User Manual
CM CANopen
CANopen Module for SIMATIC S7-1200

Doc.Id. HMSI-216-117
Rev. 1.00

Connecting Devices™

HALMSTAD • CHICAGO • KARLSRUHE • TOKYO • BEIJING • MILANO • MULHOUSE • COVENTRY • PUNE • COPENHAGEN

HMS Industrial Networks
Mailing address: Box 4126, 300 04 Halmstad, Sweden
Visiting address: Stationsgatan 37, Halmstad, Sweden

E-mail: info@hms-networks.com
Web: www.anybus.com

Important User Information

This document is intended to provide a good understanding of the functionality offered by the CM CANopen Module for SIMATIC S7-1200.

The reader of this document is expected to be familiar with high level software design, and communication systems in general. The use of advanced CANopen-specific functionality may require in-depth knowledge in CANopen networking internals and/or information from the official CANopen specifications. In such cases, the people responsible for the implementation of this product should either obtain the CANopen specification to gain sufficient knowledge or limit their implementation in such a way that this is not necessary.

Liability

Every care has been taken in the preparation of this manual. Please inform HMS Industrial Networks AB of any inaccuracies or omissions. The data and illustrations found in this document are not binding. We, HMS Industrial Networks AB, reserve the right to modify our products in line with our policy of continuous product development. The information in this document is subject to change without notice and should not be considered as a commitment by HMS Industrial Networks AB. HMS Industrial Networks AB assumes no responsibility for any errors that may appear in this document.

There are many applications of this product. Those responsible for the use of this device must ensure that all the necessary steps have been taken to verify that the applications meets all performance and safety requirements including any applicable laws, regulations, codes, and standards

HMS Industrial Networks AB will under no circumstances assume liability or responsibility for any problems that may arise as a result from the use of undocumented features, timing, or functional side effects found outside the documented scope of this product. The effects caused by any direct or indirect use of such aspects of the product are undefined, and may include e.g. compatibility issues and stability issues.

The examples and illustrations in this document are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular implementation, HMS Industrial Networks AB cannot assume responsibility for actual use based on these examples and illustrations.

Intellectual Property Rights

HMS Industrial Networks AB has intellectual property rights relating to technology embodied in the product described in this document. These intellectual property rights may include patents and pending patent applications in the US and other countries.

Trademark Acknowledgements

Anybus ® is a registered trademark of HMS Industrial Networks AB.

SIMATIC®, S7-1200® and TIA® are registered trademarks of Siemens AG.

All other trademarks are the property of their respective holders.

Warning:	This is a class A product. in a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.
ESD Note:	This product contains ESD (Electrostatic Discharge) sensitive parts that may be damaged if ESD control procedures are not followed. Static control precautions are required when handling the product. Failure to observe this may cause damage to the product.

Sep 2013 Doc Id HMSI-216-117

Table of Contents

Chapter 7 Supported Objects

Chapter 8 Interface Functions

Appendix A Technical Specification

Appendix B Status LED Timing Diagrams

Appendix C CANopen Emergency Codes

Appendix D Error Codes (RET)

P. About This Document

P.1 How To Use This Document

This document contains a general introduction as well as a description of the technical features provided by the CM CANopen Module for SIMATIC S7-1200, including configuration of the device.

The reader of this document is expected to be familiar with PLC and software design, as well as with communication systems in general. The reader is also expected to be familiar with the Microsoft Windows operating system.

P.2 Related Documents

Document name	Author
CiA 301 rev. 4.2	CAN in Automation
CiA 302 Draft Standard Proposal 302 rev. 4.1	CAN in Automation
CiA 303-1 rev. 1.7	CAN in Automation
CiA 303-3 rev. 1.3	CAN in Automation
TIA Portal configuration tool user manual	

Please visit the support pages at the HMS Industrial Networks web site at www.hms-networks.com/can-for-S7-1200/ for more documents

P.3 Document History

Summary of Recent Changes (... 1.00)

Change	Page(s)

Revision List

Revision	Date	Author	Chapter	Description
1.00	2013-09-26	KeL	All	First revision

P.4 Conventions & Terminology

The following conventions are used throughout this document:

- Numbered lists provide sequential steps
- Bulleted lists provide information, not procedural steps
- The term 'user' refers to the person or persons responsible for installing the CM CANopen module in a network.
- Hexadecimal values are written in the format NNNNh, where NNNN is the hexadecimal value.
- Decimal values are represented as NNNN where NNNN is the decimal value
- A byte always consists of 8 bits

P.5 Glossary

Term	Meaning
Byte	8 bits
COP	CANopen
User	Person or persons responsible for installing the CM CANopen module
Higher Level Network	CANopen
Network	
Fieldbus	
RO	Read only. A register, parameter or object labeled RO can only be read, not written.
RW	Read/Write. A register, parameter or object labeled RW can both be read and written
EDS	Electronic Data Sheet. A file that describes the properties of a network device, in this case a CANopen device.

P.6 Support

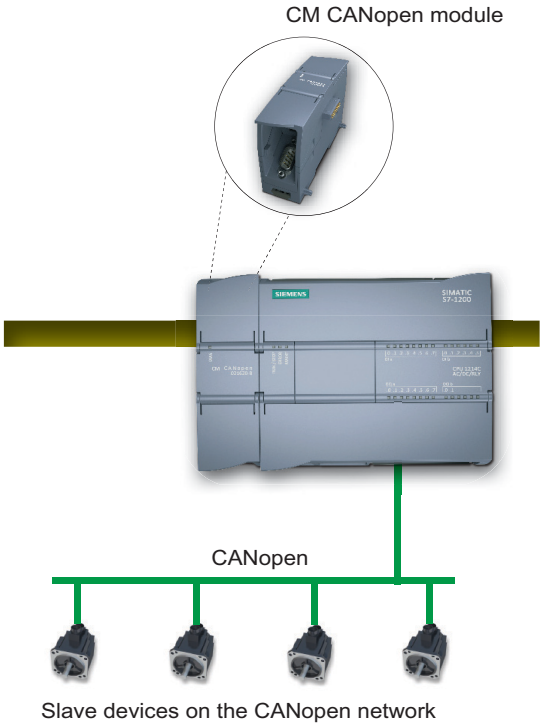
For general contact information and where to find support, please refer to the contact and support pages at www.hms-networks.com.

1. Product Overview

1.1 Introduction

The CM CANopen Module for SIMATIC S7-1200 is a plug-in IP20 compliant communication module designed to be used as an accessory to a SIMATIC S7-1200 PLC. It provides the user with the possibility to connect a CANopen network to a SIMATIC S7-1200 PLC.

The figure below shows a CM CANopen module attached to a SIMATIC S7-1200 PLC to provide the connection from the PLC to CANopen. In this case the PLC is standalone, but it is possible to connect it to a PROFINET network.



The CM CANopen Module for SIMATIC S7-1200 can be configured either as a CANopen manager or as a slave, depending on the application. The network and the module are configured using external configuration tools (TIA Portal and CM CANopen Configuration Studio). See “Configuration” on page 10.

1.2 Features

- Powerful CANopen module for SIMATIC S7-1200 PLC.
- 3 modules per CPU can be connected.
- Connects up to 16 CANopen slave nodes per module in manager mode.
- 256 byte input and 256 byte output per module.
- CANopen slave functionality supported.
- 3 LEDs provide diagnostic information on module, network and I/O status.
- The module can be integrated in the hardware catalogue of TIA Portal.
- CANopen configuration via CM CANopen Configuration Studio (external tool).
- Complies to the CANopen communication profile CiA 301 rev. 4.2 and the CiA 302 Draft Standard Proposal rev. 4.1.
- Transparent CAN supported.

2. CANopen Fieldbus Functionality

The functionality of the CM CANopen Module for SIMATIC S7-1200 is defined by the CANopen CiA 301 rev. 4.2 specification and the CANopen CiA 302 Draft Standard Proposal rev. 4.1. The module can be configured either as a manager or as a slave on the CANopen network.

2.1 Supported Fieldbus Services

Communication and parameters in the CANopen protocol are built around objects. There are different services available to communicate with the objects and to perform other CANopen tasks like supervising the network. The following message types and objects are implemented in the CM CANopen Module for SIMATIC S7-1200:

- NMT (Network Management)[1] messages trigger the NMT state transitions for the slaves and/or perform network and device monitoring, as well as handle errors. If the module is configured as a slave, startup is performed by the manager.
- The CMT (Configuration Manager)[1] configures the CANopen devices. This primarily involves PDO parameters and mapping of information. If the module is configured as a slave, the configuration is performed by the manager.
- PDOs (Process Data Objects) are used for I/O communication. There are 64 Receive PDOs and 64 Transmit PDOs implemented in the CM CANopen Module for SIMATIC S7-1200 that each can transfer up to 8 bytes. Supported PDO message types are event driven (COS or timing), Cyclic Synchronous and Acyclic Synchronous
- SDOs (Service Data Objects) are asynchronous data transmission, and are used to access objects without mapping them to an I/O (PDO) connection. Access is provided to all CANopen objects in the module and in the network nodes (manager mode). SDO messages are used to configure the modules and they can transfer more than 8 bytes, which is the upper limit for a PDO. (Expedited Upload/Download Protocol and Normal (Segmented) Upload/Download Protocol are supported)
- A SYNC (Synchronization Object) is used for synchronizing PDO communication. A manager can be either a producer or a consumer of the synchronization. A slave can only be a consumer.
- The Heartbeat Mechanism helps a device to monitor the status of another node. The module can act either as heartbeat producer or consumer or both at the same time.
- The Node Guarding Protocol provides active surveillance of a slave by the manager. Slaves can be configured to expect a node guarding request from the manager.
- An EMCY (Emergency Object) is used for error reporting when a fault has occurred in the module and for reporting when all faults in a module have been cleared.

1. Only available when the module is configured as manager.

3. Transparent CAN Mode

The CM CANopen Module for SIMATIC S7-1200 integrates Transparent CAN 2.0A (11 bit identifiers). With this protocol, the module allows transmission of any CAN frame from the PLC regardless of the overlaying CAN protocol, and is capable of receiving a number of predefined CAN frames from the CAN network. No other functionality is available.

CAN 2.0A is the underlying protocol of CANopen, directly working on the bus. During configuration in TIA Portal, Transparent CAN can be selected in the Options menu. When this configuration is downloaded to the module, the CANopen functionality is not possible to use. If CANopen functionality is to be reactivated, a new configuration from TIA Portal will have to be downloaded. It is not possible to change between CAN and CANopen at runtime.

Each CAN frame is tagged with an identifier. Only received frames with an identifier specified by the function block CAN_CTRL will be forwarded to the PLC. The data in the frame is then available for the user to interpret. When sending a frame, the user specifies the data and adds a valid identifier before sending it to the module.

To facilitate the use of Transparent CAN, separate PLC function blocks are available[1]. They are described in "Transparent CAN Function Blocks" on page 47. The blocks give the designer of the PLC program the possibility to integrate Transparent CAN functionality (sending, receiving, configuration) into the PLC program, thereby making it possible to implement customer specific CAN protocols.

Note: The receive buffer can hold up to 256 unread received messages

1. The function blocks can be downloaded from www.hms-networks.com/can-for-S7-1200.

4. Installation

The mounting and configuration of the CM CANopen Module for SIMATIC S7-1200 is done following these steps:

1. Mounting

2. Configuring the SIMATIC S7-1200 PLC to use the module (See "SIMATIC S7-1200 PLC Configuration" on page 10)

3. Setting the parameters of the module (See "SIMATIC S7-1200 PLC Configuration" on page 10)

4. If in a CANopen operation mode, configuring the CANopen network, including the module (See "CANopen Network Configuration" on page 12)

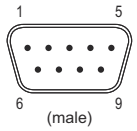
The following items are needed to perform the installation:

- TIA Portal V11, SP2 or later
- HSP (configuration file) for the module[1]
- CM CANopen Configuration Studio
- Function blocks (optional)[1]

4.1 Mounting

The CM CANopen Module for SIMATIC S7-1200 is designed to be connected directly to a SIMATIC S7-1200 PLC.

CANopen connection is provided via a DSUB contact.



No.	Name	Description
2	L	CAN_L, low level in CANopen communication
3, 6	GND	Ground, provides connections to the CANopen network ground
5	SHLD	Shield, provides connection to the CANopen network shield
7	H	CAN_H, high level in CANopen communication
1, 4, 8, 9	N.C.	Not connected[a]

a. To ensure backwards compatibility, these contacts should not be connected.

A new module is configured from scratch, using TIA Portal and CM CANopen Configuration Studio.

See also:

- "CANopen Network Configuration" on page 12

1. Can be downloaded from www.hms-networks.com/can-for-S7-1200

4.2 Status LEDs

Three status LEDs indicate the status of the module as shown in the table below. The DIAG LED is visible on the front of the module and the other two are found behind the lid at the top of the module. The behavior of the LEDs is described in "Status LED Timing Diagrams" on page 56

LED	Indication	Status
DIAG (red/green)	Green	Normal operation
	Blinking green	Hardware address not configured yet
	Blink red	CM bus error state. Note: This LED only report errors on the CM backplane bus. Diagnostic errors are indicated by the error LED on the PLC.
ERR - CANopen error (red)	Off	Normal operation
	Blinking	General configuration error
	Single flash	Warning limit reached in CAN controller, for example due to bad or no signal on the CANopen network.
	Double flash	A nodeguard event or a heartbeat event has occurred.
	Triple flash	Sync error. The sync message has not been received within the configured communication cycle time
	1 Hz	Fatal error, contact support.
	On	The CAN controller is bus off.
RUN - CANopen state (green)	Off	No power, not configured or no CANopen network
	Blinking	Pre-operational
	Single flash	Stopped
	Triple flash	Transparent CAN mode
	On	Operational

4.3 Maximum Cable Lengths

When designing the CANopen network, please take into account the maximum cable length for different baud rates.

Baud rate (kilobaud)	Maximum cable length (m)
20	2500
50	1000
125	500
250	250
500	100
800	50
1000	25

4.4 Line Termination

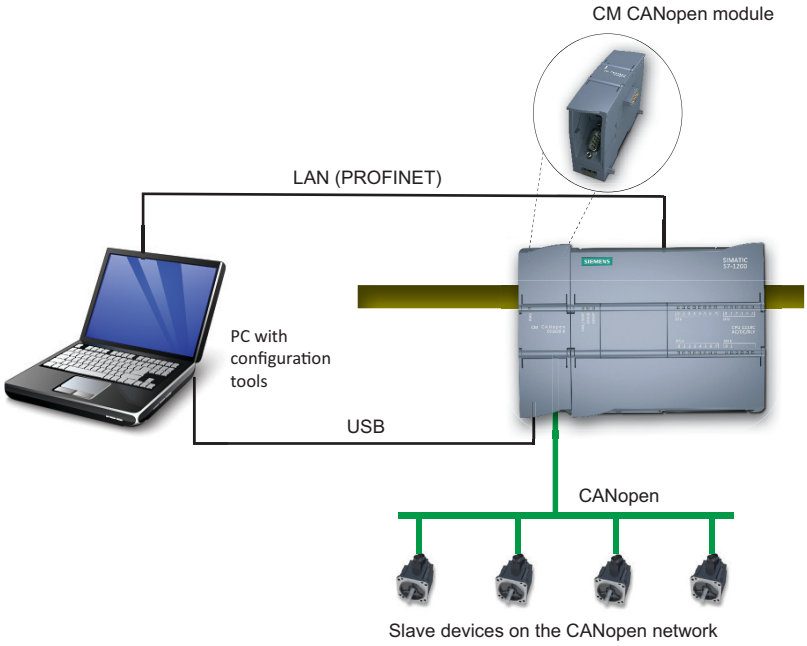
To minimize the signal's reflections from the end of the cable, a line termination is needed close to each end of the bus. If the cable you are using is not terminated, connect a line termination between contacts 2 and 7 (CAN_L and CAN_H) beneath the slot, where the module is mounted. The impedance of the termination should be 120 Ω (5%, 1/4 W max.).

5. Configuration

5.1 General

Configuration of the SIMATIC S7-1200 PLC to provide connectivity to a CAN/CANopen network, using the CM CANopen Module for SIMATIC S7-1200, can be done in different ways, depending on the application.

The CANopen network, and the behavior of the module on this network, have to be configured using an external CANopen configuration tool on the PC and the configuration is downloaded to the CM CANopen module through the module USB connection.



The module can be configured either as a manager or as a slave, depending on the application.

5.2 SIMATIC S7-1200 PLC Configuration[1]

Siemens TIA Portal is used to configure the SIMATIC S7-1200 PLC and the CM CANopen Module for SIMATIC S7-1200. The module can be imported into the tool.

