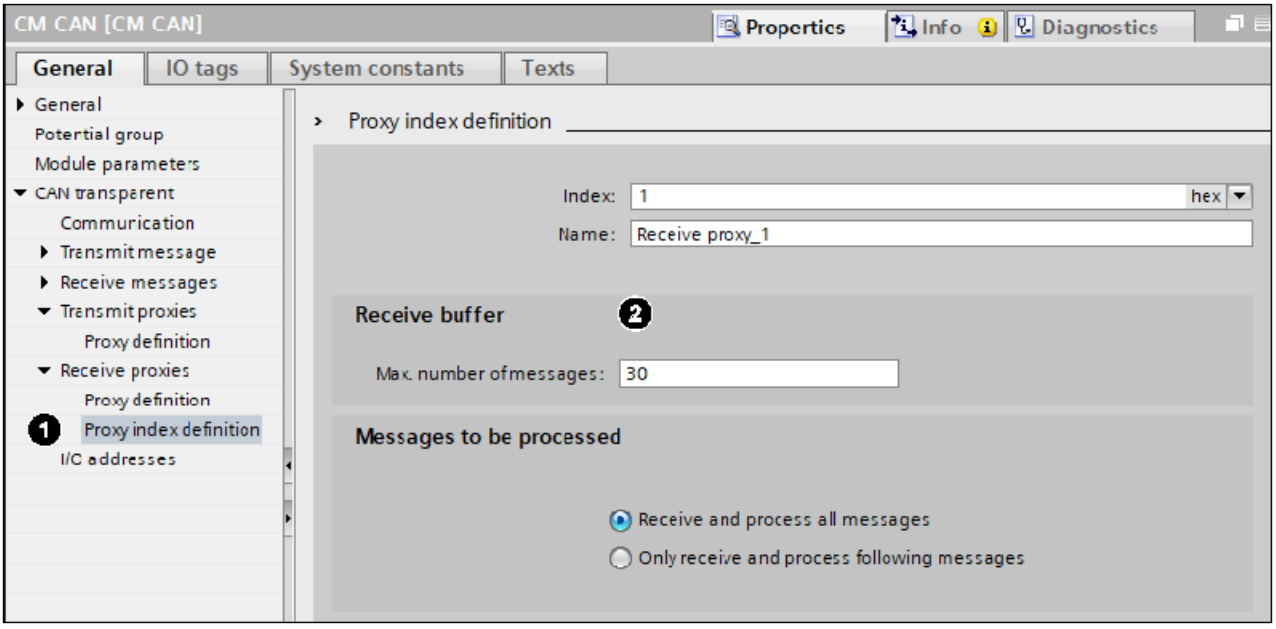


Figure 7-34 Set proxy index definition

Adapting I/O addresses

The input and output addresses of the communication module use an address space of 1-256 bytes. The I/O addresses are automatically assigned for each communication module when the device configuration is specified in TIA Portal.

Checking data consistency

You can check the consistency of the assignments for the receive data and transmit data as well as the data types used with a compilation.

Setting CAN communication to "ON" via S7 user program

To allow transmission of data between the SIMATIC S7 controller and the module, you must set a control bit to "1" with the S7 user program. This control bit is transmitted cyclically from the SIMATIC S7 controller to the module as part of the IO data.

Programming

8

8.1 PLC tags

The PLC tags are automatically created by the HSP when the module is configured in the TIA Portal. You can use the PLC tags in the user program to access the cyclic I/O image. This means the parts of the I/O image allocated to the module can be accessed from the user program using logical names. The use of explicit addresses in the user program is therefore not required.

Note

To improve the performance, switch off the "Automatic generation of PLC tags for I/O objects" when creating large configurations.

When configuration is complete, switch on the "Automatic generation of PLC tags for I/O objects" to generate the PLC tags. This process applies to all ports.

The automatically generated PLC tags of the communication module (all modes) are not deleted:

- When the copied set (IM155 and CM CAN) is switched off by the subnet or I/O system
- When the copied set (IM155 and CM CAN) is deleted

You must delete the tags manually.

The mechanism of the tag generation does not reflect switch-off of the interface module from PROFINET. You must delete the tags manually.

