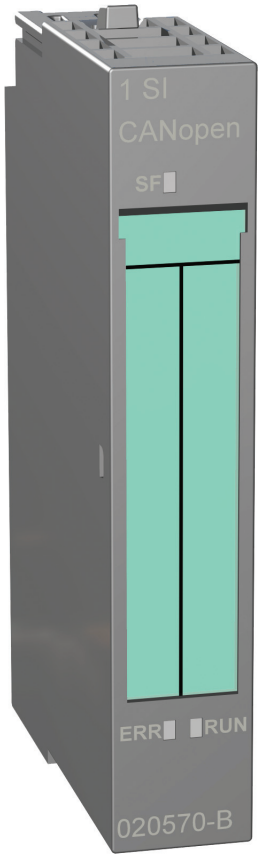


User Manual
1 SI CANopen
CANopen Module for ET200S

Doc.Id. SCM-1200-005
Rev. 1.32

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 1 SI CANopen Module for ET200S.

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

All 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.

1 SI CANopen Module for ET200S User Manual
Rev 1.32

Mar 2012 Doc Id SCM-1200-005

Table of Contents

Chapter 6 CANopen Module Specification

Chapter 7 Supported Objects

Chapter 8 CANopen Function Blocks

Chapter 9 Transparent CAN Function Blocks

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 1 SI CANopen Module for ET200S, 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 Draft Standard 301 v4.2	CAN in Automation
CiA Draft Standard Proposal 302 Part 1-5	CAN in Automation
SIMATIC STEP7 manual	Siemens
CiA Draft Standard 305	CAN in Automation
SIMATIC Isochrone Mode Function Manual, doc. id. A5E00223279-02	Siemens

Please visit the support pages at the 1 SI CANopen website at www.et200can.com for more documents.

P.3 Document History

Summary of Recent Changes (1.31 ... 1.32)

Change	Page(s)
Updated sales and support information	3, 4
References to application notes added	45, 50
Updated emergency code 6161h table	60
Updated Slave Assignment object (1F81h)	38
Corrected object 1016h in object dictionary list	33
Changed wording in Boot Time object	41
Updated chapter CANopen Emergency Codes	60

Revision List

Revision	Date	Author	Chapter	Description
1.00	2009-06-26	KeL	-	First release
1.10	2009-10-08	KeL	7, 8, 9	Misc minor updates and three new chapters
1.20	2010-06-15	KeL	All	Misc updates, new mode and revision of configuration example
1.30	2010-11-17	KeL	All	Misc updates and corrections
1.31	2011-08-31	KeL	2, 4, 6, 8, 9	Misc minor updates and corrections
1.32	2012-03-21	KeL	P, 7, 8, E	Misc updates and corrections

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 1 SI CANopen Module for ET200S 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 1 SI CANopen Module for ET200S
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

P.6 Sales Enquiries

For sales enquiries contact:

Direct Sales Germany:
Tel: +49 721 989 777 - 201
E-mail: de-sales@et200can.com

Direct Sales China:
Tel: +86 10 8532 3183
E-mail: cn-sales@et200can.com

Direct Sales Japan:
Tel: +81 45 478 5340
E-mail: jp-sales@et200can.com

Direct Sales France:
Tel: +33 3 68 368 034
E-mail: fr-sales@et200can.com

Direct Sales Italy:
Tel: +39 039 59662 27
E-mail: it-sales@et200can.com

Direct Sales USA:
Tel: +1 312 829 0601
E-mail: us-sales@et200can.com

Direct Sales India:
Tel: +91 (0) 20 40111201
E-mail: in-sales@et200can.com

Direct Sales All Other Countries:
Tel: +46 35 17 29 56
E-mail: sales@et200can.com

P.7 Support

For technical support contact:

Support Germany:
Tel: +49 721 989 777 - 301
E-mail: de-support@et200can.com

Support China:
Tel: +86 10 8532 3023
E-mail: cn-support@et200can.com

Support France:
Tel: +33 3 68 368 033
E-mail: fr-support@et200can.com

Support North America:
Tel: +1 312 829 0601
E-mail: us-support@et200can.com

Support Italy:
Tel: +39 039 59662 27
E-mail: it-support@et200can.com

Support Japan:
Tel: +81 45 478 5340
E-mail: jp-support@et200can.com

Support India:
Tel: Tel: +91 (0) 20 40111201
E-mail: in-support@et200can.com

Support All Other Countries:
Tel: +46 35 17 29 20
E-mail: support@et200can.com

P.8 All Other Issues

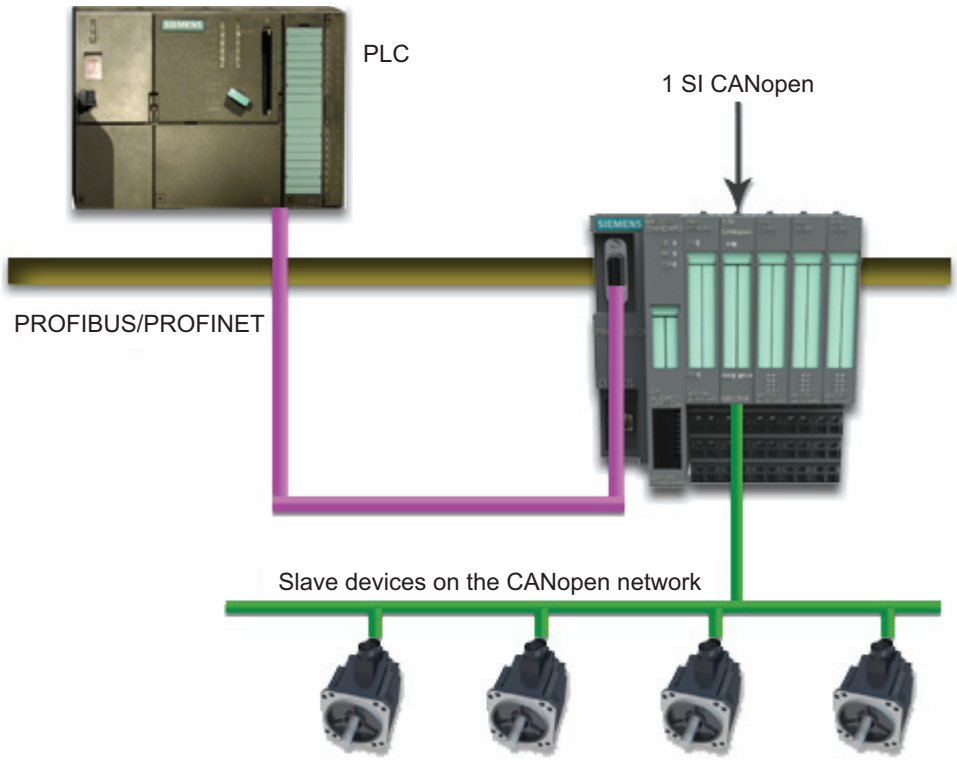
E-mail: info@et200can.com

or visit http://www.et200can.com

1. Product Overview

1.1 Introduction

The 1 SI CANopen Module for ET200S is a plug-in IP20 compliant communication module designed to be used as an accessory to an ET200S Distributed I/O System. It provides the user with the possibility to connect a CANopen network to an ET200S Distributed I/O System.The figure below shows an example, where the ET200S Distributed I/O System Rack is configured from a PROFIBUS/ PROFINET PLC to act as a slave on a PROFIBUS or PROFINET network.



Some versions of the ET200S Distributed I/O System have an internal CPU and can also be used as a standalone system, in which case this internal CPU can be used to configure the unit and the modules attached to it.

The 1 SI CANopen Module for ET200S can be configured either as a CANopen master or as a slave, depending on the application. The network and the module are configured using external configuration tools (SIMATIC Step 7 and a CANopen network configuration tool). See “Configuration” on page 12.

1.2 Features

- Powerful CANopen module for SIMATIC ET200S Distributed I/O
- Connects up to 126 CANopen slave nodes in master mode
- CANopen slave functionality supported
- 3 LEDs provide diagnostic information on module, network and I/O status
- Module “hot swap” supported which allows replacement of expansion modules while the device is active on the network without removing power
- Isochrone mode supported
- The module can be integrated in the hardware catalogue of STEP7
- CANopen configuration via external tool
- Complies to the CANopen communication profile DS301 rev. 4.2 and the CANopen DSP302 (part 1-5)
- Transparent CAN supported

2. CANopen Fieldbus Functionality

The functionality of the 1 SI CANopen Module for ET200S is defined by the CANopen DS301 Rev. 4.2 specification and the CANopen DSP302 (part 1-5). The module can be configured either as a Master 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 1 SI CANopen Module for ET200S:

- NMT (Network Management)[1] messages configure and initialize the network, as well as monitor the network and handle errors. If the module is configured as a slave, startup is performed by the master.
- CMT (Configuration Manager)[1] messages are used for configuration of 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 master.
- PDOs (Process Data Objects) are used for I/O communication. There are 128 Receive PDOs and 128 Transmit PDOs implemented in the 1 SI CANopen Module for ET200S that each can transfer up to 8 bytes. Supported PDO message types are COS (Change of state), 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 (master mode). The 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 Segmented Upload/Download Protocol are supported)
- A SYNC (Synchronization Object) is used for synchronizing PDO communication. A master 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 master. Slaves can be configured to expect a node guarding request from the master.
- An EMCY (Emergency Object) is used for error reporting when a fatal fault has occurred in the slave.
- LSS (Layer Setting Services)[1]. As LSS master, the module can configure baud rate and node ID of all slaves that support LSS (i.e. the pre-configured baud rate and node ID of a slave can be changed by a master). The module can not act as an LSS slave.

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

3. Transparent CAN Mode

The 1 SI CANopen Module for ET200S integrates Transparent CAN 2.0A (11 bit identifiers). With this functionality, 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 pre-defined CAN frames from the CAN network.

CAN 2.0A is the underlying protocol of CANopen, directly working on the bus.When the application disables CANopen functionality, it still retains access to this protocol. CANopen can be disabled at any time, but if the user wants to re-enable CANopen functionality, the 1SI CANopen module has to be restarted.

Separate PLC function blocks for this functionality are available for download from www.et200can.com. The function blocks are described in "Transparent CAN Function Blocks" on page 50. The blocks give the designer of the PLC program the possibility to interpret/generate any CAN frame directly in the PLC program, thereby making it possible to implement customer specific CAN protocols.

When CANopen functionality is disabled, the only functionality available is the transmitting and receiving of CAN data frames on the bus. The frames are tagged with an identifier. Only received frames with an identifier specified during setup 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.

Note 1: Only 16 and 32 byte modules support CAN 2.0A.

Note 2: The receive buffer can hold up to 255 unread received messages

4. Installation

The mounting and configuration of the 1 SI CANopen Module for ET200S is done following these steps:

1. Mounting

2. Configuring the ET200S Distributed I/O System to include the module (See "ET200S Distributed I/O System Configuration" on page 13)

3. Setting the parameters of the module (See "ET200S Distributed I/O System Configuration" on page 13)

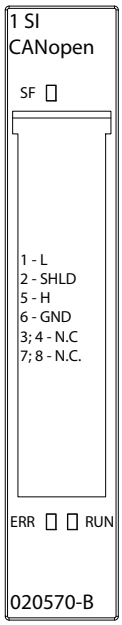
4. Configuring the CANopen network, including the module (See "CANopen Network Configuration" on page 15)

The following items are needed to perform the installation:

- Siemens SIMATIC STEP7 tool, V5.4 SP 5 or later
- HSP (STEP7 configuration file) or GSD files for the module[1],[2]
- EDS file for the 1 SI CANopen Module[1]
- CANopen configuration tool
- Function blocks (optional)[1]

4.1 Mounting

The 1 SI CANopen Module for ET200S is designed to be plugged directly into a slot in an ET200S Distributed I/O System rack. The module exchanges data and signals via the backplane bus. Power is supplied by the 24 VDC of the loadgroup voltage, coming from the power module PM-E.



Please confirm that the contacts[3] below the slot are compatible with the CANopen contacts defined on the module.

