wieland

samos® PRO

Gateways

Operating instructions

Doc. No. BA000587
Issued: 06/2012 (Rev. C)

CE

Contents

Contents

1 About this document

Please read this chapter carefully before working with these operating instructions and the ***samos***®PRO gateways.

1.1 Function of this document

These operating instructions only apply in conjunction with the other ***samos***®PRO operating instructions (see section 1.2 “The ***samos***®PRO operating instructions” below) and provide the technical personnel at the machine manufacturer or machine operating organisation information on safe mounting, adjustment, electrical installation, commissioning as well as on operation and maintenance of the ***samos***®PRO gateways.

These operating instructions do not provide information on the operation of the machine in which a ***samos***®PRO modular safety controller and a ***samos***®PRO gateway is integrated. Information on this is to be found in the appropriate operating instructions for the machine.

1.2 The *samos*®PRO operating instructions

For the ***samos***®PRO system there are three operating instructions with clearly distinguished fields of application as well as mounting instructions for each module.

- The mounting instructions (Wieland document nos. BA000572, BA000583) are enclosed with each ***samos***®PRO module. They inform on the basic technical specifications of the modules and contain simple mounting instructions. Use the mounting instructions when mounting ***samos***®PRO safety controllers.
- The ***samos***®PRO hardware operating instructions (Wieland document no. BA000497) describe all ***samos***®PRO modules and their functions in detail. Use the Hardware operating instructions in particular to configure ***samos***®PRO safety controllers.
- The ***samos***®PRO gateways operating instructions (this document) describe all ***samos***®PRO gateways and their functions in detail.
- The ***samos***®PLAN operating instructions (Wieland part no. BA000518) describe the software-supported configuration and parameterization of the ***samos***®PRO safety controllers. In addition, the software operating instructions contain the description of the diagnostics functions that are important for operation and detailed information for the identification and elimination of errors. Use the Software operating instructions in particular for the configuration, commissioning and operation of ***samos***®PRO safety controllers.

1.3 Target group

These operating instructions are addressed to planning engineers, machine designers and the operators of systems in which a ***samos***®PRO modular safety controller is integrated and who want to exchange data with a fieldbus (a controller) via a gateway.

They are also addressed to people who are placing a ***samos***®PRO gateway in operation for the first time or maintaining it.

1.4 Information depth

These operating instructions contain information on the ***samos***®PRO gateways on the following subjects:

- mounting
- implementation into a network
- configuration via ***samos***®PLAN software
- data transfer to and from the network
- status information, planning and related mapping
- part numbers

Warning!
Pay attention to the safety notes and safety measures on the ***samos***®PRO gateway!

NOTE

We also refer you to our eCatalogue in the Internet at (http://eshop.wieland-electric.com)
There you will find the following files for download on the product pages:
- SP-EN-IP EDS file for EtherNet/IP
- SP-EN-PN GSDML file for Profinet IO
- SP-PROFIBUS-DP GSD file for PROFIBUS DP
- SP-CANopen EDS file for CANopen

1.5 Scope

These operating instructions apply to the ***samos***®PRO gateway modules SP-PROFIBUS-DP, SP-CANopen, SP-EN-MOD, SP-EN-IP and SP-EN-PN.

This document is part of Wieland document number BA000587 ("***samos***®PRO gateways" operating instructions in all available languages).

This document is the original operating instruction.

1.6 Abbreviations used

EFI	Enhanced Function Interface
SINT	Short integer = 1 Byte
UDINT	Unsigned double integer = 4 Bytes = 2 Words
UINT	Unsigned integer = 2 Byte = 1 Word

1.7 Symbols used

NOTES

Refer to notes for special features of the device.

Warning!
A warning notice indicates an actual or potential risk or health hazard. They are designed to help you to prevent accidents.
Read carefully and follow the warning notices!

1.8 Trademarks

Windows 98, Windows NT 4.0, Windows 2000, Windows XP and Internet Explorer are registered trademarks of Microsoft Corporation in the USA and other countries.

SIEMENS SIMATIC Manager is a registered trademark of SIEMENS AG.

DeviceNet and DeviceNet Safety are registered trademarks of the Open DeviceNet Vendor Association, Inc. (ODVA).

Other product names and company names referenced in this manual are trademarks or registered trademarks of their respective companies.

2 On safety

This chapter deals with your own safety and the safety of the equipment operators.

➡ Please read this chapter carefully before working with a ***samos***®PRO gateway.

2.1 Qualified safety personnel

The ***samos***®PRO gateway must only be installed, commissioned and serviced by qualified safety personnel.

Qualified safety personnel are defined as persons who …

- have undergone the appropriate technical training, and
- have been instructed by the responsible machine operator in the operation of the machine and the current valid safety guidelines, and
- have access to the operating instructions of the ***samos***®PRO gateway and ***samos***®PRO modular safety controller and have read and familiarised themselves with them.

2.2 Intended use

The ***samos***®PRO gateways can only be operated with a ***samos***®PRO system. The firmware version of the connected SP-SCON must be at least V1.11.0, the version of the ***samos***®PLAN configuration software must be at least 1.3.0.

The ***samos***®PRO gateways do not have a dedicated voltage supply.

The *samos*®PRO gateways are not suitable for operation on a safety fieldbus!
These gateways only generate non-safety-related fieldbus data (status bytes) for control and diagnostics purposes.

Do not use non-safe data from a *samos*®PRO gateway for safety related applications!
With the ***samos***®PRO gateways it is possible to integrate non-safe data into the logic editor such that the safety function of the ***samos***®PRO system is compromised. Never implement the gateway into a ***samos***®PRO system without having this danger checked by a safety specialist.

These modules may only be used by qualified safety personnel and only on the machine where they have been installed and initialised by qualified safety personnel in accordance with the operating instructions.

Pay attention to the safety notes and safety measures on the *samos*®PRO gateway!
If the device is used for any other purposes or modified in any way — also during mounting and installation — any warranty claim against Wieland Electric GmbH shall become void.

NOTES

- During the mounting, installation and usage of the ***samos***®PRO gateway, observe the standards and directives applicable in your country.
- The national/international rules and regulations apply to the installation, commissioning, use and periodic technical inspection of the ***samos***®PRO modular safety controller, in particular:
  - EMC directive 2004/108/EC,
  - Provision and Use of Work Equipment Directive 2009/104/EC,
  - the work safety regulations/safety rules.
- The operating instructions must be made available to the operator of the machine where a ***samos***®PRO system is used. The machine operator is to be instructed in the use of the device by qualified safety personnel and must be instructed to read the operating instructions.

The ***samos***®PRO system complies, as per the "radiated emissions" generic standard, with the requirements of class A (industrial applications). The ***samos***®PRO system is therefore only suitable for use in an industrial environment.

WARNING

2.3 Environmental protection

The ***samos***®PRO gateways are designed for minimum impact on the environment, they consume only a minimum of energy and resources.

➡ At work, always act in an environmentally responsible manner.

2.3.1 Disposal

Unusable or irreparable devices should always be disposed as per the applicable national regulations on waste disposal (e.g. European waste code 16 02 14).

NOTE

We would be pleased to be of assistance to you on the disposal of these devices. Contact us.

2.3.2 Separation of materials

Only appropriately trained personnel are allowed to separate materials!
Caution is required when dismantling devices. There is a risk of injuries.

WARNING

Before you send the devices for appropriate recycling, it is necessary to separate the different materials of the ***samos***®PRO gateways.

➡ Separate the housing from the rest of the parts (in particular the circuit board).
➡ Send the separated parts for recycling as appropriate (see **Tab. 1**).

Components	Disposal
Product Housing, circuit boards, cables, connectors and electrical connecting pieces	Electronic recycling
Packaging Cardboard, paper	Paper/cardboard recycling

Tab. 1: Overview on disposal by components

3 Product description *samos*®PRO gateways

The ***samos***®PRO gateways allow the ***samos***®PRO System to send and receive non-safety related data to and from the external fieldbus system for control and diagnostics purposes.

NOTE

In this manual, the data exchanged between the ***samos***®PRO system and the respective network will be considered always from the network master (PLC) point of view. Therefore data sent from the ***samos***®PRO system into the network will be referred to as *input data* while data received from the network will be referred to as *output data*.

WARNING

Do not operate a *samos*®PRO gateway on a safety fieldbus!
The ***samos***®PRO gateway modules are not suitable for operation on a safety fieldbus. They do not support any safety mechanism, which would be mandatory to communicate within a safety network.

Configuration of the ***samos***®PRO gateways is performed using the ***samos***®PLAN configuration software on a PC or laptop connected to the SP-SCON over RS-232 interface or connected to the Ethernet gateways over Ethernet TCP/IP.

The safety relevant logic of the ***samos***®PRO system operates independently from the gateway. If however the ***samos***®PRO system has been configured to integrate non-safe information from the fieldbus into the logic editor, a decoupling of the gateway can result in availablity problems.

A ***samos***®PRO gateway can only be operated on a ***samos***®PRO system. It does not have a dedicated voltage supply. It is possible to use two ***samos***®PRO gateways per system.

The gateways are fitted in a 22.5 mm wide housing for 35 mm rails in accordance with EN 60715.

Ordering information can be found in section 9 “Ordering information” on page 122.

3.1 Device variants

Four ***samos***®PRO gateways are available for the different network types. Suitable for Ethernet networks are the EtherNet/IP gateway SP-EN-IP, the Modbus TCP gateway SP-EN-MOD and the Profinet IO gateway SP-EN-PN. The SP-PROFIBUS-DP and the SP-CANopen are fieldbus gateways without Ethernet functionality. With the SP-DeviceNet for DeviceNet, a further fieldbus gateway will be available in the future.

Tab. 2: Device variants and features overview

Gateway	Network type	Ethernet TCP/IP socket interface	TCP/IP configuration interface
SP-EN-IP	EtherNet/IP explicit messaging	Client/server	Available at port 9000
SP-EN-MOD	Modbus TCP master & slave receive method	Client/server	Available at port 9000
SP-EN-PN	PROFINET IO slave conformance class A	Client/server	Available at port 9000
SP-PROFIBUS-DP	PROFIBUS DP slave	–	–
SP-CANopen	CANopen slave	–	–

NOTE

You will find the device’s date of manufacture on the type label in the *S/N* field in the format ww/yyyy (yyyy = year, ww = calendar week).

3.2 Data transmitted into the network (network input data sets)

Available data

The ***samos***®PRO gateways can provide the following data:

- Operational data
  - **Logic results** from the ***samos***®PRO main unit (SP-SCON) (see section 3.2.1 on page 14)
  - **Input values** (HIGH/LOW) for all ***samos***®PRO input extension modules in the system and EFI devices connected (see section 3.2.3 on page 15)
  - **Output values** (HIGH/LOW) for all ***samos***®PRO input/output extension modules connected (see section 3.2.3 on page 15)
  - **Output data** from another network, i.e. data received by a second gateway in the ***samos***®PRO system (see section 3.2.4 on page 15)
- Diagnostics
  - **Checksums** (CRCs) (see section 3.2.5 on page 15)
  - **Error and status information** for all modules except the SA-OR-S2 and SA-OR-S1 (see section 3.2.6 on page 16)

Data sets

The physical ***samos***®PRO modules are not represented as typical hardware modules in the network. Instead, the data available from the ***samos***®PRO system has been organized into four input *data sets*.

- **Data set 1** (max. 50 bytes) contains the operational data. It can be compiled using the ***samos***®PLAN tool. Upon delivery there is a default selection for the content of data set 1 which can be freely modified. For details see Tab. 4 on page 14.
  For the SP-EN-PN and the SP-PROFIBUS-DP, data set 1 has been subdivided in five input *data blocks*, where data block 1-4 contain 12 bytes each and data block 5 contains two bytes. For detailed information see the section on the related gateway.
- **Data set 2** (32 bytes) contains the system configuration CRCs. See Tab. 4 on page 14.
- **Data set 3** (60 bytes) contains the individual module status and diagnostics data with four (4) bytes per module. For details see **Tab. 5** on page 16.
- **Data set 4** (60 bytes) is currently filled with reserved values.

Tab. 3 gives an overview which data sets are available for which gateway.

	Data set 1	Data set 2	Data set 3	Data set 4
SP-EN-IP	EtherNet/IP or TCP/IP	EtherNet/IP or TCP/IP	EtherNet/IP or TCP/IP	EtherNet/IP or TCP/IP
SP-EN-MOD	Modbus TCP or TCP/IP	Modbus TCP or TCP/IP	Modbus TCP or TCP/IP	Modbus TCP or TCP/IP
SP-EN-PN	Profinet IO or TCP/IP	Profinet IO or TCP/IP	Profinet IO or TCP/IP	Profinet IO or TCP/IP
SP-PROFIBUS-DP	PROFIBUS DP	–	–[1)]	–
SP-CANopen	CANopen	CANopen (SDOs)	CANopen (SDOs)[2)]	–

Tab. 3: Availability of data set 1-4

[1)] With the SP-PROFIBUS-DP, diagnostics data is available via PROFIBUS standard DP-V0 diagnostics. For more information on how to retrieve module status and diagnostics data via the PROFIBUS DP gateway please refer to chapter 6.1 “PROFIBUS DP gateway” on page 72.

[2)] With the SP-CANopen, diagnostics data are available via CANopen SDO (service data objects) method. For more information on how to retrieve module status and diagnostics data via the CANopen gateway please refer to chapter 6.2 “CANopen gateway” on page 83.

Tab. 4: Overview input data sets 1-3 (default settings for EtherNet IP, Modbus TCP and TCP/IP)

	Data set 1	Data set 2	Data set 3
Byte 0	Logic result 1	Overall CRC	Module status module 0. Module 0 is always the CPU. For detailed information about the module status see Tab. 5.
Byte 1	Logic result 2		
Byte 2	Logic result 3		
Byte 3	Logic result 4		
Byte 4	Input values module 1	System CRC (SCID)	Module status module 1
Byte 5	Input values module 2		
Byte 6	Input values module 3		
Byte 7	Input values module 4		
Byte 8	Input values module 5	Reserved	Module status module 2
Byte 9	Input values module 6		
Byte 10	Input values module 7		
Byte 11	Input values module 8		
Byte 12	Input values module 9		Module status module 3
Byte 13	Input values module 10		
Byte 14	Input values module 11		
Byte 15	Input values module 12		
Byte 16	Output values module 1		Module status module 4
Byte 17	Output values module 2		
Byte 18	Output values module 3		
Byte 19	Output values module 4		
Byte 20	Output values module 5		Module status module 5
Byte 21	Output values module 6		
Byte 22	Output values module 7		
Byte 23	Output values module 8		
Byte 24	Output values module 9		Module status module 6
Byte 25	Output values module 10		
Byte 26	Output values module 11		
Byte 27	Output values module 12		
Byte 28	Not assigned		Module status module 7
Byte 29	Not assigned		
Byte 30	Not assigned		
Byte 31	Not assigned		
Byte …	Not assigned		…
Byte 49	Not assigned		…
Byte …			…
Byte 56			Module status module 14. Module 13 and 14 are always the gateways.
Byte 57			
Byte 58			
Byte 59			
Length	50 bytes	32 bytes	60 bytes

NOTE If there are dual channel input or output elements configured at the IO module, then only the lowest bit represents the element's input or output status (on/off). It is represented by the element's tag name. The highest bit is not supported.

3.2.1 Logic results

Logic results generated by the logic editor of the ***samos***®PRO main unit can be made available to the network. Up to 20 bytes are available where each bit represents one logic result from the logic editor.

Data set 1 containing the logic results can be customized. For detailed information see the chapter on the related gateway and chapter 7 "Layout and content of the process image" on page 109.

3.2.2 Gateway direct output values

It is possible to write values directly from the logic editor into a gateway. There are four bytes reserved for this in the default settings for data set 1, however up to all 50 bytes of data set 1 can be defined as gateway direct output values. For more information please see section 7.3.5 “Gateway direct output values” on page 113.

NOTE

In order to use gateway direct output values, a CPU with firmware V2.00.0 or higher is required.

3.2.3 Module and EFI input and output values

The ***samos***®PRO gateways can transmit all input and output states of all samos(r)PRO modules and EFI devices connected to the ***samos***®PRO system into the network. Data set 1 containing the input and output values and the EFI information can be customized. For detailed information see the chapter on the related gateway and chapter 7 “Layout and content of the process image” on page 109.

Module input and output states

The input and output states of the modules are transmitted using one byte for each module’s inputs and one byte for each module’s outputs where each bit represents the state of one input or output (on/off).

EFI system information

The SP-SCON-NET main module has two EFI interfaces. An EFI interface is a safe communication interface between Wieland Electric devices. It allows to

- read out information from the safety devices (e.g. safety light curtains, safety laser scanners)
- transfer commands to the safety devices

The ***samos***®PRO gateways allow these EFI devices connected to the SP-SCON-NET to transmit their data into the network.

NOTE

It is only possible to select the EFI data in byte arrays. 4 byte arrays for each connected EFI device are available. Some of the data content is reserved and can not be used at the PLC.

Further information about the properties, functions and benefits of the EFI interfaces can be found in the ***samos***®PRO hardware operating instructions (Wieland Electric doc. no. BA000497).

The general EFI function description is available in the ***samos***®PLAN software manual (Wieland Electric doc. no. BA000518).

3.2.4 Routing of data from a second network

If your ***samos***®PRO system contains two gateways, it is possible to route information received by the first gateway from one network (e.g. from a Modbus PLC) into a second network via the second gateway (e.g. to a PROFIBUS master) and vice versa.

3.2.5 Configuration checksums (CRCs)

Data set 2 contains the following configuration CRCs for the ***samos***®PRO system:

- Overall CRC (same as system CRC)
- System CRC (SCID)

Each checksum is four bytes long. The overall CRC is the checksum displayed in the ***samos***®PLAN report. Data set 2 can not be customized.

3.2.6 Error and status information of the modules

Data set 3 contains the module status information transferred to the network.

Four bytes are used for each module (e.g. SP-SDIO). These four bytes are being transferred in Big Endian format, i.e. in 16 bit word format with the first byte placed in the least significant, or rightmost byte of the integer and the second byte placed in the most significant, or leftmost byte of the integer:

Data set 3 can not be customised.

Tab. 5: Meaning of the module status bits

Byte	Bit	I/O modules (e.g. SP-SDIO, SP-SDI)[3]	CPU modules (e.g. SP-SCON)	Gateway modules
0	0	Module operating state 1 = Executing 0 = Any other state	Module operating state 1 = Executing 0 = Any other state	Module operating state 1 = Executing 0 = Any other state
	1	Internal error: Internal tests failed or watchdog test failed or bad process data or self test failure 1 = No error 0 = Error	Module operating state is Critical Fault 1 = No error 0 = Critical Fault	Internal error: Internal tests failed, bad process data 1 = No error 0 = Error
	2	External error: Input test or dual channel evaluation failure, or bad output power supply range, or output(s) stuck-at-high or stuck-at-low. 1 = No error 0 = Error	Power supply out of range 1 = No error 0 = Error	External error: network connection inactive/failure 1= No error 0 = Error/inactive
	3	Reserved	Reserved	Reserved
	4	Configuration status changed to invalid. 1 = Configuration valid 0 = Conf. invalid or unknown	Configuration status changed to invalid. 1 = Configuration valid 0 = Conf. invalid or unknown	Configuration status changed to invalid. 1 = Configuration valid 0 = Conf. invalid or unknown
	5	Output power supply out of range. 1 = Power supply o.k. 0 = Power supply out of range	Output power supply out of range. 1 = Power supply o.k. 0 = Power supply out of range	Input status 1 = Valid network communication 0 = Invalid or no communication
	6	Reserved	EFI 1 communication failure 1 = No error 0 = Error	Output status 1 = Valid network communication 0 = Invalid or no communication
	7	Reserved	EFI 2 communication failure 1 = No error 0 = Error	Reserved
1	0	Input 1-2 dual channel input evaluation error 1 = No error 0 = Error	Reserved	Reserved
	1	Input 3-4 dual channel input evaluation error 1 = No error 0 = Error		
	2	Input 5-6 dual channel input evaluation error 1 = No error 0 = Error		
	3	Input 7-8 dual channel input evaluation error		

[3] The module status bits for the SP-SDIO and SP-SDI are fully supported only with firmware version 1.2.x and higher.

Byte	Bit	I/O modules (e.g. SP-SDIO, SP-SDI)[3]	CPU modules (e.g. SP-SCON)	Gateway modules
		1 = No error 0 = Error		
	4	Status output 1 fast shut off logic control time out. 1 = No error 0 = error		
	5	Status output 2 fast shut off logic control time out. 1 = No error 0 = error		
	6	Status output 3 fast shut off logic control time out. 1 = No error 0 = error		
	7	Status output 4 fast shut off logic control time out. 1 = No error 0 = error		
2	0	Input 1 external test signal failure. 1 = No error 0 = Error	Reserved	Reserved
	1	Input 2 external test signal failure. 1 = No error 0 = Error		
	2	Input 3 external test signal failure. 1 = No error 0 = Error		
	3	Input 4 external test signal failure. 1 = No error 0 = Error		
	4	Input 5 external test signal failure. 1 = No error 0 = Error		
	5	Input 6 external test signal failure. 1 = No error 0 = Error		
	6	Input 7 external test signal failure. 1 = No error 0 = Error		
	7	Input 8 external test signal failure. 1 = No error 0 = Error		
3	0	Output 1 stuck-at-high error. 1 = No error 0 = Error	Reserved	Reserved
	1	Output 1 stuck-at-low error. 1 = No error 0 = Error		
	2	Output 2 stuck-at-high error. 1 = No error 0 = Error		
	3	Output 2 stuck-at-low error. 1 = No error 0 = Error		
	4	Output 3 stuck-at-high error. 1 = No error 0 = Error		
	5	Output 3 stuck-at-low error. 1 = No error 0 = Error		
	6	Output 4 stuck-at-high error.		

Byte	Bit	I/O modules (e.g. SP-SDIO, SP-SDI)[3]	CPU modules (e.g. SP-SCON)	Gateway modules
		1 = No error 0 = Error		
	7	Output 4 stuck-at-low error. 1 = No error 0 = Error		

NOTES

- Reserved (for future use) = static 1 (no status change)
- If no module is present, all values including the reserved values are set to logical 1.

You will find an example process image in section 5.1.3 “TCP/IP process image example” on page 37.

3.3 Data received from the network (network output data sets)

The data received from the network are organised in output data sets (max. 50 bytes). These data have been subdivided in five data blocks holding 10 bytes each for the SP-EN-IP, SP-EN-MOD and SP-EN-PN; for the SP-PROFIBUS-DP output data blocks 1-4 hold 12 bytes each while output data block 5 holds 2 bytes.

Tab. 6: Output data blocks 1-5 for the different gateways

	Data block output size				
Gateway	**Block 1**	**Block 2**	**Block 3**	**Block 4**	**Block 5**
SP-EN-IP	10 bytes	10 bytes	10 bytes	10 bytes	10 bytes
SP-EN-MOD	10 bytes	10 bytes	10 bytes	10 bytes	10 bytes
SP-EN-PN	10 bytes	10 bytes	10 bytes	10 bytes	10 bytes
SP-PROFIBUS-DP	12 bytes	12 bytes	12 bytes	12 bytes	2 bytes
SP-CANopen	8 bytes	8 bytes	8 bytes	8 bytes	–

The contents of the output data blocks can be used within the ***samos***®PRO CPU logic editor and can also be made available to another network via a second ***samos***®PRO gateway in the ***samos***®PRO system.

NOTES

- In order to make the data from the network available in the logic editor or as input to another network, you will have to define a tag name for each bit that shall be used.
- Bits without a specific tag name will not be available in the logic editor nor for routing via a second gateway. For detailed information on how to define tag names for the data received please see the related section in the chapters on the different gateways.
- The status of the communication to and from the network can be monitored in the logic editor using the module input status bit for data from the network and the module output status bit for data to the network. When the gateway detects an invalid communication, the contents of the data sets will be set to zero (logical 0) and the corresponding module status bit will also be set to zero (logical 0).
- In case the communication is dropped, the data of the output data sets will be set to zero (logical 0) and the module input status bit will also be set to zero (logical 0).
- If a connection is closed while still others are available, the MS LED flashes for 10 s Red/Green and an error history entry is made. In this case, the status bits will not be affected.