The CM CANopen module offers the SIMATIC S7-1200 PLC access to a CAN/CANopen network, where the CM CANopen module either acts as a Transparent CAN module or acts as a slave or a manager on a CANopen network, depending on the settings in the parameter list. The SIMATIC S7-1200 may act as a PROFINET slave, with another PLC on the PROFINET network as master. If so, the SIMATIC S7-1200 together with the CM CANopen module can act as a gateway between PROFINET and CANopen.

A example on how to configure the system is given on page 12.

1. The configuration procedure in TIA Portal is the same for CANopen and CAN networks.

Double-clicking on the CM CANopen Module for SIMATIC S7-1200 in the list of modules in TIA Portal will open a window, where the user can set the values of the module parameters.

The following parameters, that have to be defined, can be found in the CANopen Interface tab:

Parameter	Description	Values	Default value
Node number	Node-ID on the CANopen network. Allowed values are 1 - 127	1-127	1
Operating mode	The operating mode of the module can be changed at any time using this parameter.	Transparent CAN CANopen Manager CANopen Slave	CANopen Manager
Baudrate	This parameter defines the baud rate on the CAN network.	20 kbit/s 50 kbit/s 125 kbit/s 250 kbit/s 500 kbit/s 800 kbit/s 1000 kbit/s	500 kbit/s
CANopen input data size[a]	This parameter defines the CANopen input data size. The direction is defined from the CANopen network point of view, that is input is input to the module from the CANopen network. Possible values are 1 - 256 bytes.	1 - 256	32
CANopen output data size[a]	This parameter defines the CANopen output data size. The direction is defined from the CANopen network point of view, that is output is output to the CANopen network from the module. Possible values are 1 - 256 bytes.	1 - 256	32
CANopen Manager auto configuration[b]	Auto configuration scans the network for devices at startup and tries to start them. Default PDO mapping is used. Auto configuration is enabled by checking a box in the configurator.	Disabled Enabled	Disabled

a. Only valid in CANopen mode
b. Only valid in CANopen manager mode

The parameterization is downloaded to the PLC with the HW configuration, using an Ethernet connection.

5.3 CANopen Network Configuration

An external CANopen configuration tool[1] is used to configure the nodes on the CANopen network. The resulting Concise DCF files are downloaded to the CANopen manager using a USB connection between the PC and the CM CANopen module. At the next startup the CANopen manager will configure the network, if this function was set in the configuration tool during initial configuration.

The CM CANopen Module for SIMATIC S7-1200 can be configured either as a manager or as a slave on the CANopen network. The configuration is decided by the PLC (by the Parameterization values).

5.4 Configuration Files

The Electronic Data Sheet (EDS) file for the CM CANopen Module for SIMATIC S7-1200 will be automatically installed with the CANopen configuration tool. This file is necessary to perform the configuration of the module on the CANopen network. Updated versions will be available at www.hms-networks.com/can-for-S7-1200.

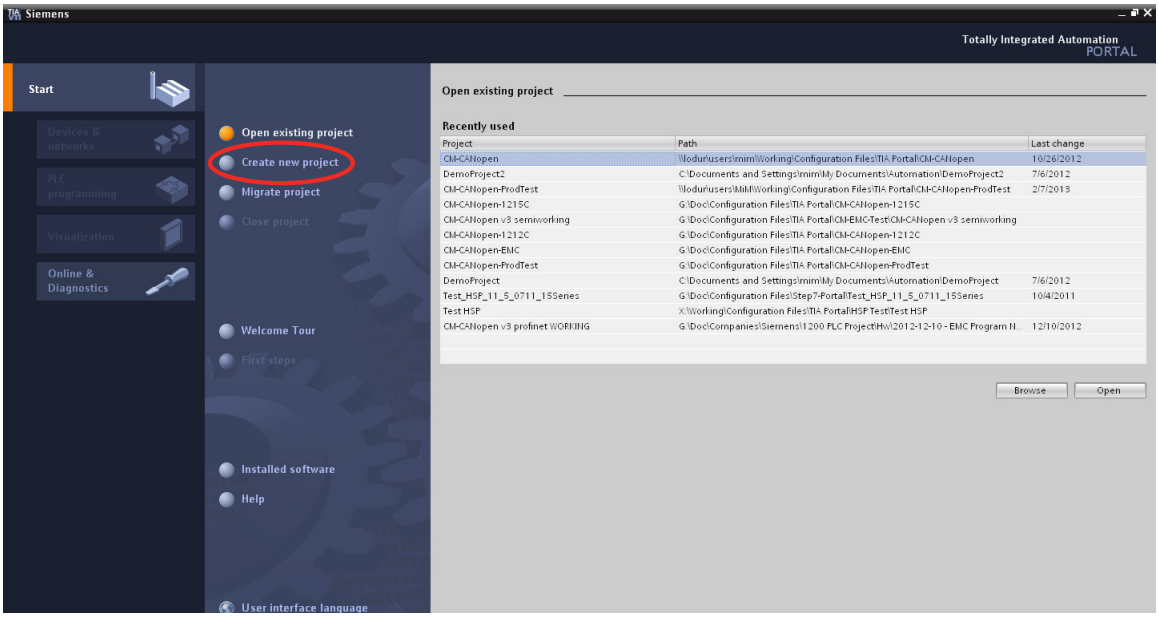
The necessary information for configuring the module with the SIMATIC S7-1200 PLC can be included in TIA portal by installing the HSP file[2].

5.5 Configuration Example

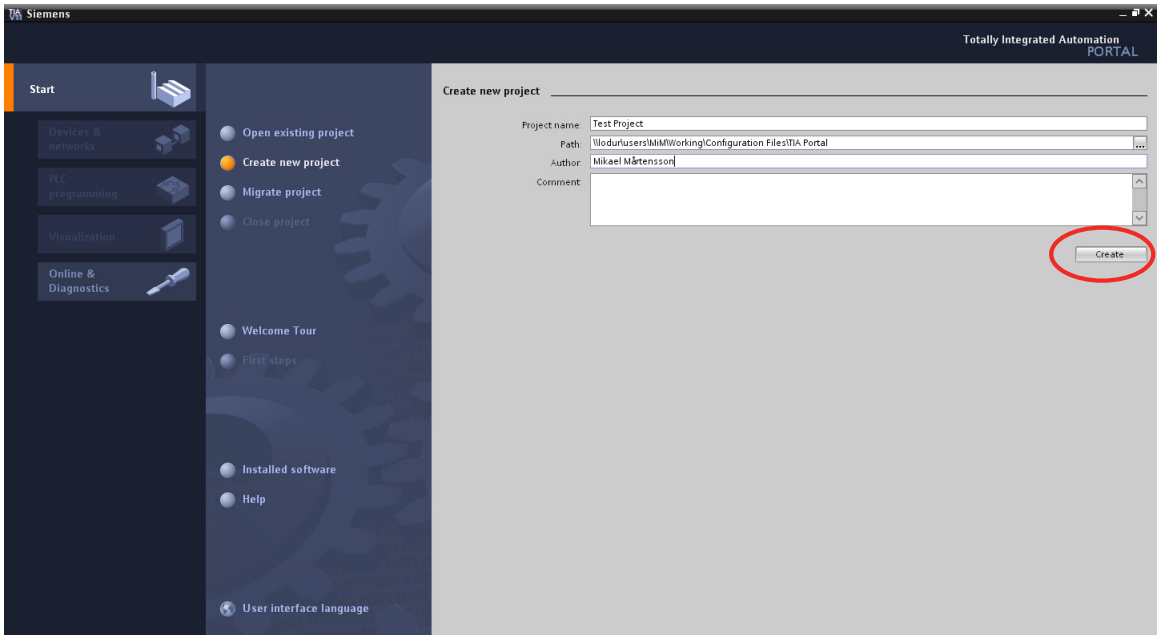
This section gives an example of a TIA Portal configuration of the CM CANopen module from HMS Industrial Networks in combination with the S7-1200 PLC from Siemens. In this example a CPU of type 1214C is used, but the example can be applied to all other PLC types as well. To run this example, the HSP file, that can be downloaded from www.hms-networks.com/can-for-S7-1200, must have been installed.

1. Please visit www.hms-networks.com/can-for-S7-1200 or contact HMS support for further information, see "Support" on page 2.
2. Available for download at www.hms-networks.com/can-for-S7-1200.

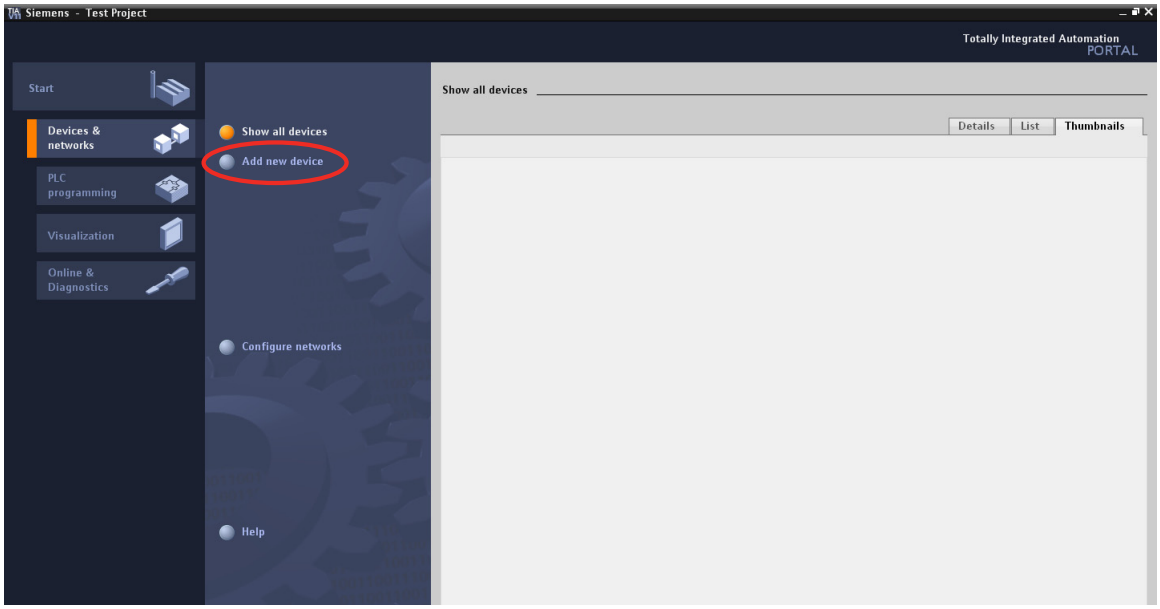
1. Open the TIA Portal program and start a new project by clicking “Create new project” on the left side of the screen.



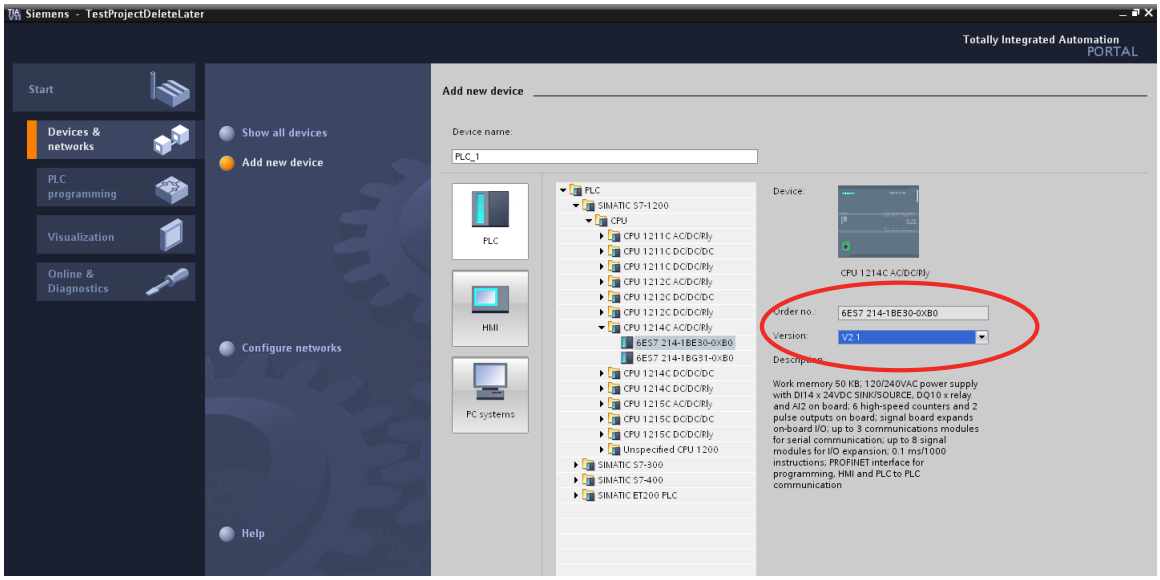
2. Enter a name for the project and the path to where the project should be stored, optionally with information like author and descriptive comments. Press “Create”.



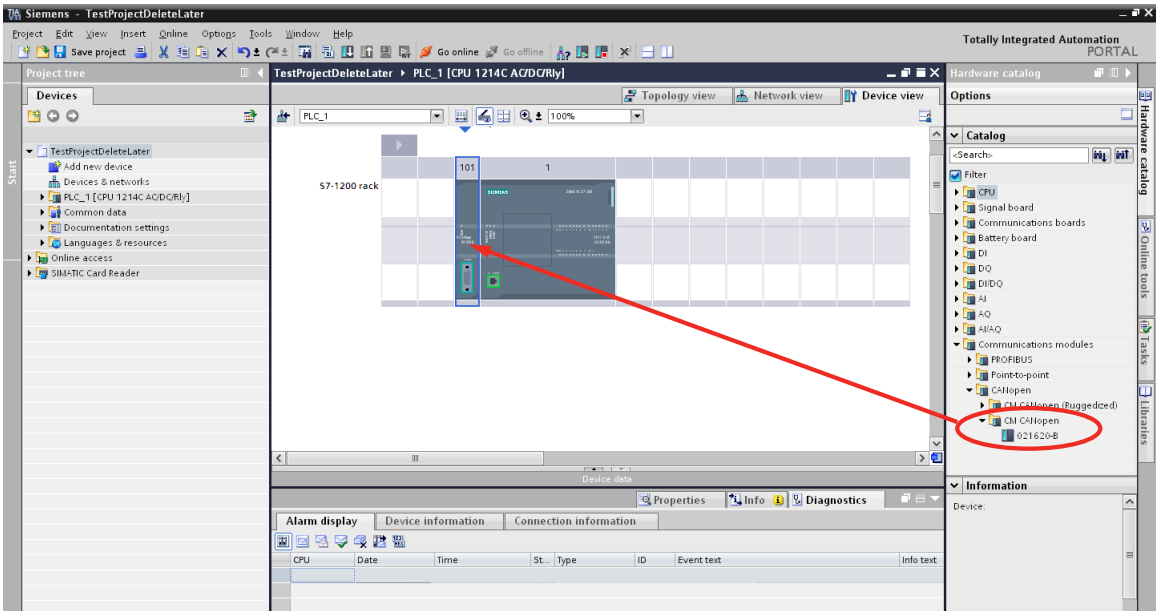
3. Select “Devices & networks” to the left and then press “Add new device”.