Naming scheme of the created PLC tags

Table 8- 1 Naming scheme of the created PLC tags

Module/ submodule		**IO data**	**Naming scheme**	**Data type**	**S7 address Mnemonics**
CANopen manager CANopen slave			**Pre-fix:**<IO_DeviceName>.<ModuleName>.<OperatingMode>		
		Status	<*Prefix*>.Status	Byte	%IB<*ByteAdr*>
		Control	<*Prefix*>.Control	Byte	%QB<*ByteAdr*>
	Transmit data object dictionary		**Prefix**: <IO_DeviceName>.<ModuleName>.<OperatingMode>.<TransmitDataObjectDictionaryName>		
		Unsigned 08	<*Prefix*>.<Subindex name>	USInt	%QB<*ByteAdr*>
		Unsigned 16	<*Prefix*>.<Subindex name>	UInt	%QW<*ByteAdr*>
		Unsigned 32	<*Prefix*>.<Subindex name>	UDInt	%QD<*ByteAdr*>
		Unsigned 64 *	<*Prefix*>.<Subindex name>	ULInt	%Q<*ByteAdr*>.0
		Integer 08	<*Prefix*>.<Subindex name>	SInt	%QB<*ByteAdr*>

Module/ submodule		**IO data**	**Naming scheme**	**Data type**	**S7 address Mnemonics**
		Integer 16	<*Prefix*>.<Subindex name>	Int	%QW<*ByteAdr*>
		Integer 32	<*Prefix*>.<Subindex name>	DInt	%QD<*ByteAdr*>
		Integer 64 *	<*Prefix*>.<Subindex name>	LInt	%Q<*ByteAdr*>.0
		Real 32	<*Prefix*>.<Subindex name>	Real	%QD<*ByteAdr*>
		Real 64 *	<*Prefix*>.<Subindex name>	LReal	%Q<*ByteAdr*>.0
		Bool	<*Prefix*>.<Subindex name>	Bool	%Q<*ByteAdr*>.0
	Receive data object dictionary		**Prefix**: <IO_DeviceName>.<ModuleName>.<OperatingMode>.<ReceiveDataObjectDictionaryName>		
		Unsigned 08	<*Prefix*>.<Subindex name>	USInt	%IB<*ByteAdr*>
		Unsigned 16	<*Prefix*>.<Subindex name>	UInt	%IW<*ByteAdr*>
		Unsigned 32	<*Prefix*>.<Subindex name>	UDInt	%ID<*ByteAdr*>
		Unsigned 64 *	<*Prefix*>.<Subindex name>	ULInt	%I<*ByteAdr*>.0
		Integer 08	<*Prefix*>.<Subindex name>	SInt	%IB<*ByteAdr*>
		Integer 16	<*Prefix*>.<Subindex name>	Int	%IW<*ByteAdr*>
		Integer 32	<*Prefix*>.<Subindex name>	DInt	%ID<*ByteAdr*>
		Integer 64 *	<*Prefix*>.<Subindex name>	LInt	%I<*ByteAdr*>.0
		Real 32	<*Prefix*>.<Subindex name>	Real	%ID<*ByteAdr*>
		Real 64 *	<*Prefix*>.<Subindex name>	LReal	%I<*ByteAdr*>.0
		Bool	<*Prefix*>.<Subindex name>	Bool	%I<*ByteAdr*>.0
CAN transparent			**Prefix:** <IO_DeviceName>.<ModuleName>.<OperatingMode>		
		Status	<*Prefix*>.Status	Byte	%IB<*ByteAdr*>
		Control	<*Prefix*>.Control	Byte	%QB<*ByteAdr*>
	Transmit message		**Prefix**: <IO_DeviceName>.<ModuleNamee>.<OperatingMode>.<TransmitMessageName>		
		Unsigned 08	<*Prefix*>.<Parameter name>	USInt	%QB<*ByteAdr*>
		Unsigned 16	<*Prefix*>.<Parameter name>	UInt	%QW<*ByteAdr*>
		Unsigned 32	<*Prefix*>.<Parameter name>	UDInt	%QD<*ByteAdr*>
		Unsigned 64 *	<*Prefix*>.<Parameter name>	ULInt	%Q<*ByteAdr*>.0
		Integer 08	<*Prefix*>.<Parameter name>	SInt	%QB<*ByteAdr*>
		Integer 16	<*Prefix*>.<Parameter name>	Int	%QW<*ByteAdr*>
		Integer 32	<*Prefix*>.<Parameter name>	DInt	%QD<*ByteAdr*>
		Integer 64 *	<*Prefix*>.<Parameter name>	LInt	%Q<*ByteAdr*>.0
		Real 32	<*Prefix*>.<Parameter name>	Real	%QD<*ByteAdr*>
		Real 64 *	<*Prefix*>.<Parameter name>	LReal	%Q<*ByteAdr*>.0
		Bool	<*Prefix*>.<Parameter name>	Bool	%Q<*ByteAdr*>.0
	Receive message		**Prefix**: <IO_DeviceName>.<ModuleName>.<OperatingMode>.<ReceiveMessageName>		
		Unsigned 08	<*Prefix*>.<Parameter name>	USInt	%IB<*ByteAdr*>
		Unsigned 16	<*Prefix*>.<Parameter name>	UInt	%IW<*ByteAdr*>
		Unsigned 32	<*Prefix*>.<Parameter name>	UDInt	%ID<*ByteAdr*>
		Unsigned 64 *	<*Prefix*>.<Parameter name>	ULInt	%I<*ByteAdr*>.0
		Integer 08	<*Prefix*>.<Parameter name>	SInt	%IB<*ByteAdr*>
		Integer 16	<*Prefix*>.<Parameter name>	Int	%IW<*ByteAdr*>