WARNING

Do not use the same output data set number for two different PLC connections or TCP/IP sockets!
The output data set can be written to the Ethernet gateways in parallel by all communication interfaces or TCP/IP sockets (e.g. Modbus TCP and Ethernet TCP/IP), if they use the same output data set number. In that case the last message overrides data received earlier.

4 Mounting and basic configuration of the gateways

4.1 Mounting/Dismantling

This chapter describes the mounting of the ***samos***®PRO gateways.

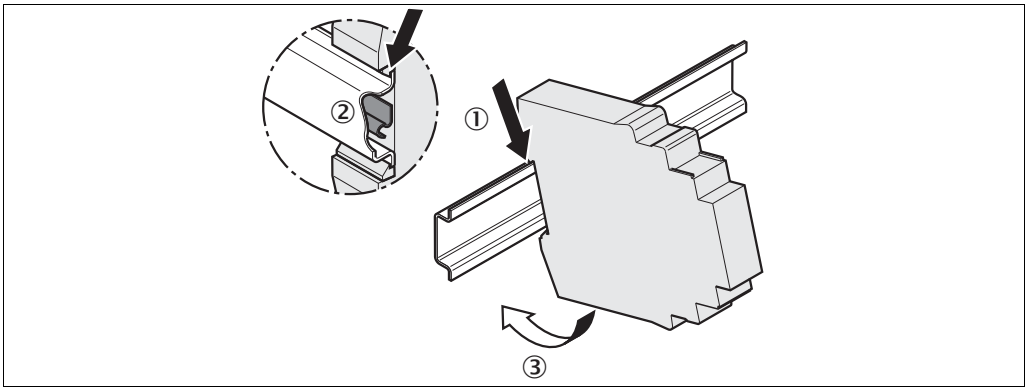
Make sure that the connection of the *samos*®PRO gateway cannot lead to hazardous situations during installation!
Ensure that connecting a ***samos***®PRO gateway cannot lead to a hazardous situation when implementing the unit on to the ***samos***®PRO system and Ethernet network. Prevent unintended start-up of equipment during connection of a ***samos***®PRO gateway.

4.1.1 Steps for mounting the modules

- The ***samos***®PRO system is only suitable for mounting in a control cabinet with at least IP 54 degree of protection.
- While supply voltage is applied, modules must not be plugged to nor be removed from the ***samos***®PRO system.
- To ensure full electromagnetic compatibility (EMC), the DIN mounting rail must be connected to functional earth (FE). Additionally connect all network cable shields directly at the control cabinet entrance to a common FE ground line.

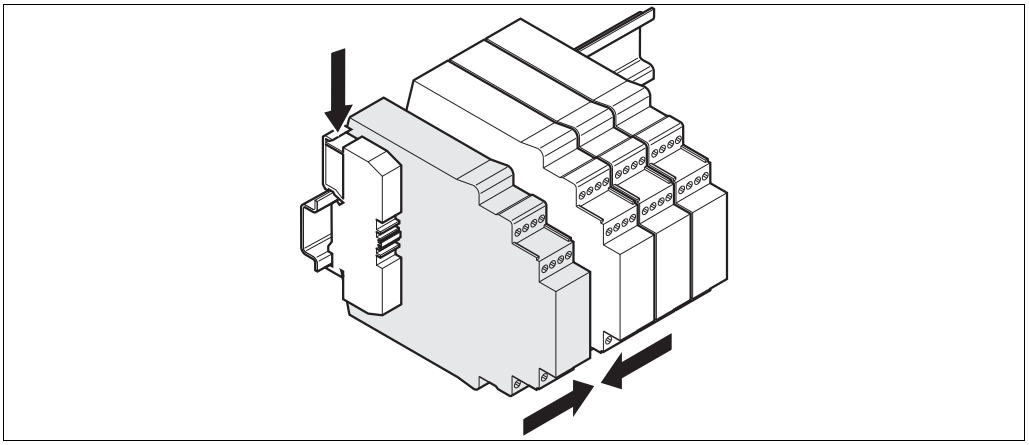
- In a ***samos***®PRO system the controller module SP-SCON is positioned at the extreme left.
- The two optional gateways follow directly to the right of the controller module.
- Connect further ***samos***®PRO extension modules (e.g. SP-SDIO or SP-SDI) onto the right side of the gateways and any additional relay modules (SA-OR-S2 or SA-OR-S1) to the extreme right of the entire ***samos***®PRO system.
- Ensure that suitable ESD protective measures are taken during mounting. Otherwise the devices may be damaged.
- The connection between the modules is effected by means of the plug connection integrated in the housing. Take into account that, when replacing a module, the ***samos***®PRO modules have to be pushed approx. 10 mm apart before the corresponding module can be removed from the DIN rail.
- Take suitable measures to ensure that foreign matter does not penetrate the connector openings, in particular that of the removable memory plug.
- Mount the modules in accordance with EN 50274.
- The modules are located in a 22.5 mm wide modular system for 35 mm DIN rails according to EN 60715.

Fig. 1: Mounting the module onto the DIN rail



- Make sure that the voltage supply of the ***samos***®PRO system is switched off.
- Hang the device onto the DIN rail (1).
- Connect the gateways directly onto the right side of the SP-SCON module of the ***samos***®PRO system. Up to two gateways per system are possible.
- Ensure that the earthing spring contact (2) contacts the DIN rail such that it can electrically conduct.
- Latch the module onto the DIN rail by pressing it lightly in the direction of the arrow (3).

Fig. 2: Installing the end clips



- If there are several modules, slide the modules together individually in the direction of the arrow until the side plug connection latches in.
- Install end clips on the left and right.

The following steps are necessary after mounting:

- Complete the electrical connections (see section 4.2 “Electrical installation” on page 30)
- Configuration (see section 4.3 “First configuration steps” on page 22 and the ***samos***®PLAN operating instructions (Wieland part no. BA000518).
- Checking the installation (see the chapter on commissioning in the ***samos***®PRO hardware operating instructions, Wieland document no. BA000497).

4.1.2 Steps for dismantling the modules

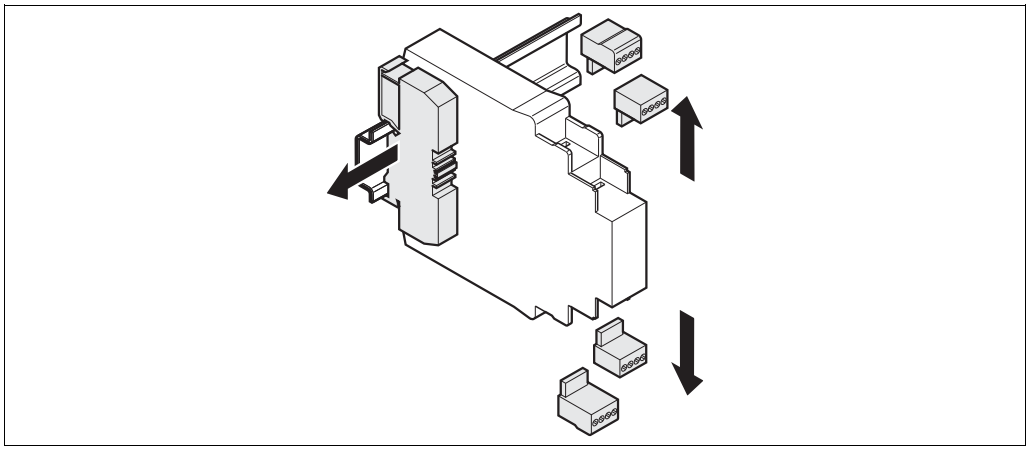


Fig. 3: Removing the removable terminals

➡ Remove the removable terminals with the wiring and the end clips.

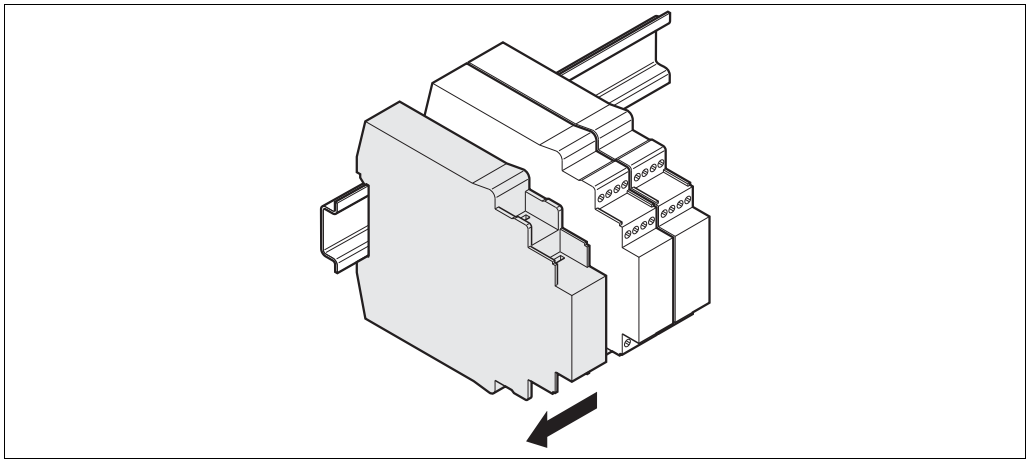


Fig. 4: Disconnecting the plug connections

➡ If there are several modules, slide the modules away from each other individually in the direction of the arrow until the side plug connection is separated.

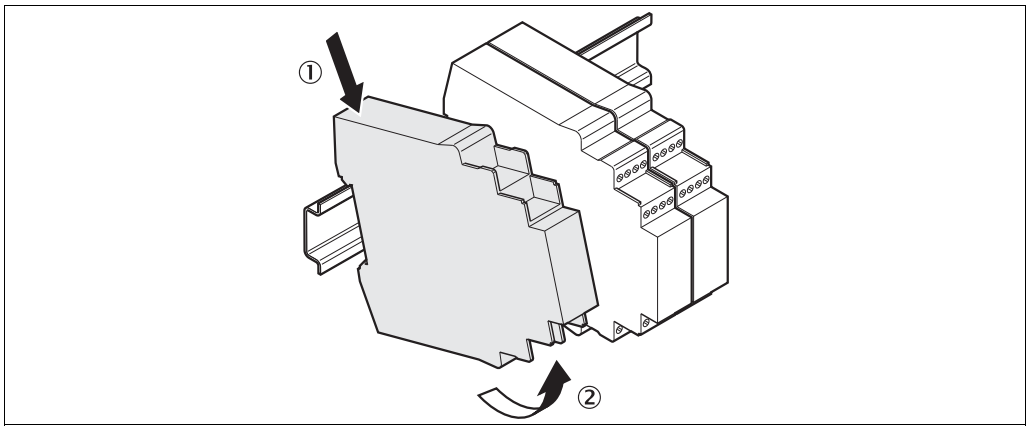


Fig. 5: Removing modules from the DIN rail

➡ Press the module downwards at the rear (1) and remove it from the DIN rail in the direction of the arrow while keeping it pressed down (2).

4.2 Electrical installation

WARNING

Switch the entire machine/system off line!
The system could start up unexpectedly while you are connecting the devices.

NOTES

- The ***samos***®PRO gateways fulfil the EMC requirements in accordance with the basic specification EN 61000-6-2 for industrial use.
- To ensure full electromagnetic compatibility (EMC), the mounting rail has to be connected to functional earth (FE).
- The control cabinet or assembly casing of the ***samos***®PRO system must comply at least with enclosure rating IP 54.
- Mounting in accordance with EN 50274.
- Electrical installation in accordance with EN 60204-1.
- The voltage supply of the devices must be capable of buffering brief mains voltage failures of 20 ms as specified in EN 60204-1.
- The voltage supply has to fulfil the regulations for extra-low voltages with safe separation (SELV, PELV) in accordance with EN 60664 and DIN 50178 (equipment of electrical power installation with electronic devices).
- Ensure that all the modules of the ***samos***®PRO system, the connected protective devices as well as the voltage supplies are connected with the same GND plane. The GND of the RS232 interface is connected internally to the GND of the supply of the controller module (A2).
- Connect all fieldbus and Ethernet cable shields directly at the control cabinet entrance to the functional earth (FE).

4.3 First configuration steps

This chapter describes the basic steps you have to perform for the configuration of the gateway:

- Establish a first connection between the gateway and a PC or laptop
- Upload or transfer of a configuration
- Verification of a configuration

4.3.1 Establishing a connection between gateway and PC

➡ Connect a PC or notebook to the RS-232 interface of the SP-SCON.
➡ Power on the ***samos***®PRO System.
➡ Open the ***samos***®PLAN configuration tool installed on the PC.

➡ Click on **Edit com. interface settings** to ensure the correct communication interface has been selected. The following dialog appears:

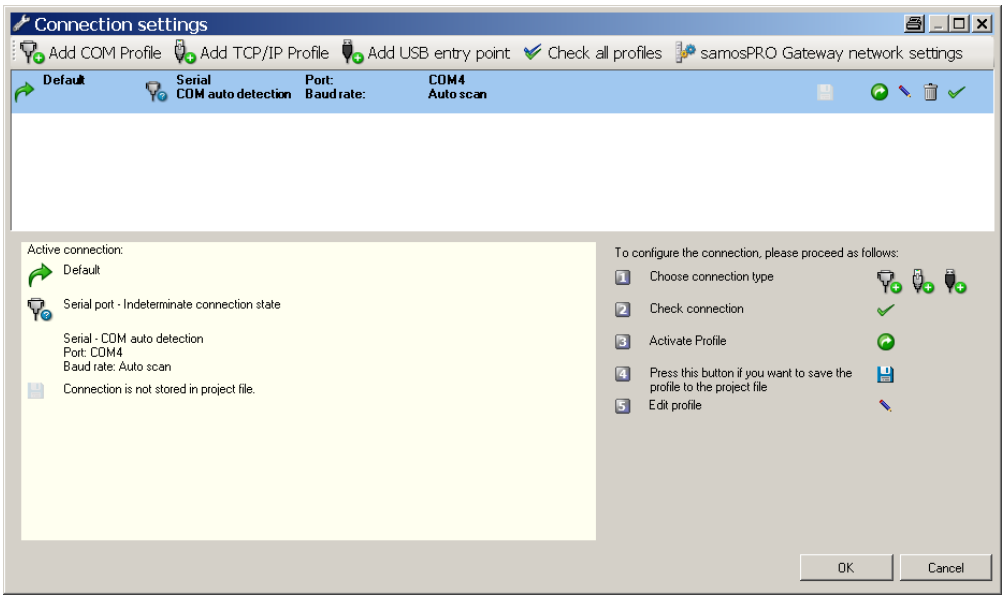


Fig. 6: Com settings dialog

➡ To edit the settings click on the pencil icon to the right. The following dialog appears:

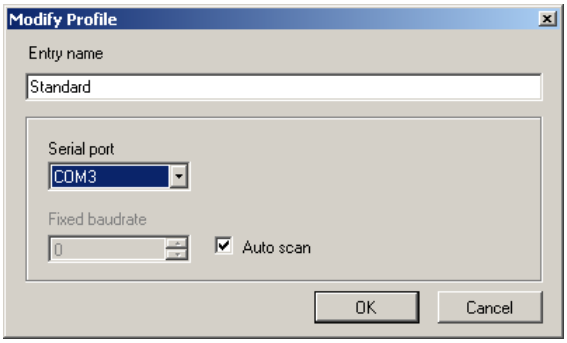


Fig. 7: Modify profile dialog

➡ Modify the settings if required and click **OK**.

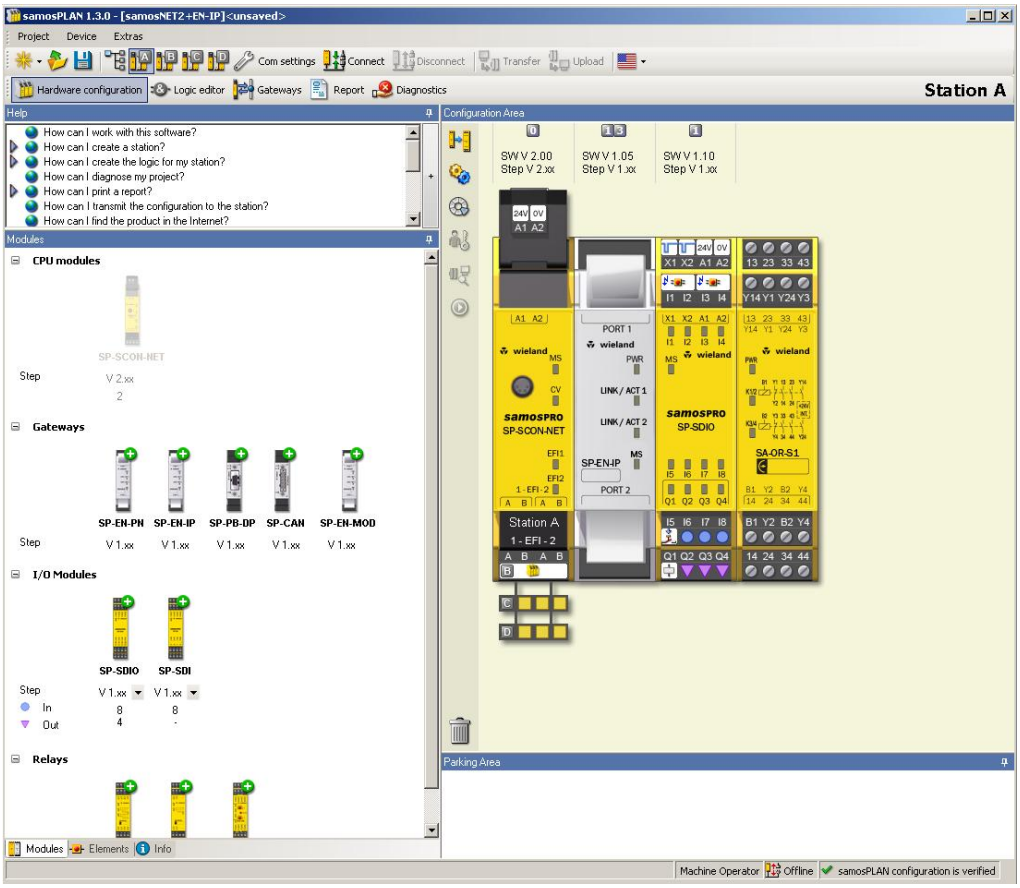
➡ Click **OK**. The dialog closes.

➡ Click on **Connect to physical device**. The ***samos***®PLAN will search for connected ***samos***®PRO devices and load the hardware configuration into the hardware configuration dialog. Once all modules have been identified correctly, the ***samos***®PLAN will ask whether the configuration shall be uploaded.

➡ Click **Yes** to upload the configuration.

As an example, the following hardware configuration may appear:

Fig. 8: Hardware configuration dialog



➡ Click **Disconnect** to go into the offline mode if you want to change the configuration of the ***samos***®PRO modules.

4.3.2 Configuration of the gateways

For the configuration of the gateways please refer to the sections on the related gateway:

- Section 5.2 “EtherNet/IP gateway” on page 38
- Section 5.3 "Modbus TCP gateway" on page 52
- Section 5.4 "PROFINET IO gateway" on page 60
- Section 6.1 “PROFIBUS DP gateway” on page 72
- Section 6.2 "CANopen gateway" on page 83

For the configuration of the TCP/IP interface of the Ethernet gateways, please refer to the following sections:

- Section 5.1.1 “TCP/IP configuration interface” on page 26
- Section 5.1.2 “Ethernet TCP/IP socket interface” on page 30

For the configuration of the operational data (data transfer from and to the network), please refer to chapter 7 “Layout and content of the process image” on page 109.

More information can be found in the ***samos***®PLAN operating instructions (Wieland doc. no. BA000518).

4.3.3 Transfer of a configuration

Once you have finished the configuration, you have to transfer the configuration to your ***samos***®PRO system. In order to transfer a configuration, perform the following steps:

➡ Click **Connect** to go online. The ***samos***®PLAN connects to the ***samos***®PRO system.
➡ Click **Transfer** to transfer the configuration to the ***samos***®PRO system.

NOTE

Depending on your current user level, you will be prompted to log on as authorized client to be able to transfer a configuration. For details please see the ***samos***®PLAN operating instructions.

➡ Once the transfer has been completed, you will be asked whether you want to run the CPU module. Depending on your choice, click **Yes** or **No** to leave the dialog.

NOTE

You can also start and stop the application in the **Hardware configuration** view using the **Run application** or **Stop application** buttons while the project is online.

More information can be found in the ***samos***®PLAN operating instructions (Wieland part no. BA000518).

4.3.4 Verification of a configuration

After the configuration has been transferred successfully, the ***samos***®PRO system can be verified. To this purpose, the downloaded configuration data are read back out from the ***samos***®PRO system and compared with the project data. If they match, the data are displayed in a report. If the user confirms that they are correct, the system is considered to be verified.

➡ In the **Hardware configuration** view, click on the **Upload and Verify configuration** button. A report of the current configuration will be generated.
➡ Click Yes below at the question **Mark device as verified?** if the displayed configuration is the expected configuration. The system is then considered to be verified.

NOTES

- You have to be logged in as authorized user in order to mark the configuration as “verified”.
- If the verification is completed successfully, a “Read in and compare” report that provides the most important project information is created subsequently. You can print out or store this report.
- The status verified/not verified is indicated in the lower right-hand corner of the ***samos***®PLAN and by the CV LED at the ***samos***®PRO controller module lighting up.
- Only if the device and the corresponding configuration have been marked as verified, the “Auto Start mode” is active in the configuration of the controller module. If the configuration is not set to verified after power up, the system stays in Idle mode (CV LED on the SP-SCON module flashing) and the system needs to be set to Run mode using the ***samos***®PLAN.
- If differences between the project data and the read-back configuration data are detected, a corresponding message including information about possible actions is displayed. Verification of the configuration is not possible then. Observe the information in the error message for the further procedure. Terminate the dialog box by clicking **Close**.
- If you change a verified configuration, the status is reset to “not verified”.
  Exception: If you make only non safety-related changes such as modifying the gateway name, the gateway’s IP address or the port number for a TCP/IP socket connection, the configuration status remains “verified”.

More information can be found in the ***samos***®PLAN operating instructions (Wieland part no. BA000518).

4.3.5 Upload of a configuration

When in online mode, you can upload a configuration from the connected ***samos***®PRO system:

➡ Click on **Upload**. The current configuration of the ***samos***®PRO system will be loaded into the ***samos***®PLAN and can be edited after going offline.

5 Ethernet gateways

This chapter describes the following ***samos***®PRO gateways:

- EtherNet/IP gateway (SP-EN-IP)
- Modbus TCP gateway (SP-EN-MOD)
- Profinet IO gateway (SP-EN-PN)

5.1 Common features

5.1.1 TCP/IP configuration interface

The ***samos***®PRO Ethernet gateways offer a TCP/IP configuration interface which allows the configuration of the ***samos***®PRO System over Ethernet TCP/IP. This runs parallel to the Ethernet TCP/IP or other Ethernet protocols.

WARNING

Do not connect to the *samos*®PRO system via the RS-232 and the Ethernet interface at the same time!
The ***samos***®PRO system can only communicate with one instance of the ***samos***®PLAN at one time. Connecting to the ***samos***®PRO system using multiple instances of samos®PLAN, either on a single PC or multiple PCs, may result in inconsistencies of the configuration and the diagnostics as well as in operational errors. This applies to both RS-232 and Ethernet connections equally.

WARNING

Consider the signal delay for remote TCP/IP connections!
Remote TCP/IP connections to the gateway may be unstable if the signal delay is too great.

➡ Check the signal delay to the gateway using the ping command. Signal delays > 300 ms may cause the connection to break down.

Possible solutions:

➡ Make sure that the connection is fast enough or make another route if this is possible.

Or:

➡ Use a software tool like *Teamviewer* or *PC anywhere* to control a local computer on which ***samos***®PLAN software is installed and connected with the ***samos***®PRO system.

Or:

➡ Contact Wieland Electric support.

In order to configure a gateway for TCP/IP configuration for the first time, perform the following steps:

Step 1: Assign an IP address

➡ Connect a PC or notebook to the RS-232 interface of the SP-SCON.
➡ Power on the ***samos***®PRO System.
➡ Open the ***samos***®PLAN configuration tool installed on the PC and load the hardware configuration including the gateway.
➡ If your project is online, click on the **Disconnect** button to go offline.
➡ Click on the **Gateway** button above the main window and select the desired gateway.