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

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

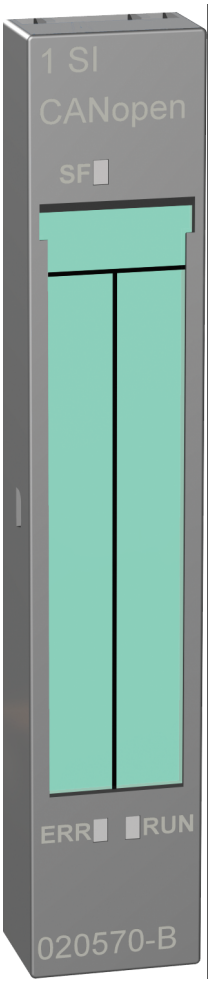
The module can be replaced with a new one without removing power from the rack or the network. The new module can be configured from scratch, or be configured using the previously stored configuration.

See also:

- "CANopen Network Configuration" on page 15
- "Configuration Upload/Download" on page 48

1. Can be downloaded from www.et200can.com
2. HSP is needed if the 1 SI CANopen Module is inserted in a CPU-based ET200S rack.
3. For usable terminal modules see "ET200S Terminal Modules" on page 59

4.2 Status LEDs



Three status LEDs on the front indicate the status of the module as shown in the table below. Their behavior is described in "Status LED Timing Diagrams" on page 57

LED	Indication	Status
SF - Group error (red)	Off	Normal operation
	On	Diagnostic error indication See section 6.4.4 on page 31 for further information.
ERR - CANopen error (red)	Off	Normal operation
	Flickering	The LSS services are in progress.
	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
	Flickering	The LSS services are in progress.
	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. This is often included in CANopen cables. If the cable you are using is not terminated, connect a line termination between contacts 1 and 5 (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.).

4.5 Implementation Notes

4.5.1 Isochrone mode

In an isochronous network, all nodes are synchronized to a system clock. Well-defined delays are applied to avoid jitter, so that any event will happen in real time with a constant well-defined time difference to other events. Time intervals for input and output data will always be constant allowing for fast and safe transmission. Isochrone mode is implemented in the ET200S Distributed I/O System to enforce this behavior.[1]

The 1 SI CANopen Module for ET200S supports isochrone mode. The module cooperates in isochrone mode with the PROFIBUS/PROFINET master and all other slave units on that network. However the module does not forward the synchronization, enforced in isochrone mode, to the CANopen network it controls.

The 1 SI CANopen Module for ET200S attempts to activate isochrone mode by default, without any action from the user.

1. Please refer to the SIMATIC Isochrone Mode Function Manual for information. This manual is available from http://support.automation.siemens.com.

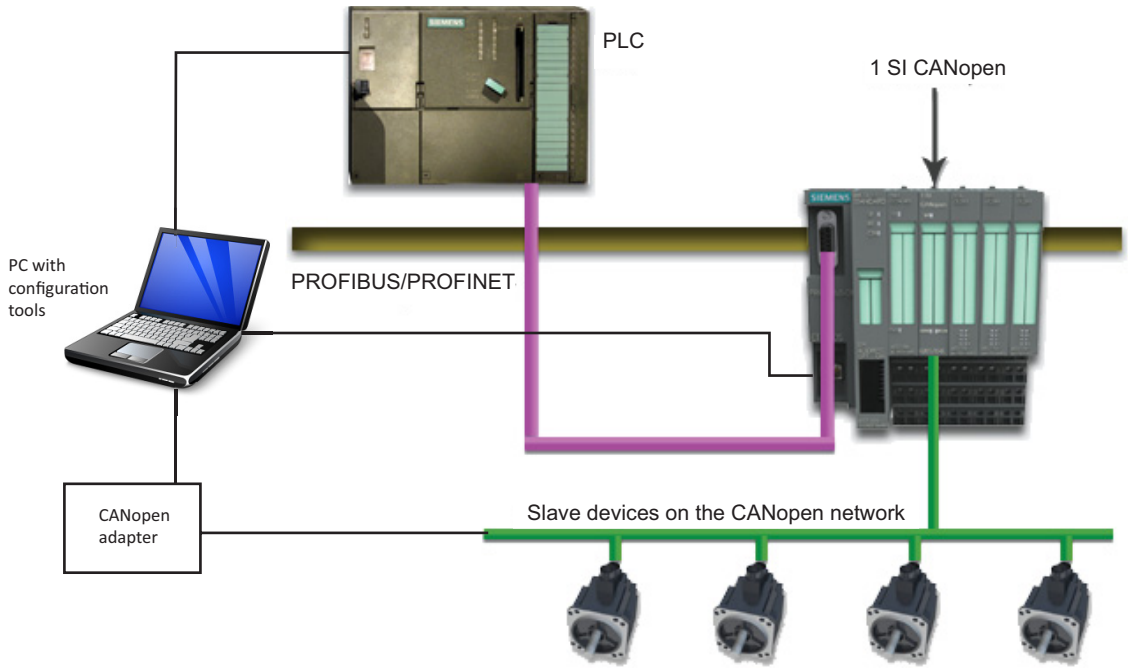
5. Configuration

5.1 General

Configuration of the ET200S Distributed I/O System rack to include the 1 SI CANopen Module for ET200S can be done in different ways, depending on the application:

1. The rack can be part of a PROFIBUS/PROFINET network. In this case the module can be configured through the PROFIBUS/PROFINET master PLC to be included in the PROFIBUS/PROFINET network.
2. If the ET200S Distributed I/O System includes an internal CPU, this can be used, in combination with the PLC, to configure the 1 SI CANopen Module for ET200S.
3. When the ET200S Distributed I/O System includes an internal CPU, it can also be used as a standalone rack. The module is configured by downloading the configuration to this CPU instead of to the PLC.

The CANopen network, and the behavior of the module on this network, have to be configured using an external CANopen configuration tool[1] on the PC and the configuration is downloaded to the CANopen network through a CANopen adapter.[1]



The size of the data block, transferred from the PLC to the 1 SI CANopen Module for ET200S, depends on the ET200S Distributed I/O System rack and will be either 4, 8, 16 or 32 bytes. If an application sends larger messages than the rack is configured for, the messages can be fragmented and sent in several packages. This implies slower communication. A function block, that can be downloaded from www.et200can.com to the PLC or to the ET200S CPU, will handle the fragmentation, see “Fragmentation Handling” on page 47.

Maximum CANopen I/O size also depends on the ET200 Distributed I/O System rack. Depending on the rack the module can be configured to different block sizes, ranging from 4 to 32 bytes as shown in

1. Please contact HMS support for further information, see “Support” on page 4.

the table below. Without fragmentation enabled, the data block size will limit the amount of data transmitted/received.

Configured Rack I/O block size (bytes)	Maximum CANopen I/O size (bytes)
4	512
8	1024
16	1024
32	1024

For example, if the configured I/O block size of the rack is 4 bytes, the maximum allowed amount of CANopen I/O without fragmentation is 4 bytes and with fragmentation 512 bytes. For more information see "Fragmentation Handling" on page 47.

For data consistency for different racks, see "Siemens Interface Modules Compatibility" on page 58.

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

5.2 ET200S Distributed I/O System Configuration

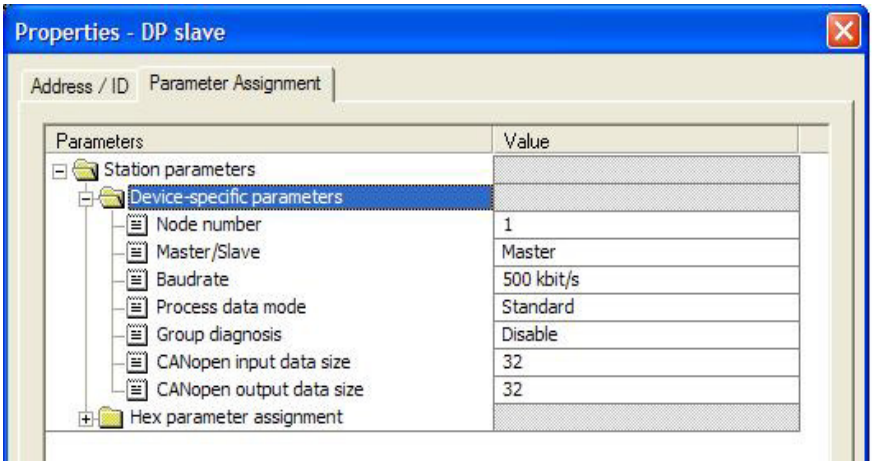
The Siemens SIMATIC STEP7[1] tool is used to configure the ET200S Distributed I/O System rack and the 1 SI CANopen Module for ET200S for integration into the rack. The module is included in the SIMATIC STEP7 tool. For information on which ET200S Distributed I/O System versions the 1 SI CANopen Module for ET200S are compatible to, see "Siemens Interface Modules Compatibility" on page 58.

The ET200S Distributed I/O System appears as a slave on the PROFIBUS/PROFINET network if it is configured from the PROFIBUS/PROFINET PLC. It can appear either as a master or as a slave on the CANopen network depending on the settings in the parameter list.

A example on how to configure the system using SIMATIC STEP7 is given on page 16.

Right-clicking on the 1 SI CANopen Module for ET200S in the list of modules in the rack will allow the user to define the parameters of the module.

1. STEP7 V5.4 SP 5 or later



The following parameters have to be defined

Parameter	Description	Values	Default value
Node Number	Node ID on the CANopen Network. Allowed values are 1 - 127	1-127	1
Master/Slave	The module can be configured either as a master or as a slave on the CANopen network	Master Slave	Master
Baud rate	This parameter defines the baud rate on the CANopen network. If “Auto” is selected the baud rate will be automatically detected.	20 kbit/s 50 kbit/s 125 kbit/s 250 kbit/s 500 kbit/s 800 kbit/s 1000 kbit/s Auto	500 kbit/s
Process data mode	Data can be processed either in standard mode or fragmented mode, see page 12 and page 47.	Standard Fragmented	Standard
Group diagnosis	This parameter enables/disables the diagnostic information from the 1 SI CANopen module, except the Parameterization error message that can not be disabled. For further information, see “PROFIBUS/PROFINET Channel Diagnostics” on page 31.	Disable Enable	Disable
CANopen input data size	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 - 1024 bytes. **NOTE**: The maximum data size is limited to 512 bytes if you use a 4 byte backplane size, see page 12.	1 - 1024	Same as configured rack I/O block size.
CANopen output data size	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 - 1024 bytes. **NOTE**: The maximum data size is limited to 512 bytes if you use a 4 byte backplane size, see page 12.	1 - 1024	Same as configured rack I/O block size.

The configuration is downloaded to the PLC or the internal CPU using an MPI or Ethernet cable.

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 network master using a CANopen adapter[1]. At the next startup the CANopen master will configure the network, if this function was set in the configuration tool during initial configuration.

The 1 SI CANopen Module for ET200S can be configured either as a master or as a slave on the CANopen network.

After the configuration is finished, the CANopen network configuration can be uploaded to and stored in the PROFIBUS/PROFINET PLC or the ET200S I/O System CPU. If the 1 SI CANopen Module for ET200S has to be replaced by another module, the previously stored configuration can be downloaded to the new module, and communication can be restarted without unnecessary delay. The change of modules can be performed with power still on. To perform upload/download of configuration you need a function block that can be downloaded from www.et200can.com. For this block to function properly, Concise DCF files have to be used. For more information see "Configuration Upload/Download" on page 48.

5.3.1 LSS Routine

If there is a missing slave on the network after the boot time-out the master will initiate the LSS routine. It will send an identify slave request. If one (and only one) slave responds to that message, the master sets the NodeID of that node to the first available NodeID. The master will then send a boot up message to the slave. See also "LSS Services" on page 29.

Note: This routine only works for one slave without Node ID at a time on the network. If there are more than one slave without Node ID on the network, they will all be assigned the same Node ID, which will cause severe problems.

5.4 Configuration Files

The EDS file for the 1 SI CANopen Module for ET200S can be downloaded from www.et200can.com. After download the file is imported into the CANopen configuration tool. This file is necessary to perform the configuration of the module on the CANopen network.