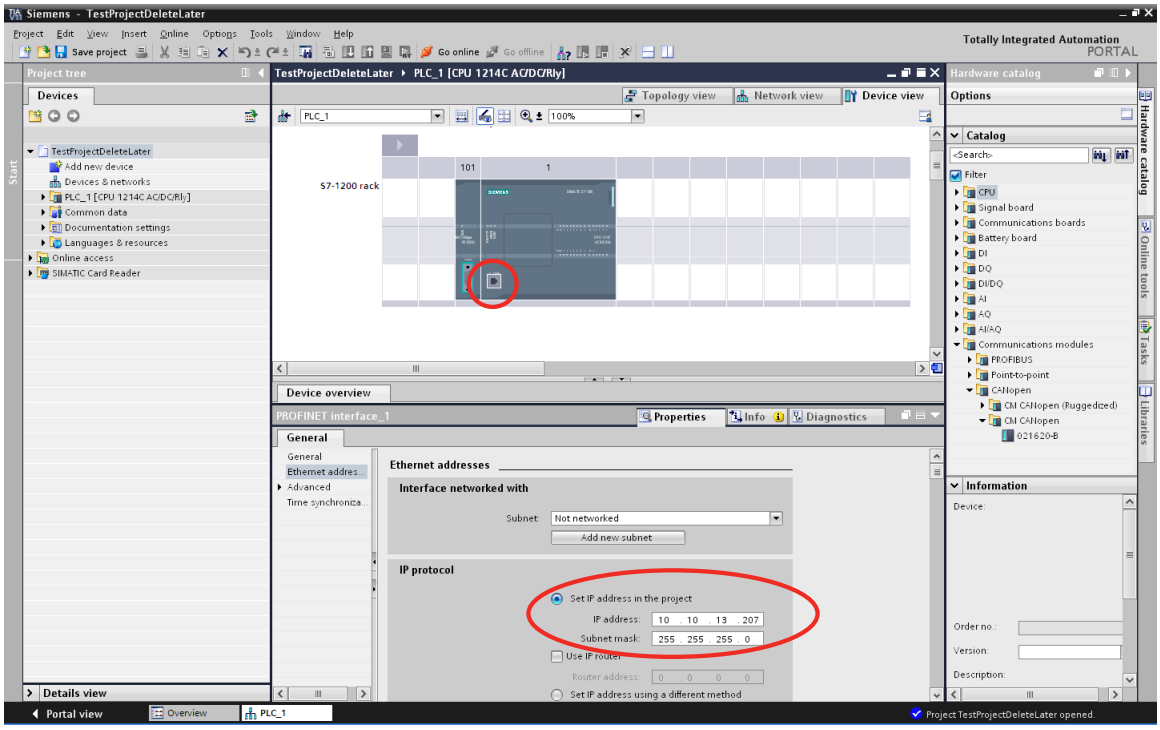
4. Select the PLC type used in the configuration. Select correct software version and optionally enter a name for the PLC (default PLC_1). Press “Add”.



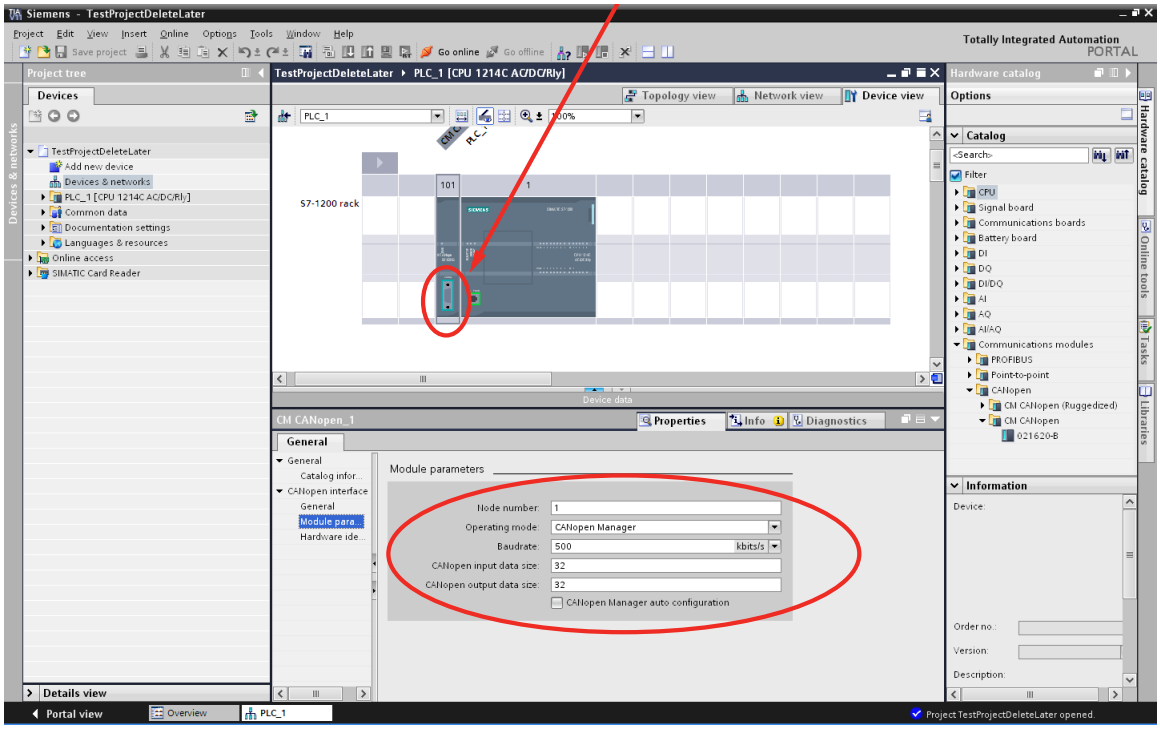
5. Select the CM CANopen module in the list as shown in the picture. Drag and drop an instance of the module to the left side of the PLC module.



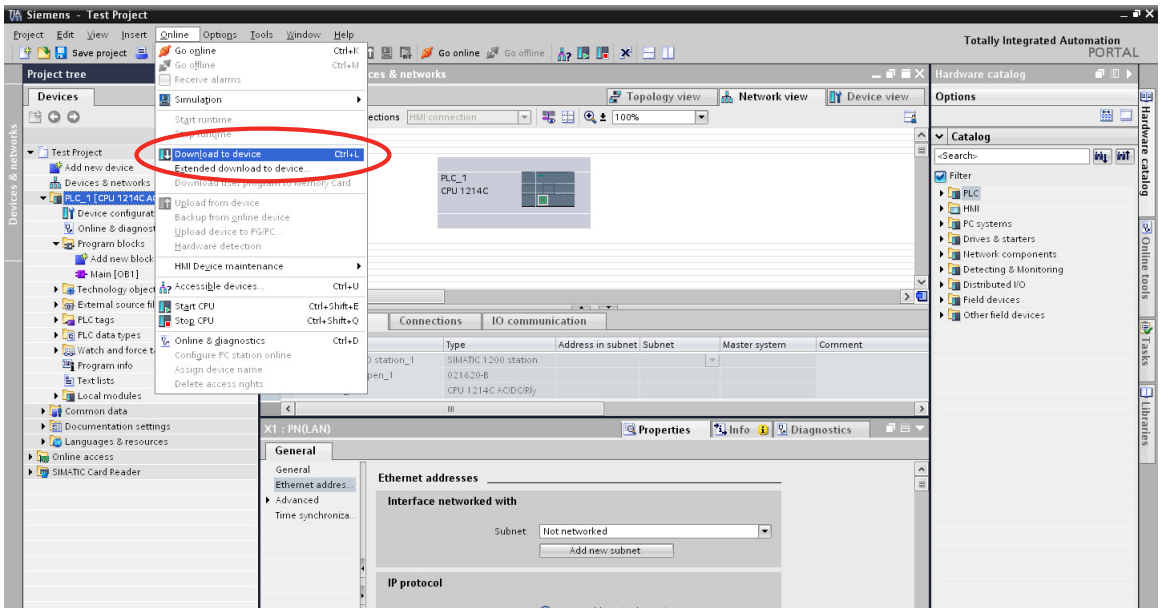
6. Double click on the Ethernet connector on the PLC and enter a proper IP address for the PLC.



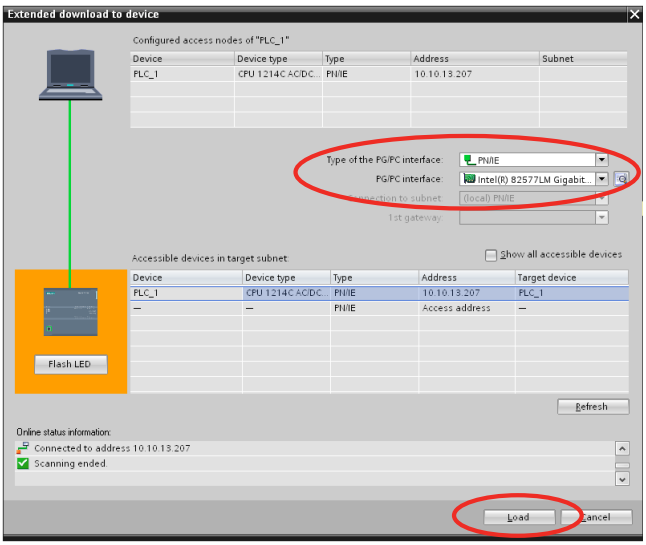
7. Double click on the connector of the CM CANopen module and choose the module parameter tab. Enter the values in the parameter list. Please note that these parameters are valid for the module on the CANopen network.
In this example, the CM CANopen module for S7-1200 is configured as a CANopen manager. The node number on the CANopen network is 1. The CANopen baud rate is set to 500 kbit/s and both the input data size and the output data size are set to 32 bytes.



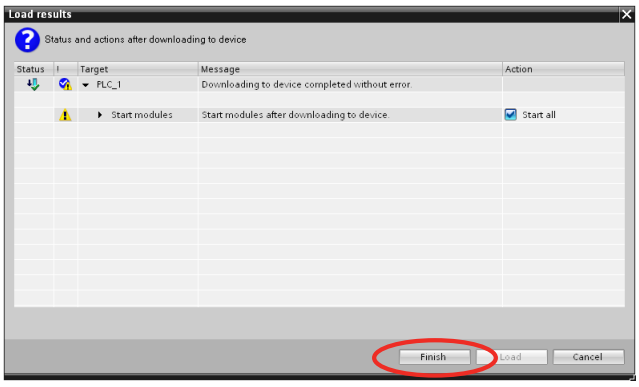
8. The configuration is now finalized and can be downloaded to the S7-1200 device. To download the software configuration select the PLC_1 on the right and choose “Online” > “Download to device”. To download the hardware configuration, right-click on the device in the tree and choose “Download to device” > “all”.



9. Select the PG/PC interface you are using (in this example PN/IE) and the Ethernet interface on your PC (in this example Intel 82577...). If the IP address, entered earlier, is found on the network press "Load" to download to your device. If not, select the correct device from the list available in the "Accessible devices in target subnet" window. If the device is on another subnet, and not available in this window, the check box "Show all accessible devices" will have to be checked.



10. If the download was successful the following screen will appear asking if the PLC should start. Press Finish and the PLC will start running the empty program that was downloaded.



11. The configuration of the module is now finished and the PLC programming can start.

Configuration of the CANopen network

The configuration of the CANopen network, including the CM CANopen Module for SIMATIC S7-1200, is done separately with CM CANopen Configuration Studio.

Important: Before downloading the configuration to the CANopen network, make sure that the PLC is set to STOP!

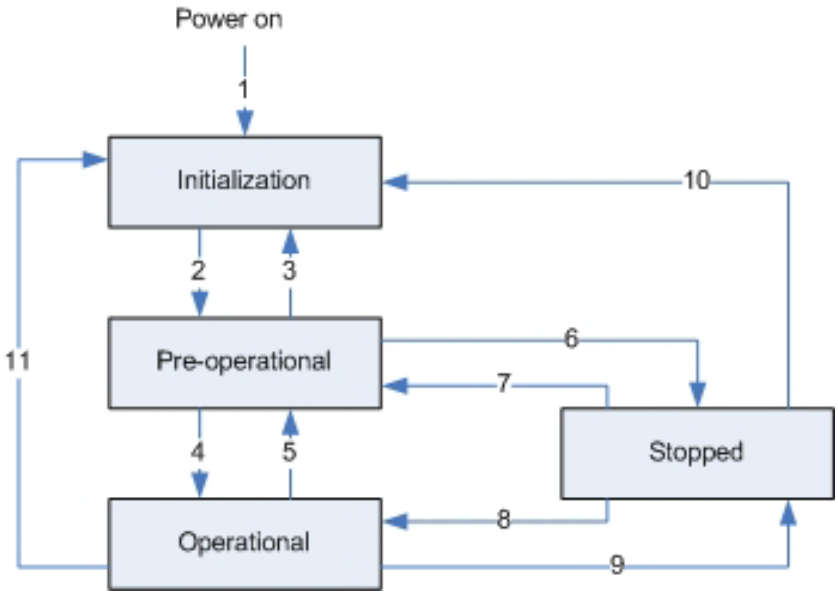
1. Open CANopen configuration tool.
2. Add nodes to CANopen network.
3. Configure each node with the necessary parameters.
4. Check that there is no conflict between the parameters downloaded from the CANopen configuration tool and the parameters set up with TIA Portal, regarding operation mode and process data size.
5. Download the configuration to the CANopen manager as Concise DCF-files. The configuration is automatically stored locally in nonvolatile memory.

Please consult the user manual for the configuration tool for details and/or contact HMS support, see "Support" on page 2.

6. CANopen Module Specification

6.1 NMT State Machine

The function of the CM CANopen Module for SIMATIC S7-1200 can be described as a state machine with four states.



State	Description
Initialization	When the power is switched on, the module starts initializing. This is done in three steps: basic initialization, reset application and reset communication. The parameters are set to the so called power-on values, which are the default values or the latest stored values. If parameter values are stored from a previous configuration, these are used. If not, or if a restore_default command is issued, the parameters are reset to the default values according to the communication and device profile specifications.
Pre-operational	Once initialized, the module enters the pre-operational state. SDO (Service Data Object) communication is allowed. A configuration application or configuration from CM CANopen Module for SIMATIC S7-1200 can configure PDOs (Process Data Objects), device parameters and allocate application objects (PDO mapping).
Operational	In the operational state all communication objects are active. Data is communicated according to the configurations made. Not all SDO:s can be changed, as some information e.g. an object may contain the application program that needs to stay fixed throughout operation.
Stopped	All communication is stopped, except node guarding and heartbeat, if active. From this state any transition to another state is possible, depending on if a restart, reconfiguration or reset of the module is wanted.

The module changes states upon reception of a request from the CANopen network, a hardware reset or following a change in the PLC RUN/STOP state.

If the CM CANopen Module for SIMATIC S7-1200 is configured as a CANopen manager, the transitions are initiated from the PLC. The module will either enter PRE-OPERATIONAL state or OPERATIONAL state directly, depending on the configuration downloaded from TIA Portal to the PLC.

The module can not enter the state STOPPED when transitions are initiated from the PLC.

If the module is configured as a slave its behavior is controlled by a CANopen manager on the network by the use of NMT telegrams. If the connection to the manager is lost, though, the module sends an emergency code (FFA0h) and enters PRE-OPERATIONAL state.

Transition no.	Description
1	The initialization state is entered automatically at power on.
2	Initialization finished. The pre-operational state is entered automatically.
3	Reset node or reset communication. New parameters have been received and a new initialization is required.
4	If configured to do so, the controlling manager changes to “Run” mode. If the node is a CANopen manager, this transition is controlled by the PLC. If it is a slave, the transition is controlled by the CANopen manager using the command ‘Start remote node’.
5	Return to pre-operational state. If the controlling manager is a PLC, this transition is initiated from the PLC.
6	Stop
7	Return to pre-operational state.
8	Start
9	Stop
10	Reset node or reset communication
11	Reset node or reset communication. New parameters have been received and a new initialization is required

Note: At a STOP from the PLC, the data last received is stored in the CM CANopen module. When the PLC issues a RUN the stored data is sent on the CANopen bus. The behavior of the module is the same if the connection is lost with the PLC on PROFINET.

6.2 Data Exchange

Process data is read and written using function blocks, see "Get Process Data In" on page 43 and "Set Process Data Out" on page 44.

6.2.1 PDO Functionality

Real-time data transfer is performed by means of PDOs (Process Data Objects). The PDOs are linked to entries in the Device Object Dictionary and provide the interface to the application objects. Number and length of PDOs in a device are node specific and have to be configured by the CANopen configuration tool.

PDOs are used both for data transmission and reception, using so called Transmit-PDOs (TPDOs) and Receive-PDOs (RPDOs). Each PDO corresponds to two entries in the Device Object Dictionary. The PDO parameter object holds information on the COB-ID, the transmission type etc. On recognition of the COB-ID the corresponding PDO mapping object can be identified, to make it possible to transmit/receive data to/from the correct object in the device. The default settings for the mapping can be changed during configuration.

Default PDO Mapping Scheme for Slave Mode[1]

The module features a simple default mapping scheme with 4 TPDOs and 4 RPDOs.