➡ Click on **Gateway configuration** on the left hand menu. The following dialog appears:

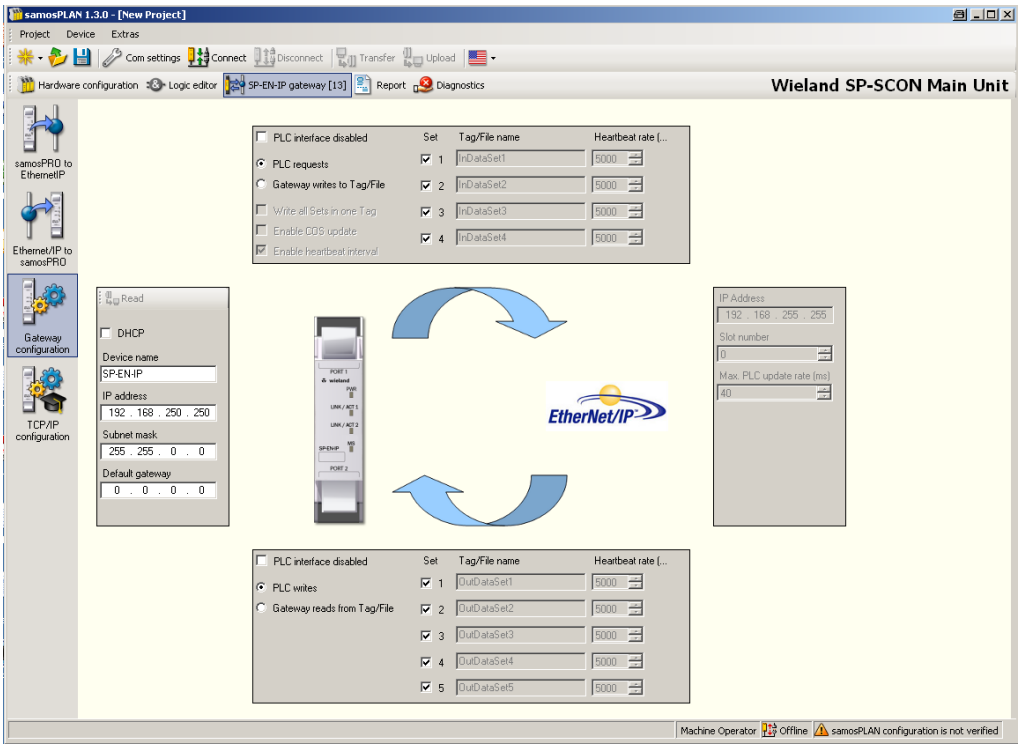


Fig. 9: Gateway configuration dialog

On the left side of the dialog you will find the area for the gateway IP configuration.

➡ If desired, enter a **Device name** for the ***samos***®PRO gateway.
➡ Enter a valid **IP address**, for the ***samos***®PRO gateway, and if required a valid **Subnet mask** and a valid IP address for a **Default gateway**.

Or:

➡ If your network uses a DHCP server, activate the **DHCP** checkbox.
➡ Click **Connect** to go online and transfer the configuration to the ***samos***®PRO system.

NOTES:

- If your project is online, you can use the **Read IP address** button at the upper left corner of the gateway IP configuration area to retrieve the current IP settings of the gateway.
- The out-of-the-box default IP address of the gateway is 192.168.250.250. You can find the default IP address also on the type label of the gateway.

Step 2: Add a TCP/IP profile to your project

➡ Connect one of the two Ethernet ports of the gateway with your Ethernet network using a shielded Ethernet cable.
➡ Connect a PC (or notebook) to the same Ethernet network. Ensure the IP address settings of the PC match the network setup.

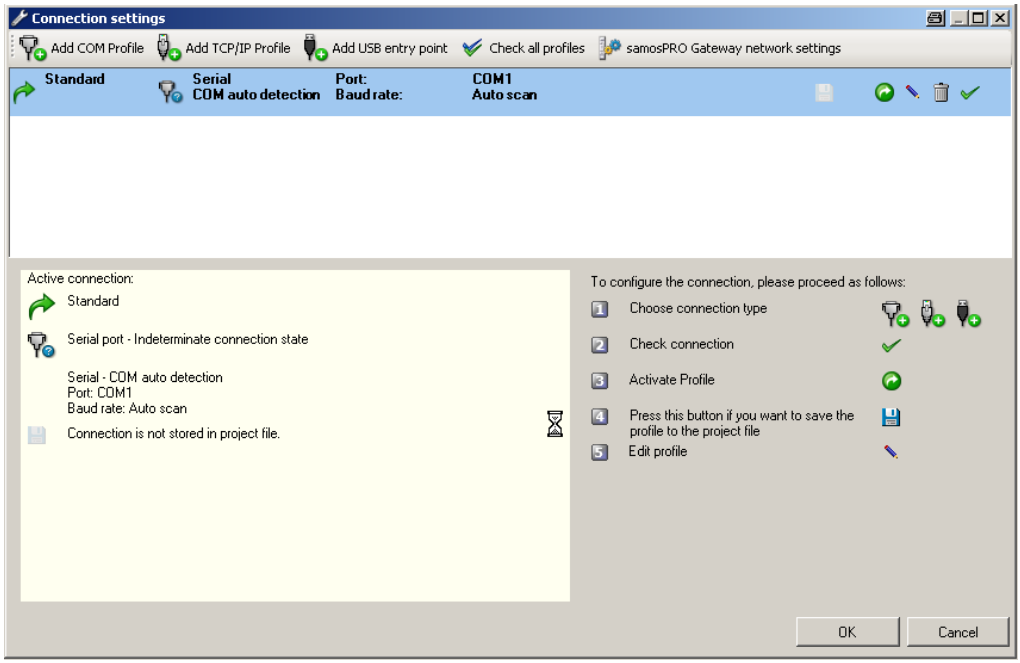
NOTE

You can also connect your PC directly to one of the two Ethernet ports of the gateway. In this case, you can either adapt the IP address settings of your PC or the IP address settings of the gateway to match the other device's IP setup.

➡ Open the ***samos***®PLAN configuration tool installed on the PC and load the hardware configuration including the gateway.
➡ If your project is online, click on the **Disconnect** button to go offline.

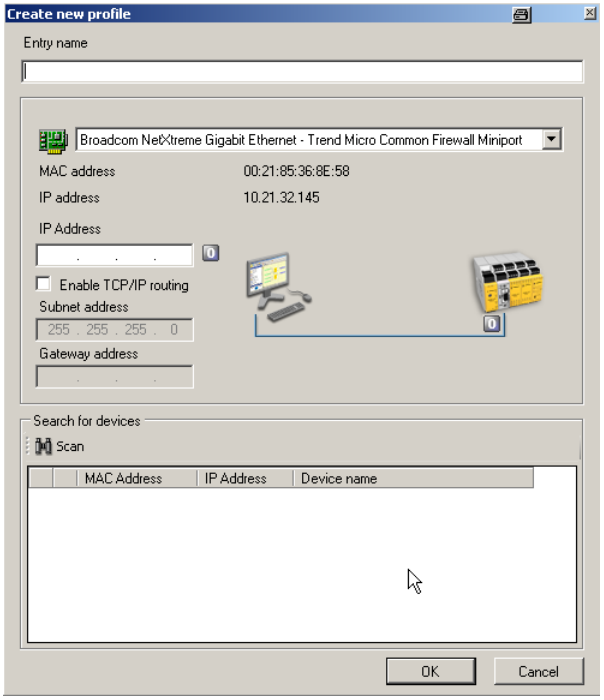
➡ Click on **Com Settings**. The following dialog appears:

Fig. 10: Connection settings dialog



➡ Click on **Add TCP/IP Profile**. The following dialog appears:

Fig. 11: Create new TCP/IP profile dialog



➡ Select your Ethernet network adaptor from the drop-down list.

➡ Click on **Scan** to search for samos®PRO gateways on your Ethernet network. Gateways located will be displayed as shown in the dialog below. The IP address will be displayed as well as MAC address and device name.

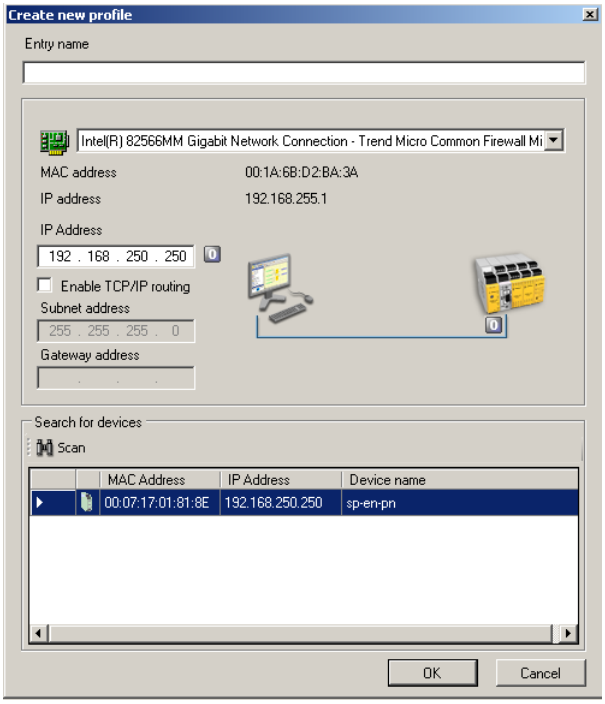


Fig. 12: Create new TCP/IP profile dialog after scan has been performed

- Select the gateway that you want to use for the new profile.
- Enter a name for the profile to the **Entry name** edit field.
- Click **OK**. The entry point has now been created and is shown in the connection dialog:

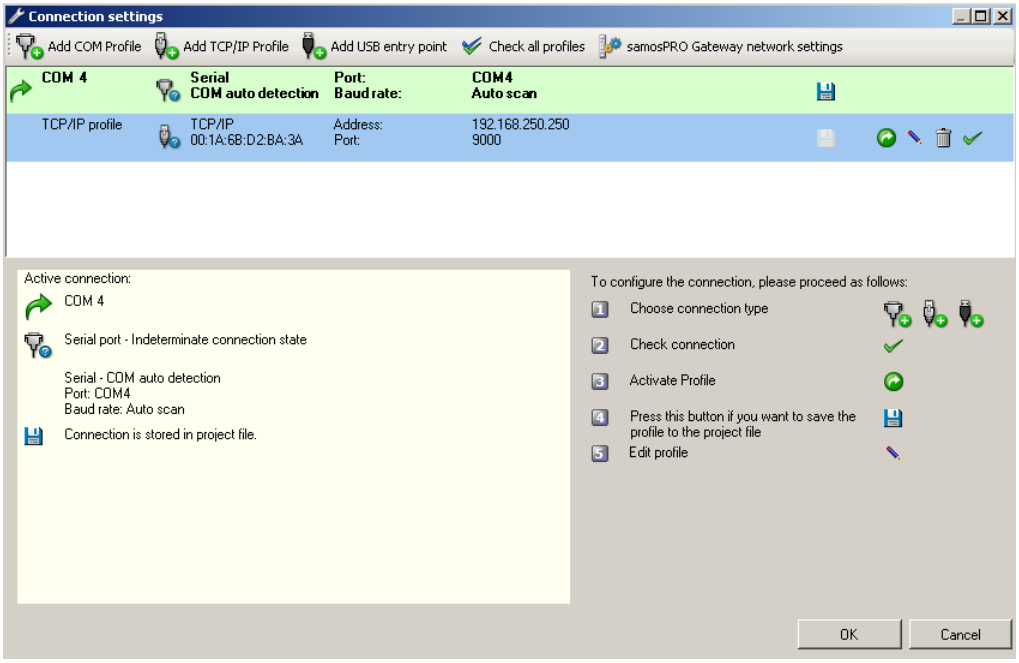
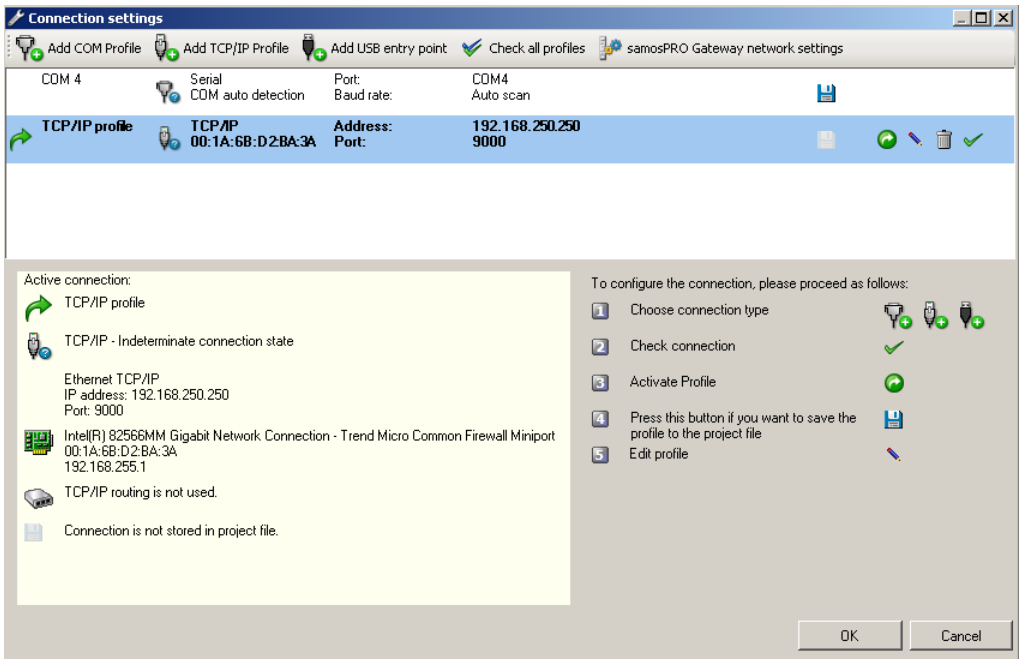


Fig. 13: Connection settings dialog with new TCP/IP profile

In order to use this entry point, it needs to be activated.

➡ Click on the **Activate Entry Point** icon (white arrow in green circle) on the far right. The entry point will then be activated and marked as such:

Fig. 14: Connection settings dialog with new TCP/IP profile activated



➡ Click **OK**. All communication to the ***samos***®PRO system will now happen via TCP/IP. In order to use the entry point via the serial interface again, you will have to re-activate it.

NOTE

The port number for the TCP/IP configuration interface is pre-set to port 9000 and can not be changed.

Step 3: Connect via TCP/IP

➡ Click on the **Connect** button to go online.

5.1.2 Ethernet TCP/IP socket interface

Each ***samos***®PRO Ethernet gateway supports a total number of four TCP/IP socket interfaces. This allows up to four different applications to communicate with the gateway at the same time over Ethernet TCP/IP. The gateway's proprietary network interface (e.g. Modbus TCP) runs in parallel and its configuration or usage does not interact with the TCP/IP socket configuration as it happens independently on separate ***samos***®PLAN pages.

WARNING

Do not use the same output data set number for two different PLC connections or TCP/IP sockets!
The output data set can be written to the Ethernet gateways in parallel by all communication interfaces or TCP/IP sockets (e.g. Modbus TCP and Ethernet TCP/IP), if they use the same output data set number. In that case the last message overrides data received earlier.

The gateway processes the data of a ***samos***®PRO system and makes it available in different compilations, the *data sets*. These data sets are available over the TCP/IP interface. For a detailed description of the data sets please refer to section 3.2 "Data transmitted into the network (network input data sets)" on page 13.

In order to configure the Ethernet TCP/IP socket interface, perform the following steps:

➡ Open the ***samos***®PLAN and load the hardware configuration including the gateway.
➡ Click on the **Gateway** button above the main window and select the respective gateway to open the gateway configuration dialog.

➧ Click on **TCP/IP configuration** on the left hand menu. The following dialog appears:

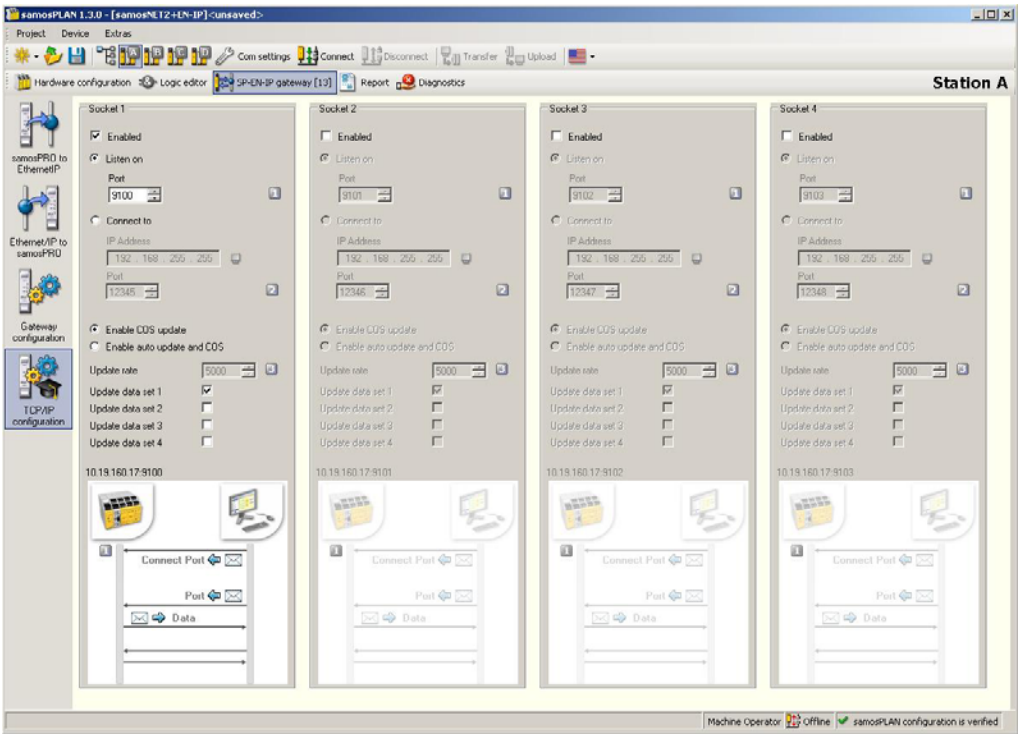


Fig. 15: TCP/IP configuration dialog

Configuration of the TCP/IP interface — who establishes the connection

If the ***samos***®PRO gateway shall connect to the external application, perform the following configuration steps:

➧ Activate the **Connect to** radio button.
➧ Set **IP Address** to the IP address of the computer the application is running on.
➧ Enter the port number of the application for **Port**.

NOTE

The configuration is considered faulty if either the connect socket port and/or the connect IP address is zero when in **Connect** mode.

If the external application shall connect to the ***samos***®PRO gateway, perform the following configuration steps:

➧ Activate the **Listen on** radio button.
➧ Enter the **Port** number for the application.

NOTES

- Suggested port numbers are 9100 to 9103 (default values).
- Port 0 and port 9000 are reserved and can not be used (faulty configuration).
- Port numbers 0 to 1023 are managed by the Internet Assigned Numbers Authority (IANA) and can not be used. See http://www.iana.org/assignments/port-numbers.

Finally, determine how the data is transferred. Follow the steps outlined in the following section.

Data transfer method — how the data is transferred

Whenever the TCP/IP socket connection has been established (either by an application on a PC or by the gateway itself), there are two possible methods how the data sets can be transferred:

- The application requests the data set(s) per command message (Application requests (Polling) mode), **or**
- the gateway auto-updates the data sets as per configuration (Gateway writes to Address/Port (Auto update) mode).

For both methods there are two update modes how the gateways update the data:

- *Change of state (COS)*: when any data of the input data set change status.
- *Automatic update*: data will be sent according to the configured update rate in ms.

NOTE If automatic update is enabled, a change of state will trigger an immediate update of the data as well, regardless of the set update interval. I.e. COS is always active.

For both methods the following structure of messages applies.

General telegram structure

The request/response message (e.g. telegram) is structured as shown below:

0	1	...	...	...	...	...	...	...	...	...	...	...	...	n
Command	Parameter(s) (content depends on type of command)											Data		

Tab. 7: Telegram structure

Parameter	Length	Description
Command	WORD	0hex = Undefined (no command) Polling mode specific 00F1hex = Input data set(s) request message 001Fhex = Input data set(s) response message Auto-update specific 00E1hex = Auto update control 001Ehex = Auto update control response 002Ehex = Auto update input data set(s) message Digital outputs read/write 00F2hex = Write output data set settings 002Fhex = Response to write output data set settings
Parameter(s)	Length determined by command	As defined in specific command
Data	Length determined by command	As defined in specific command

Error response to invalid messages

The gateway will set the most significant bit of the command word in the event that an invalid or improperly formatted message is received.

Tab. 8: Error response message

Parameter	Length	Description
Command	WORD	Bit 15 of received command will be set (i.e. command of 00F2hex would become 80F2hex)
Following data	Length determined by command	Unchanged. Returned as it was received

Application requests (Polling) mode

In this mode the gateway will only send any data upon request (e.g. polling). Therefore the application shall send request telegrams as per definition below and the gateway will respond with telegrams structured as per definition below.

Get input data set(s)

The request message is sent by an application to the gateway. The request message telegram shall be structured as shown below:

Parameter	Length	Value
Command	WORD	00F1hex = Data set(s) request message
Request data set 1	WORD	0 = Do not send data set 1 1 = Send data set 1
Request data set 2	WORD	0 = Do not send data set 2 1 = Send data set 2
Request data set 3	WORD	0 = Do not send data set 3 1 = Send data set 3
Request data set 4	WORD	0 = Do not send data set 4 1 = Send data set 4

Tab. 9: Get data set(s) request

The response message is returned to the application by the gateway. The response message telegram will be structured as shown below:

Parameter	Length	Value
Command	WORD	00F1hex = Data set(s) response message
Data set 1 length	WORD	0 = Data set not returned in data set(s) data field Non-zero = Length of data set
Data set 2 length	WORD	0 = Data set not returned in data set(s) data field Non-zero = Length of data set
Data set 3 length	WORD	0 = Data set not returned in data set(s) data field Non-zero = Length of data set
Data set 4 length	WORD	0 = Data set not returned in data set(s) data field Non-zero = Length of data set
Data set(s) data	Array of bytes	Data set(s) information

Tab. 10: Get data set(s) response

Write output data sets

The following command message is sent by the application to the gateway to write to the output data sets:

Parameter	Length	Value
Command	WORD	00F2hex = Set output data set(s) command message
Output data set 1 length	WORD	0 = Output data set not included in data set(s) data field Non-zero = Length of data set
Output data set 2 length	WORD	0 = Output data set not included in data set(s) data field Non-zero = Length of data set
Output data set 3 length	WORD	0 = Output data set not included in data set(s) data field Non-zero = Length of data set
Output data set 4 length	WORD	0 = Output data set not included in data set(s) data field Non-zero = Length of data set
Output data set 5 length	WORD	0 = Output data set not included in data set(s) data field Non-zero = Length of data set
Data set(s) data	Array of bytes	Data set(s) information

Tab. 11: Write output data set setting command

The response message is returned to the application by the gateway. The response message telegram is structured as shown below:

Tab. 12: Write output data set setting response

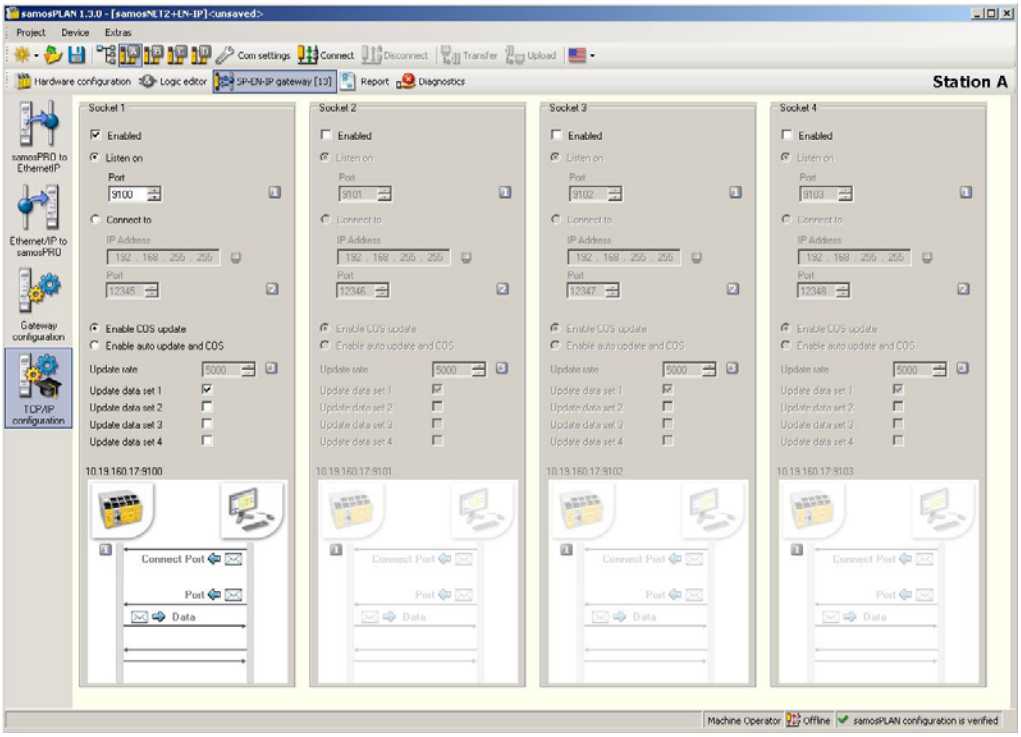
Parameter	Length	Value
Command	WORD	002Fhex = Response to write output data set settings message
Status	WORD	0 = Success. Output data sets written correctly 1 = Error — Can not write output data sets due to either: Loss of backplane communication Incorrect routing information

*Configuration via **samos**®PLAN*

In order to configure the **Application requests (Polling) mode** of the gateway via the ***samos***®PLAN tool, perform the following steps:

- Open the ***samos***®PLAN and load the hardware configuration including the gateway.
- Click on the **Gateway** button above the main window and select the respective gateway to open the gateway configuration dialog.
- Click on **TCP/IP configuration** on the left hand menu. The following dialog appears:

Fig. 16: TCP/IP configuration for Application requests (Polling) mode



- Check the **Listen on** checkbox.
- Enter the **Port** number on which the application will connect.
- Select the update mode: **Enable COS update** or **Enable auto update and COS**.
- If you have selected **Enable auto update and COS**, select the **Update rate** in ms.
- Select which data sets shall be updated: Check the **Update Dataset n** checkbox.