The necessary information for configuring the module in the ET200S Distributed I/O System can be included in the SIMATIC STEP7 tool by installing the HSP or GSD file(s)[2] into the tool.

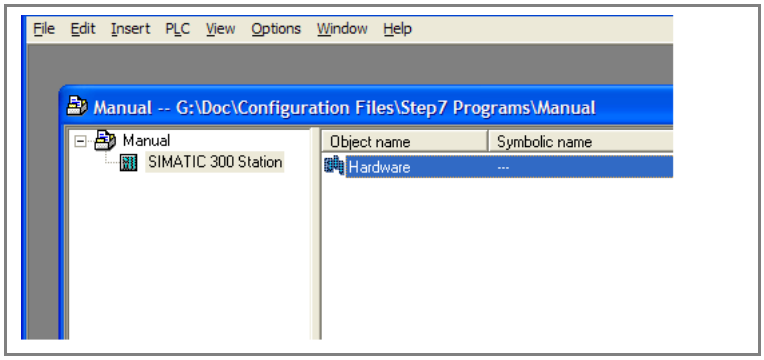
For applications using a third party PLC, please visit www.et200can.com for information and download of configuration files.

1. Please visit www.et200can.com or contact HMS support for further information, see "Support" on page 4.
2. Available for download at www.et200can.com.

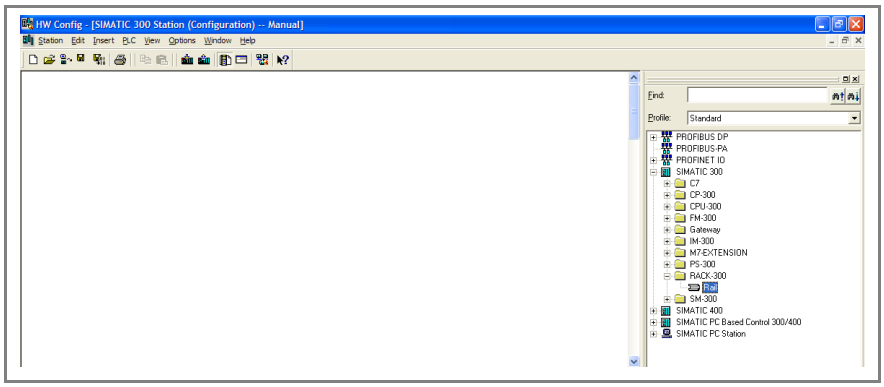
5.5 Configuration Example

This section gives an example of a STEP7[1] configuration of the ET200S Distributed I/O System and the 1 SI CANopen Module for ET200S. In the example, PROFIBUS and an ET200 rack without a CPU, but connected to a 315PLC are used, but a similar procedure can be used for PROFINET and/or for a ET200 rack including a CPU. To run this example, the HSP file must have been imported.

1. Open the SIMATIC Manager program and start a new project. Enter a name for the project.
2. Right-click on the name and choose "Insert new object".
3. In the list that appears, click on SIMATIC 300 Station. Enter a name and double-click on the new icon

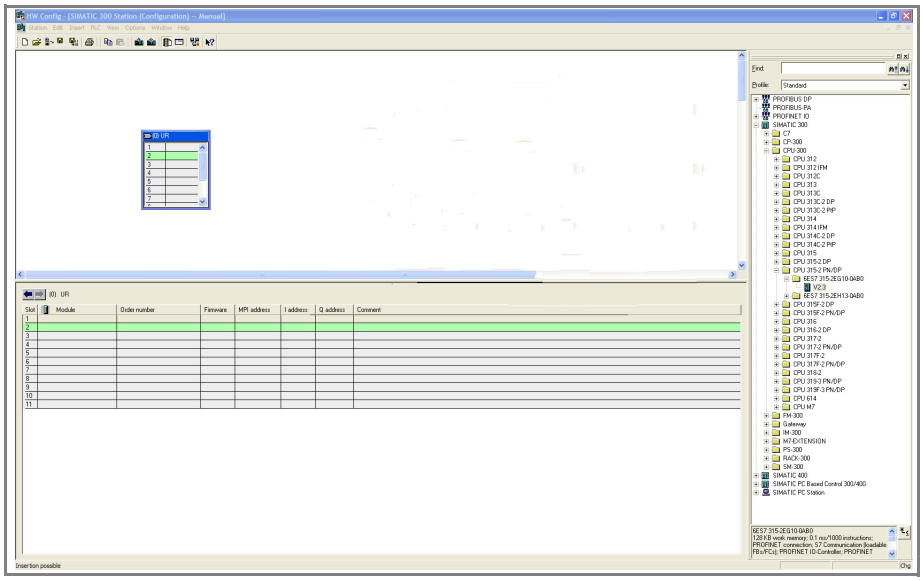


4. Double-click on "Hardware" to open the HW Config window.
5. Choose Rail under RACK-300. Add this to the configuration by dragging and dropping

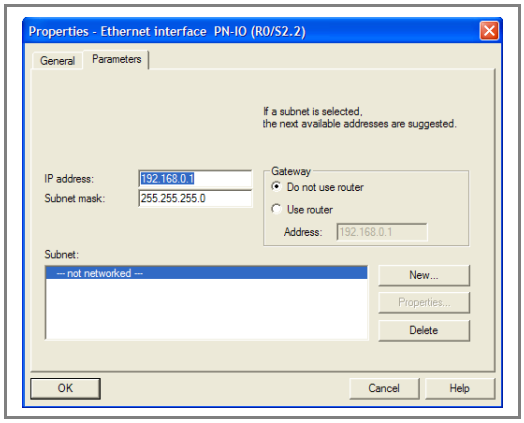


1. STEP7 V5.4 SP5 or later

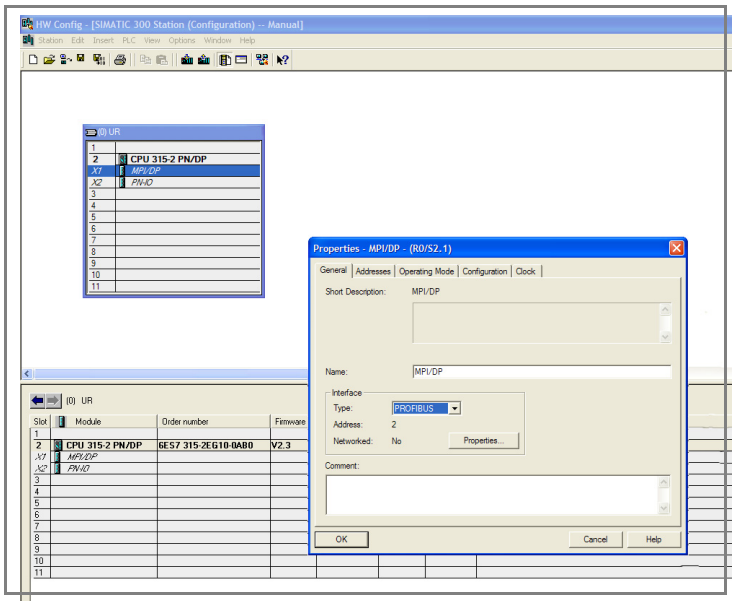
6. Choose PLC. Add this to the configuration by dragging and dropping.



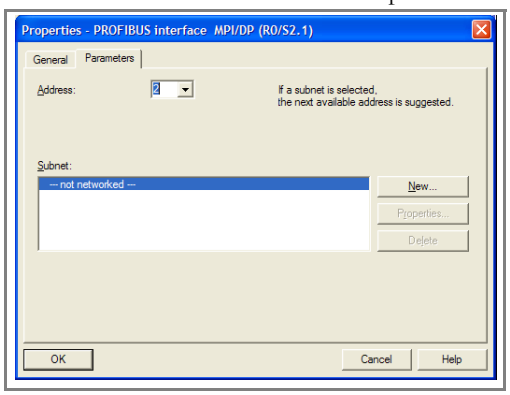
7. Select settings for the Ethernet interface that is used to download the configuration



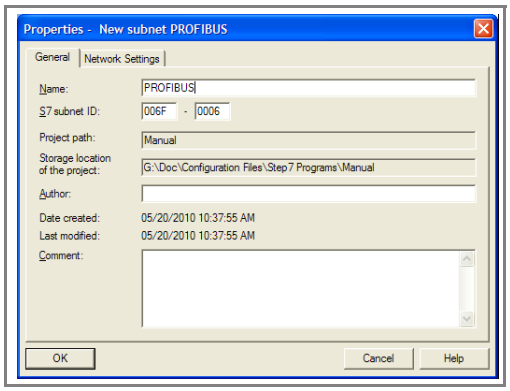
8. Configure the fieldbus interface



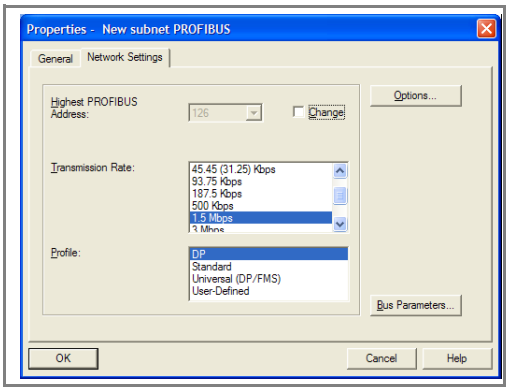
9. Set the Master node address to 2 in the Properties dialogue.