Module/ submodule		IO data	Naming scheme	Data type	S7 address Mnemonics
		Integer 32	<*Prefix*>.<Parameter name>	DInt	%ID<*ByteAdr*>
		Integer 64 *	<*Prefix*>.<Parameter name>	LInt	%I<*ByteAdr*>.0
		Real 32	<*Prefix*>.<Parameter name>	Real	%ID<*ByteAdr*>
		Real 64 *	<*Prefix*>.<Parameter name>	LReal	%I<*ByteAdr*>.0
		Bool	<*Prefix*>.<Parameter name>	Bool	%I<*ByteAdr*>.0

I/Q: Input/output bit address

IB/QB: Input/output byte address

ID/QD: Input/output double word address

***** The S7-1200 CPUs do not support 64-bit data types, so the generated tags cannot be used for the S7-1200.

Function block

A special function block was developed to assign parameters to the CM CAN. Install the supplied HSP and the additional TIA Portal library (function block).

The HSP generates a data block with the configuration of the module. The function block then transfers the configuration in the runtime process.

The configuration is transferred as follows:

- Automatically, if the status byte is used as input parameter "CAN_STATE" of the function block.
- Manually, if you set the "REQ" parameter of the function block to "1". The parameter is then set to "0".

Drag the corresponding function block from the global library to your program blocks according to the selected CPU:

- CPUs 1500: ET200SPCM_CANConfig
- CPUs 1200: ET200SPCM_CANConfig_1200
- Open Controller: ET200SPCM_CANConfig_1515_PC

The function block appears as ET200SP_CAMConfig[FB1500].

Drag the corresponding function block (ET200SPCM_CANConfig) from the program blocks into your organization blocks.

Enter the parameters as follows:

DB_NO: Configuration data block of the CAN communication module generated by the HSP

CAN_STATE: Generated PLC tag of the status byte information

REQ: Trigger of manual download

ABORT: Cancelation of download

Note

Create an independent instance of the function block that transfers the configuration for each communication module in the module.

Function description

The transfer is started with REQ=TRUE. All data records included in the "DB_NO" configuration block are written one after the other. BUSY is set to TRUE during execution of the command.

If all data records are transferred without an error, the message DONE=TRUE appears.

If a data record cannot be transferred due to a temporary error, the request is automatically repeated.

If a data record cannot be transferred due to a static error, execution of the command is stopped and displayed in ERROR and STATUS.

After the instruction is started, the bit for the activation of the REQ instruction is automatically reset. The transfer of the current data is started with ABORT=TRUE.

If the status byte of the communication module is placed in the CAN_STATE parameter, the data records for an unconfigured module are automatically written.

Send new hardware configuration

You need to send a new hardware configuration if:

- The extended parameter assignment fails. The function block shows ERROR=TRUE
- The status byte of the communication module indicates "bad parameterization" for bit setting in the module status.

You send a new hardware configuration as follows:

1. Right-click "PLC".
2. Select "Compile"
3. Click "Hardware (rebuild all)"
4. Select "Download to device"
5. Click "Software (all)"
6. Select "Download to device"
7. Click "Hardware configuration"

After successful sending of the hardware configuration, the module can receive the extended parameter assignment again. The status byte of the module shows "not configured" for bit setting in the "Module status".