Gateway writes to Address/Port (Auto update) mode

The gateway can be configured to automatically update the data set information (i.e. the application does not need to send any request messages as it would do in polling mode) once the connection to the application has been made.

The configuration settings are available via the ***samos***®PLAN configuration tool or via the TCP/IP interface itself. Using one interface does not disable the other: The auto update mode could be enabled via ***samos***®PLAN and disabled via TCP/IP command, for example.

Configuration via TCP/IP interface

This command message is sent by an application to the gateway to configure the auto update mode. This message can be used to either disable or enable the auto update mode directly through the TCP/IP interface.

Parameter	Length	Value
Command	WORD	00E1hex = Auto update control
Request data set 1	WORD	0 = Do not send data set 1 1 = Send data set 1
Request data set 2	WORD	0 = Do not send data set 2 1 = Send data set 2
Request data set 3	WORD	0 = Do not send data set 3 1 = Send data set 3
Request data set 4	WORD	0 = Do not send data set 4 1 = Send data set 4
Heartbeat mode update rate	WORD	0 = Disable heartbeat messages Non-zero = Enable heartbeat message at specified rate in ms. Minimum = 40 ms

Tab. 13: Auto update mode configuration command

NOTE

Auto update is disabled if all Request Input Data Set flags are set to zero.

The response message returned to the application by the gateway:

Parameter	Length	Value
Command	WORD	001Ehex = Response to the auto update control message

Tab. 14: Auto update mode configuration response

*Configuration via **samos**®PLAN*

In order to configure the **Gateway writes to Address/Port (Auto update)** mode of the gateway via the ***samos***®PLAN tool, perform the following steps:

- ➧ Open the ***samos***®PLAN and load the hardware configuration including the gateway.
- ➧ Click on the **Gateway** button above the main window and select the respective gateway to open the gateway configuration dialog.
- ➧ Click on **TCP/IP configuration** on the left hand menu. The following dialog appears:

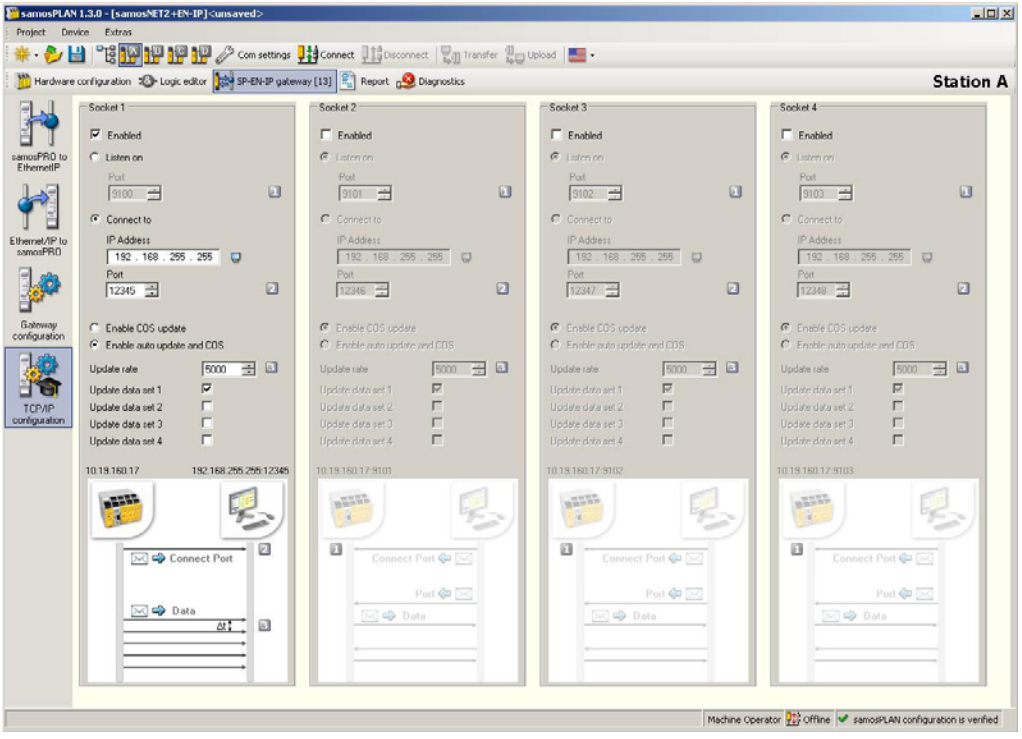


Fig. 17: TCP/IP configuration for auto update

- ➡ Check the **Connect to** checkbox.
- ➡ Enter the **IP Address** and the **Port** number the gateway shall write to.
- ➡ Select the update mode: **Enable COS update** or **Enable auto update and COS**.
- ➡ If you have selected **Enable auto update and COS**, select the **Update rate** in ms.
- ➡ Select which data sets shall be updated: Check the **Update Dataset n** checkbox.

Normal operation

The following message is sent from the gateway to the application while operating in auto update mode.

Tab. 15: Auto update mode normal operation message

Parameter	Length	Value
Command	WORD	002Ehex = Auto update data set(s) message
Data set 1 length	WORD	0 = Data set not returned in data set(s) data field Non-zero = Length of data set
Data set 2 length	WORD	0 = Data set not returned in data set(s) data field Non-zero = Length of data set
Data set 3 length	WORD	0 = Data set not returned in data set(s) data field Non-zero = Length of data set
Data set 4 length	WORD	0 = Data set not returned in data set(s) data field Non-zero = Length of data set
Data set(s) data	Array of bytes (length dependent on set definition)	Data set(s) information. Details see section 3.2 “Data transmitted into the network (network input data sets)” on page 13 and chapter 7 “Layout and content of the process image” on page 109.

5.1.3 TCP/IP process image example

The following example shows a possible process image sent by a SP-EN-IP gateway via TCP/IP in auto update mode:

Tab. 16: TCP/IP process image example

Byte values (hex)	Part of message	Meaning
00 2E	Command	Auto update data sets (see Tab. 15)
00 32	Command parameters	Length of data set 1: 50 bytes
00 20		Length of data set 2: 32 bytes
00 3C		Length of data set 3: 60 bytes
00 3C		Length of data set 4: 60 bytes
03 FF 03 03	Data set 1 (default byte assignments, see Tab. 4)	Logic results 1-4
C0		Input values module 1: C0 = 11000000 = Inputs I8 and I7 **Active**
03		Input values module 2: 03 = 00000011 = Inputs I2 and I1 **Active**
3F 05 05 05 00 00 00 00 00 00		Input values module 3-12
00 00 00 00 00 00 00 00 00 00 00 00		Output values module 1-12
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		Not assigned
52 A1 10 4C	Data set 2 (see Tab. 4)	Overall CRC (same as system CRC)
52 A1 10 4C		System CRC
00 00 00 00		Reserved
00 00 00 00		
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		
FF FF FF FF	Data set 3 (see Tab. 4 and Tab. 5) Data set 3 data is transferred in Big Endian format, i.e. in 32 bit double word format with the most significant byte placed in the leftmost position.	Status module 0 (SP-SCON): OK
FF FF FF FF		Status module 1 (e.g. SP-SDI): OK
FD FB FF FF		Status module 2 (e.g. SP-SDIO): Byte 0: FF = 11111111: No errors Byte 1: FF = 11111111: No errors Byte 2: FB = 11111011: Input 3 external test signal failure Byte 3: FD = 11111101: Output 1 stuck-at-low error
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF		Status modules 3-6: OK
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF		Status modules 7-12 (no modules present)
FF FF FF FF		Status module 13 (e.g. SP-EN-IP): OK
FF FF FF FF		Status module 14 (no module present)
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	Data set 4	Reserved

5.2 EtherNet/IP gateway

The following ***samos***®PRO gateway can be used for EtherNet/IP: SP-EN-IP.

You will find the EDS file and device icon for PLC interfacing

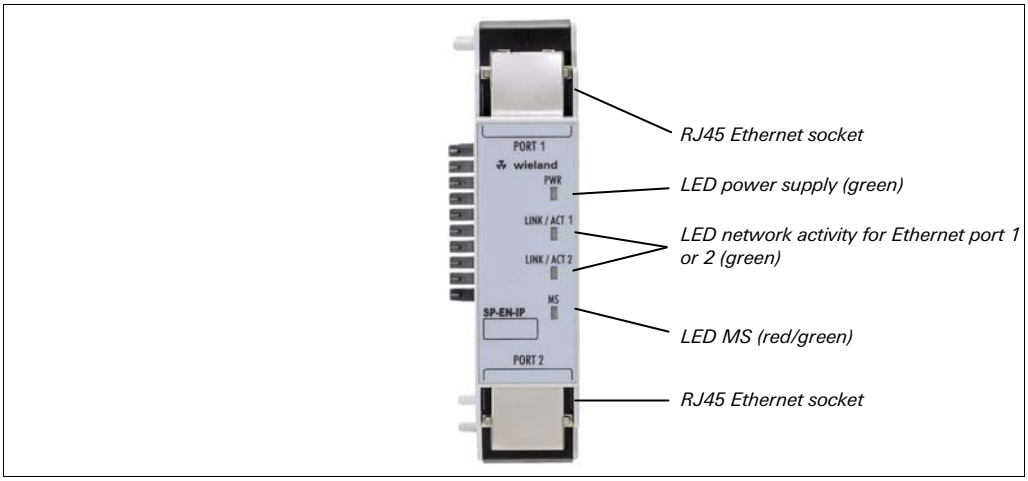
- in the Internet on the SP-EN-IP product page (http://eshop.wieland-electric.com).

The ***samos***®PRO EtherNet/IP gateway SP-EN-IP supports only EtherNet/IP explicit messaging. Class 1 messaging is not supported.

5.2.1 Interfaces and operation

The SP-EN-IP is equipped with an integrated three-port switch for connection with the Ethernet network. Two RJ45 sockets are available for the connection. The switch functionality allows the SP-EN-IP to be used for connection to another Ethernet component (e.g. connection to a notebook) without having to interrupt the Ethernet connection to the network.

Fig. 18: Interfaces and display elements of the SP-EN-IP



Tab. 17: Meaning of the LED displays of the SP-EN-IP

LED		Meaning
Symbol description: ○ – LED off, ● Green – LED lights up green, ✸ Red – LED flashes red		
PWR	○	No power supply
	● Green	Power supply switched on
LINK/ACT 1 LINK/ACT 2	○	No Ethernet connection
	● Green	Ethernet connection active, no data transmission
	✸ Green	Ethernet connection active, data transmission
MS	○	Power-up
	● Green	Executing (live process data from/to CPU)
	✸ Green	1 Hz: Idle
	✸ Red	1 Hz: Configuring/configuration required 2 Hz: Critical fault on gateway
	● Red	Critical fault on another module
	✸ Red/Green	Executing, but Ethernet communication not established or faulty

NOTE Error elimination is described in section 5.2.7 “Diagnostics and troubleshooting” on page 51.

Power-up sequence

On power up, the following LED test sequence is performed:

- LED MS ○ **Off** for 6 s.
- LED MS ● **Red** for 0.25 s.
- LED MS ● **Green** for 0.25 s.
- LED MS ○ **Off**.

5.2.2 Basic configuration — assigning a device name and IP address

Configuration of the SP-EN-IP is performed via the ***samos***®PLAN tool.

Via *samos*®PLAN tool

➡ Open the ***samos***®PLAN and load the hardware configuration including the EtherNet/IP gateway.
➡ Click on the **Gateway** button above the main window and select the SP-EN-IP to open the gateway configuration dialog.
➡ Click on **Gateway configuration** on the left hand menu. The following dialog appears:

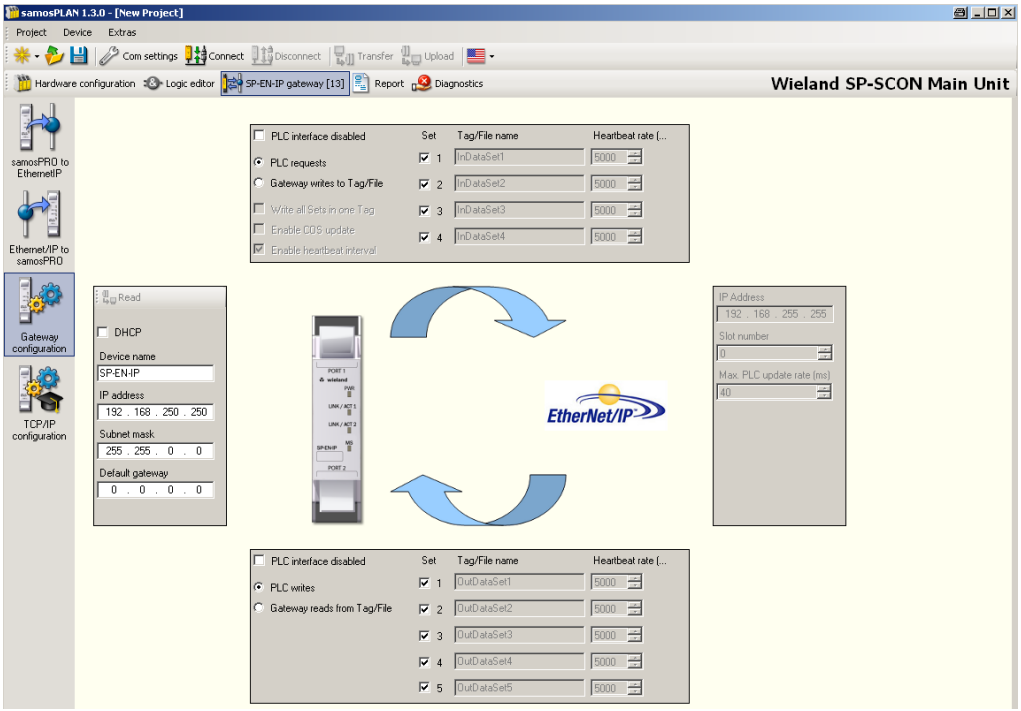


Fig. 19: Gateway configuration dialog

➡ If desired, change the **Device name** for the ***samos***®PRO gateway.
➡ Enter a valid **IP address** for the ***samos***®PRO gateway, and if required a valid **Subnet mask** and a valid IP address for a **Default gateway**.
➡ Click **Connect** to go online and download the configuration to the ***samos***®PRO system.

5.2.3 Configuration of the interface to the PLC — how the data are transferred

Transfer methods

The configuration steps in this section specify how the data to the higher-level PLC are transferred. In general, there are two different transfer *methods* available for both transfer *directions* such as ***samos***®PRO to Network and Network to ***samos***®PRO:

- **Gateway writes to Tag/File** and/or **Gateway reads from Tag/File** — The SP-EN-IP gateway operates as *master*. It writes the data into and/or reads from the PLC memory.
- **PLC requests** and/or **PLC writes** — The SP-EN-IP gateway operates as *slave*. The PLC requests the data from the gateway and/or writes the data to the gateway.

Both methods can be mixed. E.g. it is possible to configure the gateway as master for the ***samos***®PRO to Network direction (option **Gateway writes to Tag/File** activated) while it

operates at the same time as slave for the Network to ***samos***®PRO direction (option **PLC writes** activated).

Number of possible connections

The number of possible connections to the PLC depends on whether the SP-EN-IP is operated as *master* or *slave*. Depending on the setting, up to 128 PLCs can address the SP-EN-IP at the same time.

Tab. 18: Number of possible connections

Operating mode of the SP-EN-IP	Maximum connections
Rx (To PLC) Transfer mode: Gateway writes to Tag/File Tx (From PLC) Transfer mode: Gateway reads from Tag/File	Rx and Tx: 1
Rx (To PLC) Transfer mode: Gateway writes to Tag/File Tx (From PLC) Transfer mode: PLC writes	Rx: 1 Tx: 127
Rx (To PLC) Transfer mode: PLC requests Tx (From PLC) Transfer mode: Gateway reads from Tag/File	Rx: 127 Tx: 1
Rx (To PLC) Transfer mode: PLC requests Tx (From PLC) Transfer mode: PLC writes	Rx and Tx: 128

Configuration process

The following table outlines the configuration process depending on the transfer method:

Tab. 19: Configuration guideline — gateway as master

Gateway is master (Gateway writes to Tag/File and/or Gateway reads from Tag/File)	
To do in the gateway configuration (via ***samos***®PLAN tool)	To do in the PLC program and/or EtherNet/IP network configuration tool
Select which data shall be written to/read from the PLC	—
Define where in the PLC memory the selected data shall be written to: Enter tag names. Example: InDataSet1 And/or define where in the PLC memory the selected data shall be read from: Enter tag names. Example: OutDataSet1	Define exactly the same tag names in the PLC program. Example: InDataSet1 INT[25] OutDataSet1 INT[5] The data type shall be INT.
Select how often this data shall be transmitted.	—
Define where the data shall be read from/written to in the EtherNet/IP network: Enter the IP address and controller slot number of the PLC.	—

Tab. 20: Configuration guideline — gateway as slave

Gateway is slave (PLC requests and/or PLC writes)	
To do in the gateway configuration (via ***samos***®PLAN tool)	To do in the PLC program and/or EtherNet/IP network configuration tool
—	Download and install the SP-EN-IP EDS file from the SP-EN-IP product page (http://eshop.wieland-electric.com)
—	Integrate the SP-EN-IP into the EtherNet/IP network via network configuration tool (i.e. RSNetworx).
—	Program the explicit message “Get_Attribute_...” or “Set_Attribute_...” in the PLC program to read/write data from/to the gateway
—	Program the trigger for sending the explicit messages.

Method 1: Gateway writes to/reads from Tag/File — SP-EN-IP gateway writes the data into/reads the data from the PLC memory

In this operating mode the SP-EN-IP as a *master* writes the data of all activated data sets into the specified memory areas of the PLC. The only task for the PLC programmer is to de-fine a controller tag name which matches the gateway configuration tag name.

In order to configure the gateway to be *master*, perform the following steps:

- ➧ Open the ***samos***®PLAN and load the hardware configuration including the EtherNet/IP gateway.
- ➧ Click on the **Gateway** button above the main window and select the SP-EN-IP to open the gateway configuration dialog.

➡ Click on **Gateway configuration** on the left hand menu. The following dialog appears:

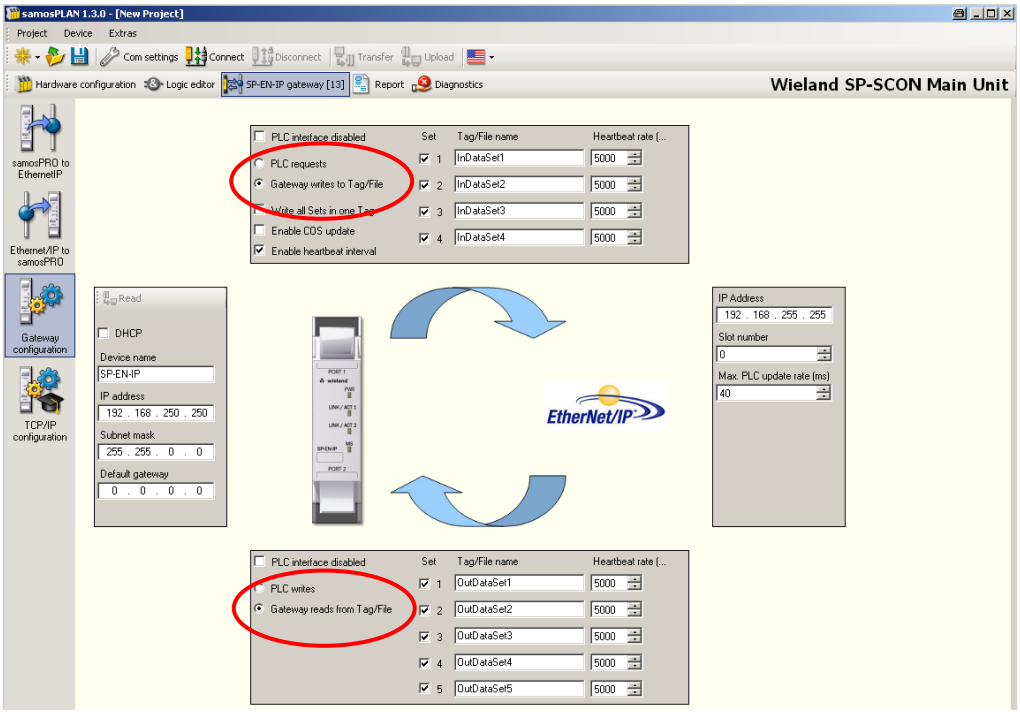


Fig. 20: Gateway configuration dialog

➡ Within the **Gateway configuration** dialog, select the transfer method by activating **Gateway writes to Tag/File** for the ***samos***®PRO to Network direction and **Gateway reads from Tag/File** for the Network to ***samos***®PRO direction.

➡ Select which data shall be written to/read from the PLC by checking the checkbox for the required data set.

➡ Define where in the PLC memory the selected data shall be written to or read from: Enter tag names into the **Tag/File name** edit fields (max. 20 characters).

➡ Select **Write all Sets in one Tag** if all data sets shall be written into one tag in the PLC memory. In this case, the tag defined for data set 1 will be used.

➡ Define how often the data shall be transmitted to the PLC:
- Select **Enable Update COS** (update on change of state) if the SP-EN-IP is to update the data in the PLC immediately when changes occur in the data sets.
- Select **Enable heartbeat interval** to activate updating of the selected data sets with the set **Heartbeat rate** in ms.
- Both options may be selected at the same time.

➡ Define how often the data shall be read from the PLC:
- Enter a **Heartbeat rate** in ms to activate updating of the selected data sets with the set time.

➡ Define where the data shall be read to/written from in the EtherNet/IP network: Enter the **IP address** and controller **Slot number** of the PLC.

NOTE

The configuration is considered faulty, if the PLC IP address is zero and either **Gateway writes to Tag/File** for the ***samos***®PRO to Network direction and/or **Gateway reads from Tag/File** for the Network to ***samos***®PRO direction is activated.