10. Click new to open the properties window for the Profibus sub network. Set a name for the Profibus network.

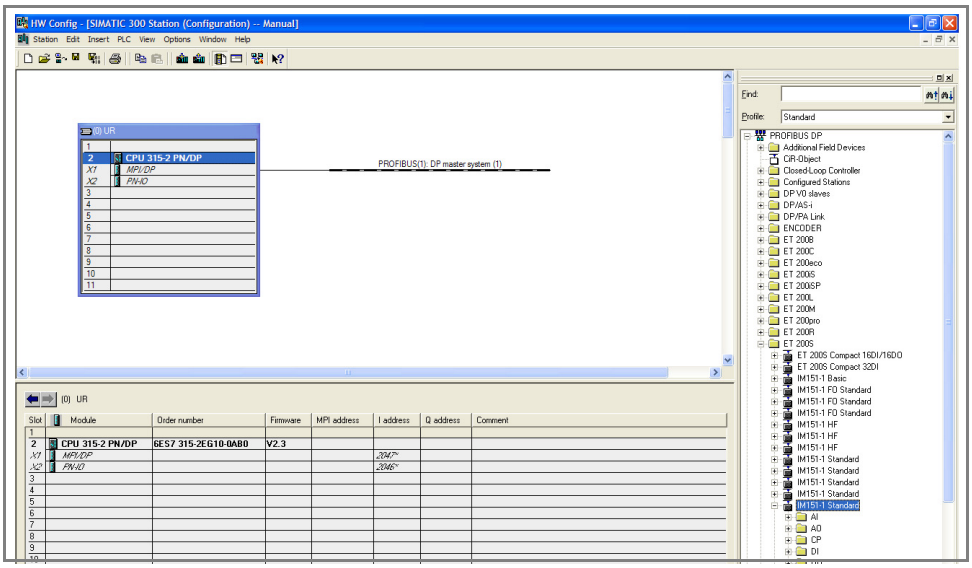


11. Select baud rate and profile under the Network Settings tab.

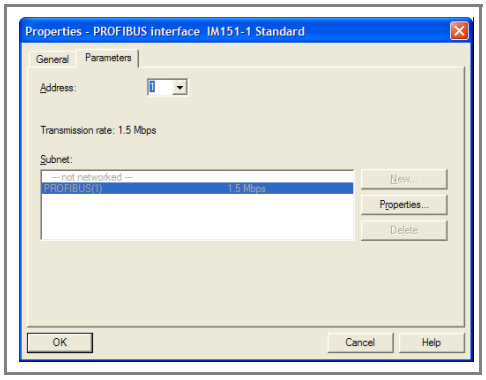


12. Press OK when prompted (three times) to return to the main configuration window.

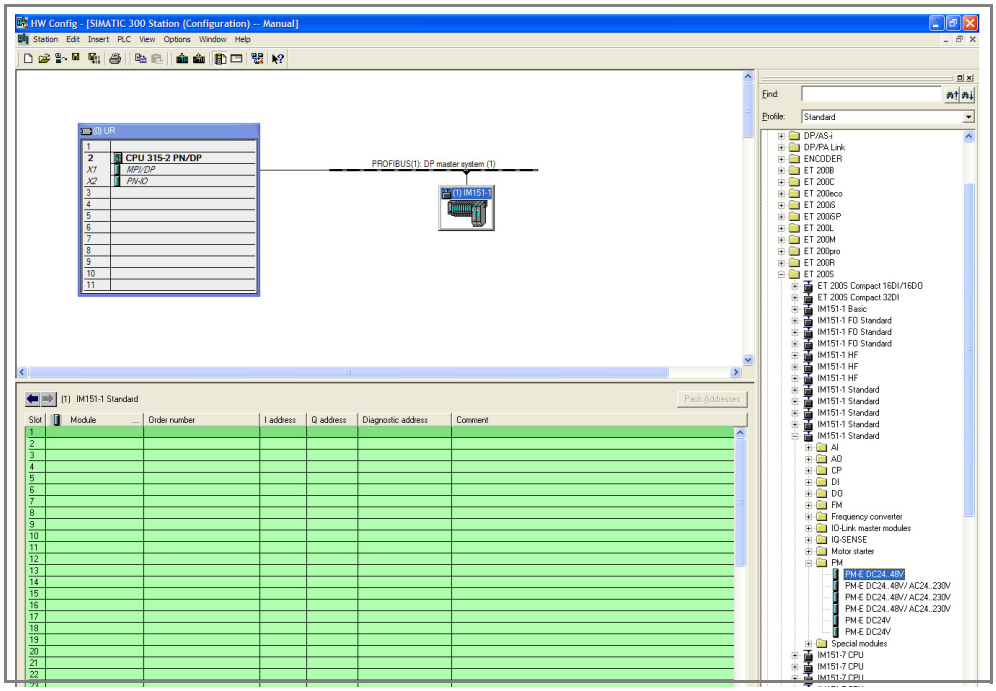
13. Drag and drop the correct type of ET200S Distributed I/O System to the network.



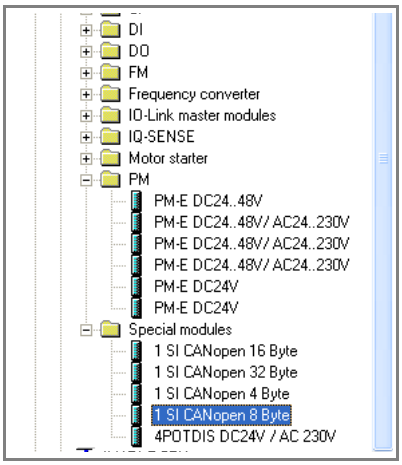
14. Select node address for the rack on the Profibus network



15. Select power module and other modules for the ET200S rack.

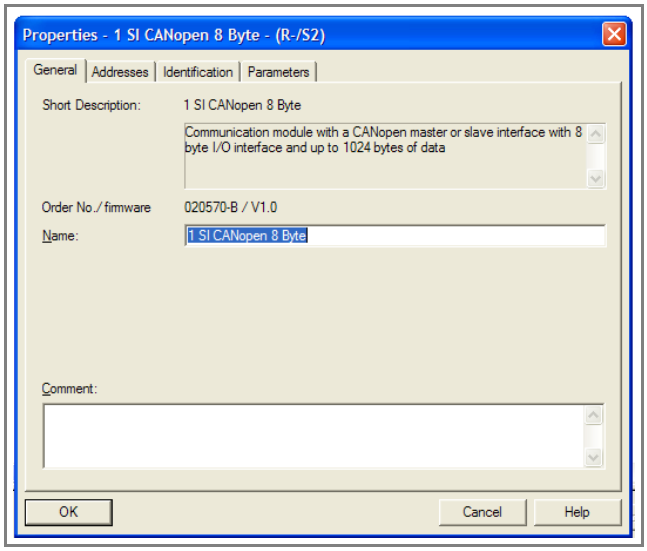


16. Click on the ET200S module in the list to the right to open the directory of possible modules. Locate the "1 SI CANopen" modules[1] under special modules. Click to add this at the appropriate place in the list of modules in the ET200S Distributed I/O System rack.



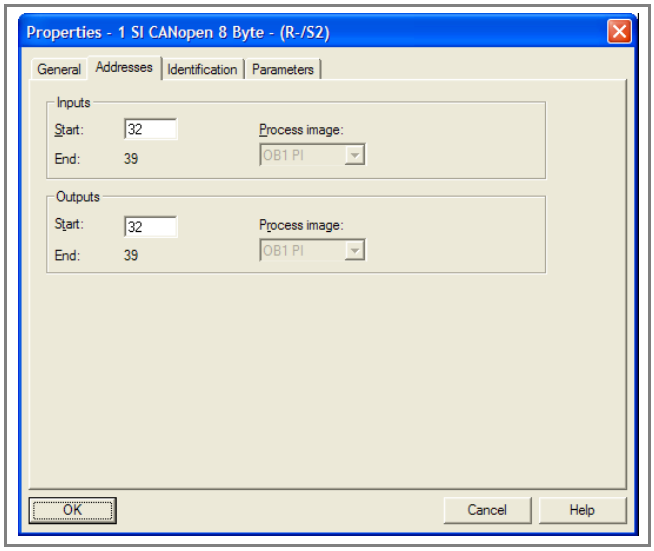
17. Right-click on the 1 SI CANopen Module for ET200S in the list and open the Object Properties window. (Alternatively, double-click on the ET200S module to open the Object Properties window.).

18. Enter name and comment for the module under the General tag.



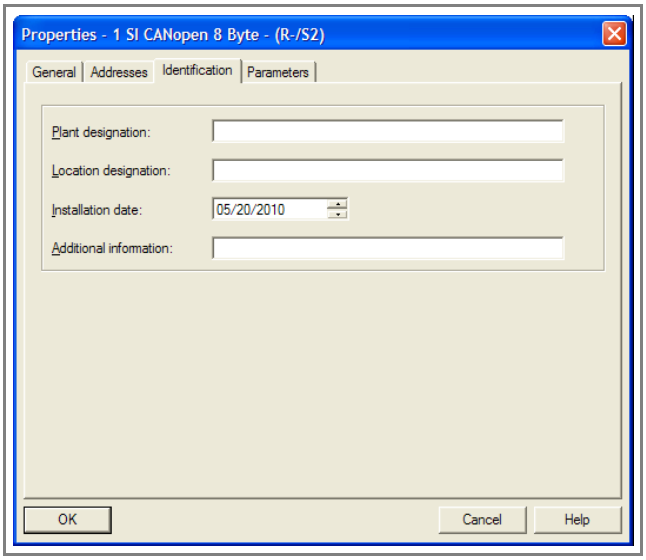
19. Choose the Addresses tag to set the I/O data address offset in the PLC. Make sure that the address change is acceptable. Input and output has to be set to the same value.

Note: Please select an address within the process image area of the PLC to ensure that data is transferred correctly.

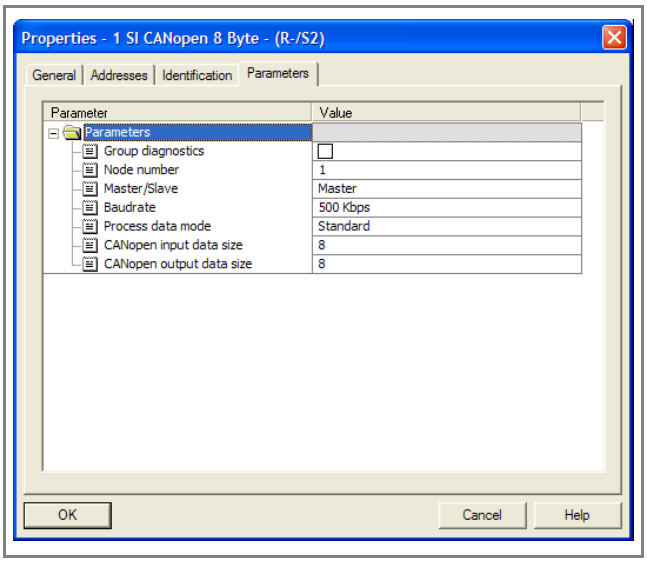


1. The physical module is always the same, but you can choose what data size to use, to fit your application.

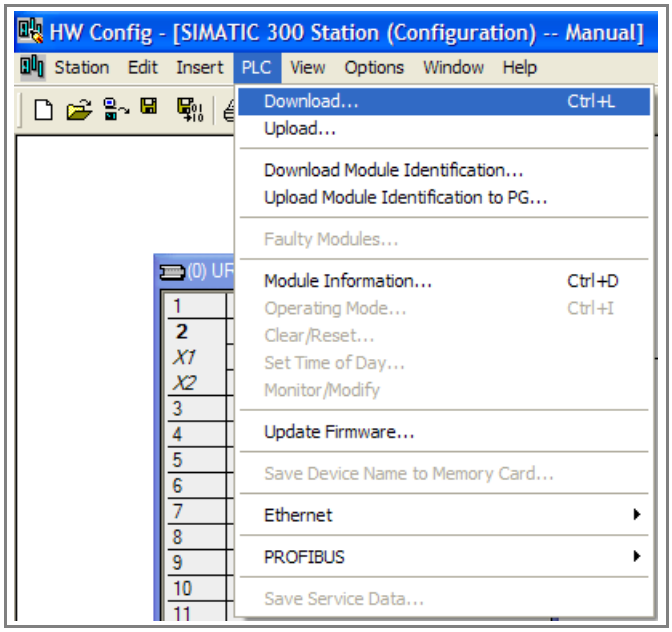
20. You can enter additional information about the module if you choose the Identification tag.



21. Choose the Parameters tag to define the values in the parameter list. Please note that these parameters are valid for the module on the CANopen network.
In this example, the 1 SI CANopen Module for ET200S is configured without Group Diagnostics. The node number on the CANopen network is 1. It is configured as a CANopen master with the baud rate set to 500 kbit/s. Process data mode is set to standard (not fragmented) and both the input data size and the output data size is set to 8 bytes. Press the OK button when finished.



22. Download the configuration to the PLC.



The 1 SI CANopen Module for ET200S is now configured as a slave on the PROFIBUS/PROFINET network, and, depending on the parameters, as a slave or a master on the CANopen network.

Configuration of the CANopen network

The configuration of the CANopen network, including the 1 SI CANopen Module for ET200S, is done separately with a CANopen network configuration tool.

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

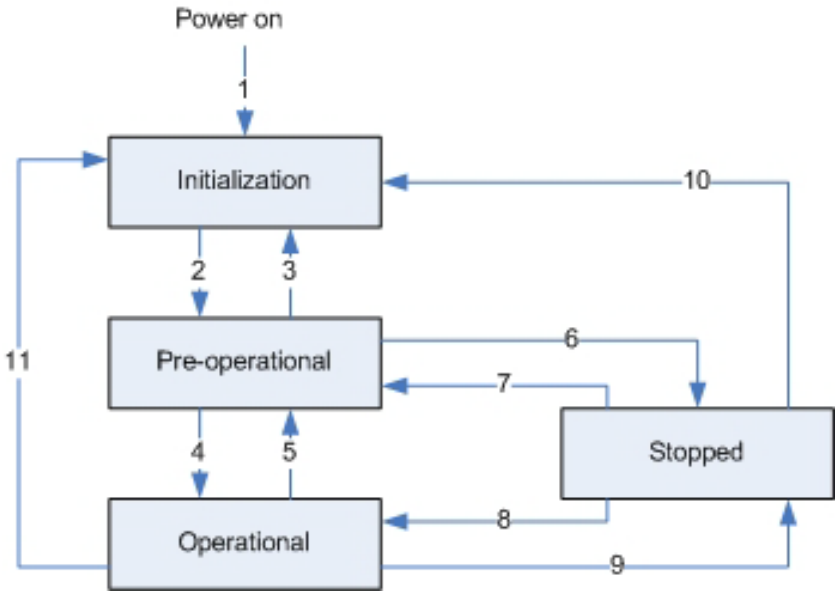
1. Download the EDS file from www.et200can.com to your PC.
2. Open CANopen configuration tool.
3. Add nodes to CANopen network.
4. Configure each node with the necessary parameters.
5. Check that there is no conflict between the parameters downloaded from the CANopen configuration tool and the parameters set up with SIMATIC STEP7.
6. Download the configuration to the CANopen network as Concise DCF-files to the master or store the configuration locally in each module's non volatile memory.