- **RPDO**

RPDO no.	Default COB-IDs	Mapped to...	Relating to...	Default State
1	200h + Node-ID	Object index A4C0h, subindex 1... 8	Output data buffer, bytes 0... 7	Enabled
2	300h + Node-ID	Object index A4C0h, subindex 9... 16	Output data buffer, bytes 8...15	
3	400h + Node-ID	Object index A4C0h, subindex 17... 24	Output data buffer, bytes 16... 23	
4	500h + Node-ID	Object index A4C0h, subindex 25... 32	Output data buffer, bytes 24... 31	
5	80000000h	Object index A4C0h, subindex 33... 40	Output data buffer, bytes 32... 39	Disabled
...		...	...	
16		Object index A4C0h, subindex 121... 128	Output data buffer, bytes 120... 127	
...		...	...	
31		Object index A4C0h, subindex 241... 248	Output data buffer, bytes 240... 247	
32		Object index A4C0h, subindex 241... 248 Object index A4C1h, subindex 1... 2	Output data buffer, bytes 248... 255	
33		No default mapping	-	
...		...	...	
64		No default mapping	-	

1. There is no default mapping for master mode.

- **TPDO**

TPDO no.	Default COB-IDs	Mapped to...	Relating to...	Default State
1	180h + Node-ID	Object index A040h, subindex 1... 8	Input data buffer, bytes 0... 7	Enabled
2	280h + Node-ID	Object index A040h, subindex 9... 16	Input data buffer, bytes 8... 15	
3	380h + Node-ID	Object index A040h, subindex 17... 24	Input data buffer, bytes 16... 23	
4	480h + Node-ID	Object index A040h, subindex 25... 32	Input data buffer, bytes 24... 31	
5	80000000h	Object index A040h, subindex 33... 40	Input data buffer, bytes 32... 39	Disabled
...		...	...	
16		Object index A040h, subindex 121... 128	Input data buffer, bytes 120... 127	
...		...	...	
31		Object index A040h, subindex 241... 248	Input data buffer, bytes 240... 247	
32		Object index A040h, subindex 241... 248 Object index A041h, subindex 1... 2	Input data buffer, bytes 248... 255	
33		No default mapping	-	
...		...	...	
64		No default mapping	-	

RPDO Transmission Types

RPDOs can be received either in synchronous or asynchronous mode. A synchronization (SYNC) object is transmitted periodically by a synchronization producer, located either in the CM CANopen Module for SIMATIC S7-1200 or in another node on the CANopen network. The data in synchronous RPDOs are not transferred to the SIMATIC S7-1200 PLC until after the next SYNC object is received. Asynchronous RPDOs will be transferred to the SIMATIC S7-1200 PLC at reception.

The transmission type parameter of a RPDO specifies the triggering mode.

Transmission type, RPDO	Mode	RPDO transmission description
0 - 240	Synchronous	A received RPDO is transferred to the SIMATIC S7-1200 PLC after a SYNC object is received.
241 - 253	-	Reserved
254 - 255	Asynchronous, Event driven	An RPDO is transferred to the SIMATIC S7-1200 PLC without any relation to the SYNC object. **Note**: All default PDOs are mapped with transmission type 255.

TPDO Transmission Types

TPDOs can be transmitted either in synchronous or asynchronous mode. A synchronization (SYNC) object is transmitted periodically by a synchronization producer, located either in the CM CANopen Module for SIMATIC S7-1200 or in another node on the CANopen network. Synchronous TPDOs are transmitted within a pre-defined time-window immediately after a configured number of SYNC objects, either always or after a CoS (Change of State event). Asynchronous TPDOs can be transmitted at any time, triggered by a CoS or a cyclic period set in the Event Timer.

The transmission type parameter of a TPDO specifies the transmission mode as well as the triggering mode.

Transmission type, TPDO	Mode	TPDO transmission description
0	Synchronous, acyclic	A TPDO is triggered by an event, but not transmitted before the occurrence of a SYNC object
1 - 240	Synchronous, cyclic	A TPDO is transmitted with every n-th SYNC object, where n is the defined number from 1 - 240.
241 - 253	-	Reserved
254 - 255	Asynchronous, Event driven	A TPDO is transmitted without any relation to the SYNC object. The transmission is triggered by a CoS event or if a specified time has elapsed without an event. **Note**: All default PDOs are mapped with transmission type 255.

6.3 Error Control

Different mechanisms exist to monitor the network. At an error event from any of these, the active I/O data is frozen, as no new data will be available.

6.3.1 Heartbeat Mechanism

The Heartbeat Mechanism is used to monitor the nodes in the network and verify that the nodes are available. A heartbeat producer periodically sends a message. The data part of the frame contains a byte indicating the node status. The heartbeat consumer reads these messages. If a message fails to arrive within a certain time limit (defined in the object directory of the devices, object 1016h, page 27), a heartbeat event is triggered by the consumer. The ERROR LED on the front of the CM CANopen Module for SIMATIC S7-1200 will indicate the event and a diagnostic error will be logged to the SIMATIC S7-

1200 PLC. If the module is configured as a slave, it will react according to the settings in object 1029h. If it is in OPERATIONAL state, it will thus go to PRE-OPERATIONAL state and wait for the user to take action. If it is in manager mode it will take action according to the settings in the manager objects.

The CM CANopen Module for SIMATIC S7-1200 can act as both heartbeat consumer and as heartbeat producer.

6.3.2 Node Guarding

When using this mechanism NMT Master transmits guarding requests. If an NMT Slave has not responded within a defined time span (node lifetime) or if the communication status of the slave has changed, the master takes appropriate action according to its configuration. In case of the CM CANopen Module for SIMATIC S7-1200 being the master, the PLC will be informed about the disappearance of the slave on the CANopen network. Any data to or from the disappeared slave will be frozen.

If Life guarding (the slave guards the master) is supported, the slave uses the guard time and lifetime factor from its Object Dictionary to determine the node lifetime. If the slave does not receive a guarding request within its lifetime, a node guard event is registered. The ERROR LED on the front of the CM CANopen Module for SIMATIC S7-1200 will indicate the event and a diagnostic error will be logged on the channel diagnostics.

If the guard time or the lifetime factor are 0 (default), the slave does not guard the master. The guarding can be initiated at boot-up or later.

Note 1: If heartbeat is enabled, node guarding is disabled.

Note 2: When node guarding is used for master supervision, a GuardTime and a RetryFactor is set for each node in its slave assignment object, see "NMT Slave Assignment, 1F81h" on page 32.

6.3.3 Emergency Object (EMCY)

The Emergency Object is used for error reporting on the CANopen network when a fault has occurred and for reporting when all faults in a module have been cleared. The error codes are saved in a list in the Communication Profile Object 1003h, see page 26 and a message is produced on the CANopen network. A list of emergency error codes, that can be produced by the module, is available in "CANopen Emergency Codes" on page 57

6.3.4 Diagnostics

When an event or alarm is generated by the CM CANopen module, the device alarm code 1Ah (external error) is transmitted and will trigger an interrupt in the PLC user program. This interrupt indicates that there is an alarm/event to be polled out from the module. Using the interface function Get Node & Network Status (CANopen mode) or Get Diagnostic Information (Transparent CAN mode) information about the alarm/event can be polled out from the module. For more information, see "Get Node & Network Status" on page 46 and "Get Diagnostic Information" on page 52.

7. Supported Objects

The following sections describe the CANopen objects, according to CiA 301 and CiA 302, implemented within the module and described in the EDS file.

7.1 Static Data Types

According to CiA 301.

7.2 Communication Profile Area

7.2.1 CiA 301 Communication Profile Objects

The table below shows the objects according to CANopen specification CiA 301 rev. 4.2.

Index	Object Name	subindex	Description	Type	Access	Notes
1000h	Device Type	00h	Type of device	U32	RO	0000 0000h (No profile)
1001h	Error register	00h	Error register, connected to the EMCY object. Bit 0 indicates a generic error	U8	RO	-
1002h	Manufacturer status register		Byte 0: 5000/02h Byte 1: 5000/03h Byte 2-3: 5000/01h	U32	RO	
1003h	Pre-defined error field	00h	Number of errors. Writing a 0 to this subindex clears the error list.	U8	RW	See “CANopen Emergency Codes” on page 57 for emergency error codes.
		01h...05h	List of errors. Most recent error at top of list.	U32	RO	
1005h	COB-ID Sync	00h	COB-ID of the sync message	U32	RW	-
1006h	Communication Cycle Period	00h	Communication cycle period (ms).	U32	RW	Min: 0 ms, max: 32767 ms. Only available if SYNC support is enabled.
1008h	Manufacturer device name	00h	The name of the CANopen module	Visible string	RO	“S7-1200 CM CANopen”
1009h	Manufacturer hardware version	00h	Manufacturer hardware version	Visible string	RO	Current hardware revision
100Ah	Manufacturer software version	00h	Manufacturer software version	Visible string	RO	Current software revision
100Ch	Guard time	00h	Used together with “Life time factor” to decide the node lifetime (ms).	U16	RW	Min: 0 ms (default), max: 32767 ms.
100Dh	Life time factor	00h	If the node has not been guarded within its lifetime (“Life time factor”*”Guard time”), an error event is logged and a remote node error is indicated	U8	RW	Min: 00h (default). max FFh

Index	Object Name	subindex	Description	Type	Access	Notes
1010h	Store Parameters[a]	00h	Largest subindex supported	U8	RO	01h
		01h	Store all parameters	U32	RW	To save a configuration, write “save” = 73 61 76 65h to this object.[b]
1011h	Restore Parameters[a]	00h	Largest subindex supported	U8	RO	01h
		01h	Restore all default parameters	U32	RW	To restore the default values of a configuration, write “load” = 6C 6F 61 64h to this object.[c]
1014h	COB-ID EMCY	00h	Defines the COB-ID of the Emergency Object	U32	RW	
1016h	Consumer Heartbeat Time	00h	Largest subindex supported	U8	RO	10h
		01h - 10h	The consumer heartbeat time defines the expected heartbeat cycle time and has to be higher than the corresponding producer heartbeat time. Monitoring starts after the reception of the first heartbeat. Not used if 0	U32	RW	Node-ID + Heartbeat Time. Bits 31-24: reserved Bits 23-16: Node-ID Bits 15-0: Heartbeat Time Value must be a multiple of 1ms. (Min: 10 ms, max 32768 ms). Up to 16 nodes can be monitored.
1017h	Producer Heartbeat Time	00h	Defines the cycle time of the heartbeat. Not used if 0	U16	RW	The time must be at least 10 ms and a multiple of 1 ms. Max: 32768 ms.
1018h	Identity object	00h	Highest subindex supported	U8	RO	04h
		01h	Vendor-ID	U32	RO	1Bh (HMS Industrial Networks)
		02h	Product Code	U32	RO	1Fh (CM CANopen Module for SIMATIC S7-1200)
		03h	Revision Number	U32	RO	Current software revision
		04h	Serial Number	U32	RO	32 bit serial number
1020h	Verify Configuration	00h	Number of entries	U8	RO	-
		01h	Configuration date	U32	RW	-
		02h	Configuration time	U32	RW	-
1029h	Error Behavior[d]	00h	Number of entries	U8	RO	01h
		01h	Communication Error. Defined by CANopen (loss of CAN messages, transmit queue overrun, RPDO length error, CAN-ID error, slave guarding by master; guarding time out, consumer heartbeat time out)	U8	RW	00h (default): Change to NMT state PRE_OPERATIONAL if currently in NMT state OPERATIONAL. 01h: No change of NMT state. 02h: Change to NMT state STOPPED.
		02h	Manufacturer Specific Error. PLC changes from RUN to STOP while CANopen manager is in state OPERATIONAL.	U8	CONST	
1400h ... 143Fh	Receive PDO parameter	00h	Largest subindex supported	U8	RO	02h
		01h	COB ID used by PDO	U32	RW	-
		02h	Transmission type	U8	RW	See “RPDO Transmission Types” on page 23

Index	Object Name	subindex	Description	Type	Access	Notes
1600h ... 163Fh	Receive PDO mapping	00h	No. of mapped application objects in PDO	U8	RW	-
		01h	Mapped object #1	U32	RW	-
		02h	Mapped object #2	U32	RW	-
		03h	Mapped object #3	U32	RW	-
		04h	Mapped object #4	U32	RW	-
		05h	Mapped object #5	U32	RW	-
		06h	Mapped object #6	U32	RW	-
		07h	Mapped object #7	U32	RW	-
		08h	Mapped object #8	U32	RW	-
1800h ... 183Fh	Transmit PDO parameter	00h	Largest subindex supported	U8	RO	05h
		01h	COB ID used by PDO	U32	RW	-
		02h	Transmission type	U8	RW	See “TPDO Transmission Types” on page 23
		03h	Inhibit time	U16	RW	In steps of 0.1 ms
		05h	Event Timer (ms)	U16	RW	-
1A00h ... 1A3Fh	Transmit PDO mapping	00h	No. of mapped application objects in PDO	U8	RW	-
		01h	Mapped object #1	U32	RW	-
		02h	Mapped object #2	U32	RW	-
		03h	Mapped object #3	U32	RW	-
		04h	Mapped object #4	U32	RW	-
		05h	Mapped object #5	U32	RW	-
		06h	Mapped object #6	U32	RW	-
		07h	Mapped object #7	U32	RW	-
		08h	Mapped object #8	U32	RW	-

a. Relevant only for communication parameters
b. If entered manually, the bytes should be entered in reversed order: 65 76 61 73h.
c. If entered manually, the bytes should be entered in reversed order: 64 61 6F 6C.
d. Only valid in slave mode.

7.2.2 Configuration Manager

CiA 302 part 3: Configuration and program download

Network Configuration Objects

Configuration of the manager and the slaves can be done using concise device configuration files. The configuration is stored in the manager by writing to object 1F22h. The nodes are configured either when the module (in manager mode) is booted or when a request to boot a slave is sent to object 1F25h.

Index	Object Name	subindex	Description	Type	Access
1F22h	Concise DCF	0 - 127	The concise/compressed DCF files information is stored in this object. The configuration is stored in the manager by writing to the subindex corresponding to the Node-ID of the module.	Domain	Sub 0: RO Sub 1 - 127: RW
1F25h	Configuration Request	0 - 128	subindex 0 is ignored. subindex i (i = 1 - 127): Request reconfiguration of slave with Node-ID equal to subindex i. subindex 128: Request to reconfigure all slaves.	UINT32[a]	Sub 0: RO Sub 1 - 128: WO

a. To configure the slave with Node-ID i, write “conf” = 63 6F 6E 66h to this object (1F25h, subindex i)

If an error occurs during configuration of a mandatory node or a detected optional node, an emergency object, with the code FF30h, with either error code 1 or 4 will be sent. Diagnostics (error code 9) will be set and the specified node will not be set to Operational.

See also ...

- “Error Control” on page 23

Check Configuration

The Configuration Manager (CMT) compares signature and configuration with the value from the DCF to decide if a reconfiguration of a CANopen slave device is to be performed or not. The comparison values are stored by the Configuration Manager in these objects:

Index	Object Name	subindex	Description	Type	Access
1F26h	Expected Configuration Date	0 - 127	The date that the Configuration Manager expects to find when comparing signature and configuration.	UINT32	RW
1F27h	Expected Configuration Time	0 - 127	The time that the Configuration Manager expects to find when comparing signature and configuration.	UINT32	RW

7.2.3 Network Management Objects

The NMT master controls the states of the connected network participants, the NMT slaves. It monitors the devices and reports to the application, for example if an NMT slave fails. Please refer to the CANopen specification, see "Related Documents" on page 1. In more complex systems several devices are able to perform as master, which means that the configuration must have an entry defining which device will act as master.

Once configured, the objects carry all information needed for the module to act on the network and the application does not need to be accessed to obtain this information. This results in a substantial reduction of the overall implementation and maintenance effort when implementing multiple applications.

Index	Object Name	Subindex	Description	Type	Access
102A	NMT inhibit time	0	This object holds the configured inhibit time between two subsequent NMT messages.	U16	RW
1F80h	NMT Start-up	0	Defining whether the device is the NMT master.	U32	RW
1F81h	NMT Slave Assignment	ARRAY	Module list: Entry of all slaves to be managed, including guarding values and the entry of actions to be taken in event of guarding errors.The array subindex corresponds to the Node-ID of the CANopen slave.	U32	Sub 0: RO Sub 1 - 127: RW
1F82h	Request NMT	ARRAY	Remote control initiation of NMT services. For example, tools can use this to request intentional start/stop of individual slaves. Remote query of the current state.	U8	Sub 0: RO Sub 1 - 127: RW Sub 128: WO
1F83h	Request Guarding	ARRAY	Remote control start/stop of guarding. Remote query of the current state	U8	Sub 0: RO Sub 1 - 127: RW Sub 128: WO
1F84h	Device Type Identification	ARRAY	Verify expected device types for the slaves.	U32	Sub 0: RO Sub 1 - 127: RW
1F85h	Vendor Identification	ARRAY	Verify vendor identifications for the slaves	U32	Sub 0: RO Sub 1 - 127: RW
1F86h	Product Code	ARRAY	Verify product codes for the slaves	U32	Sub 0: RO Sub 1 - 127: RW
1F87h	Revision Number	ARRAY	Verify revision numbers for the slaves	U32	Sub 0: RO Sub 1 - 127: RW
1F88h	Serial Number	ARRAY	Verify expected serial numbers for the slaves	U32	Sub 0: RO Sub 1 - 127: RW
1F89h	Boot Time	VAR	Maximum slave boot time before the configuration manager indicates boot error.	U32	RW
1F8A	Restore Configuration	ARRAY	The values in this array define whether the corresponding node address (subindex) should be restored during startup or not. If the entry value is 0, the no restore command should be sent to the device.	U8	Sub 0: const Sub 1 - 127: RW

NMT Start-up, 1F80h

This object configures the startup behavior of the device, and how it will manage the slaves. If the module is set up as an NMT master in TIA Portal hardware configuration, the module will enable the master functionality by forcing bit 0 and bit 2 in this object to TRUE. If there is an attempt to set bit 0 and bit 2 to different values from the CANopen bus, an error response is received.