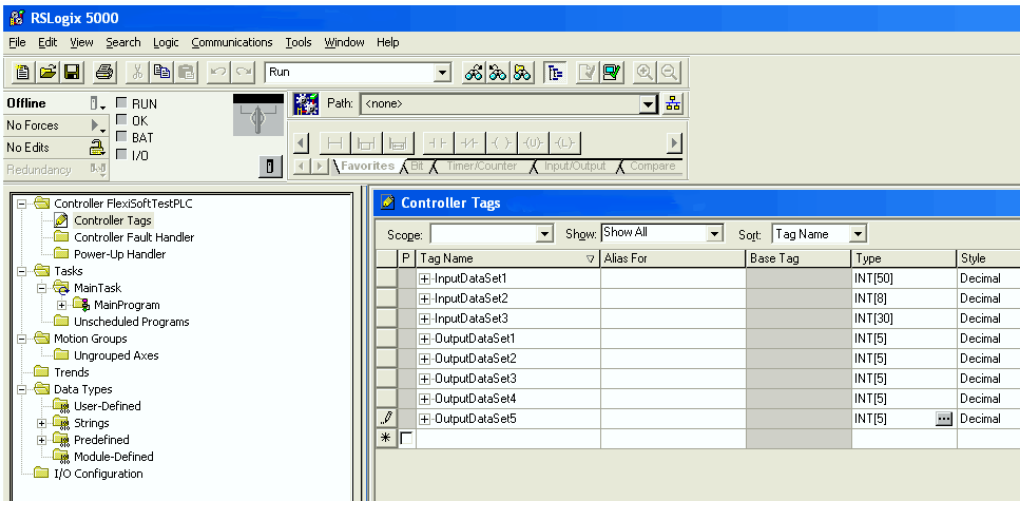
➡ The **Max. PLC update rate (ms)** defines the maximum rate (the minimum time interval) for transferring the data sets to and from the PLC. Settings occur dependent on the PLC processing speed. Minimum = 10 ms, maximum = 65535 ms. The default value of 40 ms is suitable for most PLCs.

NOTES

- If the value entered for the **Max. PLC update rate** is greater than the **Heartbeat rate** set for writing to or reading from the PLC, the heartbeat rate will be automatically increased (i.e. slowed down) to this value.
- All data sets are transferred to the PLC in 16 bit integer format with the first byte placed in the most significant, or leftmost byte of the integer.

➡ Click **Connect** to go online and download the configuration to the ***samos***®PRO system.
➡ Open the PLC programming tool.
➡ Define the PLC tag names as previously configured in the ***samos***®PRO EtherNet/IP gateway. Fig. 21 shows an example for the definition of tag names in a PLC program written with RSLogix:

Fig. 21: Example of tag names in a PLC program



NOTES

- Tag names for Allen Bradley SLC/PLC-5 PLCs must begin with a "$" (i.e. $N10:0).
- Tag names for Allen Bradley MicroLogix PLCs must begin with a "#" (i.e. #N10:0).

Method 2: Polling mode — PLC requests the data from/PLC writes the data to the SP-EN-IP

In this operating mode the SP-EN-IP operates as *slave*. It sends the data to the PLC upon request and the PLC writes the data to the gateway. If this operating mode is desired:

➡ Open the ***samos***®PLAN and load the hardware configuration including the EtherNet/IP gateway.
➡ Click on the **Gateway** button above the main window and select the SP-EN-IP to open the gateway configuration dialog.

➡ Click on **Gateway configuration** on the left hand menu. The following dialog appears:

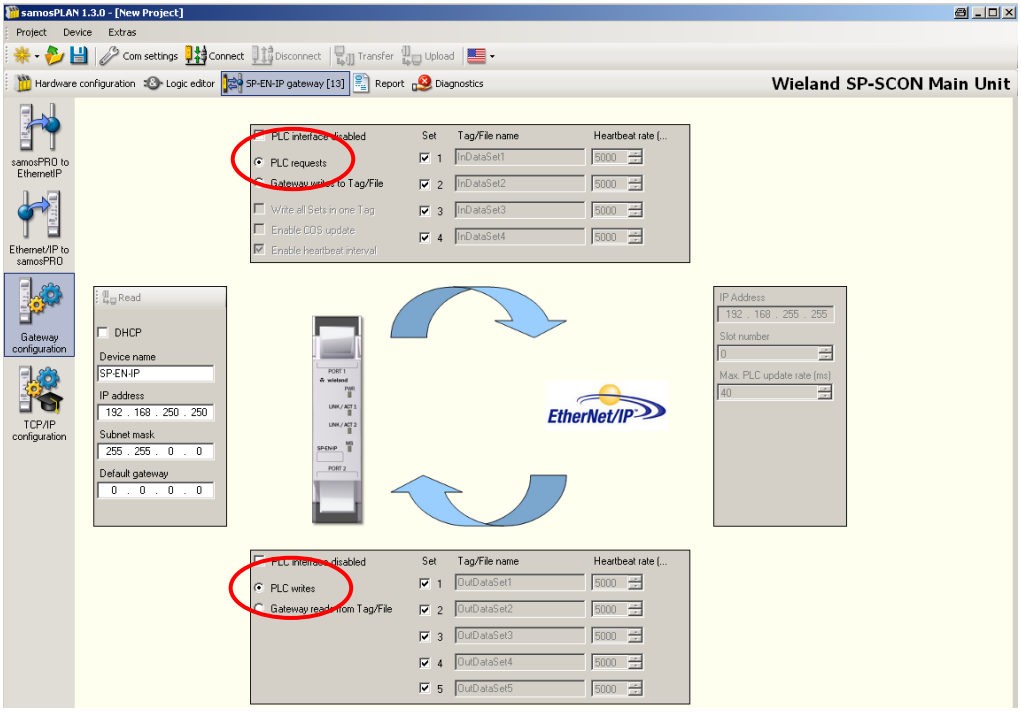


Fig. 22: Gateway configuration dialog

➡ Within the **Gateway configuration** dialog, select the transfer method by activating **PLC requests** for the ***samos***®PRO to Network direction, **PLC writes** for the Network to ***samos***®PRO direction.
➡ Select which data shall be requested or written by the PLC by checking the checkboxes for the required data sets.
➡ Click **Connect** to go online and download the configuration to the ***samos***®PRO system.
➡ Program the explicit messaging in the PLC.

Polling data sets via explicit messaging

The SP-EN-IP supports two vendor specific objects which can be polled via explicit messaging:

- The *Full Data Set Transfer* object allows to poll for each of the data sets. There is one instance where each attribute represents a data set.
- The *Individual Input Data Set Transfer* object allows to poll for the individual data set parameters. There is one instance per data set and each attribute represents one byte of the data set.

Full Data Set Transfer object definition (72 HEX – one instance)

The vendor specific **Full Data Set Transfer** object defines the attributes by which the PLC can:

- Request the complete input data set information from the SP-EN-IP.
- Write the complete output data set information to the SP-EN-IP.

Class attributes

Attribute ID	Name	Data type	Data value(s)	Access rule
1	Revision	UINT	1	Get
2	Max. instance	UINT	1	Get
3	Num. instances	UINT	1	Get

Tab. 21: Full Data Set Transfer object (72hex) class attributes

Instance attributes

These attributes provide access to input and output data sets. *Get Attribute Single* requests for a specific input data set will return the input data set information. *Get Attribute All* requests will return all enabled input data sets.

All data set information will be returned in integer (16 bit word) format. For byte data, the first byte will be placed in the most significant or leftmost byte of the integer and the second byte will be placed in the least significant or rightmost byte of the integer.

Tab. 22: Full Data Set Transfer object (72hex) instance attributes

Attribute ID	Name	Data type	Data value(s)	Access rule
samos®PRO to Network				
1	Request input data set 1 specific data	Array of UINT	0-255	Get
2	Request input data set 2 specific data	Array of UINT	0-255	Get
3	Request input data set 3 specific data	Array of UINT	0-255	Get
4	Request input data set 4 specific data	Array of UINT	0-255	Get
Network to ***samos***®PRO				
5	Write the output data set 1 specific data	Array of UINT	0-255	Set
6	Write the output data set 2 specific data	Array of UINT	0-255	Set
7	Write the output data set 3 specific data	Array of UINT	0-255	Set
8	Write the output data set 4 specific data	Array of UINT	0-255	Set
9	Write the output data set 5 specific data	Array of UINT	0-255	Set

Common services

Tab. 23: Full Data Set Transfer object (72hex) common services

Service code	Implemented in class	Implemented in instance	Service name
01hex	Yes	Yes	Get_Attributes_All
0Ehex	Yes	Yes	Get_Attribute_Single
10hex	No	Yes	Set_Attribute_Single
02hex	No	Yes	Set_Attribute_All

Individual Input Data Set Transfer object definition (73 HEX – one instance per data set)

The vendor specific **Individual Input Data Set Transfer** object defines the attributes by which the PLC can request either full input data sets or individual parameters within an input data set.

Class attributes

Tab. 24: Individual Input Data Set Transfer object (73hex) class attributes

Attribute ID	Name	Data type	Data value(s)	Access rule
1	Revision	UINT	1	Get
2	Max instance	UINT	4	Get
3	Num instances	UINT	4	Get

Instance attributes

Tab. 25: Individual Input Data Set Transfer object (73hex) instance attributes

Attribute ID	Name	Data type	Data value(s)	Access rule
1 to *n* (dependent on data set definition)	Request input data set specific data	SINT	0-255	Get

Common services

Service code	Implemented in class	Implemented in instance	Service name
01hex	Yes	Yes	Get_Attributes_All
0Ehex	Yes	Yes	Get_Attribute_Single

Tab. 26: Individual Input Data Set Transfer object (73hex) common services

Instance attribute definitions

Attribute 1 to *n* — Request input data set specific parameters
These attributes return the input data set specific data arrays. **Get Attribute Single** requests for a specific input data set will return only the requested data set parameter information. **Get Attribute All** requests will return the entire data set.

The data set attributes, numbered from 1 to N, refer to each individual attribute of each individual input data set. Each instance refers to a unique input data set and each input data set has a unique attribute numbering scheme. The following tables reflect the attribute definitions for each input data set.

Get All Data Set Attributes request

All data set information will be returned in integer (16 bit word) format. For byte data, the first byte will be placed in the least significant or rightmost byte of the integer and the second byte will be placed in the most significant or leftmost byte of the integer.

Example:

For an input data set, the data will be returned as follows:

- IntegerArray[0]: AABB (hex) – AA = BYTE1; BB = BYTE2
- IntegerArray[1]: CCDD (hex) – CC = MED1; BB = MED2
- …
- IntegerArray[6]: MMNN (hex) – MM = BYTE13; NN = BYTE14

NOTE

The typical PC tools of Rockwell/Allen Bradley change this data format back to BBAA (hex) format for visualisation purposes. Check your data for plausibility before putting your ***samos***®PRO system into operation.

Instance 1 – Input data set 1 attribute definitions

Attribute number	Data set parameter	Size
1	Byte 0	SINT
2	Byte 1	SINT
…	…	…
50	Byte 49	SINT

Tab. 27: Individual Input Data Set Transfer object (73hex) instance 1 attribute definitions

Instance 2 – Input data set 2 attribute definitions

Attribute number	Data set parameter	Size
1	Overall CRC	UDINT
2	System CRC (SCID)	UDINT
3	Reserved	UDINT
4	Reserved	UDINT
5	Reserved	UDINT
6	Reserved	UDINT
7	Reserved	UDINT
8	Reserved	UDINT

Tab. 28: Individual Input Data Set Transfer object (73hex) instance 2 attribute definitions

Instance 3 - Input data set 3 attribute definitions

Tab. 29: Individual Input Data Set Transfer object (73hex) instance 3 attribute definitions

Attribute number	Data set parameter	Size
1	Module status module 0	UINT[2]
2	Module status module 1	UINT[2]
…	…	…
15	Module status module 14	UINT[2]

Instance 4 - Input data set 4 attribute definitions

Tab. 30: Individual Input Data Set Transfer object (73hex) instance 4 attribute definitions

Attribute number	Data set parameter	Size
1	Reserved	UINT[2]
2	Reserved	UINT[2]
…	…	…
15	Reserved	UINT[2]

PLC-5/SLC/MicroLogix interface

The PLC-5, SLC and MicroLogix PLC interfaces are supported by:

- The same write to PLC functionality as provided to ControlLogix PLCs provided in the Write-to-File receive method.
- PCCC based messages transferred via the PCCC object.
  - SLC Typed Read Message.
  - SLC Typed Write Message.
  - PLC-5 Typed Read Message (Logical ASCII and Logical Binary address format).
  - PLC-5 Typed Write Message (Logical ASCII and Logic binary address format).
- Normal PLC-5/SLC file naming conventions are used.

The primary differences between the PLC-5/SLC/MicroLogix interface and the ControlLogix interfaces are:

- Polling is performed through the SLC and PLC-5 specific messages instead of accessing the Data Transfer object.
- Data is written into files on the PLC, instead of tags as on ControlLogix PLCs.

NOTE While ControlLogix PLCs support the SLC and PLC-5 messages, using those messages on ControlLogix PLCs is not recommended due to data size and performance considerations.

Receive communication methods

- Polling Receive Method
  This method provides a polling method that allows the PLC to request data on a periodic basis.
  In this method, the input data set information is returned in the response to the data request message. The PLC requests data by accessing the corresponding data file address on the SP-EN-IP with either a SLC typed read or PLC-5 typed read message.
  The following restrictions apply to this method:
  - The file location to receive the input data set on the PLC must be of type INTEGER and large enough to contain the input data set table(s).
  - If no data has been received on the SBUS+ for the specified module, all zeros will be returned.
- Unsolicited – Write to File Receive Method
  When it is determined that data received on the ***samos***®PRO gateway's SBUS+ interface is to be sent to the PLC, the data is immediately written to a file location on the PLC.
  The following restrictions apply to this method:
  - The Receive Data Area File Name must have the same name as the file defined on the PLC. For SLC and PLC-5 PLCs, all file names must be configured with a preceding "$" (i.e $N10:0). For MicroLogix PLCs, all file names must be configured with a preceding "#" (i.e # N10:0).
  - The file on the PLC must be of type INTEGER and must be large enough to contain the input data set table(s).

– Data will be written with the first byte placed in the MS byte location of the integer.
Example: aabb, ccdd, eeff, etc. where aa = byte 1, bb = byte 2, cc = byte 3, etc.

Transmit (From PLC) Data Transfer Methods

The SP-EN-IP will support the following methods of receiving or retrieving the output data set(s) from the PLC.

- PLC Writes Method
  This is the standard method where the PLC uses a message instruction to write the output data sets to the SP-EN-IP. With this method, the output data sets can be updated via a PCCC message written to the corresponding file/address location on the SP-EN-IP.
- Read-from-File Transmit Method (Poll the PLC)
  With this method, the SP-EN-IP will monitor the configured PLC memory location for changes to the output data set(s). When a change is detected, the output data sets will be processed accordingly.
  The following restrictions apply to this method:
  – The output data set file locations must be of INTEGER (16 bit word) format and must be of sufficient length to contain the entire output data set.
  – Data in the INTEGER file must be formatted with the first byte placed in the MS byte location.
  Example: aabb, ccdd, eeff, etc. where aa = byte 1, bb = byte 2, cc = byte 3, etc.

PLC-5 and SLC Messages

The following PCCC messages are supported for the PLC-5, SLC and MicroLogix PLCs:

Message type	PCCC message	Maximum message size
SLC Typed Read	162	CLX: 242 SINTs (121 INTs) SLC: 206 SINTs (103 INTs)
SLC Typed Write	170	CLX: 220 SINTs (110 INTs) SLC: 206 SINTs (103 INTs)
PLC-5 Typed Read	104	CLX: 234 SINTs (117 INTs) SLC: 252 SINTs (126 INTs)
PLC-5 Typed Write	103	CLX: 226 SINTs (113 INTs) SLC: 226 SINTs (113 INTs)

Tab. 31: Supported PCCC messages for the PLC-5, SLC and MicroLogix PLCs

NOTE Both the PLC-5 and SLC Typed Read message can be used to retrieve all input data sets.

Address	Description	Access rule	Data size (words)
N10:0	All enabled input data sets data	Get	16-101[4)]
N11:0	Request input data set 1 data	Get	25
N12:0	Request input data set 2 data	Get	16
N13:0	Request input data set 3 data	Get	30
N14:0	Request input data set 4 data	Get	30
N20:0	Write all enabled output data sets	Set	5-25[5)]
N21:0	Write output data set 1 data	Set	5
N22:0	Write output data set 2 data	Set	5
N23:0	Write output data set 3 data	Set	5
N24:0	Write output data set 4 data	Set	5
N25:0	Write output data set 5 data	Set	5

Tab. 32: Addressing for the PLC-5/SLC messages

[4)] Will correspond to all enabled input data sets.
[5)] Must correspond to all enabled output data sets. Example: If only output data sets 1 and 2 are enabled, then 10 words, (20 bytes), must be written. If all output data sets are enabled, then 25 words, (50 bytes), must be written.

PLC-5/SLC Receive Data Message

The Receive Input Data Set Message format is as defined for each individual input data set. Please refer to Tab. 4 and Tab. 5 in section 3.2 “Data transmitted into the network (network input data sets)” on page 13 for further details.

PCCC object (67 HEX – 1 instance)

The **PCCC** object provides the ability to encapsulate and then transmit and receive PCCC messages between devices on an EtherNet/IP network. This object is used to communicate to SLC 5/05 and PLC-5 PLCs over EtherNet/IP.

Class attributes

Not supported.

Instance attributes

Not supported.

Instances

Supports instance 1.

Common services

Tab. 33: PCCC object (67hex) common services

Service code	Implemented in class	Implemented in instance	Service name
4Bhex	No	Yes	Execute_PCCC

Message structure for Execute_PCCC

Tab. 34: PCCC object (67hex) request message

Name	Data type	Description
Length	USINT	Length of requestor ID
Vendor	UINT	Vendor number of requestor
Serial number	UDINT	ASA serial number of requestor
CMD	USINT	Command byte
STS	USINT	0
TNSW	UINT	Transport word
FNC	USINT	Function code
PCCC_params	Array of USINT	CMD/FMC specific parameters

Tab. 35: PCCC object (67hex) response message

Name	Data type	Description
Length	USINT	Length of requestor ID
Vendor	UINT	Vendor number of requestor
Serial number	UDINT	ASA serial number of requestor
CMD	USINT	Command byte
STS	USINT	Status byte
TNSW	UINT	Transport word. Same value as request
EXT_STS	USINT	Extended status (if error)
PCCC_params	Array of USINT	CMD/FMC specific result data

Tab. 36: PCCC object (67hex) supported PCCC command types

CMD	FNC	Description
0Fhex	67hex	PLC-5 write
0Fhex	68hex	PLC-5 read
0Fhex	A2hex	SLC 500 protected read with 3 address fields
0Fhex	AAhex	SLC 500 protected write with 3 address fields

5.2.4 How to configure Explicit Messaging

This section gives an example how to configure Explicit Messaging using RSLogix.

Required gateway settings

In the **Gateway configuration** dialog of samos®PLAN, the following settings have to be activated:

- PLC requests
- PLC writes

Required RSLogix settings

In RSLogix, the following settings must be made:

- PLC is Active (Explicit Messaging enabled)
- 128 connections possible
- Each Data Set must have the correct size.
- Main program sends a message with command to SET or GET (either Get_Attribute_Single or Get_Attributes_All, see Tab. 26).

Step 1:

➡ Create two tags. One for the MESSAGE and one to store the data from the GET command.

The MESSAGE tag will be for the MSG block which is used for explicit messaging. The MSG command in this example will request Data Set 1. The received Data Set 1 will then be placed into a tag of our choosing, called “WhatIWant”.

NOTE

The destination tag must be set to the size of the Data Set. In this case, it would be 50 Bytes or INT[25].

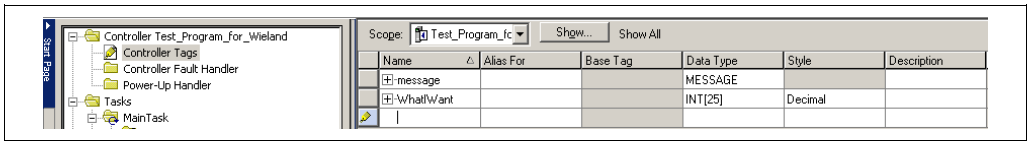


Fig. 23: Creating tags for Explicit Messaging

Step 2:

➡ On one line in the program, create a NOT connected to the MSG command

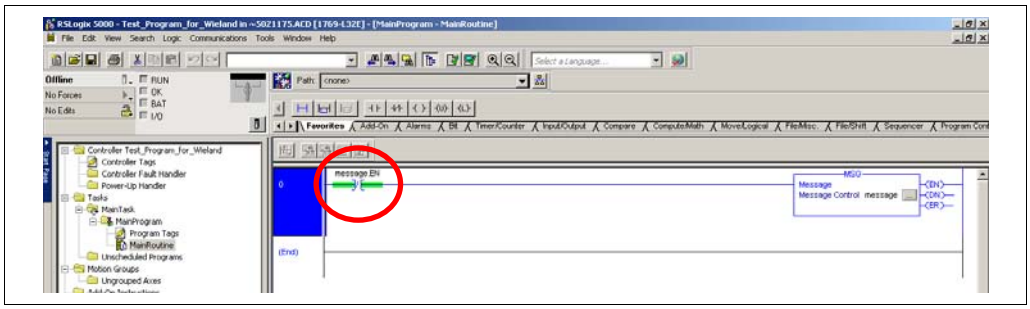


Fig. 24: Main routine programming for Explicit Messaging

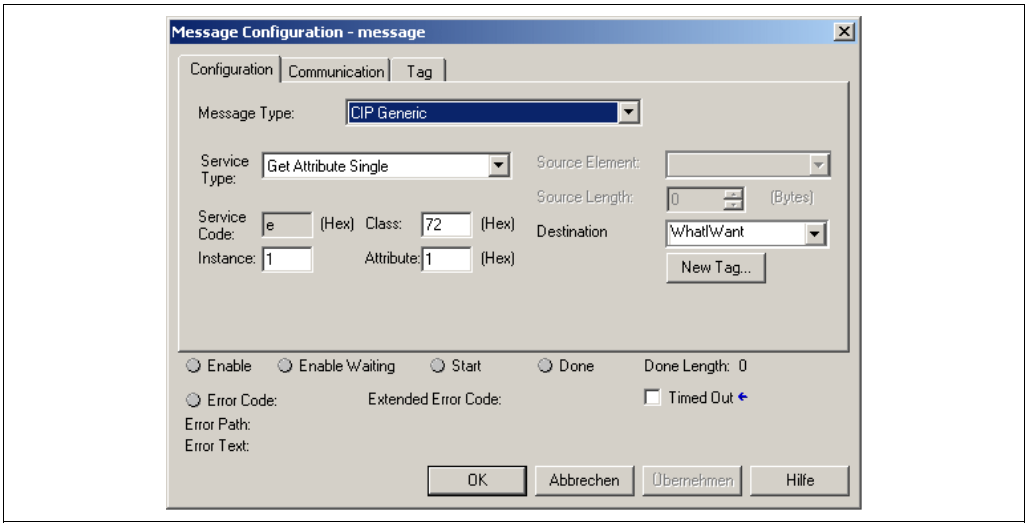
➡ Select “message.EN” for the NOT symbol. This ensures that the message will be requested repeatedly.