Please consult the user manual for the configuration tool for details and/or contact HMS support, see "Support" on page 4.

6. CANopen Module Specification

6.1 NMT State Machine

The function of the 1 SI CANopen Module for ET200S 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 1 SI CANopen Module for ET200S 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 an NMT object, a hardware reset or Module Control Services locally initiated by application events.

If the 1 SI CANopen Module for ET200S is configured as a CANopen master, the transitions are initiated from the PLC, using the RUN/STOP switch of the PLC. If the switch is in position STOP at startup, the module will enter PRE-OPERATIONAL state and if it is in position RUN at startup, the

module will automatically enter OPERATIONAL state. The module will however not enter OPERATIONAL state until all data has been updated from 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 master on the network by the use of NMT telegrams. If the data valid signal disappears, though, the module sends an emergency code (FF10h) 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	The controlling master changes to "Run" mode. If the node is a CANopen master, this transition is controlled by the PROFIBUS/PROFINET PLC. If it is a slave, the transition is controlled by the CANopen master using the command 'Start remote node'.
5	Return to pre-operational state. If the controlling master is a PROFIBUS/PROFINET PLC, this transition is initiated when the STOP/RUN switch is flipped from RUN to STOP.
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 1 SI 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 or PROFIBUS.

Transition to NMT state Operational

To safely detect that the CANopen network is up and running, the module will report diagnostics. A "parameterization error" will be active until the module has received a telegram with the parameterization information. Then the diagnostic "Error"[1] is active until the bootup routine is completed. The following criteria will have to be fulfilled for the bootup routine to be completed:

- All mandatory nodes have to be available on the network and configured without errors.
- If heartbeat or node guarding is enabled, all nodes in operational state have to run these services.
- All data have to be updated once (when fragmentation is used, the fragmentation block will have to be running to complete the update).
- If configuration using concise files is used, and the configuration of mandatory nodes and detected optional nodes does not succeed, an ode guarding is enabled, all nodes in operational state have to run these services.
- All data have to be updated once (when fragmentation is used, the fragmentation block will have to be running to complete the update).

If configuration using concise files is used, and the configuration of mandatory nodes and detected optional nodes does not succeed, an EMCY object (6161h) is sent on the CANopen network.

If the module is in slave mode, the diagnostic "Error" will be removed as soon as the module has booted.

1. See "PROFIBUS/PROFINET Channel Diagnostics" on page 31.

object (6161h) is sent on the CANopen network.

If the module is in slave mode, the diagnostic "Error" will be removed as soon as the module has booted.

For information about error control, see "Error Control" on page 29.

6.2 Data Exchange

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

The module features a simple default mapping scheme with 4 TPDO's and 4 RPDO's.

- **RPDO**

RPDO no.	Default COB IDs	Mapped to...	Relating to...	Default State
1	200h + Node ID	Object index 2100h, sub-index 1... 8	Output Data buffer, bytes 0... 7	Enabled
2	300h + Node ID	Object index 2100h, sub-index 9... 16	Output Data buffer, bytes 8...15	
3	400h + Node ID	Object index 2100h, sub-index 17... 24	Output Data buffer, bytes 16... 23	
4	500h + Node ID	Object index 2100h, sub-index 25... 32	Output Data buffer, bytes 24... 31	
5	80000000h	Object index 2101h, sub-index 1... 8	Output Data buffer, bytes 32... 39	Disabled
...		...	...	
128		Object index 211Fh, sub-index 25... 32	Output Data buffer, bytes 1016... 1023	

- **TPDO**

TPDO no.	Default COB IDs	Mapped to...	Relating to...	Default State
1	180h + Node ID	Object index 2000h, sub-index 1... 8	Input Data buffer, bytes 0... 7	Enabled
2	280h + Node ID	Object index 2000h, sub-index 9... 16	Input Data buffer, bytes 8... 15	
3	380h + Node ID	Object index 2000h, sub-index 17... 24	Input Data buffer, bytes 16... 23	
4	480h + Node ID	Object index 2000h, sub-index 25... 32	Input Data buffer, bytes 24... 31	
5	80000000h	Object index 2001h, sub-index 1... 8	Input Data buffer, bytes 32... 39	Disabled
...		...	...	
128		Object index 201Fh, sub-index 25... 32	Input Data buffer, bytes 1016... 1023	

For more information on the mapping see "Vendor Specific Objects" on page 42

RPDO Transmission Types

The RPDOs can be received either in synchronous or asynchronous mode. A synchronization (SYNC) object is transmitted periodically by a synchronization master, located either in the 1 SI CANopen Module for ET200S or in another node on the CANopen network. The data in synchronous RPDOs are not transferred to the ET200S system until after the next SYNC object is received. Asynchronous RPDOs will be transferred to the ET200S system 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 ET200 system after a SYNC object is received.
241 - 253	-	Reserved
254 - 255	Asynchronous, Event driven	An RPDO is transferred to the ET200 system without any relation to the SYNC object. **Note**: All default PDOs are mapped with transmission type 255.

TPDO Transmission Types

The TPDOs can be transmitted either in synchronous or asynchronous mode. A synchronization (SYNC) object is transmitted periodically by a synchronization master, located either in the 1 SI CANopen Module for ET200S 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 LSS Services

LSS master functionality according to CANopen DS305 specification is supported by the module. The module can configure the node ID of slaves that support LSS (i.e. the pre-configured node ID of a slave can be changed by a master). The module can not act as an LSS slave.

An LSS Slave is identified upon its LSS address, that consists of Vendor ID, Product Code, Software Version and Serial Number of the LSS slave module. If there is a missing slave on the network after the boot time out, see "Boot Time, 1F89h" on page 41, the master will initiate the LSS routine. It will send an identify slave request, using the LSS address of the slave. If one (and only one) slave responds to this request, the master will set the NodeID on that node to the first missing NodeID. It will then send a bootup message to the node.

Note: This routine only works for one slave without Node ID at the time on the network. If there are more than one slave without Node ID on the network, they will all be assigned the same Node ID, which will cause severe problems.

6.4 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.

6.4.1 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, objects 1016h and 1017h, page 33), a heartbeat event is registered by the consumer. The ERROR LED on the front of the 1 SI CANopen Module for ET200S will indicate the event and a communication error will be logged on the PROFIBUS/PROFINET channel diagnostics. An EMCY object (8130h) is also transmitted on the CANopen fieldbus. If the module is configured as a slave and is in OPERATIONAL state, it will go to PRE-OPERATIONAL state and wait for the user to take action. If it is in master mode it will take action according to the settings in the master objects.

The1 SI CANopen Module for ET200S can act as both heartbeat consumer an as heartbeat producer.

6.4.2 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 1 SI CANopen Module for ET200S being the master, the PROFIBUS/PROFINET master will be informed about the disappearance of the slave on the CANopen network through the PROFIBUS/PROFINET channel diagnostics. 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 1 SI CANopen Module for ET200S will indicate the event and a communication error will be logged on the PROFIBUS/PROFINET channel diagnostics. An EMCY object (8130h) is also transmitted on the CANopen fieldbus.

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 "Slave Assignment, 1F81h" on page 38. The master handles/evaluates each new node guarding request (RTR) in a timely manner and in numerical node order.

6.4.3 Emergency Object (EMCY)

The Emergency Object is used for error reporting on the CANopen network when a fault has occurred. The error codes are saved in a list in the Communication Profile Object 1003h, see page 32 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 60

6.4.4 PROFIBUS/PROFINET Channel Diagnostics

The following PROFIBUS/PROFINET channel diagnostics are generated from the 1 SI CANopen Module for ET200S. The diagnostic information can be read from the ET200S system at the time of a diagnostic error. When the OB82 interrupt is triggered, the address of the fault module can be read out. This information can then be used to trigger a read request to the ET200S system using the SFC51 function block. For more information about the specific parameters to the SFC51 function block, please consult the SIMATIC STEP7 manual.

Code	Name	Description	Comment
9	Error	CANopen configuration in process: The master is configuring the CANopen network according to saved configuration. This diagnostic will remain until all mandatory CANopen slaves have been correctly configured and have entered full I/O communication. In bootup mode.	If Group Diagnosis is enabled, see table on page 14
16	Parameterization error	Initialization failed due to wrong parameters.	Always
19	Communication error	Node guarding or Heartbeat event. This error indicates that one or more nodes has not responded correctly to the supervision protocols.	If Group Diagnosis is enabled, see table on page 14
26	External error	BUS OFF: This error indicates that the CAN controller of the 1SI CANopen has reached a BUS OFF state, e.g. due to a bad network connection or a faulty baud rate or Transparent CAN event: When running Transparent CAN mode this diagnostic is triggered by an event in the 1SI CANopen module. The user can then use the CAN Status function block to determine the reason for the event indication.	
27	Unknown error	Error when reading from external memory (non volatile storage). This error only occurs if the internal structure of the memory has changed (e.g. through software update) so that it is not possible for the software to correctly evaluate the memory. If this error appears the user should delete the current configuration, perform a new CANopen configuration download to the module and then restart the module.	

7. Supported Objects

The following sections describe the CANopen objects, according to DS301 and DS302, implemented within the module and described in the .EDS file.

7.1 Static Data Types

According to DS302

7.2 Communication Profile Area

7.2.1 DS301 Communication Profile Objects

The table below shows the objects according to CANopen specification DS301 rev. 4.2.

Index	Object Name	Sub-Index	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	-
1003h	Pre-defined error field	00h	Number of errors. Writing a 0 to this sub-index clears the error list.	U8	RW	See “CANopen Emergency Codes” on page 60 for emergency error codes.
		01h...10h	List of errors. Most recent error at top of list.	U32	RO	
1005h	COB-ID Sync	00h	ID of the sync message	U32	RW	-
1006h	Communication Cycle Period	00h	Communication cycle period	U32	RW	Only available if SYNC support is enabled
1007h	Synchronous Window Length	00h	Synchronous Window Length	U32	RW	Only available if SYNC support is enabled
1008h	Manufacturer device name	00h	The name of the CANopen module	Visible string	RO	“1 SI 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 in (ms)	U16	RW	0000h (default)
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	00h (default)

Index	Object Name	Sub-Index	Description	Type	Access	Notes
1010h	Store Parameters[a]	00h	Largest sub-index supported	U8	RO	01h
		01h	Store all parameters	U32	RW	To save a configuration, write “save” = 73 61 76 65h to this object.
1011h	Restore Parameters[1]	00h	Largest sub index 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.
1014h	COB ID EMCY	00h	Defines the COB ID of the Emergency Object	U32	RO	80h+NodeId
1016h	Consumer Heartbeat Time	00h	Largest sub-index supported	U8	RO	7Fh
		01h - 7Fh	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. Up to 127 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
1018h	Identity object	00h	Number of entries	U8	RO	04h
		01h	Vendor ID	U32	RO	1Bh (HMS Industrial Networks)
		02h	Product Code	U32	RO	17h (1 SI CANopen Module for ET200S)
		03h	Revision Number	U32	RO	Current software revision
		04h	Serial Number	U32	RO	32 bit serial number
1029h	Error Behavior	00h	Number of Error Classes	U8	RO	01h
		01h	Communication Error	U8	CONST	00h. Change to NMT state Pre-Operational (only if currently in NMT state Operational).
		02h	Specific Error Class	U8	CONST	00h. Change to NMT state Pre-Operational (only if currently in NMT state Operational).
1400h ... 147Fh	Receive PDO parameter	00h	Largest sub-index supported	U8	RO	02h
		01h	COB ID used by PDO	U32	RW	-
		02h	Transmission type	U8	RW	See “RPDO Transmission Types” on page 28
1600h ... 167Fh	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	-