Parameters

The following table shows the parameters:

Parameters	Information	Data type	Memory area	Description
DB_NO	Input	Variant	D	Data block with data records that are to be written (symbolic name or "%DBxy")
CAN_STATE	Input	Byte	I, Q, M, D, L	Status byte of the CAN module
REQ	InOut	Bool	I, Q, M, D, L	TRUE: Start of the data transfer
ABORT	InOut	Bool	I, Q, M, D, L	TRUE: Data transfer interrupted
BUSY	Output	Bool	I, Q, M, D, L	TRUE: Data transfer active
DONE	Output	Bool	I, Q, M, D, L	TRUE: All data records are transferred
ERROR	Output	Bool	I, Q, M, D, L	TRUE: At least one data record was not transferred
STATUS	Output	DWORD	I, Q, M, D, L	Information about an error

Status parameters

The following table shows the status parameters:

Error code	Description
0000 8001	The instruction is already running.
0000 8002	The block in parameter DB_NO is not a global data block. You enter a global data block with a symbolic name or "%DBxy".
0000 8003	Configure a data block number in parameter DB_NO.
0000 8004	The data block is too short.
0000 8005	There are too few data records in the data block. There needs to be at least one data record.
0000 8006	The data record is too large. Maximum size of a data record is 4096 bytes.
DF80 B500	Extended parameter assignment is ignored because the module was not in "not configured" state.
DF80 B800	Incorrect parameter assignment data was received.
DF80 B600	The data record was not accepted.

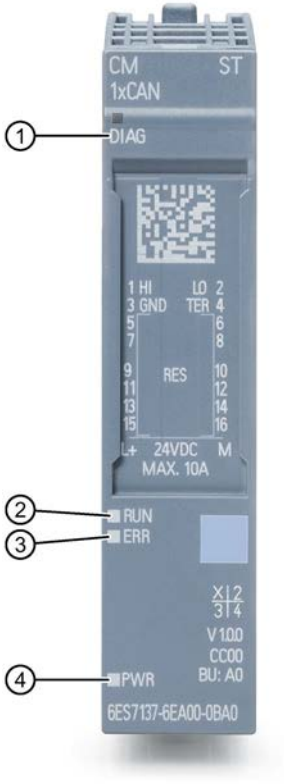
Interrupts/diagnostics alarms

9

9.1 Status and error display

LED display

The figure below shows the LED displays (status and error displays) of the CM CAN.



① DIAG LED (green/red)
② RUN LED (green)
③ ERROR LED (red)
④ PWR LED (green)

Figure 9-1 SIMATIC ET 200SP CM CAN

Meaning of the LED displays

The following tables show the meaning of the status and error displays. Measures for dealing with diagnostics alarms can be found in the section Diagnostics alarms (Page 124).

DIAG LED

Table 9- 1 Behavior of the DIAG LED

DIAG LED	Meaning
Off	• Supply voltage of the ET 200SP via the backplane bus is not present.
Flashes	• Module parameters are not assigned.
On	• Module parameters are assigned and there are no diagnostics.
Flashes	• Module parameters are assigned and there are diagnostics.

RUN LED

Table 9- 2 Behavior of the RUN LED

RUN LED	Meaning	Note
Off	• The module is in the initialization phase.	After power-on of the device or after reset
Flashes	• The NMT state of the module is "Pre-Operational".	-
One-time brief flashing	• The NMT state of the module is "Stopped".	-
On	• The NMT state of the module is "Operational".	-

ERR LED

Table 9- 3 Behavior of the ERR LED

ERR LED	Meaning	Note
▫ Off	• No error on the CANopen bus	-
☼ One-time brief flashing	• At least one error counter in the CAN controller has reached its warning threshold.	-
☼ Two-time brief flashing	• A Heartbeat or Node Guarding error has occurred.	-
■ On	• The CAN controller is in "Bus-Off" state.	-

PWR LED

Table 9- 4 Behavior of the PWR LED

PWR LED	Meaning
▫ Off	• Supply voltage L+ is too low or missing. • Invalid BaseUnit (A1) used
■ On	• Supply voltage L+ present

9.2 Interrupts

Diagnostics

The module sends diagnostic information to the SIMATIC S7 controller.

A distinction is made here between diagnostic messages that refer to the entire ET 200SP CM CAN communication module and those that refer to individual modules.

Many diagnostic messages are the result of error events and not error states. Therefore, the diagnostics are removed immediately after they have been reported to the SIMATIC S7 controller ("Incoming" message and "Outgoing" message).

The tables below provide an overview of the following for the various modes of the module:

- The events that trigger a diagnostic message
- The events that cause a pending diagnostic message to be withdrawn
- The associated component

The diagnostic messages including description are listed in the section Diagnostics alarms (Page 124). The descriptions also include instructions and information on error correction.