Step 3:

➧ Edit the message command as follows:

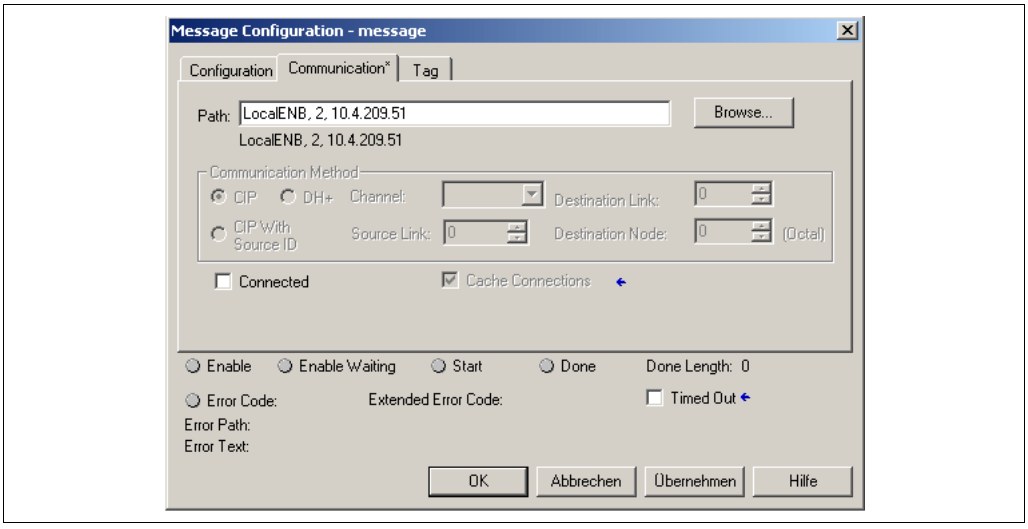
– Configuration: Select **CIP Generic**. The **Class** describes the object. **Class 72** is for *Full Data Set Transfer*. **Instance 1** and **Attribute 1** determine the data type which in this case is **Data Set 1** (see Tab. 22). As **Destination** the “WhatIWant” tag must be chosen.

Fig. 25: Explicit messaging – Message configuration



– Communication: The message has to contain the **Path** to the gateway. In this example the path is 10.4.209.51.

Fig. 26: Explicit messaging – Communication configuration



5.2.5 TCP/IP configuration interface

See section 5.1.1 “TCP/IP configuration interface” on page 26.

5.2.6 Ethernet TCP/IP socket interface

See section 5.1.2 “Ethernet TCP/IP socket interface” on page 30.

5.2.7 Diagnostics and troubleshooting

For information how to perform diagnostics on the ***samos***®PRO system please refer to the operating instructions for the ***samos***®PLAN software (Wieland part no. BA000518).

Tab. 37: Troubleshooting for the SP-EN-IP

Error	Cause	Possible remedy
Symbol description: O – LED off, ● Green – LED lights up green, ✸ Red – LED flashes red		
The ***samos***®PLAN tool does not connect to the ***samos***®PRO gateway module	SP-EN-IP has no power supply. SP-EN-IP is not in the same physical network as the PC. The PC is configured to another subnet mask in the TCP/IP settings. SP-EN-IP has already been configured once and has a fixed set IP address or an IP address assigned by a DHCP server that is not recognised.	Establish the power supply. Check the Ethernet wiring and network settings on the PC and correct if necessary. Set the subnet mask on the PC to 255.255.0.0 (factory setting of the SP-EN-IP). Check the communication settings in the ***samos***®PLAN.
SP-EN-IP does not supply any data. LED PWR ● Green LED LINK/ACT ●/✸ Green LED MS ✸ Red/Green	SP-EN-IP is configured for data transfer to PLC, but Ethernet communication is not yet established or faulty. Duplicate IP address detected. Another device on the network has the same IP address.	Minimum one Ethernet connection needs to be established. Set up Ethernet connection on PLC side, check Ethernet cabling, check Ethernet connection settings on PLC and in the ***samos***®PLAN. If no Ethernet communication is required, disable the Ethernet connections/PLC interfaces on the SP-EN-IP. Adjust IP address and power cycle device.
SP-EN-IP does not supply any data. LED PWR ● Green LED LINK/ACT ●/✸ Green LED MS ✸ Red (1 Hz)	Configuration required. Configuration download is not completed.	Configure the SP-EN-IP and download the configuration to the device. Wait until the configuration download has been completed.
SP-EN-IP does not supply any data. LED PWR ● Green LED LINK/ACT ●/✸ Green LED MS ● Green	No data set is activated. No Ethernet communication interface is enabled.	Activate at least one data set.
SP-EN-IP does not supply any data. LED PWR ● Green LED LINK/ACT ●/✸ Green LED MS ✸ Green (1 Hz)	SP-EN-IP is in Idle mode.	CPU/application is stopped. Start CPU (change into Run mode) .
SP-EN-IP functioned correctly after configuration, but suddenly no longer supplies data. LED PWR ● Green LED LINK/ACT ●/✸ Green LED MS ✸ Red/Green	SP-EN-IP is operated in slave mode, the IP address is assigned from a DHCP server. After the SP-EN-IP or the DHCP server has been restarted, a different IP address that is unknown to the PLC has been assigned to the SP-EN-IP.	Either assign a fixed IP address to the SP-EN-IP, or reserve a fixed IP address for the SP-EN-IP in the DHCP server (manual assignment by means of the MAC address of the SP-EN-IP).
SP-EN-IP/ ***samos***®PRO system is in Critical fault mode. LED PWR ● Green LED LINK/ACT ✸ Green LED MS ● Red	SP-EN-IP is not plugged properly into the other ***samos***®PRO module. Module connection plug is soiled or damaged. Other ***samos***®PRO module has internal critical error.	Plug the SP-EN-IP in correctly. Clean the connecting socket/plug. Repower the system. Check the other ***samos***®PRO modules.
SP-EN-IP is in Critical Fault mode. LED PWR ● Green LED LINK/ACT ●/✸ Green LED MS ✸ Red (2 Hz)	SP-EN-IP internal device error CPU firmware version does not support ***samos***®PRO gateways.	Switch off the power supply of the ***samos***®PRO system and switch it on again. Check the diagnostics messages with the ***samos***®PLAN. Use a CPU with the required firmware version (see section 2.2 “Intended use” on page 10). If the error remains, replace the gateway.

5.3 Modbus TCP gateway

The following ***samos***®PRO gateway can be used for Modbus/TCP: SP-EN-MOD.

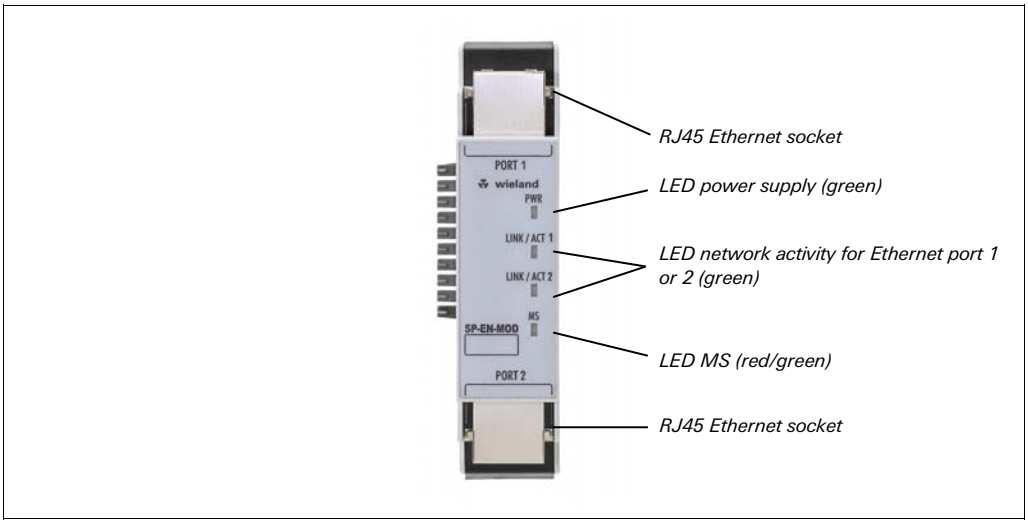
The ***samos***®PRO Modbus TCP gateway supports:

- Modbus TCP master and slave receive methods
- Ethernet TCP/IP socket interface, polling and auto update function

5.3.1 Interfaces and operation

The SP-EN-MOD is equipped with an integrated three-port switch for connection with the Ethernet network. Two RJ45 sockets are available for the connection. The switch functionality allows the SP-EN-MOD to be used for connection to another Ethernet component (e.g. connection to a notebook) without having to interrupt the Ethernet connection to the network.

Fig. 27: Interfaces and display elements of the SP-EN-MOD



Tab. 38: Meaning of the LED displays

LED		Meaning
Symbol description: ○ – LED off, ● Green – LED lights up green, ✸ Red – LED flashes red		
PWR	○	No power supply
	● Green	Power supply switched on
LINK/ACT 1 LINK/ACT 2	○	No Ethernet connection
	● Green	Ethernet connection active, no data transmission
	✸ Green	Ethernet connection active, data transmission
MS	○	Power-up
	● Green	Executing (live process data from/to CPU)
	✸ Green	1 Hz: Idle
	✸ Red	1 Hz: Configuring/configuration required 2 Hz: Critical fault on gateway
	● Red	Critical fault on another module
	✸ Red/Green	Executing, but Ethernet communication not established or faulty

NOTE Error elimination is described in section 5.3.6 “Diagnostics and troubleshooting” on page 59.

Power-up sequence

On power up, the following LED test sequence is performed:

- LED MS ○ **Off** for 6 s.
- LED MS ● **Red** for 0.25 s.
- LED MS ● **Green** for 0.25 s.
- LED MS ○ **Off**.

5.3.2 Basic configuration — assigning an IP address

Configuration of the SP-EN-MOD is performed via the ***samos***®PLAN tool.

Via *samos*®PLAN tool

➡ Open the ***samos***®PLAN and load the hardware configuration including the Modbus TCP gateway.
➡ Click on the **Gateway** button above the main window and select the SP-EN-MOD to open the gateway configuration dialog.
➡ Click on **Gateway Configuration** on the left hand menu. The following dialog appears:

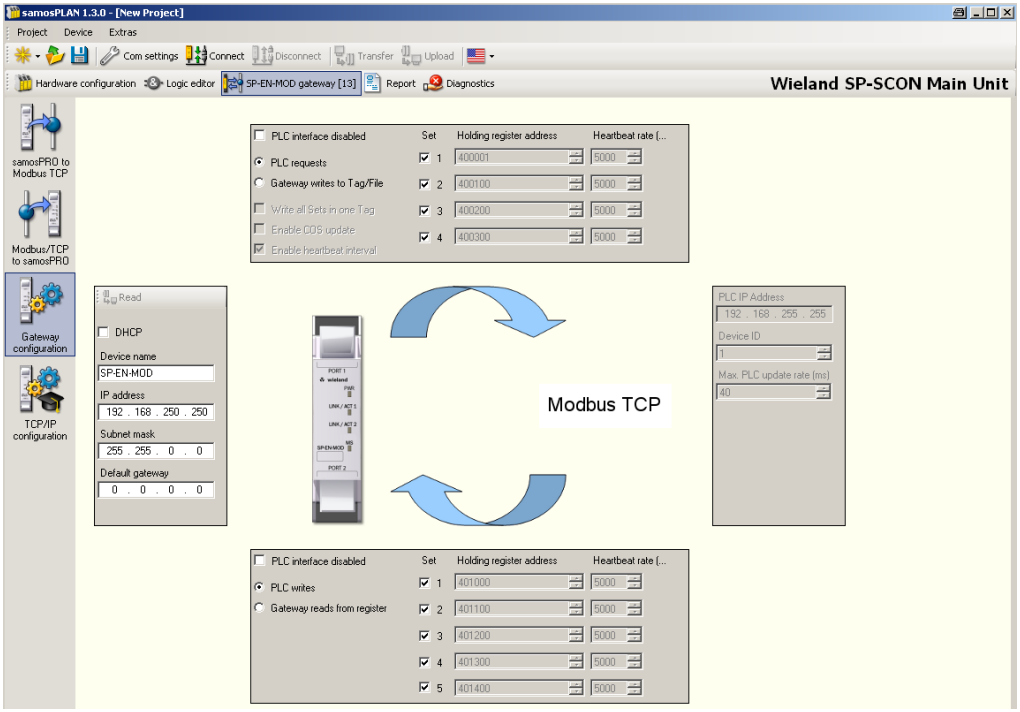


Fig. 28: Gateway configuration dialog

➡ Enter a valid **IP address**, **Subnet mask** and if required a valid IP address for a **Default gateway**.
➡ Click **OK**.
➡ Click **Connect** to go online and download the configuration to the ***samos***®PRO system.

5.3.3 Configuration of the Modbus TCP interface to the PLC — how the data is transferred

Modbus TCP application characteristics:

- Support of standard Modbus TCP addressing conventions.
- Master and Slave receive methods

Modbus TCP PLC requirements:

- The PLC must support the Modbus TCP protocol.
- The PLC must support the Read Holding Registers and Write Multiple Registers commands or, alternatively, the Read/Write Multiple Registers command.

The configuration steps in this section specify how the data to the higher-level PLC are transferred.

In general, there are two different transfer methods available for each transfer direction such as ***samos***®PRO *to Network* and *Network to* ***samos***®PRO:

- Polling receive method/PLC requests (gateway as slave)
  This method provides a polling method that allows the PLC to request data on a periodic basis. In this method, the data is returned in the response to the data request message. The PLC requests data by accessing the receive data address on the SP-EN-MOD module with a Read Holding Registers message.
- Master receive method – Gateway writes to PLC (auto-update, gateway as master)
  When it is determined that data received on the SP-EN-MOD module backplane interface is to be sent to the PLC, the data is immediately written to a data memory location on the PLC.
- Slave transmit method - PLC writes (gateway as slave)
  In this method, the PLC will send write messages to the SP-EN-MOD module to set the output data sets. To write to the output data sets, the PLC writes the data to specified addresses.
- Master transmit method - Gateway reads from PLC (auto-update, gateway as master)
- In the master transmit mode, the SP-EN-MOD module will poll the PLC for the output data set settings.

NOTE The configuration is considered faulty, if the PLC IP address is zero and either the Read Transfer mode and/or the Write Transfer mode is set to Master.

The number of possible connections to the PLC depends on whether the SP-EN-MOD is operated as a master or slave. Depending on the setting, up to 32 PLCs can address the SP-EN-MOD at the same time.

Tab. 39: Number of possible connections

Operating mode of the SP-EN-MOD	Maximum connections
Rx (To PLC) transfer mode: Master Tx (From PLC) transfer mode: Master	Rx and Tx: 1
Rx (To PLC) transfer mode: Master Tx (From PLC) transfer mode: Slave	Rx: 1 Tx: 31
Rx (To PLC) transfer mode: Slave Tx (From PLC) transfer mode: Master	Rx: 31 Tx: 1
Rx (To PLC) transfer mode: Slave Tx (From PLC) transfer mode: Slave	Rx and Tx: 32

The following table outlines a guideline to the configuration process depending on the transfer method:

Gateway is master

Tab. 40: Configuration guideline — gateway as master

To do in the gateway configuration (via ***samos***®PLAN tool)	To do in the PLC program and/or Modbus TCP network configuration tool
Select **Gateway writes to Tag/File** and/or **Gateway reads from register** to configure gateway as master..	–
Select which data shall be written to/read from the PLC	–
Define where in the PLC memory the selected data shall be written to: Enter holding register address(es). Example: "400001" And/or define where in the PLC memory the selected data shall be read from: Enter holding register addresses.	Ensure the addresses defined in the ***samos***®PLAN configuration are available and contain the data determined for the ***samos***®PRO system.
Select how often this data shall be transmitted.	–
Define where the data shall be read from/written to in the Modbus TCP network: Enter the IP address and controller slot number of the PLC.	–

Gateway is slave

To do in the gateway configuration (via *samos*®PLAN tool)	To do in the PLC program and/or Modbus TCP network configuration tool
Select **PLC requests** and **PLC writes** in the gateway configuration dialog	–
–	Define which data shall be written to/read from the gateway. Ensure the PLC program writes the data into the addresses defined for the gateway (refer to section "SP-EN-MOD as slave — data addressing").

Tab. 41: Configuration guideline — gateway as slave

NOTE

The Modbus TCP gateway address setting is based 1. Please add 1 to the holding register address set in the ***samos***®PLAN for an address setting based 0.

Master mode — SP-EN-MOD reads from/writes to the PLC

In order to configure the gateway to be *master*, perform the following steps:

- Open the ***samos***®PLAN and load the hardware configuration including the Modbus TCP gateway.
- Click on the **Gateway** button above the main window and select the SP-EN-MOD to open the gateway configuration dialog.
- Click on **Gateway Configuration** on the left hand menu. The following dialog appears:

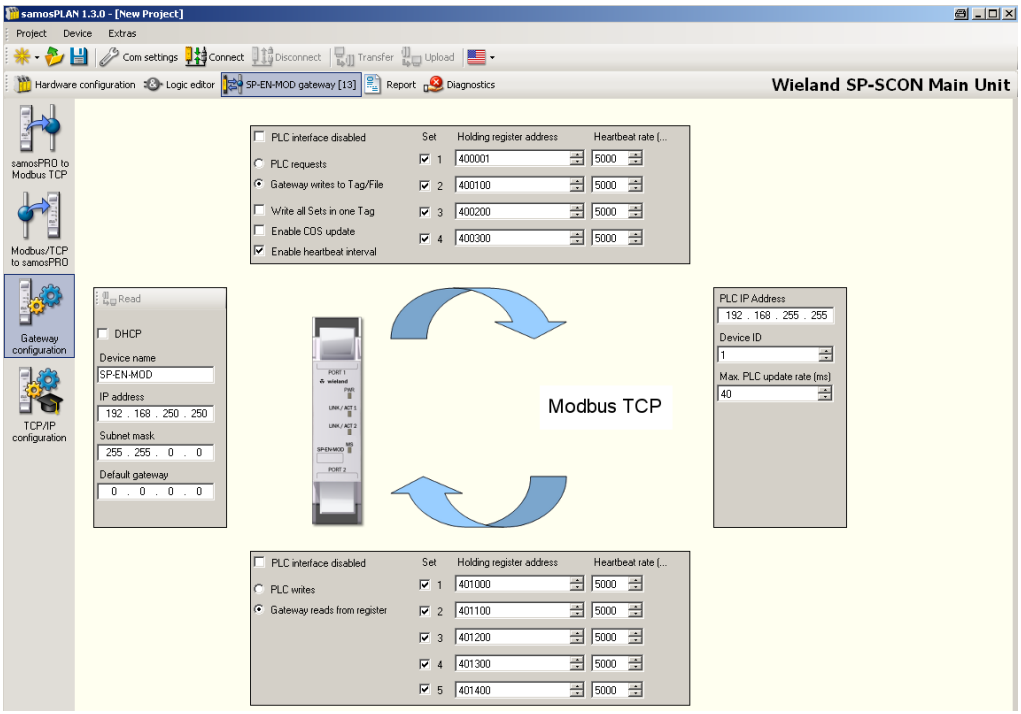


Fig. 29: Gateway configuration dialog

- Within the **Gateway Configuration** dialog, select the transfer method by checking **Gateway writes to Tag/File** for the ***samos***®PRO to Network direction and **Gateway reads from register** for the Network to ***samos***®PRO direction.
- Select which data shall be written to/read from the PLC by checking the checkbox for the required data set. For the exact description of the data sets please refer to section 3.2 "Data transmitted into the network (network input data sets)" on page 13.
- Define where in the PLC memory the selected data shall be written to or read from: Enter addresses into the **Holding register address** field (max. 20 characters).
- Select **Write all Sets in one tag** if all data sets shall be written into one address in the PLC memory. In this case, the tag/file defined for data set 1 will be used.
- For the ***samos***®PRO **to Network** direction, define how often the data shall be transmitted:
  - Select **Enable Update COS** if the SP-EN-MOD is to update the data in the PLC immediately when changes occur in the data sets.

– Select **Enable heartbeat interval** to activate updating of the selected data sets with the set Heartbeat rate.
– Both options may be selected at the same time.

➡ For the **Network to *samos*®PRO** direction, define how often the data shall be read:
– Enter a **Heartbeat rate** to activate updating of the selected data sets with the set time interval.

➡ Define where the data shall be read from/written to in the Modbus TCP network: Enter the **PLC IP address** and the Modbus **Device ID** of the PLC.

➡ **Max. PLC update rate** defines the maximum rate (the minimum time interval) for sending the data sets to the PLC. Settings occur dependent on the PLC processing speed. Minimum = 10 ms, maximum = 65535 ms. The default value of 40 ms is suitable for most PLCs.

NOTE If this value is greater than the **Heartbeat rate**, the heartbeat rate is slowed down to this value.

➡ Click **OK**.

➡ Go online and download the configuration to the ***samos*®PRO** system.

Write to PLC

NOTE The following restrictions apply when the gateway is master and writes the input data sets to the PLC:

- The input data set address (set via ***samos*®PLAN** Tool) must be the same as that defined on the PLC.
- The variable to receive the data on the PLC must be:
  – in the 40xxxx address range (for Schneider Modicon type PLCs)
  – an array of 16 bit words
  – long enough to contain the specified input data set array.
- All input data sets are transferred to the PLC in 16 bit word format with the first byte placed in the least significant, or rightmost byte of the integer and the second byte placed in the most significant, or leftmost byte of the integer.

Read from PLC

NOTE The following restrictions apply when the gateway is master and reads the output data sets from the PLC:

- The output data set addresses must be the same as those defined on the PLC.
- The variables to request the data on the PLC must be:
  – In the 40xxxx address range (for Schneider Modicon type PLCs)
  – for the output data set settings, an array of 16 bit words long enough to contain the entire output data set.
- All output data sets are transferred from the PLC in 16 bit word format and the first byte must be placed in the least significant, or rightmost byte of the integer and the second byte placed in the most significant, or leftmost byte of the integer.

Slave mode — PLC reads from/writes to the SP-EN-MOD

In this operating mode the SP-EN-MOD sends the data as *slave* upon request from the PLC. If this operating mode is desired:

➡ Open the ***samos*®PLAN** and load the hardware configuration including the Modbus TCP gateway.

➡ Click on the **Gateway** button above the main window and select the SP-EN-MOD to open the gateway configuration dialog.

➡ Click on **Gateway Configuration** on the left hand menu. The following dialog appears:

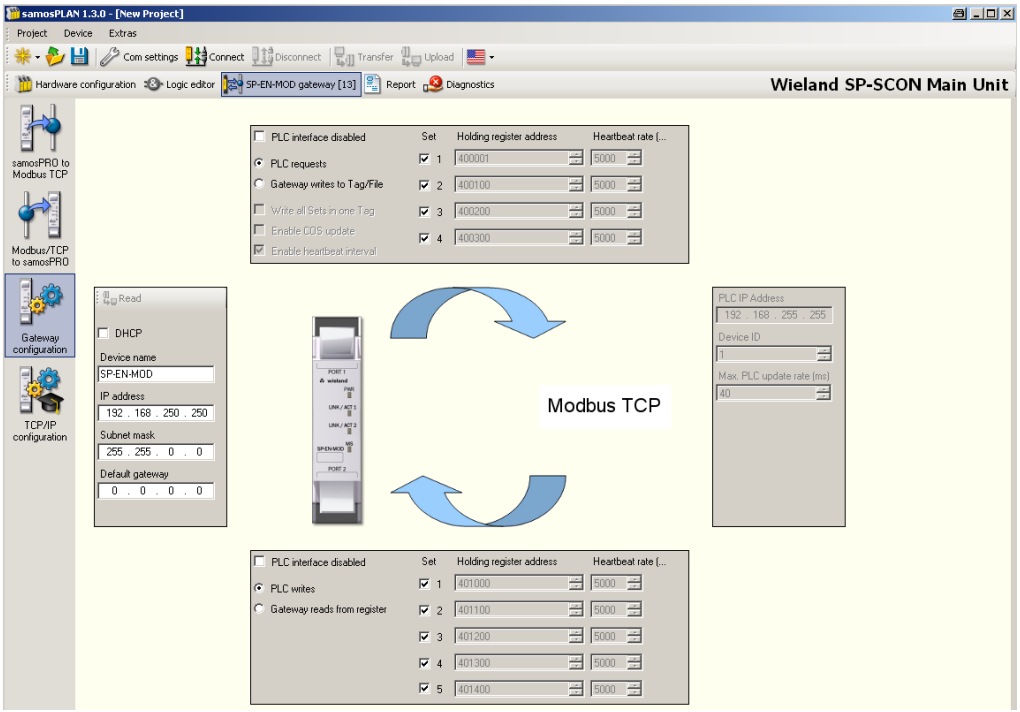


Fig. 30: Gateway configuration dialog

➡ Within the **Gateway Configuration** dialog, select the transfer method by checking **PLC requests** for the ***samos***®PRO to Network direction and **PLC writes** for the Network to ***samos***®PRO direction.