Bit No.	Value	Description	Notes
0	0	NMT master functionality is disabled. Ignore the rest of the object except bit 1 and 3. Ignore the entries of object 1F81h.	
	1	NMT master functionality is enabled. The device is master	
1	0	Start only explicitly assigned slaves (if bit 3 = 0)	
	1	After boot-up, perform the service NMT Start Remote Node All Nodes (if bit 3 = 0)	
2	0	Automatically enter Operational state	Not supported
	1	Do not enter Operational state automatically. The SIMATIC S7-1200 PLC will decide when to enter Operational state	
3	0	The NMT master shall start the slaves	
	1	The NMT master is not allowed to send NMT Start Remote Node command. The application will start the slaves	
4	0	If a mandatory slave generates an Error Control Event, treat the slave individually	If bit 6 = 1, ignore bit 4
	1	If a mandatory slave generates an Error Control Event, perform NMT Reset All Nodes, see bit 6 and object 1F81h, bit 3.	
5	-	Not implemented	
6	0	If a mandatory slave generates an Error Control Event, treat the slave according to bit 4	
	1	If a mandatory slave generates an Error Control Event, send NMT Stop All Nodes (including self). Ignore bit 4	
7 - 31	-	Reserved (0)	

NMT Slave Assignment, 1F81h

This object defines which slaves the master should monitor, control and/or configure. One entry is made for each assigned slave, with the subindex corresponding to the slave's Node-ID.

Byte 0

Bit No	Value	Description
0	0	Node with this ID is not a slave
	1	Node with this ID is a slave. After configuration the node will be set to Operational
1	-	Reserved
2	0	On an Error Control Event or on detection of a new slave, inform the application, but do NOT configure and start the slave
	1	On an Error Control Event or on detection of a new slave, inform the application and start the process "Start Boot Slave"
3	0	Optional slave. The network may be started even if this node could not be contacted.
	1	Mandatory slave. The network must not be started if this node could not be contacted during the boot slave process
4	-	Not implemented
5	-	Not implemented
6	-	Not implemented
7	0	CANopen device may be used without prior resetting
	1	CANopen device shall be reset to factory default by issuing a restore to default (object 1011h)

Byte 1

8 bit value for the RetryFactor

Byte 2... 3

16 bit value for the GuardTime

If a slave does not answer, the master will retry the request RetryFactor-1 times with an interval of GuardTime. Guarding will be performed only if non-zero values are entered for RetryFactor and GuardTime.

See also ...

- "Node Guarding" on page 24

Request NMT, 1F82h

Each node on the CANopen network can be controlled individually from the PLC by sending this object. The subindex indicates what nodes the request affects:

subindex	Description
0	Largest subindex supported (128)
i (with i = 1...127)	Request NMT Service for the slave with Node-ID i.
128	Request NMT Service for all nodes

The desired state is given as a numeric value when writing to or reading from the local object dictionary:

Value	Write access	Read access
0	-	NMT state unknown. The node is not configured or otherwise not part of the network.
1	-	CANopen device is missing. The node at this Node-ID is configured but not available on the network.
4	STOP remote node	NMT state STOPPED
5	START remote node	NMT state OPERATIONAL
6	RESET NODE	-
7	RESET COMMUNICATION	-
127	Enter PRE-OPERATIONAL	NMT state PRE-OPERATIONAL

The entire network can be started with one command (subindex 128)

Examples

- Node 5 should be transferred to the OPERATIONAL state:
  An SDO write access with the value 5 is executed to object 1F82h subindex 5 in the local object dictionary.
- All the nodes in the network should be transferred to the PRE-OPERATIONAL state:
  An SDO write access with the value 127 is executed to object 1F82h subindex 128 in the local object dictionary.

Request Guarding, 1F83h

Guarding can be initiated from the object dictionary in a similar way. Guarding is initiated with the values stored in "NMT Slave Assignment, 1F81h" on page 32, provided that at the same time no parameters are entered for that node as a Heartbeat Consumer

Note: This functionality is only supported in master mode.

subindex	Description	Access
0	Largest subindex supported (128)	RO
i (with i = 1...127)	Request Guarding for the slave with Node-ID i	RW
128	Request Start/Stop Guarding for all nodes.	WO

Value	Write access	Read access
1	Start guarding	Slave is guarded
0	Stop guarding	Slave is not guarded

Example:

- Guarding should be started for node 5 (500 ms, Life Time Factor 3):
  An SDO write access with the value 00000001h is executed to object 1F81h subindex 5 in the local object dictionary. Guarding is activated by an SDO write access with the value 1 to object 1F83h subindex 5 in the local object dictionary.

Device Type Identification, 1F84h

Each node on the CANopen network is checked against its expected device type. The subindex indicates which node is checked:

subindex	Description
0	Largest subindex supported (127)
i (with i = 1...127)	Compares expected device type (entered into this object) with actual device type (object 1000h, subindex 0) for the slave with Node-ID i. If the expected device type is 0, this only gives information about the existence of a node, not which device type it is. If the value is not 0, it is compared to the value read from the node, and boot up is continued if they match.

Vendor Identification, 1F85h

Each node on the CANopen network is checked against its expected vendor. The subindex indicates which node is checked. The node in question is only checked if data is other than zero.

subindex	Description
0	Largest subindex supported (127)
i (with i = 1...127)	Compares expected vendor (entered into this object) with actual vendor (object 1018h, subindex 1) for the slave with Node-ID i. Boot up of slave is continued only if they match. If not, the slave is not put in Operational.

Product Code, 1F86h

Each node on the CANopen network is checked against its expected product code. The subindex indicates which node is checked. The node in question is only checked if data is other than zero.

subindex	Description
0	Largest subindex supported (127)
i (with i = 1...127)	Compares expected product code (entered into this object) with actual product code (object 1018h, subindex 2) for the slave with Node-ID i. Boot up of slave is continued only if they match. If not, the slave is not put in Operational.

Revision Number, 1F87h

Each node on the CANopen network is checked against its expected revision number. The revision number includes major and minor revision. For a match to occur the major revision has to be exactly the same and the minor revision of the module has to be greater than or equal to the expected minor revision number. The subindex indicates which node is checked. The node in question is only checked if data is other than zero.

subindex	Description
0	Largest subindex supported (127)
i (with i = 1...127)	Compares expected revision number (entered into this object) with actual revision number (object 1018h, subindex 3) for the slave with Node-ID i. Boot up of slave is continued only if they match. If not, the slave is not put in Operational.

Serial Number, 1F88h

Each node on the CANopen network is checked against its expected serial number. The subindex indicates which node is checked. The node in question is only checked if data is other than zero.

subindex	Description
0	Largest subindex supported (127)
i (with i = 1...127)	Compares expected serial number (entered into this object) with actual serial number (object 1018h, subindex 4) for the slave with Node-ID i. Boot up of slave is continued only if they match. If not, the slave is not put in Operational.

Boot Time, 1F89h

The network master will wait the assigned time (in ms) for all mandatory slaves to boot. An error will be indicated to the application if the mandatory slaves have not booted within the assigned time. If the assigned time is 0, the timer is disabled and the master will wait endlessly.

Value (ms)	Description
0	Default. No time limit for mandatory slaves to boot
> 0	Time limit for mandatory slave to boot

7.3 Manufacturer Specific Objects

One or several variable data objects are connected to each PDO. This is configured during the configuration phase. There are 64 Receive PDOs and 64 Transmit PDOs.

Index	Subindex	Type	Access	Name and description	Comment
5000h	-	RECORD		Status CM CANopen	
	00h	U8	RO	Largest subindex supported	
	01h	U16	RO	Event indication CM CANopen	For more information, see page 38.
	02h	U8	RO	CM CANopen state	For more information, see page 39.
	03h	U8	RO	CM CANopen communication status	For more information, see page 39.
	04h	U16	RO	CM CANopen configuration bits	For more information, see page 40.
5001h	-	ARRAY		Assigned slaves bit list	
	00h	U8	RO	Largest subindex supported	
	01h	U32	RO	Node-ID 1-32	Bit n represents Node-ID (n+1).
	02h	U32	RO	Node-ID 33-64	Bit n represents Node-ID (n+33).
	03h	U32	RO	Node-ID 65-96	Bit n represents Node-ID (n+65).
	04h	U32	RO	Node-ID 97-127	Bit n represents Node-ID (n+97). Bit 31 is always 0.
5002h	-	ARRAY		Configured slaves bit list	
	00h	U8	RO	Largest subindex supported	
	01h	U32	RO	Node-ID 1-32	Bit n represents Node-ID (n+1).
	02h	U32	RO	Node-ID 33-64	Bit n represents Node-ID (n+33).
	03h	U32	RO	Node-ID 65-96	Bit n represents Node-ID (n+65).
	04h	U32	RO	Node-ID 97-127	Bit n represents Node-ID (n+97). Bit 31 is always 0.
5003h	-	ARRAY		Fault slaves bit list	
	00h	U8	RO	Largest subindex supported	
	01h	U32	RO	Node-ID 1-32	Bit n represents Node-ID (n+1).
	02h	U32	RO	Node-ID 33-64	Bit n represents Node-ID (n+33).
	03h	U32	RO	Node-ID 65-96	Bit n represents Node-ID (n+65).
	04h	U32	RO	Node-ID 97-127	Bit n represents Node-ID (n+97). Bit 31 is always 0.
5004h	-	ARRAY		Operational slaves bit list	
	00h	U8	RO	Largest subindex supported	
	01h	U32	RO	Node-ID 1-32	Bit n represents Node-ID (n+1).
	02h	U32	RO	Node-ID 33-64	Bit n represents Node-ID (n+33).
	03h	U32	RO	Node-ID 65-96	Bit n represents Node-ID (n+65).
	04h	U32	RO	Node-ID 97-127	Bit n represents Node-ID (n+97). Bit 31 is always 0.
5005h	-	ARRAY		Stopped slaves bit list	
	00h	U8	RO	Largest subindex supported	
	01h	U32	RO	Node-ID 1-32	Bit n represents Node-ID (n+1).
	02h	U32	RO	Node-ID 33-64	Bit n represents Node-ID (n+33).
	03h	U32	RO	Node-ID 65-96	Bit n represents Node-ID (n+65).
	04h	U32	RO	Node-ID 97-127	Bit n represents Node-ID (n+97). Bit 31 is always 0.

Index	Subindex	Type	Access	Name and description	Comment
5006h	-	ARRAY		Pre-operational slaves bit list	
	00h	U8	RO	Largest subindex supported	
	01h	U32	RO	Node-ID 1-32	Bit n represents Node-ID (n+1).
	02h	U32	RO	Node-ID 33-64	Bit n represents Node-ID (n+33).
	03h	U32	RO	Node-ID 65-96	Bit n represents Node-ID (n+65).
	04h	U32	RO	Node-ID 97-127	Bit n represents Node-ID (n+97). Bit 31 is always 0.
5007h	-	ARRAY		Error state bit list	This list reflects the state of bit 0 in object 1001h of the slave devices.
	00h	U8	RO	Largest subindex supported	
	01h	U32	RO	Node-ID 1-32	Bit n represents Node-ID (n+1).
	02h	U32	RO	Node-ID 33-64	Bit n represents Node-ID (n+33).
	03h	U32	RO	Node-ID 65-96	Bit n represents Node-ID (n+65).
	04h	U32	RO	Node-ID 97-127	Bit n represents Node-ID (n+97). Bit 31 is always 0.
5008h	-	ARRAY		Inconsistent concise DCF bit list	
	00h	U8	RO	Largest subindex supported	
	01h	U32	RO	Node-ID 1-32	Bit n represents Node-ID (n+1).
	02h	U32	RO	Node-ID 33-64	Bit n represents Node-ID (n+33).
	03h	U32	RO	Node-ID 65-96	Bit n represents Node-ID (n+65).
	04h	U32	RO	Node-ID 97-127	Bit n represents Node-ID (n+97). Bit 31 is always 0.
5009h	-	ARRAY		Wrong concise DCF bit list	
	00h	U8	RO	Largest subindex supported	
	01h	U32	RO	Node-ID 1-32	Bit n represents Node-ID (n+1).
	02h	U32	RO	Node-ID 33-64	Bit n represents Node-ID (n+33).
	03h	U32	RO	Node-ID 65-96	Bit n represents Node-ID (n+65).
	04h	U32	RO	Node-ID 97-127	Bit n represents Node-ID (n+97). Bit 31 is always 0.
500Ah	-	ARRAY		Identity error bit list	
	00h	U8	RO	Largest subindex supported	
	01h	U32	RO	Node-ID 1-32	Bit n represents Node-ID (n+1).
	02h	U32	RO	Node-ID 33-64	Bit n represents Node-ID (n+33).
	03h	U32	RO	Node-ID 65-96	Bit n represents Node-ID (n+65).
	04h	U32	RO	Node-ID 97-127	Bit n represents Node-ID (n+97). Bit 31 is always 0.
5010h	-	ARRAY		Node error count	Contains the number of emergency messages currently buffered in 5018h. Writing 0 to this object resets the DOMAIN entries in 5018h.
	00h	U8	RO	Highest subindex supported	
	01h	U16	RW	Node error count Node-ID 1	
	...	...	...	...	
	7Fh	U16	RW	Node error count Node-ID 127	
5011h	-	U16	RO	Generic error count	Counts emergency messages caused by emergency error code class 10XXh.
5012h	-	U16	RO	Device hardware error count	Counts emergency messages caused by emergency error code class 50XXh.
5013h	-	U16	RO	Device software error count	Counts emergency messages caused by emergency error code class 60XXh.
5014h	-	U16	RO	Communication error count	Counts emergency messages caused by emergency error code class 81XXh.

Index	Subindex	Type	Access	Name and description	Comment
5015h	-	U16	RO	Protocol error count	Counts emergency messages caused by emergency error code class 82XXh.
5016h	-	U16	RO	External error count	Counts emergency messages caused by emergency error code class 90XXh
5017h	-	U16	RO	Device specific	Counts emergency messages caused by emergency error code class FFXXh.
5018h	-	ARRAY		Node emergency history	
	00h	U8	RO	Highest subindex supported	
	01h	DOMAIN	RO	Emergency history Node-ID 1	
	...	...	...	...	
	7Fh	DOMAIN	RO	Emergency history Node-ID 127	
5500h	-	ARRAY		Fatal event log	
	00h	U8	RO	Highest subindex supported	
	01h	DOMAIN	RO	Fatal event log	

Status CM CANopen, 5000h, subindex 01h

The table below gives the event indications for the CM CANopen that can be read from subindex 01h of the Status CM CANopen object. The bits are valid either only in manager mode or in both manager and slave mode.

Bit No	Name	Mode	Description
0[a]	FATE	Both	The CM CANopen is in Fatal Error state.
1	NIDE	Manager	A device in the network uses the Node-ID of the CM CANopen. The CM CANopen is set to Fatal Error state.
2	MSE	Manager	Error control event of a mandatory device.
3	MNCE	Manager	Identity error of at least one mandatory device.
4	OIE	Manager	Identity error of at least one optional device.
5	(reserved)	Both	Always 0.
6	(reserved)	Both	Always 0.
7[a]	NMTE	Both	Set if any bit in any of the diagnostics bit lists has been updated.
8	ASE	Manager	The slave assignment entry of a device contains features that are not supported by the CM CANopen (e.g. bits 4-6 of the slave assignment object 1F81h).
9[a]	PDOLEN_ERR	Both	The CM CANopen has received an RPDO with too few data bytes.
10	CONFIG_ERR	Manager	On download of a concise DCF by the boot slave process, the CM CANopen has detected that the concise DCF is either inconsistent or does not match the object dictionary of the slave module.
11	(reserved)	Both	Always 0.
12[a]	SAVE_ERR	Both	Processing of the save command failed.
13[a]	ALONE	Both	The CM CANopen is alone in the network or not connected to the network.
14	(reserved)	Both	Always 0.
15	(reserved)	Both	Always 0.

a. Valid in slave mode.