Diagnostics-relevant events in "CANopen manager" operating mode

Table 9- 5 Diagnostics-relevant events in "CANopen manager" operating mode

Event for triggering	Event for withdrawal	Reference	Additional information*
Transition of the CAN controller to "Bus Off" state	CAN controller exits "Bus Off" state again	Module	---
Transition of the CAN controller to "Error Passive" state	CAN controller exits "Error Passive" state again	Module	---
Internal communication error	Diagnostic information was transmitted to the SIMATIC S7 controller	Module	---
Buffer overflow when sending CAN packets	Diagnostic information was transmitted to the SIMATIC S7 controller	Module	---
Buffer overflow when receiving CAN packets	Diagnostic information was transmitted to the SIMATIC S7 controller	Module	---
Receipt of PDO with incorrect length	Diagnostic information was transmitted to the SIMATIC S7 controller	Module	COB ID of the PDO
Heartbeat / Node Guarding error	When the slave is available again and in the expected state	Corresponding slave module	---
Unexpected slave state	When the slave is in the expected state again	Corresponding slave module	---

Event for triggering	Event for withdrawal	Reference	Additional information*
Error during boot-up of a slave	When the slave was booted successfully	Corresponding slave module	Cause of error, if known
Receipt of an EMCY message	Diagnostic information was transmitted to the SIMATIC S7 controller	Corresponding slave module	Error codes of the EMCY message

* Additional information can only be read with the special data record, as described in the section Advanced diagnostics alarms (Page 127).

Diagnostics-relevant events in "CANopen slave" operating mode

Table 9- 6 Diagnostics-relevant events in "CANopen slave" operating mode

Event for triggering	Event for withdrawal	Reference	Additional information
Transition of the CAN controller to "Bus Off" state	CAN controller exits "Bus Off" state again	Module	---
Transition of the CAN controller to "Error Passive" state	CAN controller exits "Error Passive" state again	Module	---
Internal communication error	Diagnostic information was transmitted to the SIMATIC S7 controller	Module	---
Buffer overflow when sending CAN packets	Diagnostic information was transmitted to the SIMATIC S7 controller	Module	---
Buffer overflow when receiving CAN packets	Diagnostic information was transmitted to the SIMATIC S7 controller	Module	---
Receipt of PDO with incorrect length	Diagnostic information was transmitted to the SIMATIC S7 controller	Module	COB ID of the PDO
Heartbeat / Node Guarding error	Diagnostic information was transmitted to the SIMATIC S7 controller	Module	ID of failed node

Diagnostic-relevant events in "CAN transparent" mode

Table 9- 7 Diagnostic-relevant events in "CAN transparent" mode

Event for triggering	Event for withdrawal	Reference
Transition of the CAN controller to "Bus Off" state	CAN controller exits "Bus Off" state again	Module
Transition of the CAN controller to "Error Passive" state	CAN controller exits "Error Passive" state again	Module
Internal communication error	Diagnostic information was transmitted to the SIMATIC S7 controller	Module
Buffer overflow when sending CAN packets	Diagnostic information was transmitted to the SIMATIC S7 controller	Module
Buffer overflow when receiving CAN packets	Diagnostic information was transmitted to the SIMATIC S7 controller	Module
Receipt of a CAN packet with incorrect length	Diagnostic information was transmitted to the SIMATIC S7 controller	Corresponding input module

9.3 Diagnostics alarms

Diagnostics

ID	Channel diagnostics error	Cause of error and possible solution	Mode
1232, 0x4D0	CAN in "Bus Off" mode	No more CAN messages can be sent or received. Check for errors in the hardware installation (e.g. terminating resistor, etc.) or communication settings (e.g. transmission rate, etc.).	Manager, Slave, Transparent
1233, 0x4D1	CAN in "Error Passive" mode	In this state, the device can no longer initiate repetition of defective CAN messages. Check for errors in the hardware installation (e.g. terminating resistor, etc.) or communication settings (e.g. transmission rate, etc.). The error may also occur if no other devices are connected to the CAN bus.	Manager, Slave, Transparent
1234, 0x4D2	CAN receive buffer overflow	Received CAN messages were lost. Reduce the CAN transmission rate, increase the PROFINET cycle time or reduce the PROFINET IO data to be transmitted.	Manager, Slave, Transparent
1235, 0x4D3	CAN transmit buffer overflow	The CAN messages to be transmitted could not be sent and were discarded because: • CAN communication volume too high • Further problems in CAN communication ("Bus Off" or "Error Passive" state). If neither "Bus off" nor "Error passive" state is present, try reducing the amount of communication traffic or increasing the CAN bit rate.	Manager, Slave, Transparent
1236, 0x4D4	Received PDO with wrong length	The received PDO has a length that differs from the one configured and will therefore be discarded. Correct the configuration of the PDOs.	Manager, Slave
1237, 0x4D5	Heartbeat error	No heartbeat message was received from node within the specified time. The error reaction configured in OD 1029 will be performed. Check whether: • Whether the node has failed • Whether the monitoring time is correctly configured.	Slave
1238, 0x4D6	Node Guarding error	No Node Guarding request was received from the manager within the specified time. The error reaction configured in OD 1029 will be performed. Check whether: • Whether the master has failed • Whether the monitoring time is correctly configured.	Slave