➡ Select which data shall be written/read to/from the PLC by checking the checkbox for the required data set. For the exact description of the data sets please refer to section 3.2 “Data transmitted into the network (network input data sets)” on page 13.

➡ Click **OK**.

➡ Go online and download the configuration to the ***samos***®PRO system.

PLC writes output data sets

The following restrictions apply when the PLC writes the output data sets:

- The device index must be 1.
- The message must be sent in word format.
- All output data sets are transferred from the PLC in 16 bit word format and the first byte must be placed in the least significant, or rightmost byte of the integer and the second byte placed in the most significant, or leftmost byte of the integer.

PLC polls input data sets

The following restrictions apply to this method:

- The device index must be 1.
- The variable to receive the data on the PLC must be:
  – in the 40xxxx address range (for Modicon type PLCs)
  – an array of 16 bit words
  – long enough to contain the data set array(s)
- All input data sets are transferred to the PLC in 16 bit word format with the first byte placed in the least significant, or rightmost byte of the integer and the second byte placed in the most significant, or leftmost byte of the integer.

SP-EN-MOD as slave — data addressing

The following table lists the address to read out the data sets.

Unit ID 1

Tab. 42: Data addressing for SP-EN-MOD as receiver

Address (Base 1)	Description	Access	Scope (words)
1000	Request all enabled input data sets	Get	16-101 [6]
1100	Request input data set 1 data	Get	25
1200	Request input data set 2 data	Get	16
1300	Request input data set 3 data	Get	30
1400	Request input data set 4 data	Get	30
2000	Write all enabled output data sets data	Set	5-25 [7]
2100	Write output data set 1 data	Set	5
2200	Write output data set 2 data	Set	5
2300	Write output data set 3 data	Set	5
2400	Write output data set 4 data	Set	5
2500	Write output data set 5 data	Set	5

Modbus commands and error messages

The SP-EN-MOD supports the following Modbus commands and error messages:

Tab. 43: Modbus commands

Modbus command	Value
Read holding registers	3
Write multiple registers	16 (10hex)
Read/write multiple registers	23 (17hex)

Tab. 44: Modbus error messages

Modbus error response	Description
1 Illegal function	The requested function is not supported
2 Illegal data address	Undefined data address received
3 Illegal data value	Request with illegal data values, for example not enough data requested for a data set
10 Gateway paths not available	Invalid configuration, for example polling or setting of the digital outputs via PLC during operation of the SP-EN-MOD in master mode

5.3.4 TCP/IP configuration interface

See section 5.1.1 “TCP/IP configuration interface” on page 26.

5.3.5 Ethernet TCP/IP socket interface

See section 5.1.2 “Ethernet TCP/IP socket interface” on page 30.

[6] Will correspond to all enabled input data sets.

[7] Must correspond to all enabled output data sets. Example: If only output data sets 1 and 2 are enabled, then 10 words (20 bytes) must be written. If all output data sets are enabled, then 25 words (50 bytes) must be written.

5.3.6 Diagnostics and troubleshooting

For information how to perform diagnostics on the ***samos***®PRO system please refer to the operating instructions for the ***samos***®PLAN software (Wieland part no. BA000518).

Tab. 45: Troubleshooting for the SP-EN-MOD

Error	Cause	Possible remedy
Symbol description: O – LED off, ● Green – LED lights up green, ✸ Red – LED flashes red		
The ***samos***®PLAN tool does not connect to the ***samos***®PRO gateway module	SP-EN-MOD has no power supply. SP-EN-MOD is not in the same physical network as the PC. The PC is configured to another subnet mask in the TCP/IP settings. SP-EN-MOD has already been configured once and has a fixed set IP address or an IP address assigned by a DHCP server that is not recognised.	Establish the power supply. Check the Ethernet wiring and network settings on the PC and correct if necessary. Set the subnet mask on the PC to 255.255.0.0 (factory setting of the SP-EN-MOD). Check the communication settings in the ***samos***®PLAN.
SP-EN-MOD does not supply any data. LED PWR ● Green LED LINK/ACT ●/✸ Green LED MS ✸ Red/Green	SP-EN-MOD is configured for data transfer to PLC, but Ethernet communication is not yet established or faulty. Duplicate IP address detected. Another device on the network has the same IP address.	Minimum one Ethernet connection needs to be established. Set up Ethernet connection on PLC side, check Ethernet cabling, check Ethernet connection settings on PLC and in the ***samos***®PLAN. If no Ethernet communication is required, disable the Ethernet connections/PLC interfaces on the SP-EN-MOD. Adjust IP address and power cycle device.
SP-EN-MOD does not supply any data. LED PWR ● Green LED LINK/ACT ●/✸ Green LED MS ✸ Red (1 Hz)	Configuration required. Configuration download is not completed.	Configure the SP-EN-MOD and download the configuration to the device. Wait until the configuration download has been completed.
SP-EN-MOD does not supply any data. LED PWR ● Green LED LINK/ACT ●/✸ Green LED MS ● Green	No data set is activated. No Ethernet communication interface is enabled.	Activate at least one data set.
SP-EN-MOD does not supply any data. LED PWR ● Green LED LINK/ACT ●/✸ Green LED MS ✸ Green (1 Hz)	SP-EN-MOD is in Idle mode.	CPU/application is stopped. Start CPU (change into Run mode).
SP-EN-MOD functioned correctly after configuration, but suddenly no longer supplies data. LED PWR ● Green LED LINK/ACT ●/✸ Green LED MS ✸ Red/Green	SP-EN-MOD is operated in slave mode, the IP address is assigned from a DHCP server. After the SP-EN-MOD or the DHCP server has been re-started, a different IP address that is unknown to the PLC has been assigned to the SP-EN-MOD.	Either assign a fixed IP address to the SP-EN-MOD, or reserve a fixed IP address for the SP-EN-MOD in the DHCP server (manual assignment by means of the MAC address of the SP-EN-MOD).
SP-EN-MOD/ ***samos***®PRO system is in Critical fault mode. LED PWR ● Green LED LINK/ACT ✸ Green LED MS ● Red	SP-EN-MOD is not plugged properly into the other ***samos***®PRO module. Module connection plug is soiled or damaged. Other ***samos***®PRO module has internal critical error.	Plug the SP-EN-MOD in correctly. Clean the connecting socket/plug. Repower the system. Check the other ***samos***®PRO modules.
SP-EN-MOD is in Critical fault mode. LED PWR ● Green LED LINK/ACT ●/✸ Green LED MS ✸ Red (2 Hz)	SP-EN-MOD internal device error CPU firmware version does not support ***samos***®PRO gateways.	Switch off the power supply of the ***samos***®PRO system and switch it on again. Check the diagnostics messages with the ***samos***®PLAN. Use a CPU with the required firmware version (see section 2.2 “Intended use” on page 10). If the error remains, replace the gateway.

5.4 PROFINET IO gateway

The following ***samos***®PRO gateway can be used for PROFINET IO: SP-EN-PN.

You will find the GSDML file and device icon for PLC interfacing with PROFIBUS support

- in the Internet on the SP-EN-PN product page (http://eshop.wieland-electric.com).

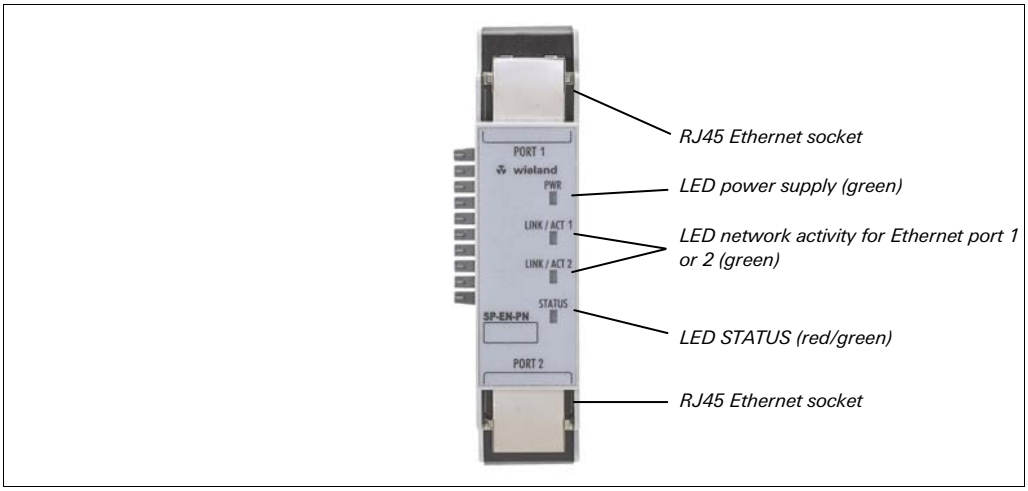
The SP-EN-PN supports

- PROFINET IO conformance class A
- LLDP
- SNMP
- MIB II
- Fast integrated switching
- Auto-MDI
- Auto negotiation
- Cyclic IO communication

5.4.1 Interfaces and operation

The SP-EN-PN is equipped with an integrated 3-port switch for connection with the Ethernet network. Two RJ45 sockets are available for the connection. The switch functionality allows the SP-EN-PN to be used for connection to another Ethernet component (e.g. connection to a notebook) without having to interrupt the Ethernet connection to the network.

Fig. 31: Interfaces and display elements of the SP-EN-PN



Tab. 46: Meaning of the LED displays of the SP-EN-PN

LED		Meaning
Symbol description: ○ – LED off, ● Green – LED lights up green, ✸ Red – LED flashes red		
PWR	○	No power supply
	● Green	Power supply switched on
LINK/ACT 1 LINK/ACT 2	○	No Ethernet connection
	● Green	Ethernet connection active, no data transmission
	✸ Green	Ethernet connection active, data transmission
STATUS	○	Power-up
	● Green	Executing (live process data from/to CPU)
	✸ Green	1 Hz: Idle 2 Hz: Profinet master requested LED flashing for physical device identification
	✸ Red	1 Hz: Configuring/configuration required 2 Hz: Critical fault on gateway
	● Red	Critical fault on another module
	✸ Red/Green	Executing, but Ethernet communication not established or faulty

NOTE Error elimination is described in section 5.4.7 “Diagnostics and troubleshooting” on page 70.

Power-up sequence

On power up, the following LED test sequence is performed:

- LED MS ○ Off for 6 s.
- LED MS ● Red for 0.25 s.
- LED MS ● Green for 0.25 s.
- LED MS ○ Off.

5.4.2 Basic configuration – assigning a device name and IP address

Configuration and diagnostic of the SP-EN-PN may be performed via the ***samos***®PLAN tool as well as with the PROFINET IO network programming tool (e.g. SIEMENS SIMATIC).

Configuration via PROFINET IO

In the out-of-the-box configuration, each PROFINET IO field device, e.g. the SP-EN-PN has a MAC address and a symbolic name stored.

NOTES

- The symbolic name for the gateway is **SP-EN-PN**.
- This name is used by the IO controller (i.e. PLC) to assign the IP address to the field device.
- If the IP address is also used for other Ethernet communications like TCP/IP or configuration over Ethernet, remember that the PLC can change the IP address so these can be interupted.

Address assignment is performed in two steps.

➡ Assign a unique plant specific name to the gateway by using either the network configuration tool, e.g. SIEMENS SIMATIC Manager or the ***samos***®PLAN tool.
➡ Using the plant specific (unique) name, the IO controller (i.e. PLC) can now assign the IP address to the gateway before system boot.

NOTE

The SP-EN-PN MAC address is printed on the device type label (example: 00:06:77:02:00:A7).

Device name set via SIEMENS SIMATIC Manager

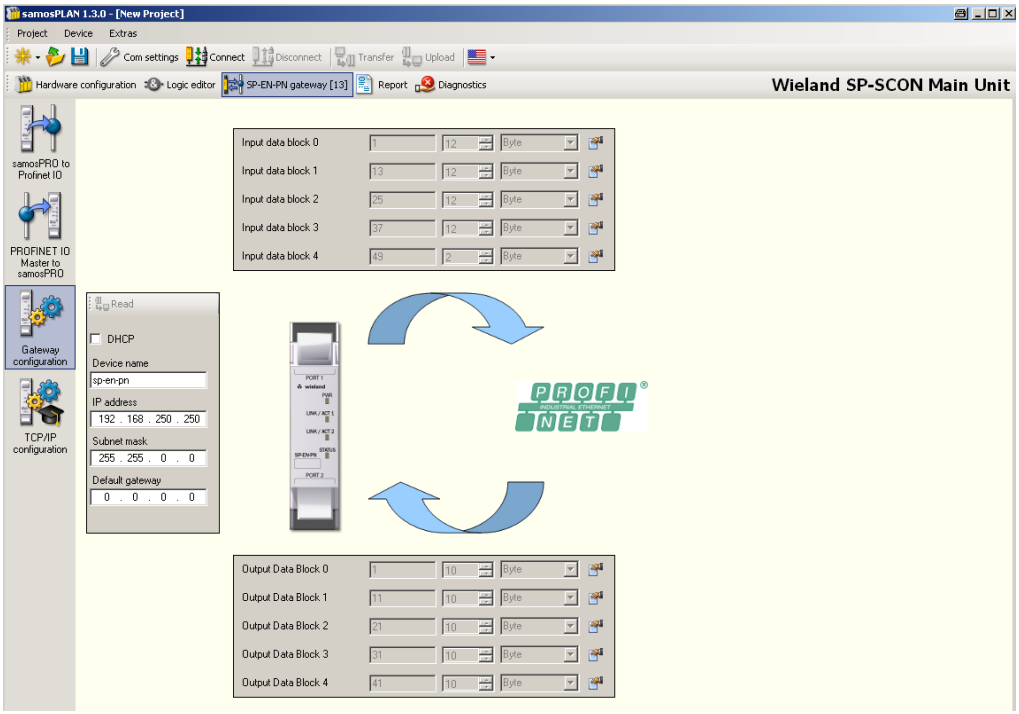
Refer to section “STEP 4 – Assign the device name” on page 63.

Device name set via *samos*®PLAN

➡ Open the ***samos***®PLAN and load the hardware configuration including the PROFINET IO gateway. Ensure your project is offline.
➡ Click on the **Gateway** button above the main window and select the SP-EN-PN to open the gateway configuration dialog.

➡ Click on **Gateway configuration** on the left hand menu. The following dialog appears:

Fig. 32: Gateway configuration dialog



➡ Enter the device name in the **Device name** field (maximum length 255 characters).

NOTES

- The device name format must apply to the PROFINET standard specification.
- Ensure that the default gateway address corresponds to the address set by the PLC for the gateway. If there is no router used, Siemens Step 7 use as default gateway address the same address as the IP address for the SP-EN-PN.

IP address set via *samos*®PLAN tool

Usually the IP address will be assigned by the PROFINET IO controller (e.g. PLC). However, the SP-EN-PN allows configuration of the entire ***samos***®PRO system over Ethernet TCP/IP. In this case, it may be necessary to assign an IP address to the gateway even before the PROFINET IO network has been setup. This can be done on the configuration page shown in Fig. 32 as well.

5.4.3 PROFINET configuration of the gateway – how the data is transferred

The following steps need to be taken in order to configure the communication between PLC and gateway.

NOTE

This document does not cover the creation of the PROFINET IO network or the rest of the automation system project in the network configuration tool. It is assumed the PROFINET project has already been set up in the configuration program, e.g. SIEMENS SIMATIC Manager. Examples refer to configurations performed with SIEMENS SIMATIC manager.

STEP 1 – Install the generic station description file (GSDML file)

Before the SP-EN-PN can be used as device in the network configuration tool, e.g. SIEMENS SIMATIC Manager, for the first time, the generic station description (GSDML) of the gateway must be installed into the hardware catalogue of the tool.

➡ Download the GSDML file and device icon from the product page of SP-EN-PN (http://eshop.wieland-electric.com).
➡ Follow the instructions in the online help or in the user manual of the PROFINET network configuration tool for installing generic station description files.

Using the SIEMENS SIMATIC Manager – HW Config, the gateway then appears in the hardware catalogue under **>>PROFINET IO > Additional Field Devices > Gateway > Wieland**.

STEP 2 – Add the gateway to the project

In order to have the ***samos***®PRO system data available in the PLC process image, the gateway must be added to the hardware configuration first. The procedure associated with this depends on the hardware configuration program of the PLC being used. On this topic, please also read the documentation for the corresponding program.

The example below shows how to add the gateway to a SIEMENS SIMATIC Manager project.

In the SIEMENS SIMATIC Hardware Manager, the gateway can be found in the hardware catalogue under **>>PROFINET IO > Additional Field Devices > Gateway > Wieland**.

➡ Drag & drop the device into the Ethernet PROFINET IO network. Example:

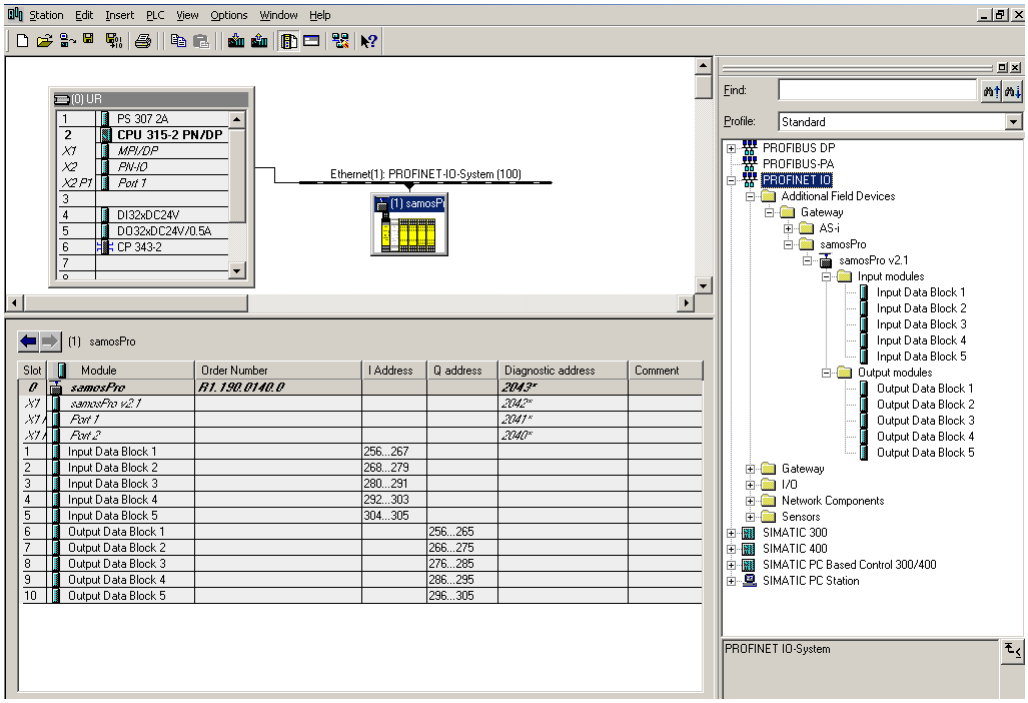


Fig. 33: Gateway in the PROFINET IO HW Config

After adding the device to the automation network it is required to configure which of the cyclic data blocks will be used and where they will be addressed in memory. For details refer to section 5.4.4 “PROFINET configuration of the gateway — which data are transferred” on page 64.

STEP 3 – Configure the properties of the gateway

➡ Double click on the gateway hardware symbol.
➡ Configure the update time of the cyclic IO data exchange. To do this click on the **IO Cycle** tab and select the desired rate from the update time pull-down menu.

STEP 4 – Assign the device name

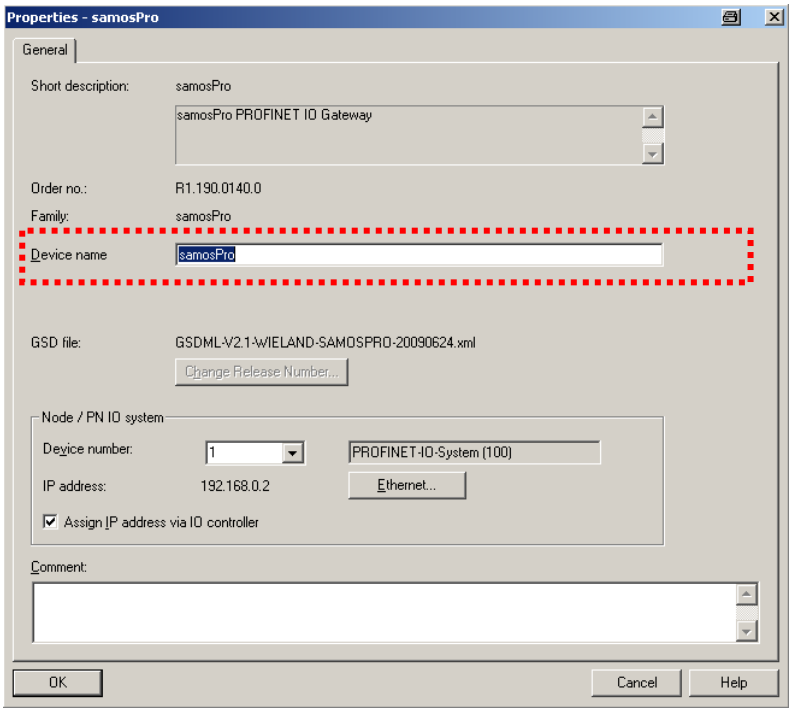
In order for the PLC to communicate with the SP-EN-PN, the PLC software and the gateway must agree on the name of the gateway.

Specify the gateway's PROFINET IO device name

➡ Double click on the gateway hardware symbol.
➡ Select the **General** tab.

➡ Enter the desired device name in the dialog as shown below:

Fig. 34: Properties dialog of the SP-EN-PN



NOTE The device name format shall apply to the PROFINET standard specification.

Assign the device name to the gateway.

➡ Select **PLC > Ethernet > Assign Device Name**. The **Assign device name** dialog opens.
➡ From the **Assign device name** dialog, find and select the Wieland gateway that you wish to assign the device name to in the list.
➡ Click the **Assign name** button.

5.4.4 PROFINET configuration of the gateway — which data are transferred

Cyclic data

The physical ***samos***®PRO IO modules are not represented as typical hardware modules in the PROFINET IO hardware catalogue. Instead, the data available from the ***samos***®PRO system has been organized into data blocks. Each data block represents a “hardware” module in the PROFINET IO hardware catalogue. The ***samos***®PRO PROFINET IO gateway GSDML supports ten (10) slots (see Fig. 35) where the modules can be placed into. This allows each data block to be mapped once.

Process data from the *samos*®PRO system to the PLC

The SP-EN-PN provides 5 input data blocks (virtual I/O device modules) containing the process image. These must be projected in a hardware configurator (e.g. SIEMENS HW Config) in natural order (1, 2, 3, 4, 5). No other sequence is possible.

NOTES

- Depending on the type of PLC used, further modules may be displayed (e.g.“universal module”). These modules are not needed and should be ignored.
- The input data blocks 1-4 contain 12 bytes each, input data block 5 contains 2 bytes.

The contents of the input data blocks are freely selectable, but are preconfigured in the ***samos***®PLAN configuration software:

	Data block 1	Data block 2	Data block 3	Data block 4	Data block 5
	Input data	Input data	Input data	Input data	Input data
Byte 0	Input values module 1	Output values module 1	Logic result 1	Not assigned	Not assigned
Byte 1	Input values module 2	Output values module 2	Logic result 2	Not assigned	Not assigned
Byte 2	Input values module 3	Output values module 3	Logic result 3	Not assigned	Not available
Byte 3	Input values module 4	Output values module 4	Logic result 4	Not assigned	
Byte 4	Input values module 5	Output values module 5	Not assigned	Not assigned	
Byte 5	Input values module 6	Output values module 6	Not assigned	Not assigned	
Byte 6	Input values module 7	Output values module 7	Not assigned	Not assigned	
Byte 7	Input values module 8	Output values module 8	Not assigned	Not assigned	
Byte 8	Input values module 9	Output values module 9	Not assigned	Not assigned	
Byte 9	Input values module 10	Output values module 10	Not assigned	Not assigned	
Byte 10	Input values module 11	Output values module 11	Not assigned	Not assigned	
Byte 11	Input values module 12	Output values module 12	Not assigned	Not assigned	
Length	12 bytes	12 bytes	12 bytes	12 bytes	2 bytes

Tab. 47: Default content of input data block 1-5 of the SP-EN-PN

For detailed information about the content of the process image please see section 3.2 “Data transmitted into the network” on page 13.