Index	Object Name	Sub-Index	Description	Type	Access	Notes
1800h ... 187Fh	Transmit PDO parameter	00h	Largest sub-index supported	U8	RO	05h
		01h	COB ID used by PDO	U32	RW	-
		02h	Transmission type	U8	RW	See “TPDO Transmission Types” on page 28
		03h	Inhibit time	U16	RW	In steps of 0.1 ms
		05h	Event Timer (ms)	U16	RW	-
1A00h ... 1A7Fh	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

7.2.2 Configuration Manager

DS302 part 3: Configuration and program download

Network Configuration Objects

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

Index	Object Name	Sub-Index	Description	Type	Access
1F22h	Concise DCF	-	The concise/compressed DCF files information is stored in this object. The configuration is stored in the master by writing to the sub-index corresponding to the Node ID of the module.	Domain	Sub 0: RO Sub 1 - 128: WO
1F25h	Configure Slave	0 - 128	Sub-index 0 is ignored. Sub-index i (i = 1 - 127): Request reconfiguration of slave with Node ID equal to sub-index i. Sub-index 128: Request to reconfigure all slaves.	U32[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, sub-index i)

If an error occurs during configuration of a mandatory node or a detected optional node, an emergency object, with the code 6161h, with either error code 7 or 8 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 29

Check Configuration

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

Index	Object Name	Sub-Index	Description	Type	Access
1F26h	Expected Configuration Date	0 - 127	The date that the Configuration Manager expects to find when comparing signature and configuration.	UNIT32	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	Sub-Index	Description	Type	Access
1F80h	NMT Start-up	-	Defining whether the device is the NMT Master	U32	RW
1F81h	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.	U32	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	RW
1F85h	Vendor Identification	ARRAY	Verify vendor identifications for the slaves	U32	RW
1F86h	Product Code	ARRAY	Verify product codes for the slaves	U32	RW
1F87h	Revision Number	ARRAY	Verify revision numbers for the slaves	U32	RW
1F88h	Serial Number	ARRAY	Verify expected serial numbers for the slaves	U32	RW
1F89h	Boot Time	VAR	Maximum slave boot time before master indicates boot error and attempts to find the slave with LSS.	U32	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 STEP7 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	
	1	Do not enter Operational state automatically. The PLC will decide when to enter Operational state	
3	0	Start-up of slaves allowed (i.e. allowed to send NMT Start Remote Node command)	
	1	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 (including self), 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)	

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 sub-index 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	0	The slave can be reset by the NMT RESET COMMUNICATION command independent of state, see "Request NMT, 1F82h" on page 39.
	1	If the slave is set in mode OPERATIONAL it will not be reset during startup.
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 Retry Factor and GuardTime.

See also ...

- "Node Guarding" on page 30

Request NMT, 1F82h

Each node on the CANopen network can be controlled individually from the PROFIBUS/PROFINET application by sending this object. The sub-index indicates what nodes the request affects:

Sub-Index	Description
0	Largest sub index 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 (sub-index 128)

Examples

- Node 5 should be transferred to the OPERATIONAL state:
  An SDO write access with the value 5 is executed to object 1F82h sub-index 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 sub-index 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 "Slave Assignment, 1F81h" on page 38, 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.

Sub-Index	Description	Access
0	Largest sub index 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 01F40301h is executed to object 1F81h sub-index 5 in the local object dictionary. Guarding is activated by an SDO write access with the value 1 to object 1F83h sub-index 5 in the local object dictionary.

Device Type Identification, 1F84h

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

Sub-Index	Description
0	Largest sub index supported (127)
i (with i = 1...127)	Compares expected device type (entered into this object) with actual device type (object 1000h, sub-index 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 sub-index indicates which node is checked. The node in question is only checked if data is other than zero.

Sub-Index	Description
0	Largest sub index supported (127)
i (with i = 1...127)	Compares expected vendor (entered into this object) with actual vendor (object 1018h, sub-index 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 sub-index indicates which node is checked. The node in question is only checked if data is other than zero.

Sub-Index	Description
0	Largest sub index supported (127)
i (with i = 1...127)	Compares expected product code (entered into this object) with actual product code (object 1018h, sub-index 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 sub-index indicates which node is checked. The node in question is only checked if data is other than zero.

Sub-Index	Description
0	Largest sub index supported (127)
i (with i = 1...127)	Compares expected revision number (entered into this object) with actual revision number (object 1018h, sub-index 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 sub-index indicates which node is checked. The node in question is only checked if data is other than zero.

Sub-Index	Description
0	Largest sub index supported (127)
i (with i = 1...127)	Compares expected serial number (entered into this object) with actual serial number (object 1018h, sub-index 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 slaves (nodes), mandatory and optional, to boot. If not all slaves are ready after this time, an EMCY object (8000h) will be generated and the LSS routine will be started, see “LSS Services” on page 29. If the assigned time is 0, the master will wait endlessly.

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

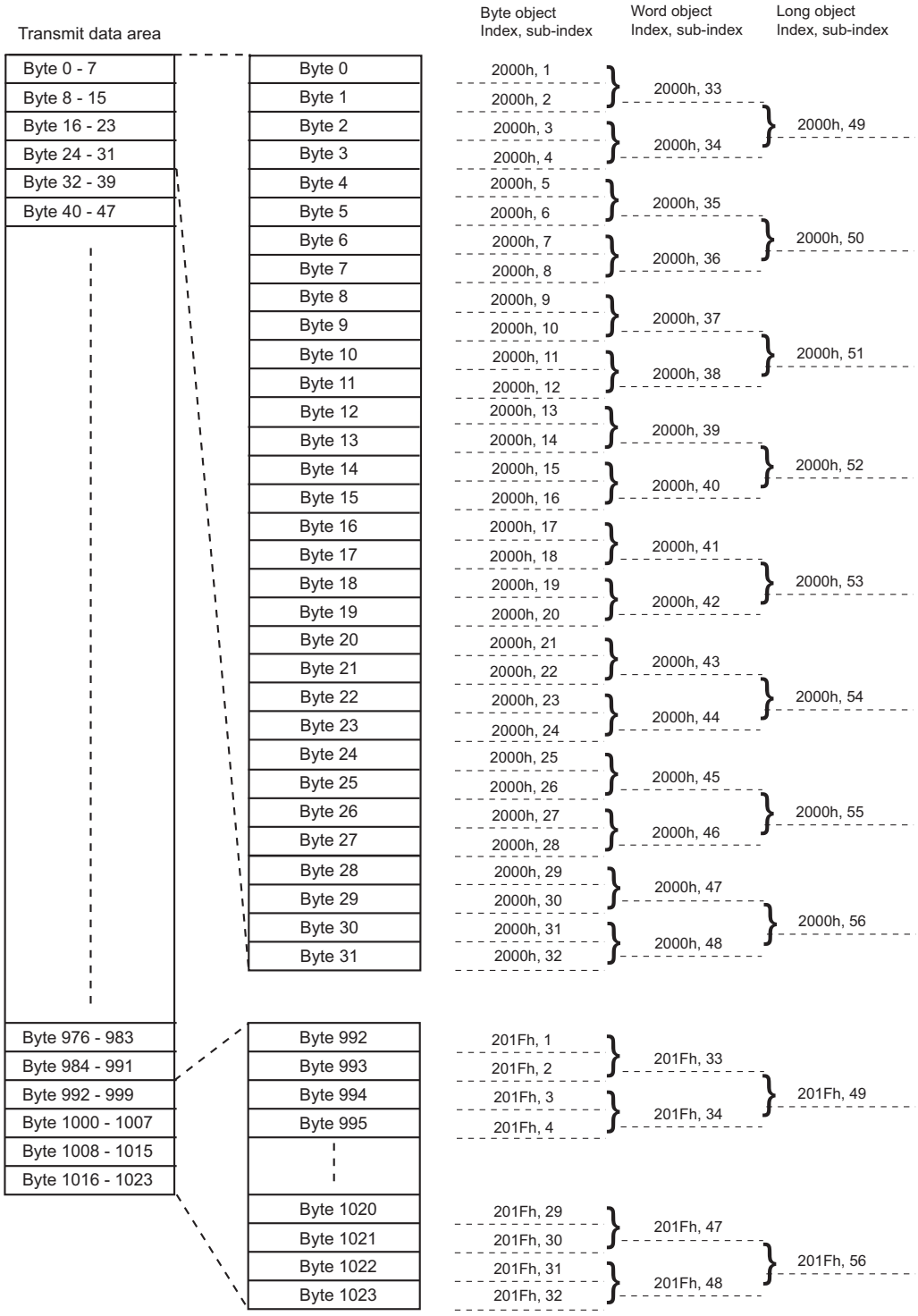
7.3 Vendor Specific Objects

One or several generic data object are connected to each PDO. This is configured during the configuration phase. There are 128 Receive PDOs and 128 Transmit PDOs.

Index	Sub-Index	Type	Access	Name and description	Comment
2000h - 201Fh	-	STRUCT		Generic Transmit Object #	
	0	U8	RO	Largest sub-index supported	
	1..32	U8	RO	Generic Byte Transmit Object 1..32	
	33..48	U16	RO	Generic Word Transmit Object 1..16	Same data as sub-index 1..32
	49..56	U32	RO	Generic Long Transmit Object 1..8	Same data as sub-index 1..32
2100h - 211Fh	-	STRUCT		Generic Receive Object #	
	0	U8	RO	Largest sub-index supported	
	1..32	U8	RW	Generic Byte Receive Object 1..32	
	33..48	U16	RW	Generic Word Receive Object 1..16	Same data as sub-index 1..32
	49..56	U32	RW	Generic Long Receive Object 1..8	Same data as sub-index 1..32
3000h	0	U8	RW	Swap data to big endian	0: Little endian (CANopen style, default) 1: Swap data to big endian, see “Swap Data to Big Endian (3000h)” on page 44

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 in different data types. For example: byte to index 2000h, sub-index 1 - 32, word to index 2000h, sub-index 33 - 48 and double-word (long) to index 2000h, sub-index 49 - 56.



Note: The picture shows the Transmit data area, but the Receive data area (objects 2100 to 211F) is structured in the same way

7.3.2 Swap Data to Big Endian (3000h)

PROFIBUS and PROFINET use big endian and CANopen uses little endian. The data swapping object (3000h) makes it possible to swap data to big endian in the 1 SI CANopen module. Data is swapped according to mapping, i.e. UINT16 data will be swapped over 2 bytes and UINT32 over 4 bytes. The mapping is decided from objects 1600h-167Fh (receive PDO mapping) or 1A00h-1A7Fh (transmit PDO mapping) depending on direction of data.

8. CANopen Function Blocks

Application notes, describing how to use the function blocks, are available for download at www.et200can.com.

8.1 SDO Read/Write

8.1.1 General

Reading (uploading) and writing (downloading) SDOs is performed by function blocks, available for download from www.et200can.com.[1]

Note: These functions are only supported in master mode.

8.1.2 Parameters

Input Parameters

Name	Data type	Description
REQ	Bool	Start request. Default is FALSE. Upload/download is started if REQ = TRUE for at least one clock cycle.
ID	DWord	Logical address for the position where the INPUT data starts for the 1 SI CANopen module in the rack.
SLOT	Byte	Defines which slot is used on the PROFIBUS/PROFINET/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 master, even when the CANopen master has another Node ID.
INDEX	Word	SDO index to be read/written
SUB	Byte	SDO sub index to be read/written
DATA	Any	• Pointer to the area where to save read data or to get data to be written. • Size of the area.

Output Parameters

Name	Data type	Description
SIZE	Word	Number of bytes that have been read (SDO Read only)
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	Int	Error code, see "Error Codes (RET)" on page 61. Available when BUSY turns FALSE, until REQ turns TRUE.