ID	Channel diagnostics error	Cause of error and possible solution	Mode
1239, 0x4D7	Heartbeat or Node Guarding error	No heartbeat message has been received from the node within the specified time or the node has not responded to a Node Guarding request within the set monitoring time. If the message occurred during startup, the start procedure is repeated for all nodes. In all other cases the reaction configured under "Reaction to failure of node" is executed.	Manager
1240, 0x4D8	Incorrect NMT status	The node reports an unexpected network status (NMT status). The reported NMT status can be read from the advanced diagnostics data record. If the message occurred during startup, the start procedure is repeated for all nodes. In all other cases the reaction configured under "Reaction to failure of node" is executed.	Manager
1241, 0x4D9	Error when booting	Errors were detected when starting the node. Details are output in one or more additional diagnostic messages. Depending on the setting "Node is required in the network", the node or all nodes are restarted.	Manager
1242, 0x4DA	Booting: Node is not responding	The node has not responded to a request within 30 seconds (OD 1000). Check if the node is correctly connected to the CAN bus and is not in the "Stopped" state. The start operation is continued.	Manager
1243, 0x4DB	Booting	Additional information on diagnostic message "Error when booting". Read the information from the advanced diagnostics data record.	Manager
1244, 0x4DC	Receive EMCY message	An EMCY message was received from the node. The emergency error code and the reported value of the error register can be read from the advanced diagnostics data record. Note: If too many EMCY messages are received in too short a time, not all messages are displayed in the diagnostic message.	Manager
1245, 0x4DD	Received message has incorrect length	The received CAN message has a different length than the configured length and is therefore discarded. Correct the configured length.	Transparent
1246, 0x4DE	Error in extended parameter assignment	An error occurred when executing the extended parameter assignment with a function block. Check whether the device configuration is valid.	Manager, Slave, Transparent
9, 0x009	Faulty module	The SSI communication is missing, but not the external power supply. STM32 seems to be faulty. Remedy: Switch off the power supply and replace the module.	Manager, Slave, Transparent

ID	Channel diagnostics error	Cause of error and possible solution	Mode
16, 0x010	Invalid parameter assignment	The module has detected a parameter assignment error. Parameter assignment errors are: -The module cannot evaluate any parameters, e.g. due to unknown parameters, invalid parameter combination. -The module does not yet have configured parameters. -The user calibration does not match the parameter assignment. -Calibration error Remedy: Check and correct the parameters. Then load the parameters into the module.	Manager, Slave, Transparent
17, 0x011	Missing supply voltage	No encoder or supply voltage present. Remedy: Check the encoder connection or the supply voltage connection. Correct the wrong connection. Ensure that the power supply is switched on.	Manager, Slave, Transparent
22, 0x016	Hardware interrupt lost	The module signals the loss of the hardware interrupt. The hardware interrupt cannot be signaled because the previous hardware interrupt was not acknowledged. An error may occur in the configuration. Remedy: Change the alarm behavior in the CPU. If needed, change the parameter settings of the module. Note: The error can be corrected, for example, by restarting the module (the module receives the new parameters from the restart).	Manager, Slave, Transparent
31, 0x01F	Firmware update	The channel is temporarily not available. Wait until the firmware update is finished.	Manager, Slave, Transparent

9.4 Advanced diagnostics alarms

Advanced diagnostics alarms

Advanced diagnostics data record

Since the ET 200SP module cannot write the advanced diagnostics, a workaround has been created. A special advanced diagnostics data record has been created so that the user receives the advanced diagnostics data. When the advanced diagnostics is signaled, the user automatically sees only the normal diagnostics ID and its description. To get the advanced diagnostics ID and the advanced diagnostics data, the user must manually read the advanced diagnostics data record.