Status CM CANopen, 5000h, subindex 02h

The table below gives the state of CM CANopen that can be read from subindex 02h of the Status CM CANopen object. The bits are valid either in manager or slave mode or in both.

Value	Name	Mode	Description
00h	INIT	Both	The CM CANopen is not initialized
40h	MASTER_STATE_RESET	Both	The CM CANopen is configured as a manager in the NMT startup object (1F80h). The object dictionary of the CM CANopen may be configured by SDOs via the CAN bus or the SDO command interface.
41h	SLAVE_STOPPED	Slave	The CM CANopen is in the NMT state Stopped.
42h	SLAVE_PREOP	Slave	The CM CANopen is in the NMT state Pre-operational.
43h	SLAVE_OP	Slave	The CM CANopen is in the NMT state Operational.
60h	PREPARE_NET_INIT	Manager	Bootup according to CiA 302. The CM CANopen verifies the NMT slave assignment.
61h	NTW_RESET	Manager	The network is reset by NMT reset communication all nodes.
62h	NTW_WAIT	Manager	The CM CANopen waits for a previously defined time (default 2 s), to enable the other nodes on the network to execute the NMT reset communication command.
64h	BOOT_CONF	Manager	The CM CANopen carries out the initialization of the individual devices according to CiA 302.
70h	BOOT_END_MISSING_MAND	Manager	The network has been scanned. At least one mandatory module is missing and the boot time has not expired.
8Xh[a]	CLEAR	Manager	The network has been scanned, but no commands to start the CM CANopen or the CANopen network has yet arrived.
AXh[a]	RUN	Manager	The network has been set to the NMT state Operational.
CXh[a]	STOP	Manager	The network has been stopped.
EXh[a]	PRE-OPERATIONAL	Manager	The network has been set to NMT state Pre-operational
90h	FATAL_ERROR	Manager	A fatal error has occurred. The CM CANopen must be reinitialized.

a. The last four bits contain additional information:

Bit No	Description
0	Error bit for optional or unexpected modules. 0: No error. 1: At least one optional or unexpected module does not correspond to the expected network configuration.
1	Error bit for mandatory modules. 0: No error. 1: At least one mandatory module does not match the expected state.
2	General operational bit. 0: No CANopen slave is in NMT state Operational 1: At least one CANopen slave is in NMT state Operational
3	CM CANopen operational bit. 0: The CM CANopen is not in NMT state Operational. 1: The CM CANopen is in NMT state Operational.

Status CM CANopen, 5000h, subindex 03h

The table below gives the CM CANopen communication state that can be read from subindex 03h of the Status CM CANopen object.

Bit No	Description
0	Not supported, always 0.
1	An overrun of the CAN controller occurred.
2	The CAN controller is bus off.
3	0: The CAN controller is in error active state. 1: The CAN controller is in error passive state.
4	The CAN controller error state has changed. Bit is reset after a read access.

Bit No	Description
5	An overrun of the low priority transmit queue occurred. Bit is reset after a read access. The CM CANopen transmits heartbeat, emergency messages, node guarding requests, SSDOs and CSDOs, asynchronous TPDOs, and NMT commands via the low priority transmit queue.
6	A receive queue overrun occurred. Bit is reset after a read access.
7	An overrun of the high priority transmit queue occurred. Bit is reset after a read access. The CM CANopen transmits synchronous TPDOs and the SYNC message via the high priority transmit queue.

Status CM CANopen, 5000h, subindex 04h

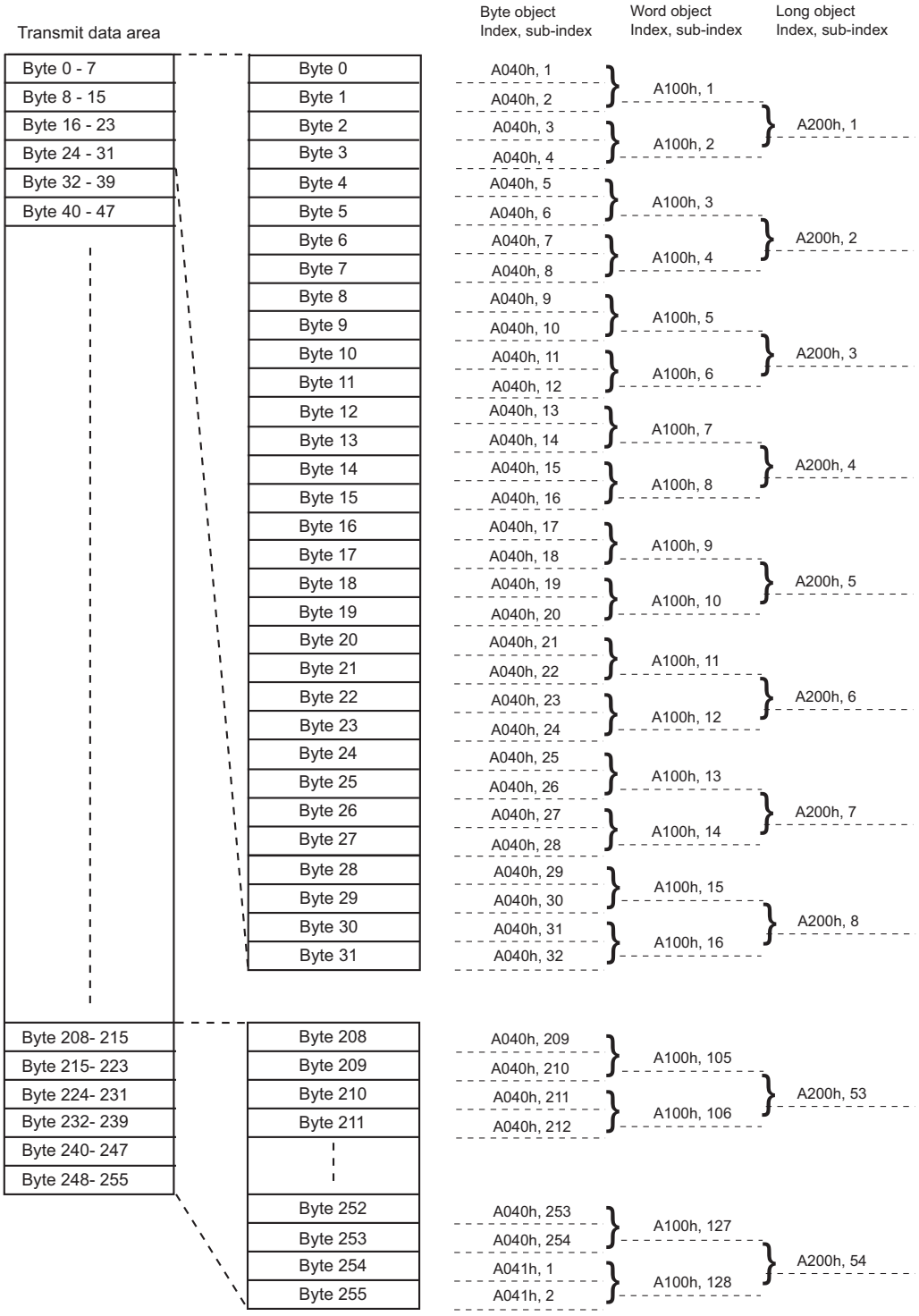
The table below gives definitions of the CM CANopen configuration bits that can be read from subindex 04h of the Status CM CANopen object. Bits 0 - 3 correspond to bits 0 to 3 of object 1F80h.

Bit No	Name	Mode	Description
0	NMT master	Both	0: The CANopen device is not NMT master. Ignore the entries of object 1F81. 1: The CANopen device is the master.
1[a]	Start all nodes	Manager	0: The CANopen slave devices are started with the NMT service Start Remote Node for each Node-ID. 1: The CANopen network is started with the NTM service Start Remote Node with Node-ID = 0.
2[a]	NMT master start	Both	0: The CANopen manager automatically enters NMT state Operational at startup. 1: The CANopen manager does not automatically enter NMT state Operational.
3[a]	Start node	Manager	0: The CANopen manager (the NMT master) will start the NMT slaves. 1: The NMT slaves will be started by the application
4	Remote error handling	Both	This bit indicates who reacts to an error control event of a mandatory device. It is always 0 if the CM CANopen module is in slave mode. 0: The application handles remote guarding or heartbeat error. 1: The CANopen manager handles remote errors, how is configured i object 1F80h, see “NMT Start-up, 1F80h” on page 31.
5	(reserved)		Always 0.
6[b]	Synchronization bit	Both	0: The value of object 1006h, Communication Cycle Period, is 0. 1: The value of object 1006h is unequal to 0.
7[b]	SYNC consumer/producer bit	Both	0: The CANopen manager is configured as SYNC consumer. 1: The CANopen manager is configured as SYNC producer.
8-9	(reserved)		Time object, not supported. Always 0.
10[c]	Node guarding stopped	Slave	0: Life guarding is configured. 1: Life guarding is not configured and not active.
11[c]	Heartbeat configured	Slave	0: The CANopen device (the module in slave mode) does not transmit heartbeat messages. 1: The CANopen device transmits heartbeat messages. Life guarding is not active.
12[c]	Node guarding configured	Slave	0: Life guarding is not configured. 1: Life guarding is configured, no guarding request has been received from the NMT master.
13[c]	Node guarding active	Slave	0: Life guarding is not configured. 1: Life guarding is configured and active.
14[c]	Life guarding status	Slave	0: Guarding request received within node lifetime. 1: Life guarding error.
15	(reserved)		Always 0.

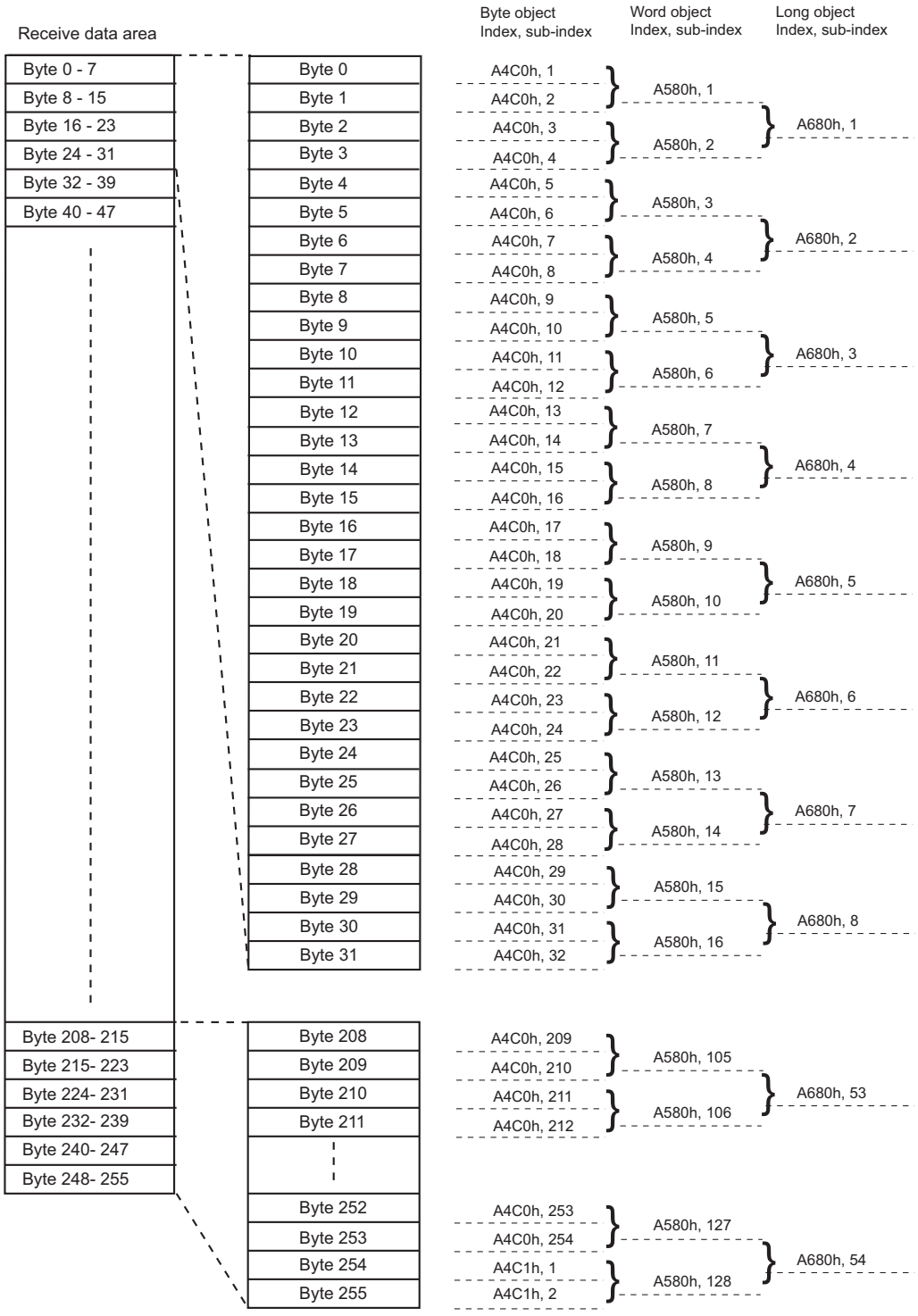
a. Bits 1 - 3 describe the startup behavior of the CANopen manager during the NMT startup procedure.
b. The synchronization mechanism is described by the values of bits 6 - 7.
c. Bits 10-14 describe life guarding status. Only one of these bits can be set at a time.

7.3.1 I/O Buffer Addresses and Object Dictionary Indices Relation

The (Q-area) application data bytes 0 - 31 are mapped to three different areas in the Local Object Dictionary. The same data is mapped to each area, but as different data types. For example: byte to index A040h, subindex 1 - 32, word to index A100h, subindex 1 - 16 and double-word (long) to indexA200h, subindex 1 - 8.



The picture shows the Transmit data area. The Receive data area is structured in the same way, see picture on next page.


Receive data area
Byte object Index, sub-index
Word object Index, sub-index
Long object Index, sub-index
Byte 0 - 7
Byte 8 - 15
Byte 16 - 23
Byte 24 - 31
Byte 32 - 39
Byte 40 - 47
Byte 208- 215
Byte 215- 223
Byte 224- 231
Byte 232- 239
Byte 240- 247
Byte 248- 255
Byte 0
Byte 1
Byte 2
Byte 3
Byte 4
Byte 5
Byte 6
Byte 7
Byte 8
Byte 9
Byte 10
Byte 11
Byte 12
Byte 13
Byte 14
Byte 15
Byte 16
Byte 17
Byte 18
Byte 19
Byte 20
Byte 21
Byte 22
Byte 23
Byte 24
Byte 25
Byte 26
Byte 27
Byte 28
Byte 29
Byte 30
Byte 31
Byte 208
Byte 209
Byte 210
Byte 211
Byte 252
Byte 253
Byte 254
Byte 255
A4C0h, 1
A4C0h, 2
A4C0h, 3
A4C0h, 4
A4C0h, 5
A4C0h, 6
A4C0h, 7
A4C0h, 8
A4C0h, 9
A4C0h, 10
A4C0h, 11
A4C0h, 12
A4C0h, 13
A4C0h, 14
A4C0h, 15
A4C0h, 16
A4C0h, 17
A4C0h, 18
A4C0h, 19
A4C0h, 20
A4C0h, 21
A4C0h, 22
A4C0h, 23
A4C0h, 24
A4C0h, 25
A4C0h, 26
A4C0h, 27
A4C0h, 28
A4C0h, 29
A4C0h, 30
A4C0h, 31
A4C0h, 32
A580h, 1
A580h, 2
A580h, 3
A580h, 4
A580h, 5
A580h, 6
A580h, 7
A580h, 8
A580h, 9
A580h, 10
A580h, 11
A580h, 12
A580h, 13
A580h, 14
A580h, 15
A580h, 16
A680h, 1
A680h, 2
A680h, 3
A680h, 4
A680h, 5
A680h, 6
A680h, 7
A680h, 8
A4C0h, 209
A4C0h, 210
A4C0h, 211
A4C0h, 212
A580h, 105
A580h, 106
A680h, 53
A4C0h, 253
A4C0h, 254
A4C1h, 1
A4C1h, 2
A580h, 127
A580h, 128
A680h, 54

8. Interface Functions

All interface functions and function blocks are available for download from www.hms-networks.com/can-for-S7-1200.