1. Both function blocks use RD_REC and WR_REC for the data transfer.

8.1.3 Description

SDO Read

Performs an SDO read on the node, index and sub index that is defined in the parameters. Returned data is saved in the area that DATA points to. The block will continue to read the SDO even when the data area is filled, until the ET200S System Rack indicates that the complete SDO is read. Only the predefined amount of data will be saved (given as size in input parameter DATA). SIZE holds the complete SDO size. If the size given in the parameter DATA is too small, RET will return error 2200h.

SDO Write

Performs an SDO write on the node, index and sub index that is defined in the parameters. Data written is fetched from the area pointed to by DATA. The amount of data written is given as size in input parameter DATA.

8.2 Fragmentation Handling

8.2.1 General

There is an upper limit on how much I/O data can be distributed in one data block. Depending on ET200S IO System rack, see "Siemens Interface Modules Compatibility" on page 58, the block size will be either 32, 16, 8 or 4 bytes. If the data message is larger, the data can be fragmented and sent in several packages. Fragmentation is enabled by setting the parameter "Process Data Mode" during configuration, see "ET200S Distributed I/O System Configuration" on page 13. The function block described below will handle the fragmentation. This block can be downloaded from www.et200can.com.[1]

Note: When data is sent fragmented, the fragmentation function block needs to be called cyclically from the user code, if the module is to work properly.

8.2.2 Parameters

Input Parameters

Name	Data type (STEP7)	Description
IN	Any	Address and size of the area in the PLC containing input I/O data, that is, data read to the module from the CANopen network. The Any pointer must indicate the data in byte format.
OUT	Any	Address and size of the area in the PLC containing output I/O data, that is data written from the module to the CANopen network. The Any pointer must indicate the data in byte format.
LADDR_IN	Word	Logical address to the module, defining where the module input starts
LADDR_OUT	Word	Logical address to the module, defining where the module output starts
BPSIZE	Word	Back plane size of selected module, allowed values are 4, 8, 16 and 32
RESET	Bool	If RESET = TRUE the function block updates all outputs to, and all inputs from, the 1 SI CANopen module. All I/O in the OUT-array is then sent to the PLC.

Output Parameter

Name	Data type (STEP7)	Description
RET	Int	Error code, see "Error Codes (RET)" on page 61. Holds result from latest execution of block. New data is available after every time the block has been executed.

8.2.3 Description

The parameters IN and OUT specify the two memory areas where input and output data are located. The fragmentation block continuously checks the OUT area and if there are any changes. Only changed data segments will be transferred. 2 bytes of the actual backplane size is used for control of fragmentation, the rest transfers data., i.e. in a 4 byte module, 2 bytes of data is transferred for every scan cycle of the PLC program.

New input data is saved in a buffer until the module signals that all data has arrived. The buffer content is then transferred to the area specified by the IN parameter.

1. The function blocks use the IO data area in the module for the data transfer.

8.3 Configuration Upload/Download

8.3.1 General

A 1 SI CANopen Module for ET200S can be removed and replaced with other hardware without the power being turned off. To be able to restart accurately and fast, the configuration can be uploaded to the PLC after configuration and downloaded to the new hardware once this is attached to the rack. The upload and download of the configuration is performed by PLC function blocks, available for download from www.et200can.com.[1]

Note: These functions are only supported in master mode.

8.3.2 Configuration Upload

Input Parameters

Name	Data type (STEP7)	Description
REQ	Bool	Start request. Default is FALSE. Upload is started if REQ = TRUE for at least one clock cycle.
ID	DWord	Logical address of the CANopen module.
DBSTART	Int	Number of the first DB (Data Block) of a maximum of four, that can be used to store data. DBSTART indicates the first DB to be used, the following three are DBSTART+1, DBSTART+2 and DBSTART+3 **NOTE**: The Upload function block will overwrite any data that is present in the requested area.
DBSIZE	Word	Size of the data blocks that are used to store data. Valid values are 262 - 50000. Max number of bytes that can be saved are DBSIZE*4 - 262.

Output Parameters

Name	Data type (STEP7)	Description
BUSY	Bool	If the request is not finished within one clock cycle, BUSY turns TRUE and stays TRUE until the request is finished, when it returns to FALSE.
RET	Int	Error code, see “Error Codes (RET)” on page 61. Available when BUSY turns FALSE, until REQ turns TRUE.

Description

If there is no configuration available in the PLC at DBSTART, or if it is of the wrong size (correct size is DBSIZE), a new data block with a new configuration is uploaded to the PLC. Any old data block at the specified address will be overwritten.

If there is a valid data block, the CRC is read from the module, and compared to the CRC that is stored in the PLC with the configuration. If the CRCs are equivalent, the configuration stored in the PLC and the configuration running in the 1 SI CANopen Module for ET200S are equivalent. Execution of the function block is stopped, the error code 0 (no error) is written to RET, and BUSY is set to false.

If the CRCs are different, a new configuration will be uploaded to the PLC. The function block fetches the configuration from object 1F22h, sub indices 1-127, and stores it in data blocks in the PLC. If the

1. Both function blocks use RD_REC and WR_REC for the data transfer.

configuration is too large for one data block, an additional one is created, where the rest of the configuration is stored.

If an error should occur during upload, the CRC in the data block is set to 00h 00h 00h 00h 00h 00h, to ensure that a new configuration will be uploaded the next time the function is called.

8.3.3 Configuration Download

Input Parameters

Name	Data type (STEP7)	Description
REQ	Bool	Start request. Default is FALSE. Download is started if REQ = TRUE for at least one clock cycle.
ID	DWord	Logical address of the CANopen module.
DBSTART	Int	Number of the first DB (Data Block) of a maximum of four, that are used for retrieving data saved at configuration upload.
DBSIZE	Word	Size of the data blocks that are used to store data. Has to be equal to the size specified at configuration upload.

Output Parameters

Name	Data type (STEP7)	Description
BUSY	Bool	If the request is not finished within one clock cycle, BUSY turns TRUE and stays TRUE until the request is finished, when it returns to FALSE.
RET	Int	Error code, see “Error Codes (RET)” on page 61. Available when BUSY turns FALSE, until REQ turns TRUE.

Description

The function block reads the configuration CRC from the 1 SI CANopen Module for ET200S and compares it to the CRC stored in the data block at DBSTART in the PLC. If this data block does not exist or if it is of the wrong size, execution is stopped and an error code is returned. If the CRCs are equivalent, execution is stopped, RET is set to 0 (no error), and BUSY is set to FALSE.

If the CRCs differ, the configuration stored in the PLC is downloaded to sub-indices 1 - 127 in the object 1F22h in the module. If any data block is missing or of the wrong size, execution is stopped and an error code returned. If the download succeeds, the configuration is saved by writing ‘save’ to object 1010h, sub index 1.

9. Transparent CAN Function Blocks

9.1 General

There are 4 function blocks handling the transparent CAN functionality. These are available for download from www.et200can.com. The functionality is described in "Transparent CAN Mode" on page 8.

CAN Send and CAN Receive handles the frames that are sent or received, CAN Control controls the state and the settings of the CAN layer, and CAN Status is used to request the status of the 1 SI CANopen Module. The transparent mode is turned on by sending a CAN Control message with FCN = 0.

Application notes, describing how to use the function blocks, are available for download at www.et200can.com/support.html.

9.2 CAN Send

9.2.1 Parameters

Input Parameters

Name	Data type (STEP7)	Description
REQ	Bool	Starts the block. Has to be set to TRUE for at least one scan cycle. Transmissions will occur until REQ is set to false.
LADDR_IN	Word	Logical address of the input area of the CANopen master module
LADDR_OUT	Word	Logical address of the output area of the CANopen master module
BPSIZE	Byte	Size of the backplane, valid values are 16 and 32
RTR	Bool	RTR on the CAN net
COBID	Word	COB ID on the CAN net
DATA	Any	Data to be sent on the CAN net
CANSIZE	Byte	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 (STEP7)	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	Int	Error code, see "Error Codes (RET)" on page 61. Valid once the BUSY signal turns FALSE until the next time the function block is started, i.e. when REQ is set to TRUE.

9.2.2 Description

When REQ turns TRUE, a CAN frame is sent according to the parameters defined by the user. When the frame has been acknowledged by the module, BUSY will go FALSE, and a new frame can be sent. This will continue as long as REQ is set to TRUE.[1]

9.3 CAN Receive

9.3.1 Parameters

Input Parameters

Name	Data type (STEP7)	Description
REQ	Bool	Starts the block. Has to be set to TRUE for at least one scan cycle
LADDR_IN	Word	Logical address of the input area of the CANopen master module
LADDR_OUT	Word	Logical address of the output area of the CANopen master module
BPSIZE	Byte	Size of the backplane, valid values are 16 and 32
ACK	Bool	Set to TRUE for one scan cycle to acknowledge a newly read CAN frame
DATA	Any	Pointer where to store the data read from the CAN frame.

Output Parameters

Name	Data type (STEP7)	Description
NEW	Bool	Set to TRUE when a new CAN frame has been read from the module.
COBID	Word	COB ID of the CAN frame
RTR	Bool	RTR of the CAN frame
SIZE	Byte	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	Int	Error code, see “Error Codes (RET)” on page 61. Valid once the BUSY signal turns FALSE until the next time the function block is started, i.e. when REQ is set to TRUE.

9.3.2 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. 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 255 frames.[1]

For polling operation, set REQ to TRUE permanently.

Note: ACK must be reset to FALSE after one scan cycle. Otherwise incoming frames will automatically be acknowledged as read, before they have been accepted by the PLC. This may result in data loss, as new frames may overwrite frames that haven’t been handled completely.

1. The function blocks use the IO data area in the module for the data transfer.
1. The function block use the IO data area in the module for the data transfer.

9.4 CAN Control

9.4.1 Parameters

Input Parameters

Name	Data type (STEP7)	Description
REQ	Bool	Starts the block. Has to be set to TRUE for at least one scan cycle to start the block.
ID	DWord	Logical address of the CANopen module.
FCN	Byte	Function code: 0: Use transparent mode 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: Clear Bus Off
LEN	Word	Only valid when FCN = 1, otherwise ignored. Defines the size of the acceptance filter array that is sent in PARAM.
PARAM	Any	Parameters 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 (STEP7)	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	Int	Error code, see “Error Codes (RET)” on page 61. Valid once the BUSY signal turns FALSE until the next time the function block is started, i.e. when REQ is set to TRUE.

9.4.2 Description

This block is used to control the state of the transparent CAN layer and to set its parameters.[1]

- **FCN = 0**

  Enables transparent mode. All ordinary CANopen functionality is disabled, and the module has to be restarted to re-enable CANopen functionality. The rest of the function codes in CAN Control can only be used if the transparent mode is enabled.

1. The function blocks use RD_REC and WR_REC for the data transfer.

- **FCN = 1**

  Configures the CAN Receive acceptance filter in the module i.e. what COB-IDs will be accepted when receiving data frames. The module will not listen to the CAN bus, if no COB-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 COB-IDs sent in the message (read from PARAM). One or more COB-IDs can be sent in one message (except for enable or disable all), also several messages with COB-IDs can be sent after one another.

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

Bit 15	Bit 14	Bit 0
Set bit	COB-ID	

  COB-ID 0000h with LEN set to 0000h will disable all COB-IDs. If LEN is set to 1 only COB-ID 000h is disabled.

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

- **FCN = 2**

  Configures the CAN Receive buffer limit warning value. It gives the opportunity to define what number of frames will be stored in the receive buffer, before bit BUFFER LIMIT REACHED in the status code will be set, see "Status Codes" on page 54). Initially the buffer limit is set to 255, 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 255.

- **FCN = 3**

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

- **FCN = 5**

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

9.5 CAN Status

9.5.1 Parameters

Input Parameters

Name	Data type (STEP7)	Description
REQ	Bool	Starts the block. Has to be set to TRUE for at least one scan cycle
ID	DWord	Logical address of the CANopen module.