The user reads the advanced diagnostics alarms with RDREC no. 0x212.

The structure of the receive data is as follows:

Table 9- 8 Structure of the receive data

Item no.	Data record user data
Element [0]	2 bytes: Normal diagnostic ID
	2 bytes: Advanced diagnostics ID
	4 bytes: Advanced diagnostics data
Element [1]	2 bytes: Normal diagnostic ID
	2 bytes: Advanced diagnostics ID
	4 bytes: Advanced diagnostics data
Element [2]	2 bytes: Normal diagnostic ID
	2 bytes: Advanced diagnostics ID
	4 bytes: Advanced diagnostics data
Up to element [31]	2 bytes: Normal diagnostic ID
	2 bytes: Advanced diagnostics ID
	4 bytes: Advanced diagnostics data

Each advanced diagnostics requires 8 bytes. The data record contains a maximum of 32 advanced diagnostics, which the user can read simultaneously.

If a higher number of advanced diagnostics is signaled at the same time, only the first 32 diagnostic items are written to the data record.

Additional diagnostics

Table 9- 9 Definition of the advanced diagnostics alarms

Diag-nostics ID	Advanced diagnostics ID				Structure of the advanced diagnostics data
	Data type	Bit	Value	Meaning	
1236, 0x4D4	Unsigned 16-bit integer:	15 ... 0	Must be "1"	Received PDO with wrong length	Unsigned 32-bit integer: COB ID (the COB ID of the PDO with the wrong length)
1237, 0x4D5	Unsigned 16-bit integer:	15 ... 0	Must be "1"	Heartbeat error	Unsigned 32-bit integer: Node ID (the ID of the node in which the error occurred)
1238, 0x4D6	Unsigned 16-bit integer:	15 ... 0	Must be "1"	Node Guarding error	Unsigned 32-bit integer: Node ID (the ID of the node in which the error occurred)
1239, 0x4D7	Unsigned 16-bit integer:	15 ... 0	Must be "1"	Heartbeat or Node Guarding error	Unsigned 32-bit integer: Node ID (the ID of the node in which the error occurred)
1240, 0x4D8	Unsigned 16-bit integer:	15 ... 0	Must be "1"	Incorrect NMT status	Unsigned 32-bit integer: 31 ... 24: NMT status (the NMT status reported by the node) 23 ... 16: Node ID (the ID of the node in which the error occurred) 15 ... 0: Reserved
1241, 0x4D9	Unsigned 16-bit integer:	15 ... 0	Must be "1"	Error when booting	Unsigned 32-bit integer: Node ID (the ID of the node in which the error occurred)
1242, 0x4DA	Unsigned 16-bit integer:	15 ... 0	Must be "1"	Booting: Node is not responding	Unsigned 32-bit integer: Node ID (the ID of the node in which the error occurred)
1243, 0x4DB	Unsigned 16-bit integer:	7 ... 0	Must be "1"	Booting: Other error	Unsigned 32-bit integer: Additional error code
		15 ... 8	1 ... 127	Node ID	
1243, 0x4DB	Unsigned 16-bit integer:	7 ... 0	Must be "2"	Booting: Heartbeat / Guarding - no reaction	Unsigned 32-bit integer: Reserved
		15 ... 8	1 ... 127	Node ID	
1243, 0x4DB	Unsigned 16-bit integer:	7 ... 0	Must be "3"	Booting: Different device type ID	Unsigned 32-bit integer: Reserved
		15 ... 8	1 ... 127	Node ID	
1243, 0x4DB	Unsigned 16-bit integer:	7 ... 0	Must be "4"	Booting: Different vendor ID	Unsigned 32-bit integer: Reserved
		15 ... 8	1 ... 127	Node ID	
1243, 0x4DB	Unsigned 16-bit integer:	7 ... 0	Must be "5"	Booting: Different product code	Unsigned 32-bit integer: Reserved
		15 ... 8	1 ... 127	Node ID	
1243, 0x4DB	Unsigned 16-bit integer:	7 ... 0	Must be "6"	Booting: Deviating revision number	Unsigned 32-bit integer: Reserved
		15 ... 8	1 ... 127	Node ID	
1243, 0x4DB	Unsigned 16-bit integer:	7 ... 0	Must be "7"	Booting: Different serial number	Unsigned 32-bit integer: Reserved
		15 ... 8	1 ... 127	Node ID	