8.1 CANopen Interface Functions

8.1.1 Get Process Data In

Input Parameters

Name	Data type	Description
REQ	Bool	Starts the function. Has to be set to TRUE for one scan cycle.
ID	HW_IO	Hardware address of the CANopen module. Can be read in TIA Portal.
INDEX	Byte, Dint, Int, Sint, Uint, Usint, Word	Data record number. Should be set to 0090h.
MLEN	Byte, Usint, Usint	Maximum length in bytes of the data to be read.
RECORD	Variant	Destination area for the read data.

Output Parameters

Name	Data type	Description
BUSY	Bool	The reading process is not yet complete, if this parameter is set to TRUE.
VALID	Bool	If set to TRUE, new data record was received and is valid.
ERROR	Bool	If TRUE, an error occurred during the reading process.
STATUS	Dword	Block status or error information.[a]
LEN	Uint	Length of the data that is read.

a. For information, see the TIA Information System in TIA Portal.

Description

To read the process data from the CANopen network to the S7-1200 CPU, use the RDREC function block, and issue an request to record number 0090h. You will receive all data configured as process data on the CANopen network. The position of the data depends on the CANopen network configuration.

8.1.2 Set Process Data Out

Input Parameters

Name	Data type	Description
REQ	Bool	Starts the function. Has to be set to TRUE for one scan cycle.
ID	HW_IO	Hardware address of the CANopen module. Can be read in TIA Portal.
INDEX	Byte, Dint, Int, Sint, Uint, Usint, Word	Data record number. Should be set to 0091h.
LEN	Byte, Usint, Usint	Length in bytes of the data to be written.
RECORD	variant	Source area for the data to be written.

Output Parameters

Name	Data type	Description
BUSY	Bool	The writing process is not yet complete, if this parameter is set to TRUE.
DONE	Bool	If set to TRUE, data was transferred to the module.
ERROR	Bool	If TRUE, an error occurred during the writing process.
STATUS	Dword	Block status or error information.[a] The user has to ensure that the data area is able to hold all data as defined in the parameterization.

a. For information, see the TIA Information System in TIA Portal.

Description

To write the process data to the CANopen network from the S7-1200 CPU, use the WRREC function block, and issue an request to record number 0091h. All data will be written as process data on the CANopen network. The position of the data depends on the CANopen network configuration.

8.1.3 SDO Read/Write

General

Reading (uploading) and writing (downloading) SDOs is performed by function blocks, available for download from www.hms-networks.com/can-for-S7-1200.[1] Please note that these functions are only supported in CANopen manager mode.

Input Parameters

Name	Data type	Description
REQ	Bool	Start request. Upload/download is started if REQ = TRUE for exactly one function block call.
ID	Word	Hardware address for the module. Can be read in TIA Portal.
SLOT	Byte	Defines which channel is used on the backplane network. This parameter has to be unique for each of the SDO requests running simultaneously, as the answer to the request is put in the slot where the request was placed. Valid values: 0 - 7. **Note**: This slot is not the same as the physical slot of the module
NODE	Int	Node-ID of the CANopen module where SDO read/write is to be performed. If you use Node-ID 0, you always address the CANopen manager, even when the CANopen manager has another Node-ID.
INDEX	Word	SDO index to be read/written
SUB	Byte	SDO subindex to be read/written
DB	Dint	The number of the data block, pointing to the area where to save read data or to get data to be written. **Note**: Data blocks have to be created as Standard S7-300 compatible data blocks (not optimized).
SIZE	Uint	Number of bytes to be written. Only valid for WriteSDO.

Output Parameters

Name	Data type	Description
SIZE	Uint	Number of bytes that have been read. Only valid for ReadSDO.
BUSY	Bool	If the request isn't finished within one scan cycle, BUSY turns TRUE and stays TRUE until the request is finished, when it returns to FALSE.
RET	Uint	Error code, see "Error Codes (RET)" on page 59. Available when BUSY turns FALSE, until REQ turns TRUE.

Description

ReadSDO

Performs an SDO read on the node, index and subindex that is defined in the parameters. Returned data is saved in the area that DB points to. The block will continue to read the SDO even when the data area is filled, until the SIMATIC S7-1200 PLC indicates that the complete SDO is read. Only the predefined amount of data will be saved. SIZE holds the complete SDO size. If the size given in the parameter DATA is too small, RET will return error 2200h.

WriteSDO

Performs an SDO write on the node, index and subindex that is defined in the parameters. Data written is fetched from the area pointed to by DB.

1. Both function blocks use RDREC and WRREC for the data transfer.

8.1.4 Get Node & Network Status

Input Parameters

Name	Data type	Description
REQ	Bool	Starts the function. Has to be set to TRUE for one scan cycle.
ID	HW_IO	Logical address of the CANopen module.
INDEX	Byte, Dint, Int, Sint, Uint, Usint, Word	Data record number. Should be set to 0099h.
MLEN	Byte, Usint, Usint	Length in bytes of the data to be read. The value should correspond to the length of the data area in RECORD.
RECORD	Variant	Destination area for the read data, see table below for contents. The user has to ensure that the data area is able to hold all data as defined in the parameterization.

Output Parameters

Name	Data type	Description
BUSY	Bool	The reading process is not yet complete, if this parameter is set to TRUE.
VALID	Bool	If set to TRUE, new data record was received and is valid.
ERROR	Bool	If TRUE, an error occurred during the reading process.
STATUS	Dword	Block status or error information, see “Message Error Codes” on page 60.
LEN	Uint	Length of the data that is read.

Description

If a diagnostic message, related to the CANopen network status, is active, information on the cause of the message can be read by using the RDREC function block to issue a request to record number 0099h. You will receive the information in the parameter RECORD in the response.

Contents of Parameter RECORD

The parameter RECORD holds all diagnostic information, as described in the table below.

Offset	Data type	Description	Definition	
0	Uint8	Message error	See “Message Error Codes” on page 60	
1	Uint8	Error flags (module)	Bit:	Definition:
			0:	Bus Off
			1:	Configuration download error
			2:	Parameterization error
			3:	NVS error in module
			4-7:	(reserved)
2	Uint8	Error flags (network)	Bit:	Definition:
			0:	Network not ready
			1:	Node error control event
			2-7:	(reserved)
3	Uint8	CANopen Module mode	Value:	Definition:
			00h	Master
			01h	Slave
			02h	Transparent CAN

Offset	Data type	Description	Definition
4	Uint8	CANopen Node status	Node NMT status 00h: NMT State Unknown 04h: NMT State Stopped 05h: NMT State Operational 06h: Reset Node 07h: Reset Communication 7Fh: NMT Pre-operational
5	Array of Uint8 (up to 127 elements)	Network status (only available if module is manager)	Mainly reflects object 1F82h, subindices. 01h to 7F. Each entry reflects a node ID. Bits 0-3 Node State 0h: NMT State Unknown 1h: CANopen Device missing 4h: NMT State Stopped 5h: NMT State Operational Fh: NMT Pre-operational Bit 4: Configuration Error bit 0: No error 1: Error Bit 5: Node mandatory bit 0: Node optional 1: Node mandatory Bits 6-7: (reserved)

8.2 Transparent CAN Interface Functions

There are separate function blocks handling the transparent CAN functionality. These are available for download from www.hms-networks.com/can-for-S7-1200. The functionality is described in "Transparent CAN Mode" on page 6.

CAN_SEND and CAN_RCV handle the frames that are sent or received and CAN_CTRL controls the state and the settings of the CAN layer. The transparent mode is turned on by choosing "Transparent CAN" as operating mode during configuration in TIA Portal.

Get Diagnostic Information is used to read diagnostic information from the CM CANopen module to the PLC.

8.2.1 CAN_SEND

Input Parameters

Name	Data type	Description
REQ	Bool	Starts the block. Has to be set to TRUE for one scan cycle. Transmissions will occur until REQ is set to false.
ID	Word	Logical address of the CM CANopen module.
RTR	Bool	RTR bit of the CAN frame to send.
COBID	Word	CAN ID on the CAN net
DB	Dint	Data to be sent on the CAN net
DATASIZE	Uint	Size of the data to be sent on the CAN net
ABORT	Bool	Aborts the current transmission when set to TRUE.

Output Parameters

Name	Data type	Description
BUSY	Bool	If the function block needs more than one scan cycle, to send the frame, this output is set to TRUE. It stays TRUE until the function block is done, when it is cleared to FALSE.
RET	Uint	Error code, see "Error Codes (RET)" on page 59. Valid once the BUSY signal turns FALSE until the next time the function block is started, i.e. when REQ is set to TRUE.

Description

When REQ turns TRUE, a CAN frame is sent according to the parameters defined by the user. When the frame has been transferred to the module, BUSY will go FALSE, and a new frame can be sent.

8.2.2 CAN_RCV

Input Parameters

Name	Data type	Description
REQ	Bool	Starts the block. Has to be set to TRUE for one scan cycle.
ID	Word	Logical address of the CM CANopen module.
ACK	Bool	Set to TRUE for one scan cycle to acknowledge a newly read CAN frame.
DB	Dint	Number of data block.

Output Parameters

Name	Data type	Description
NEW	Bool	Set to TRUE when a new CAN frame has been received by the module and is ready to be read by the function block.
COBID	Word	CAN ID of the CAN frame
RTR	Bool	RTR-bit of the CAN frame
SIZE	Uint	Data size of the CAN frame
BUSY	Bool	Set to TRUE if there is data in the receive buffer when REQ is set. BUSY will be TRUE until buffer is empty.
RET	Uint	Error code, see “Error Codes (RET)” on page 59. Valid once the BUSY signal turns FALSE until the next time the function block is started, i.e. when REQ is set to TRUE.

Description

When REQ is TRUE, the block looks for new frames waiting in the CAN buffer of the module. If there are any, the module will read the oldest frame, set NEW to TRUE, and wait for an ACK from the function block. This will be repeated until all frames have been read. When all frames have been read BUSY will go FALSE. The received frames will be buffered until they are read by the PLC. The buffer can hold up to 256 frames.

For polling operation, set REQ to TRUE permanently.

8.2.3 CAN_CTRL

Input Parameters

Name	Data type	Description
REQ	Bool	Starts the block. Has to be set to TRUE for at least one scan cycle to start the block.
ID	Word	Hardware address of the CM CANopen module.
FCN	Int	Function code: 1: Set acceptance filter in the module 2: Set number of frames to store before warning (BUFFER LIMIT REACHED) 3: Clear RX buffer. No data is used 4: (reserved) 5: Reset CAN controller (implies “Clear Bus Off”). No data is used,
LEN	Uint	Only valid when FCN = 1, otherwise ignored. Defines the size of the acceptance filter array that is sent in PARAM.
DB	Dint	Parameter data block location to be sent along with the function. Only valid when FCN = 1 or 2: FCN: Contents: 1 Acceptance filter array 2 Buffer limit

Output Parameters

Name	Data type	Description
BUSY	Bool	If the function block needs more than one scan cycle to complete a command, this output is set to TRUE. It stays TRUE until the function block is done, when it is cleared to FALSE.
RET	Uint	Error code, see “Error Codes (RET)” on page 59. Valid once the BUSY signal turns FALSE until the next time the function block is started, i.e. when REQ is set to TRUE.

Description

This block is used to control the state of the transparent CAN layer and to set its parameters.

- **FCN = 1**

  Configures the CAN_RCV acceptance filter in the module i.e. what CAN-IDs will be accepted when receiving data frames. The module will not listen to the CAN bus, if no CAN-ID in the filter is enabled. The acceptance filter can be changed at any time.

  The parameter LEN is set to n, where n is the total number of valid CAN-IDs transmitted in the message (read from PARAM). One or more CAN-IDs can be transmitted in one message (ex-

cept for enable or disable all) or CAN-IDs can be transmitted in several messages after one another.

Each CAN-ID starts with a set bit, telling whether to set or reset the CAN-ID, followed by the actual CAN-ID. If the set bit is 1, the CAN-ID will be enabled in the acceptance filter, if 0 it will be disabled.

Bit 15	Bit 14	Bit 0
Set bit	CAN-ID	

CAN-ID 0000h with LEN set to 0000h will disable all CAN-IDs. If LEN is set to 1 only CAN-ID 0000h is disabled.

CAN-ID FFFFh with LEN set to 0001h will enable all CAN-IDs and the module will forward all messages to the PLC.

- **FCN = 2**

  Configures the CAN_RCV buffer limit warning value. It gives the opportunity to define what number of frames will be stored in the receive buffer, before status code bit 4 (BUFFER LIMIT REACHED) will be set, see "Transparent CAN Status Field" on page 53. Initially the buffer limit is set to 256, which means that the status bit will be set when the last place in the buffer is used. The user can change the buffer limit to any value between 1 and 256.

- **FCN = 3**

  Empties the receive buffer. CAN_RCV still holds the old frame that hasn't been acknowledged (NEW is cleared by holding ACK high for on scan cycle)

- **FCN = 5**

  Resets the CAN controller. Clears Bus Off condition, if present.[1]

1. If a Bus Off condition is generated, the CAN Controller has to be reset, before communication can be resumed

8.2.4 Get Diagnostic Information

Input Parameters

Name	Data type	Description
REQ	Bool	Starts the function. Has to be set to TRUE for one scan cycle.
ID	HW_IO	Logical address of the CANopen module.
INDEX	Byte, Dint, Int, Sint, Uint, Usint, Word	Data record number. Should be set to 0092h.
MLEN	Byte, Usint, Usint	Length in bytes of the data to be read. The value should correspond to the length of the data area in RECORD.

Output Parameters

Name	Data type	Description
BUSY	Bool	The reading process is not yet complete, if this parameter is set to TRUE.
VALID	Bool	If set to TRUE, new data record was received and is valid.
ERROR	Bool	If TRUE, an error occurred during the reading process.
STATUS	Dword	Block status or error information, see “Message Error Codes” on page 60.
LEN	Uint	Length of the data that is read.
RECORD	Variant	Destination area for the read data, see table below for contents. The user has to ensure that the data area is able to hold all data.

Description

If a diagnostic message, related to the Transparent CAN network status, is active, information on the cause of the message can be read by using the RDREC function block to issue a request to record number 0092h. You will receive the information in the parameter RECORD in the response.

Contents of Parameter RECORD

The parameter RECORD holds all diagnostic information, as described in the table below.

Offset	Data type	Description	Definition
0	Uint16	Number of Errors	Current number of errors in the module
2	Uint16	Diagnostic Error 1	Error code, see “Error Codes” on page 52
4	Uint16	Data length Error 1	Length of additional information
6 -	Unit16	Additional error information	Additional information for Error 1
...	...	...	Additional errors with the same structure as Error 1.

Error Codes

Error Code	Description	Definition	Additional Information
3	BUSOFF error	The CAN controller of the CM CANopen has reached a BUS OFF state, e.g. due to a bad network connection or a faulty baud rate.	
10	Transparent CAN event	When running Transparent CAN mode this diagnostic is triggered by an event in the CM CANopen module.	See “Transparent CAN Status Field” on page 53 for additional error information in the status field.

Transparent CAN Status Field

The transparent CAN status field consists of 16 bits, where each bit represents a status code. Multiple bits can be set at the same time to indicate multiple status codes.

Bit no.	Description
0	BUS OFF
1	ERROR PASSIVE
2	BUFFER FULL
3	BUFFER HALF FULL
4	BUFFER LIMIT REACHED
5	RECEIVE MESSAGE LOST
6	TRANSMIT QUEUE: HALF FULL
7	TRANSMIT QUEUE: WARNING LIMIT
8-15	(reserved, set to 0)

The status code 0000h indicates 'SUCCESS', while e.g. 0024h indicates 'RECEIVE MESSAGE LOST' and 'BUFFER FULL'.

A. Technical Specification

A.1 Mechanical Properties

Housing

Plastic housing, plug-in module, protection class IP20

Dimensions

Width: 30 mm (1.18")

Weight: 120 g (0.26 lb.)