Output Parameters

Name	Data type (STEP7)	Description
STAT	DWord	Status of the CANopen master, see table below.
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	Int	Error code, see “Error Codes (RET)” on page 61. 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[1]

This block is used to request the status of the module. The status value is found in the parameter STAT.

Status Codes

The status code is represented by 32 bits, where each bit represents a status code.

If Group Diagnosis is enabled in the module vi HW configuration in STEP7, the channel diagnostics interrupt will report “External error” as long as any bit in the Status Code is set.

Status code	Name	Description
0x00000000	NO ERRORS	-
0x00000001	BUS OFF	The CAN error counter have passed Bus Off level. If there is no contact with the bus, a Bus Off signal is issued
0x00000002	ERROR PAS-SIVE	The CAN error counter has passed the warning level.
0x00000004	BUFFER FULL	255 messages are stored, and the receive buffer is full. Reset by reading out one message from the buffer.
0x00000008	BUFFER HALF FULL	Set when 128 messages are stored in the receive buffer, i.e. the buffer is half full.
0x00000010	BUFFER LIMIT REACHED	Set at 255 initially, can be changed by FCN 2, see“CAN Control” on page 52)

1. The function blocks use RD_REC and WR_REC for the data transfer.

A. Technical Specification

A.1 Mechanical Properties

Housing

Plastic housing, plug-in module, protection class IP20

Dimensions

Width: 15 mm (0.59")

Weight: 40 g (0.09 lb.)

A.2 Electrical Characteristics

Module Voltage

24 VDC +20/-15%, supplied from Power module PM-E.

Please provide adequate lightning protection, for example Blitzductor BVT AD24

Current Consumption

Max 0.8 W - 33 mA @ 24 V

A.3 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.4 Regulatory Compliance

CANopen Conformance

DS301 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

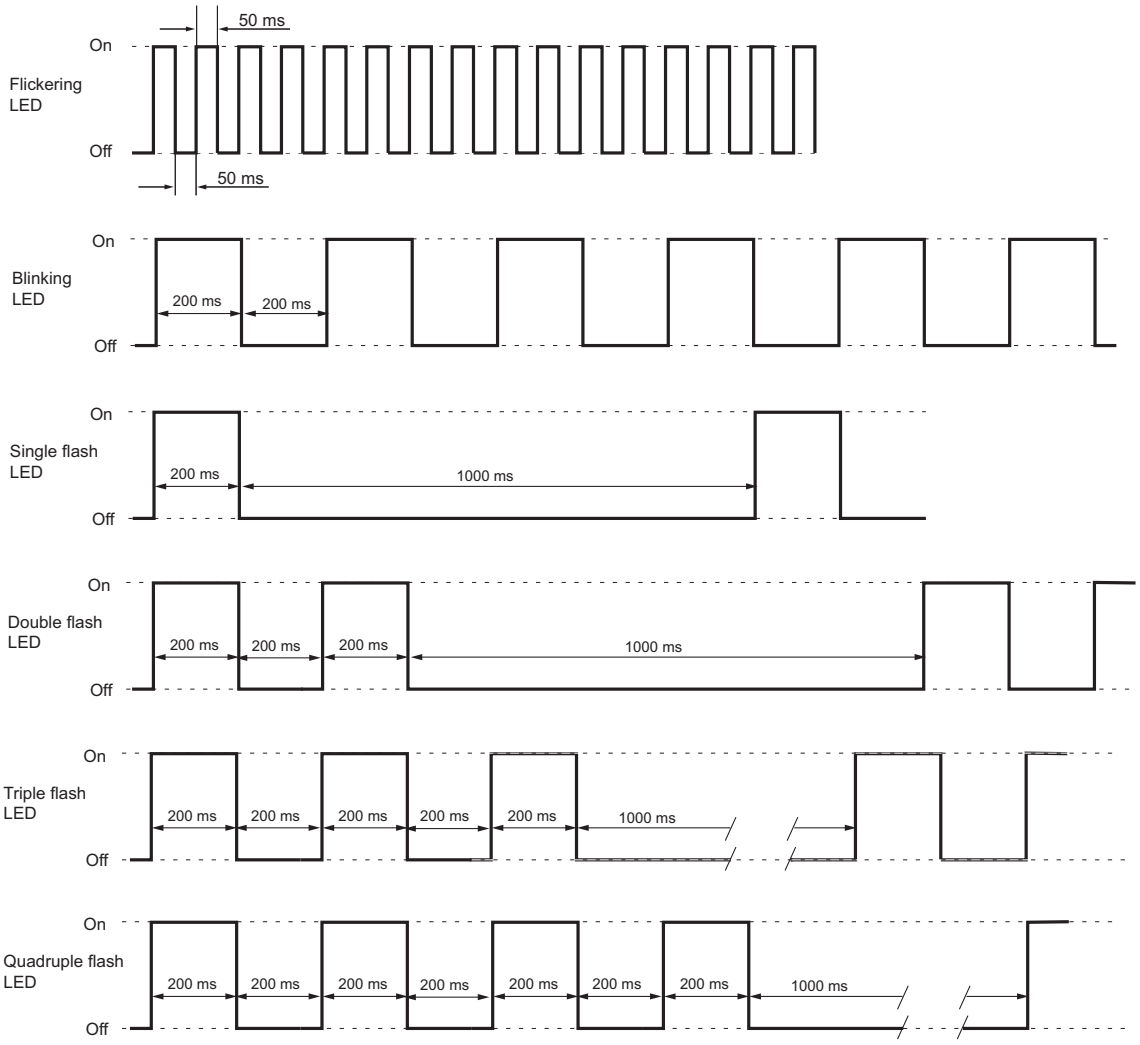
The certification has been documented by UL in file E214107.

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 10.



When LSS services are in progress, both the ERR LED (red) and the RUN LED (green) are flickering.

C. Siemens Interface Modules Compatibility

Below is a list of Siemens Interface Modules that support the 1 SI CANopen Module for ET200S.

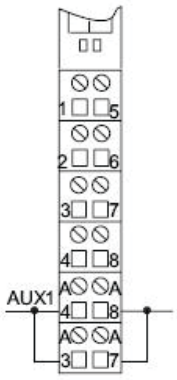
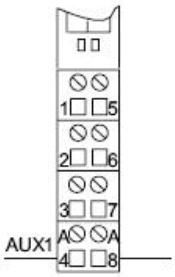
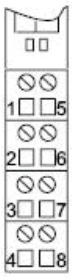
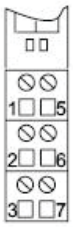
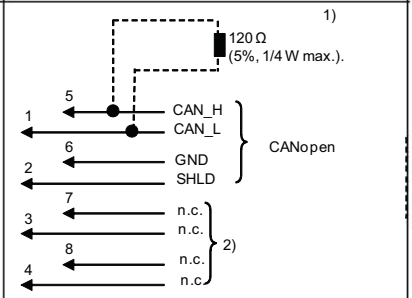
Order Number	ET 200S Interface Module	Max IO-data per cycle	Max. consistency length of IO-data	Comments
PROFIBUS-IM				
6ES7 151-1AA05-0AB0	IM 151-1 Standard, DP	32	32	
6ES7 151-1AB05-0AB0	IM 151-1 FO Standard	32	32	
6ES7 151-1BA02-0AB0	IM 151-1 High Feature	32	32	
PROFIBUS-CPU				
6ES7 151-7AB00-0AB0	IM 151-7 CPU FO	4	4	See footnote[a]
6ES7 151-7AA20-0AB0	IM 151-7 CPU (96K)	16	16	
6ES7 151-7FA20-0AB0	IM 151-7 F-CPU (128K) PROFIsafe	16	16	
PROFINET-IM				
6ES7 151-3AA23-0AB0	IM 151-3 PN IO Standard	32	32	Firmware V6.1
6ES7 151-3BA23-0AB0	IM 151-3 PN IO High Feature	32	32	Firmware V6.1
6ES7 151-3BA60-0AB0	IM 151-3 PN IO High Speed	32	32	Firmware V2.1
6ES7 151-3BB23-0AB0	IM 151-3 PN IO FO High Feature	32	32	Firmware V6.1
PROFINET-CPU				
6ES7 151-8AB00-0AB0	IM 151-8PN CPU	16	16	
6ES7 151-8AB01-0AB0	IM 151-8PN CPU	32	32	
6ES7 151-8FB00-0AB0	IM 151-8PN F-CPU PROFIsafe	16	16	
6ES7 151-8FB01-0AB0	IM 151-8PN F-CPU PROFIsafe	32	32	

a. As the max. IO-data per cycle is not higher than 4 for this CPU, The 1 SI CANopen can not be used as a CAN2.0A-device, where minimum size is 16 bytes.

D. ET200S Terminal Modules

Below is a list of ET200S terminal modules that can be used with the 1 SI CANopen Module for ET200S.

Usable terminal modules

TM-E15C26-A1 (6ES7193-4CA50-0AA0)	TM-E15C24-A1 (6ES7193-4CA30-0AA0)	TM-E15C24-01 (6ES7193-4CB30-0AA0)	TM-E15C23-01 (6ES7193-4CB10-0AA0)	← Spring terminal
TM-E15S26-A1 (6ES7193-4CA40-0AA0)	TM-E15S24-A1 (6ES7193-4CA20-0AA0)	TM-E15S24-01 (6ES7193-4CB20-0AA0)	TM-E15S23-01 (6ES7193-4CB00-0AA0)	← Screw-type terminal
TM-E15N26-A1 (6ES7193-4CA80-0AA0)	TM-E15N24-A1 (6ES7193-4CA70-0AA0)	TM-E15N24-01 (6ES7193-4CB70-0AA0)	TM-E15N23-01 (6ES7193-4CB60-0AA0)	← Fast Connect
				

1) To minimize the signal's reflections from the end of the cable, a line termination is needed close to each end of the bus. This is often included in CANopen cables. If the cable you are using is not terminated, connect a line termination between contacts 1 and 5 (CAN_L and CAN_H) beneath the slot, where the module is mounted.

2) To ensure backwards compatibility, these contacts should not be connected

E. CANopen Emergency Codes

Below is a list of the CANopen emergency codes that can be produced by the 1 SI CANopen Module for ET200S. The five latest emergencies, reported by the module, can be read from the list in object 1003h, see page 32.

Emergency Code	Description	Reason
0000h	Error reset or no error	Bootup or all errors cleared.
3100h	Mains voltage – generic error	Communication problem on backplane.
6161h	Internal software error	Internal software error. See table below.
8000h	Monitoring– generic error	Boot time error
8110h	Monitoring - CAN overrun (message lost)	
8120h	Monitoring - CAN error passive	
8130h	Monitoring - error control failure	Life guard error or heartbeat error.
8140h	Monitoring - recovered from bus off	
8220h	Protocol error - PDO length exceeded	
FF10h	Device specific - generic error	State of PLC is different from state of network.

These codes conform to the CANopen standard.

Emergency code 6161h

Byte 0 contains an error code and byte 1 contains the Node ID (if available).

The first two bytes of the manufacturer specific information in this emergency will be stored in object 1003h (see page 32) as additional information.

Byte #	Description	
0	Error code	
	1 - 2	(reserved)
	3	Could not transmit SDO
	4	Invalid concise entry (index < 1000h or subindex is > 255)
	5	(reserved)
	6	Data could not be written to non-volatile memory
	7	SDO time out
	8	SDO error
	9	Invalid object dictionary stored in flash
1	Node ID (if available)	

F. Error Codes (RET)

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		
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

Error code[a]	Additional error code[a]	Description	SFC/SFB number[b]	RET_VAL [b]
38	-	Error while communicating with the CANopen master. The additional error code is equal to the error code from the CANopen master.	-	-
	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 sub index		
	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		
39	-	Acyclic error. Additional error code equals error code from the CANopen master.	-	-
	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		
	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)

Error code[a]	Additional error code[a]	Description	SFC/SFB number[b]	RET_VAL [b]
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	-	-
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 1 SI CANopen function blocks if the error originates from outside the module. Please consult the SIMATIC STEP7 manual for further information.