Diag-	Advanced diagnostics ID				Structure of the advanced diagnostics data
1243, 0x4DB	Unsigned 16-bit integer:	7 ... 0	Must be "8"	Booting: SDO abort for OD	Unsigned 32-bit integer: Reserved
		15 ... 8	1 ... 127	Node ID	
1243, 0x4DB	Unsigned 16-bit integer:	7 ... 0	Must be "9"	Booting: SDO timeout for OD	Unsigned 32-bit integer: Reserved
		15 ... 8	1 ... 127	Node ID	
1244, 0x4DC	Unsigned 16-bit integer:	15 ... 0	Must be "1"	Receive EMCY message	Unsigned 32-bit integer: 31 ... 16: EMCY code 15 ... 8: Error register 7 ... 0: The first byte of the manufacturer error
1245, 0x4DD	Unsigned 16-bit integer:	15 ... 0	Must be "1"	Received message has incorrect length	Unsigned 32-bit integer: Wrong length

Technical specifications 10

10.1 Technical specifications

Technical specifications of the CAN communications module

The following table shows the technical specifications as of 02/2020. You can find a data sheet including daily updated technical specifications on the Internet (https://support.industry.siemens.com/cs/de/en/pv/6ES7137-6EA00-0BA0/td?dl=en).

Article number	6ES7137-6EA00-0BA0
General information	
Product type designation	CM 1x CAN ST
Firmware version	V1.0.0
• FW update possible	Yes
usable BaseUnits	BU type A0
Color code for module-specific color identification plate	CC00
Product function	
• I&M data	Yes; I&M0 to I&M3
• Module swapping during operation (hot swapping)	Yes
• Isochronous mode	No
Engineering with	
• STEP 7 TIA Portal configurable/integrated from version	STEP 7 V15.1 or higher
Supply voltage	
Rated value (DC)	24 V
permissible range, lower limit (DC)	19.2 V
permissible range, upper limit (DC)	28.8 V
Reverse polarity protection	Yes
Input current	
Current consumption, typ.	20 mA
Current consumption, max.	25 mA
Power loss	
Power loss, typ.	0.5 W
Address area	
Address space per module	
• Address space per module, max.	256 byte
1. Interface	
Interface type	CAN according to CiA 303-1
Isolated	Yes; 500 V AC or 707 V DC

Article number	**6ES7137-6EA00-0BA0**
Interface types	
• Number of ports	1
• Design of the connection	Push-in terminal
CAN	
• CAN operating modes	CAN Standard CAN 2.0A/B; CANopen Manager / Slave acc. to CiA
• Specification acc. to CiA	CiA 301 & CiA 302
• Transmission rate, min.	10 kbit/s
• Transmission rate, max.	1 000 kbit/s
• Number of slaves, max.	60
• Number of SDOs in parallel	16; Parallel
• Number of PDOs	128; Send / receive
Services	
– Node/life-guarding	Yes
– Heartbeat	Yes
– SYNC	Yes
Interrupts/diagnostics/status information	
Alarms	Yes
Diagnostics function	Yes
Diagnostics indication LED	
• RUN LED	Yes
• ERROR LED	Yes
• MAINT LED	No
• Monitoring of the supply voltage (PWR-LED)	Yes; green PWR LED
Potential separation	
between backplane bus and interface	Yes
Isolation	
Isolation tested with	707 V DC (type test)
Standards, approvals, certificates	
CE mark	Yes
UL approval	Yes
RCM (formerly C-TICK)	Yes
KC approval	Yes; Reg. No.: R-R-S49-ET200SPCMCAN
EAC (formerly Gost-R)	Yes
RoHS conformity	Yes
Ambient conditions	
Ambient temperature during operation	
• horizontal installation, min.	-30 °C
• horizontal installation, max.	60 °C
• vertical installation, min.	-30 °C

Article number	**6ES7137-6EA00-0BA0**
• vertical installation, max.	50 °C
• ceiling installation, min.	-30 °C
• ceiling installation, max.	50 °C
• floor installation, min.	-30 °C
• floor installation, max.	50 °C
Altitude during operation relating to sea level	
• Installation altitude above sea level, max.	5 000 m
Decentralized operation	
to SIMATIC S7-300	No
to SIMATIC S7-400	No
to SIMATIC S7-1200	Yes
to SIMATIC S7-1500	Yes
Dimensions	
Width	15 mm
Height	73 mm
Depth	58 mm
Weights	
Weight, approx.	32 g

Dimension drawing

See Equipment Manual ET 200SP BaseUnits
(https://support.industry.siemens.com/cs/ww/en/view/59753521)