A.2 Environmental Characteristics

Relative Humidity

Operating: 15% to 95% non-condensing
Non operating: 5% to 95% non-condensing

Temperature

Operating:	0 °C to +60 °C (+32°F to +140 °F)
Non operating:	-40 °C to +70 °C (-40 °F to +158 °F)

A.3 Regulatory Compliance

CANopen Conformance

CiA 301 rev. 4.2

CAN Standard

Supports CAN 2.0A

EMC Compliance

This product is in accordance with the EMC directive 2004/108/EC, through conformance with the following standards:

- SS-EN 55016-2-3:2007 Class A
- SS-EN 61131-2:2007 Clause 8

UL/c-UL compliance

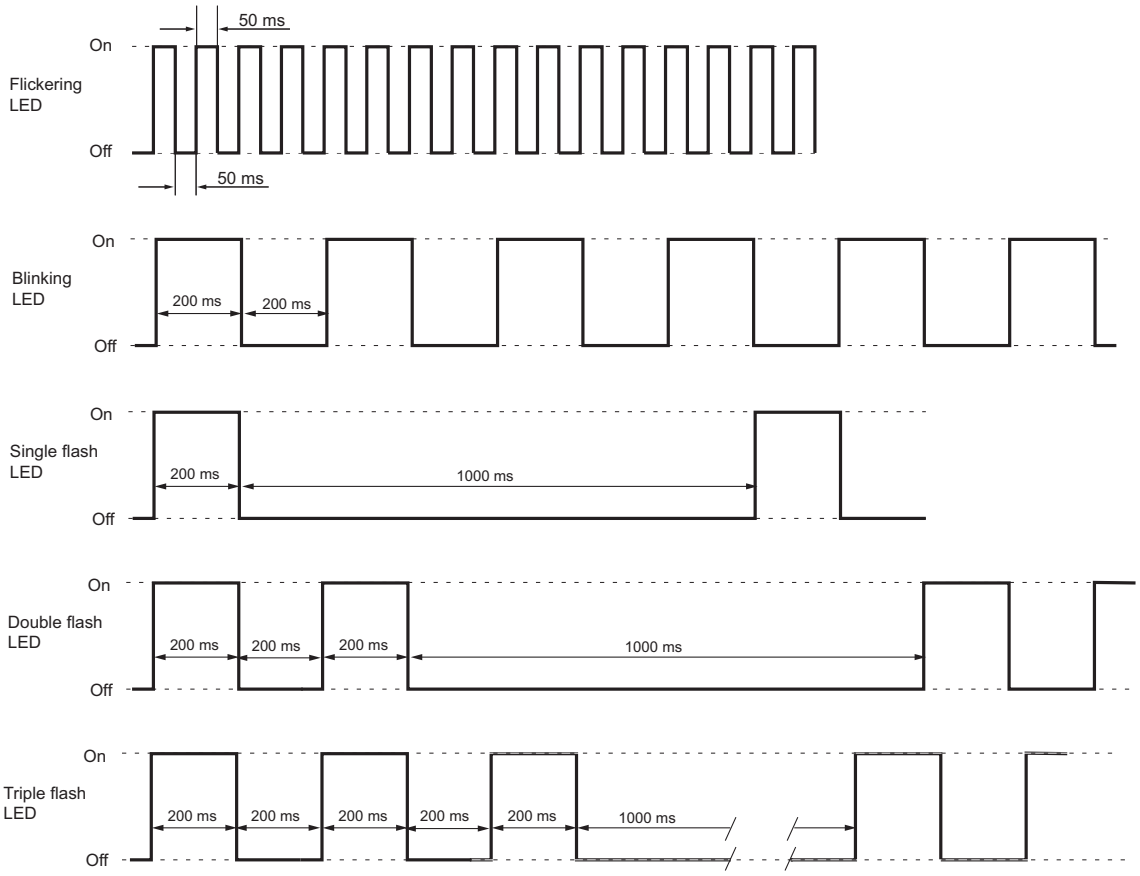
The certification has been documented by UL in file E214107.

Galvanic isolation on sub-network interface

- SS-EN 61131-2:2007:
  Pollution Degree 2
  Material Group IIIb
  250 V_S or 250 VDC Working voltage
  500 V Secondary circuit transient rating

B. Status LED Timing Diagrams

The LEDs on the front of the module change their behavior according to the status of the module. This appendix gives the timing diagrams for the different indications, described in "Status LEDs" on page 8.



C. CANopen Emergency Codes

Below is a list of the CANopen emergency codes that can be produced by the CM CANopen Module for SIMATIC S7-1200. The five latest emergencies, reported by the module, can be read from the list in object 1003h, see page 26.

Emergency Code	Description
0000h	Error reset or no error
1000h	Generic error
2000h	Current – generic error
2100h	Current, CANopen device input side – generic
2200h	Current inside the CANopen device – generic
2300h	Current, CANopen device output side – generic
3000h	Voltage – generic error
3100h	Mains voltage – generic
3200h	Voltage inside the CANopen device – generic
3300h	Output voltage – generic
4000h	Temperature – generic error
4100h	Ambient temperature – generic
4200h	Device temperature – generic
5000h	CANopen device hardware – generic error
6000h	CANopen device software – generic error
6100h	Internal software – generic
6200h	User software – generic
6300h	Data set – generic
7000h	Additional modules – generic error
8000h	Monitoring– generic error
8100h	Communication – generic
8110h	CAN overrun (objects lost)
8120h	CAN in error passive mode
8130h	Life guard error or heartbeat error
8140h	Recovered from bus off
8150h	CAN Id error
8200h	Protocol error – generic
8210h	PDO not processed due to length error
8220h	PDO length exceeded
8F00h	Error Control or Heartbeat Consume error
9000h	External error – generic error
F000h	Additional functions – generic error
FF00h	Device specific – generic error
FF10h	Missing an assigned slave
FF20h	Identity error of an assigned slave
FF30h	cDCF Mismatch: concise DCF of 1F22h does not match the object dictionary of the booted slave
FF40h	Formal inconsistent cDCF in 1F22h of an assigned slave
FF50h	SDO error detected by the boot slave process: SDO time out or SDO error except cDCF mismatch: See FF30h
FF51h	SDO timeout detected: A request to write an SDO to the module timed out
FF60h	Reset timeout of a booted slave
FF70h	Presence of a device that uses the node id of the CM
FF80	Presence of a device that is not assigned as slave

Emergency Code	Description
FFA0h	PLC has entered STOP state while network is in OPERATIONAL. Module goes to PRE-OPERATIONAL. (Only in slave mode)

These codes conform to the CANopen standard.

D. Error Codes (RET)

D.1 Function Block Error Codes

Each function block has a separate set of error codes, that are returned when an error is detected.

CAN_CTRL	
Error code[a]	**Description**
109B	Function code (FCN) not allowed
109C	If FCN=1: Length of filter is too big (> 2048)
1013	Read response failed after 1000 tries
3100-3799	Internal read/write error
39XX	CM message error (XX = Message error code, see “Message Error Codes” on page 60)
FF00	Error code missing

a. All error codes are given in hexadecimal format.

CAN_RCV	
Error code[a]	**Description**
1013	Read response failed after 1000 tries
3100-3799	Internal read/write error
38XX	Too many frames (XX = No of frames)
39XX	CM message error (XX = Message error code, see “Message Error Codes” on page 60)
FF00	Error code missing

a. All error codes are given in hexadecimal format.

CAN_SEND	
Error code[a]	**Description**
1013	Read response failed after 1000 tries
1098	Wrong COB-ID
109A	Wrong data size
3100-3799	Internal read/write error
39XX	CM message error (XX = Message error code, see “Message Error Codes” on page 60)
FF00	Error code missing

a. All error codes are given in hexadecimal format.

ReadSDO	
Error code[a]	**Description**
1013	Read response failed after 1000 tries
1022	Too much SDO data
104A	Wrong slot number
3100-3799	Internal read/write error
38XX	SDO error (XX = SDO error code)
39XX	CM message error (XX = Message error code, see “Message Error Codes” on page 60)
FF00	Error code missing

a. All error codes are given in hexadecimal format.

WriteSDO	
Error code[a]	**Description**
1013	Read response failed after 1000 tries
104A	Wrong slot number
3100-3799	Internal read/write error
38XX	SDO error (XX = SDO error code)
39XX	CM message error (XX = Message error code, see “Message Error Codes” on page 60)
FF00	Error code missing

a. All error codes are given in hexadecimal format.

D.2 Message Error Codes

Error code (hex)	**Description**
00	No error
01	Invalid command
02	Module not initialized
03	Invalid Node ID
04	Invalid length
05	Invalid baud rate
06	General stack init error
07	Invalid NMT command
08	Module in slave mode
09	SDO fragmentation error
0A	Wrong Mode
0B	Wrong Backplane
0C	Invalid Function Code
0D	Invalid communication channel
0E	Communication channel is busy - try again
0F	State error (Command cannot be requested in the current state)
10	Parameter error (FCN 1: CAN-ID > 0x7FF FCN 2: Value > maximum Number of storable CAN Receive Messages CAN_SND: CAN-ID > 0x7FF)
11	Send error (CAN_SND: Message cannot be sent)
12	No command pending - RDREC was received without preceding WRREC
FE	Command pending - WRREC was received but no answer is ready yet
FF	General error

Error code[a]	**Additional error code**[a]	**Description**	**SFC/SFB number**[b]	**RET_VAL**[b]
00	00	No error	-	-
11	00	Nesting depth exceeded	SFC20	8091
12	00	Temporary resource bottle neck	SFC23	80C1
13	00	Number of retries to read SDO exceeded	-	-
14	00	Number of retries to write acyclic request exceeded		
21	00	Failed to create DB		
22	00	Data did not fit in the specified data area. Try increasing the size		

Error code[a]	Additional error code[a]	Description	SFC/SFB number[b]	RET_VAL [b]
31	00	Specified logical address is invalid. There is no assignment in SDB1/SDB2x or there is no base address.	SFB52, SFB53, SFC14, SFC15	8090
32	00	ID mismatch (the following modules are permitted: S7-300 modules for an S7-300, S7-400 modules for an S7-400, S7-DP modules for an S7-300 and S7-400).	SFB52, SFB53	8093
33	00	Negative acknowledgement when communicating with the module. • The module was removed during the communication or is defective • Unilateral I/O of the standby CPU not available (with H-systems). For example, standby CPU is in state STOP.		80A0
34	00	The configured slot is not occupied		80B2
35	00	Actual module type does not match the required module type in SDB1		80B3
36	00	• PROFIBUS DP: DP protocol error with user interface/user • PROFINET IO: General CM error		80A3
37	01	• DP protocol error at layer 2 (for example, slave failure or bus problems) • Data record can not be read in DPV0 mode (ET200S)		80A2
	02	Communication problem on the communication bus		80A4
	03	Previous write job for the same data record has not been processed yet		80C1
	04	Module is processing maximum number of jobs for a CPU		80C2
	05	The required resources (memory etc.) are currently occupied		80C3
	06	Internal temporary error. The job could not be processed. • Repeat the job. If this error occurs frequently check your system for electrical disturbances		80C4
	07	Distributed I/Os not available		80C5
	08	Data record transfer was stopped due to a priority class abort (restart or background)		80C6
38	-	Error while communicating with the CANopen manager. The additional error code is equal to the error code from the CANopen manager.	-	-
	00h	No error		
	01h	Time out		
	02h	Out of memory		
	03h	Unsupported access		
	04h	Write only		
	05h	Read only		
	06h	Unsupported index		
	07h	No PDO mapping		
	08h	Exceeded PDO length		
	09h	Unsupported subindex		
	0Ah	Invalid data type		
	0Bh	Value too high		
	0Ch	Value too low		
	0Dh	Bad device state		
	0Eh	General error		
	0Fh	Can not be stored		
	10h	Can not be stored locally		
	11h	OD file generation failed		
	3Ah	Transparent CAN function block error		

Error code[a]	Additional error code[a]	Description	SFC/SFB number[b]	RET_VAL [b]
39	-	Acyclic error. Additional error code equals error code from the CANopen manager.	-	-
	00h	No error		
	01h	Invalid command		
	02h	Module not initialized		
	03h	Invalid Node-ID		
	04h	Invalid length		
	05h	Invalid baud rate		
	06h	General stack init error		
	07h	Invalid NMT command		
	08h	Module in slave mode		
	09h	SDO fragmentation error		
	0Ah	Wrong Mode		
	0Bh	Wrong Backplane		
	0Ch	Invalid function code		
	0Dh	Invalid communication channel		
	0Eh	Communication channel is busy - try again		
	0Fh	State error (command can not be requested in the current state)		
	10h	Parameter error FCN 1: CAN-ID > 7FFh FCN 2: Value > maximum number of storable CAN receive messages CAN_SND: CAN-ID > 7FFh		
	11h	Send error CAN_SND: Message can not be sent)		
	FFh	General error		
41	0X	DB could not be deleted because • the “compress user memory” is currently active • the DB to be deleted is currently being copied from the CPU to an offline project • the H CPU is running coupling or update functions • the WinAC Software CPU has detected an error in the operating system of the computer where WinAC is installed X specifies the offset from DBSTART where the error occurred	SFC23 (for 4100)	8092 (for 4100)
42	0X	DB could not be created because • the “compress user memory” is currently active • the number of DBs in the CPU has already reached the maximum possible number • the H CPU is running coupling or update functions • the WinAC Software CPU has detected an error in the operating system of the computer where WinAC is installed • the previous delete action is not completed X specifies the offset from DBSTART where the error occurred	SFC22 (for 4200)	8092 (for 4200)
43	0X	DB number exceeds the maximum number permitted for the CPU. X specifies the offset from DBSTART where the error occurred	SFC24	80A1
44	00	DB number entered is less than or equal to 0	-	-

Error code[a]	Additional error code[a]	Description	SFC/SFB number[b]	RET_VAL [b]
45	1X	DB could not be deleted because it was created with the keyword UNLINKED (S7-400 only). X specifies the offset from DBSTART where the error occurred	SFC24	80B2
	2X	DB could not be deleted because it is stored on a flash card. X specifies the offset from DBSTART where the error occurred	SFC24	80B3
	3X	DB can not be deleted. Possible causes: • It belongs to an F-program • It is an instance DB of a block for S7 communication (S7-400 only) • It is a technology DB X specifies the offset from DBSTART where the error occurred	SFC24	80B4
	4X	The DB with the specified number does not exist in the CPU. X specifies the offset from DBSTART where the error occurred	SFC24	80B1
46	1X	There is not enough free memory available. X specifies the offset from DBSTART where the error occurred	SFC22	80B2
	2X	There is not enough continuous memory space available. Compress memory. X specifies the offset from DBSTART where the error occurred	SFC22	80B3
47	0X	The length of the DB is not equal to the DBSIZE input parameter and has not been created using the download block. X specifies the offset from DBSTART where the error occurred	-	-
48	00	DBSIZE must be in the range 262 - 50000	-	-
49	00	DBSIZE must be an even number	-	-
4A	00	Slot must be in range 0 - 7	-	-
51	00	No DP module/PROFINET IO device from which you can read consistent data exists at the logical address specified in LADDR.	SFC14, SFC15	8093
52	00	Access error detected while I/O devices were being accessed.	SFC14	80A0
			SFC15	80A1
53	00	Slave failure on external DP interface module.	SFC14, SFC15	80B0
54	00	The length of the specified destination area is not identical to the user data length configured with STEP 7.	SFC14, SFC15	80B1
55	00	The data of the previous read/write job on the module have not yet been processed by the module.	SFC14	80C0
			SFC15	80C1
56	00	System error with external DP interface module.	SFC14, SFC15	808X, 80B2, 80B3, 80C2, 80FX, 87XX

Error code[a]	Additional error code[a]	Description	SFC/SFB number[b]	RET_VAL [b]
91	00	Parameter BPSIZE contains invalid value	-	-
92	00	IN parameter is not of type BYTE		
93	00	OUT parameter is not of type BYTE		
94	00	Illegal syntax ID of the IN parameter (valid value is 10h)		
95	00	Illegal syntax ID of the OUT parameter (valid value is 10h)		
96	00	Either • the size of the IN parameter exceeds 1024 bytes or • the size of the IN parameter exceeds the maximum size for the chosen backplane size (512 for 4 byte backplane, 1024 for the rest).		
97	00	Either • the size of the OUT parameter exceeds 1024 bytes or • the size of the OUT parameter exceeds the maximum size for the chosen backplane size (512 for 4 byte backplane, 1024 for the rest)		
98	00	COBID parameter contains invalid value		
99	00	DATA parameter is not of type BYTE		
9A	00	DATA parameter is of illegal size		
9B	00	Parameter FCN contains invalid value		
9C	00	Parameter LEN contains invalid value		
9D	00	PARAM parameter is not of type WORD		
9E	00	PARAM parameter is of illegal size		
FF	00	Internal error		

a. All error codes are given in hexadecimal format.
RET high byte: Error code
RET low byte: Additional error code

b. The SFC/SFB number and the RET_VAL value gives the origin of the error code that is generated by the CM CANopen function blocks if the error originates from outside the module. Please consult the configuration tool manual for further information.