For information on how to configure the process image, see chapter 7 “Layout and content of the process image” on page 109 and the ***samos***®PLAN operating instructions (Wieland part no. BA000518).

Data from the PLC to the *samos*®PRO CPU

There are five (5) output data blocks, 10 bytes each.

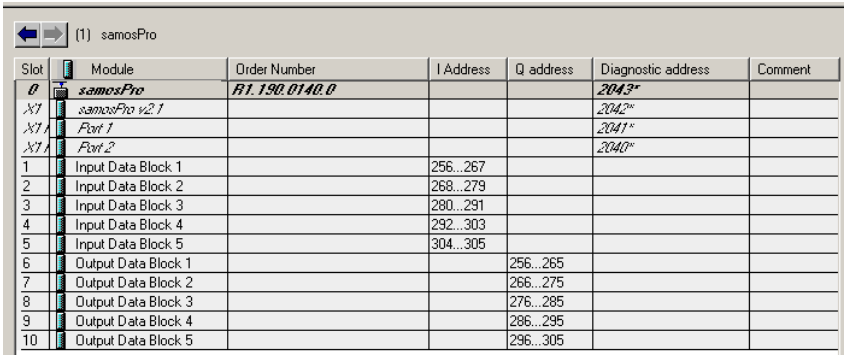
The content of these data blocks can be used as input in the ***samos***®PRO logic editor or can be routed via a second gateway into another network. In order to have the desired bits available for routing or in the logic editor, tag names have to be defined for each bit that shall be used. Bits without a tag name will not be available.

For detailed information how to define and customize the content and tag names of the input and output data please see chapter 7 “Layout and content of the process image” on page 109 and the operating instructions for the ***samos***®PLAN software (Wieland part no. BA000518).

Settings within the PROFINET IO network configuration tool

➡ Drag the data blocks from the SIEMENS SIMATIC Manager – HW Config hardware catalogue under **>>PROFINET IO > Additional Field Devices > Gateway > Wieland > *samos*®PRO... > data blocks** into the slots of the SP-EN-PN shown in the SIEMENS SIMATIC Manager – HW Config configuration table.

Fig. 35: Projecting the SP-EN-PN



NOTES

- The I and Q addresses reflect where in memory the cyclic data will be available.
- Each input data block can only be placed into the slot of the same number. Example: Input data block **4** can only be mapped into slot **4**.

Acyclic data and alarms

Record read

samos®PRO system diagnostic data is available as data record to be read by the PLC. There are three data sets, data set 2, 3 and 4, providing diagnostic information:

- Data Set 2 contains the system CRCs.
- Data Set 3 contains the individual module status with four (4) bytes per module.
- Data Set 4 is currently filled with reserved values.

The format of the data sets is as specified in the tables below.

To access the acyclic data sets, a record read must be performed on the appropriate address as shown in the following table.

Tab. 48: Memory location for data set 2, 3 and 4

	Data set 2	**Data set 3**	**Data set 4**
Location	1200-1231	1300-1359	1400-1459
Size in bytes	32 bytes	60 bytes	60 bytes

NOTE

Data set 1 is mapped into the cyclic transferred PROFINET modules of the device. The content may be defined by the user. Refer to chapter 7 “Layout and content of the process image” on page 109 for details.

	Data set 2	Data set 3	Data set 4
Byte 0	Overall CRC	Module status module 0	Reserved
Byte 1			
Byte 2			
Byte 3			
Byte 4	System CRC (SCID)	Module status module 1	
Byte 5			
Byte 6			
Byte 7			
Byte 8	Reserved	Module status module 2	
Byte 9			
Byte 10			
Byte 11			
Byte 12		Module status module 3	
Byte 13			
Byte 14			
Byte 15			
Byte 16		Module status module 4	
Byte 17			
Byte 18			
Byte 19			
Byte 20		Module status module 5	
Byte 21			
Byte 22			
Byte 23			
Byte 24		Module status module 6	
Byte 25			
Byte 26			
Byte 27			
Byte 28		Module status module 7	
Byte 29			
Byte 30			
Byte 31			
Byte ...		...	
Byte 49		...	
Byte ...		...	
Byte 56		Module status module 14. Module 13 and 14 are always the gateways.	
Byte 57			
Byte 58			
Byte 59			
Length	32 bytes	60 bytes	60 bytes

Tab. 49: Default content of input data set 2-4 of the SP-EN-PN

For the interpretation of the module status bit in data set 3 please see Tab. 5 on page 16.

Alarms

Alarms can be read acyclically through the Profinet IO alarms infrastructure. Once an error occurs on any ***samos***®PRO module, the Profinet IO gateway raises the appropriate diagnostic alarm to the network. This will trigger the fault LED on the PLC, and make the specifics (text and help) of the diagnostic alarm available through the SIMATIC PLC interface. The RALRM function block (SFB54) in OB82 (the diagnostic interrupt) allows the user to retrieve specifics on the alarm raised within the PLC program itself.

NOTES

- All alarms are output to module 0.
- The subslot number indicates the ***samos***®PRO module that has caused the alarm. Number 0 = CPU, 1 = 1st XT module, 2 = 2nd XT module ... 13 = 1st gateway, 14 = 2nd gateway.

- The reason for the alarm is being identified by an error text message from the GSDML file. Up to 32 different error messages per ***samos***®PRO module type are possible.
- For the possible causes for an alarm please refer to Tab. 50.
- The same diagnostic information is available through a record read to data set 3.

The following table matches the PROFINET IO error type (as defined by the GSDML) to the appropriate message.

Tab. 50: PROFINET IO error type definitions

	Message	
Error type	**Error origin**	**Error definition**
0100	CPU	Reserved
0101		Module operating state is Critical Fault
0102		Power supply out of range or EFI communication failure
0103		Reserved
0104		Configuration status of a module within the system changed to invalid
0105		Power supply out of specified range
0106		EFI 1 communication failure
0107		EFI 2 communication failure
0200	I/O module	Reserved
0201		Internal error: Internal tests failed or watchdog test failed or bad process data or self test failure
0202		Reserved
0203		Error history item existing: Access via configuration tool
0204		Configuration status changed to invalid
0205		Output power supply out of range
0206		Reserved
0207		Reserved
0208		Input 1-2 dual channel input evaluation: error detected
0209		Input 3-4 dual channel input evaluation: error detected
0210		Input 5-6 dual channel input evaluation: error detected
0211		Input 7-8 dual channel input evaluation: error detected
0212		Status output 1 fast shut off logic control time out
0213		Status output 2 fast shut off logic control time out
0214		Status output 3 fast shut off logic control time out
0215		Status output 4 fast shut off logic control time out
0216		Input 1 external test signal failure. Check for stuck-at-high or cabling
0217		Input 2 external test signal failure. Check for stuck-at-high or cabling
0218		Input 3 external test signal failure. Check for stuck-at-high or cabling
0219		Input 4 external test signal failure. Check for stuck-at-high or cabling
0220		Input 5 external test signal failure. Check for stuck-at-high or cabling
0221		Input 6 external test signal failure. Check for stuck-at-high or cabling
0222		Input 7 external test signal failure. Check for stuck-at-high or cabling
0223		Input 8 external test signal failure. Check for stuck-at-high or cabling
0224		Output 1 stuck-at-high error
0225		Output 1 stuck-at-low error
0226		Output 2 stuck-at-high error
0227		Output 2 stuck-at-low error
0228		Output 3 stuck-at-high error
0229		Output 3 stuck-at-low error
0230		Output 4 stuck-at-high error
0231		Output 4 stuck-at-low error
0300	PROFIBUS gateway	Reserved
0301		Internal error: Internal tests failed
0302		Reserved
0303		Reserved
0304		Configuration status changed to invalid
0305		Reserved
0306		Reserved

	Message	
Error type	Error origin	Error definition
0307 ... 0331		Reserved
0400	CANopen gateway	Reserved
0401		Internal error: Internal tests failed
0402		Reserved
0403		Reserved
0404		Configuration status changed to invalid
0405		Reserved
0406		Reserved
0407 ... 0431		Reserved
0500	DeviceNet gateway	Reserved
0501		Internal error: Internal tests failed
0502		Reserved
0503		Reserved
0504		Configuration status changed to invalid
0505		Reserved
0506		Reserved
0507 ... 0531		Reserved
0600	Modbus gateway	Reserved
0601		Internal error: Internal tests failed
0602		Reserved
0603		Reserved
0604		Configuration status changed to invalid
0605		Reserved
0606		Reserved
0607 ... 0631		Reserved
0700	Ethernet/IP gateway	Reserved
0701		Internal error: Internal tests failed
0702		Reserved
0703		Reserved
0704		Configuration status changed to invalid
0705		Reserved
0706		Reserved
0707 ... 0731		Reserved
0800	ProfiNet gateway	Reserved
0801		Internal error: Internal tests failed
0802		Reserved
0803		Reserved
0804		Configuration status changed to invalid
0805		Reserved
0806		Reserved
0807 ... 0831		Reserved
0900	Other module	Reserved
0901		Internal error: Internal tests failed
0902		Reserved
0903		Reserved
0904		Configuration status changed to invalid
0905 ... 0931		Reserved
1200	Other module	Reserved
1201		Internal error: Internal tests failed

	Message	
Error type	**Error origin**	**Error definition**
1202		Reserved
1203		Reserved
1204		Configuration status changed to invalid
1205 ... 1231		Reserved

5.4.5 TCP/IP configuration interface

See section 5.1.1 “TCP/IP configuration interface” on page 26.

5.4.6 Ethernet TCP/IP socket interface

See section 5.1.2 “Ethernet TCP/IP socket interface” on page 30.

5.4.7 Diagnostics and troubleshooting

For information how to perform diagnostics on the ***samos***®PRO system please refer to the operating instructions for the ***samos***®PLAN software (Wieland part no. BA000518).

Tab. 51: Troubleshooting for the SP-EN-PN

Error	**Cause**	**Possible remedy**
Symbol description: ○ – LED off, ● Green – LED lights up green, ✸ Red – LED flashes red		
The ***samos***®PLAN tool does not connect to the ***samos***®PRO gateway module	SP-EN-PN has no power supply. SP-EN-PN is not in the same physical network as the PC. The PC is configured to another subnet mask in the TCP/IP settings. SP-EN-PN has already been configured once and has a fixed set IP address or an IP address assigned by a DHCP server that is not recognised.	Establish the power supply. Check the Ethernet wiring and network settings on the PC and correct if necessary. Set the subnet mask on the PC to 255.255.0.0 (factory setting of the SP-EN-PN). Check the communication settings in the ***samos***®PLAN.
SP-EN-PN does not supply any data. LED PWR ● Green LED LINK/ACT ●/✸ Green LED STATUS ✸ Red/Green	SP-EN-PN is configured for data transfer to PLC, but Ethernet communication is not yet established or faulty. Duplicate IP address detected. Another device on the network has the same IP address. Improperly formatted Profinet device name.	Minimum one Ethernet connection needs to be established. Set up Ethernet connection on PLC side, check Ethernet cabling, check Ethernet connection settings on PLC and in the ***samos***®PLAN. If no Ethernet communication is required, disable the Ethernet connections/PLC interfaces on the SP-EN-PN. Adjust IP address and power cycle device. Adjust device name between Profinet Master and SP-EN-PN.
SP-EN-PN does not supply any data. LED PWR ● Green LED LINK/ACT ●/✸ Green LED STATUS ✸ Red (1 Hz)	Configuration required. Configuration download is not completed.	Configure the SP-EN-PN and download the configuration to the device. Wait until the configuration download has been completed.
SP-EN-PN does not supply any data. LED PWR ● Green LED LINK/ACT ●/✸ Green LED STATUS ✸ Green (1 Hz)	No data set is activated. ***samos***®PRO system is in Idle mode.	Activate at least one data set. Start CPU (change into Run mode) .
SP-EN-PN does not supply any data. LED PWR ● Green LED LINK/ACT ●/✸ Green LED STATUS ✸ Green (2 Hz)	Profinet Master requested LED flashing for physical device identification.	Stop flashing with Simatic Manager or power cycle ***samos***®PRO system to clear.

Error	Cause	Possible remedy
Symbol description: ○ – LED off, ● Green – LED lights up green, ✸ Red – LED flashes red		
SP-EN-PN functioned correctly after configuration, but suddenly no longer supplies data. LED PWR ● Green LED LINK/ACT ●/✸ Green LED STATUS ● Red/Green	SP-EN-PN is operated in slave mode, the IP address is assigned from a DHCP server. After the SP-EN-PN or the DHCP server has been restarted, a different IP address that is unknown to the PLC has been assigned to the SP-EN-PN.	Either assign a fixed IP address to the SP-EN-PN, or reserve a fixed IP address for the SP-EN-PN in the DHCP server (manual assignment by means of the MAC address of the SP-EN-PN).
SP-EN-PN/ ***samos***®PRO system is in Critical fault mode. LED PWR ● Green LED LINK/ACT ✸ Green LED STATUS ● Red	SP-EN-PN is not plugged properly into the other ***samos***®PRO module. Module connecting plug is soiled or damaged. Other ***samos***®PRO module has internal critical error.	Plug the SP-EN-PN in correctly Clean the connecting socket/plug. Repower the system. Check the other ***samos***®PRO modules.
SP-EN-PN is in Critical fault mode. LED PWR ● Green LED LINK/ACT ●/✸ Green LED STATUS ✸ Red (2 Hz)	SP-EN-PN internal device error CPU firmware version does not support ***samos***®PRO gateways.	Switch off the power supply of the ***samos***®PRO system and switch it on again. Check the diagnostics messages with ***samos***®PLAN. Use a CPU with the required firmware version (see section 2.2 “Intended use” on page 10). If the error remains, replace the gateway.

6 Fieldbus gateways

6.1 PROFIBUS DP gateway

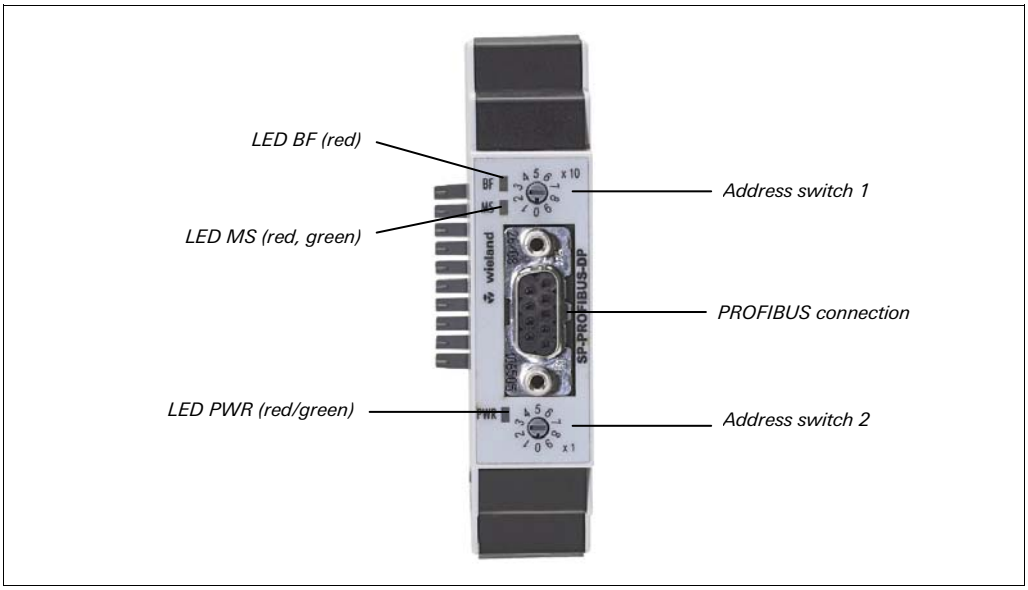
The following ***samos***®PRO gateways can be used for PROFIBUS DP:

- SP-PROFIBUS-DP

6.1.1 Interfaces and operation

Controls and status indicators

Fig. 36: Controls and status indicators SP-PROFIBUS-DP



Tab. 52: Meaning of the status LEDs of the SP-PROFIBUS-DP

LED		Meaning
Symbol description: ○ – LED off, ● Green – LED lights up green, ✸ Red – LED flashes red		
BF	○	Connection to DP master established
	● Red	No bus connection: Fieldbus cable break, address fault or master is not (or no longer) writing to the bus
MS	○	Power up, waiting for bus off
	● Green	Executing
	✸ Green	Idle
	✸Red/Green	Executing, but there is an error at the gateway
	✸Red	1 Hz: Configuration required or in progress 2 Hz: Critical fault on gateway
	● Red	Critical fault on another module
PWR	○	No power supply
	● Green	Power supply switched on, no error
	● Red	Critical fault

Tab. 53: Address switch SP-PROFIBUS-DP

Switch/button	Function
× 10	Address switch 1 10-position rotary switch for setting the module address (tens)
× 1	Address switch 2 10-position rotary switch for setting the module address (ones)

How to set the PROFIBUS DP address via hardware address switches:

➡ Set the PROFIBUS DP address using the hardware address switches on the device front. Then switch the ***samos***®PRO system off and back on again.

How to set the PROFIBUS DP address via software using the *samos*®PLAN:

➡ Set both hardware address switches on the device front to 0.
➡ Open the ***samos***®PLAN and load the hardware configuration including the PROFIBUS DP gateway. Ensure your project is offline.
➡ Click on the **Gateway** button above the main window and select the SP-PROFIBUS-DP to open the gateway configuration dialog.
➡ Click on **Gateway configuration** on the left hand menu. The following dialog appears:

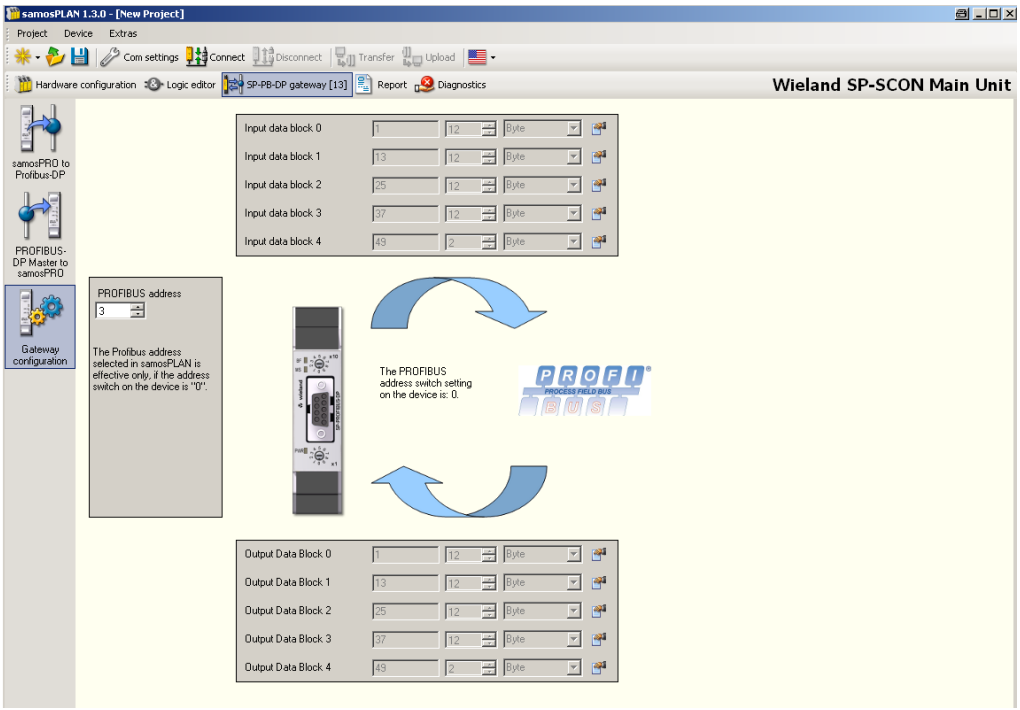


Fig. 37: Setting the PROFIBUS address for the SP-PROFIBUS-DP

➡ Select the PROFIBUS address in the **PROFIBUS address** field.

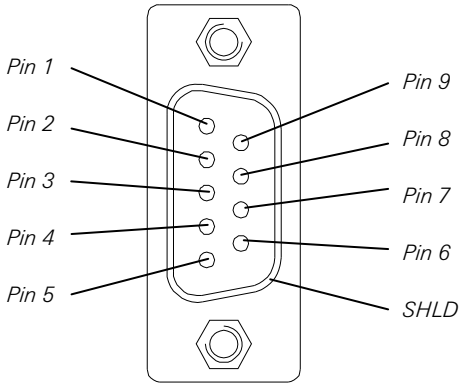
NOTES

- The address that can be set via the hardware address switch ranges from 1 … 99.
- The address that can be set via the ***samos***®PLAN software ranges from 3 … 125.
- The PROFIBUS master cannot overwrite the address.
- A modified address setting only becomes effective after switching off and switching on the ***samos***®PRO system.
- In online mode, you can read the address set on the PROFIBUS DP gateway by clicking on the **Read** button above the **PROFIBUS address** field.

Plug assignment

The connection to the PROFIBUS DP fieldbus is made using a 9 pin Sub-D socket.

Fig. 38: D-Sub socket and plug pin assignments SP-PROFIBUS-DP

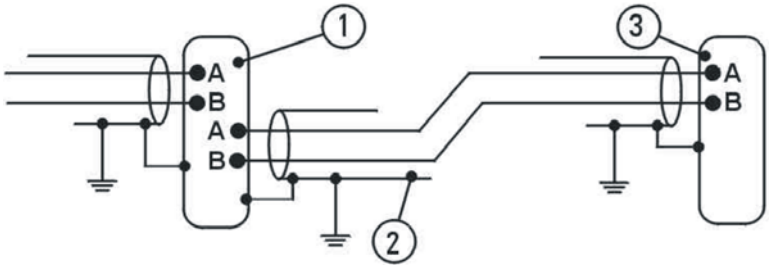


Pin	Description
1	NC
2	NC
3	RxD/TxD-P
4	CNTR-P
5	GND-EXT
6	+5V-EXT
7	NC
8	RxD/TxD-N
9	CNTR-N (GND-EXT)
SHLD	Shield

Bus cable

The bus topology for PROFIBUS DP is a linear structure comprising a screened, twisted 2-core cable with active bus termination at both ends. The possible bus lengths are 100 m at 12 MBit/s up to 1,200 m at 94 KBit/s.

Fig. 39: Bus cable SP-PROFIBUS-DP



Tab. 54: Explanation bus cable SP-PROFIBUS-DP

Position	Description
1	PROFIBUS user grey
2	Shielded bus cable
3	PROFIBUS termination yellow (with integrated terminating resistors)

Cable parameters

The properties of the bus cable are specified in EN 50□170 as cable type A.

Tab. 55: Cable parameters SP-PROFIBUS-DP

Property	Value
Characteristic impedance	135-165 Ω (at a frequency of 3-20 MHz)
Capacitance per unit length	< 30 pF/m
Loop resistance	≤ 110 Ω/km
Core diameter	> 0.64 mm
Core cross-section	> 0.34 mm^2

With these cable parameters, the following maximum physical sizes are possible for a bus segment:

Baud rate (Kbit/s)	Max. cable length (m)
9.6	1200
19.2	1200
93.75	1200
187.5	1000
500	400
1500	200
12000	100

Tab. 56: Maximum cable lengths SP-PROFIBUS-DP

Data transmission rate

The data transmission rate is set automatically.

The maximum baud rate is 12 MBit/s.

6.1.2 Planning

GSD file

In the normal case the SP-PROFIBUS-DP is used on a DP master that looks up the device characteristics in the GSD file.

You will find the GSD file and device icon for PLC interfacing with PROFIBUS support

- in the Internet on the SP-PROFIBUS-DP product page (http://eshop.wieland-electric.com).

Operational data transmitted by the SP-PROFIBUS-DP

The SP-PROFIBUS-DP GSD file provides input/output data blocks (virtual I/O device modules) containing the operational data. These five blocks must be projected in a DP configurator in natural order (1, 2, 3, 4, 5). No other sequence is possible.

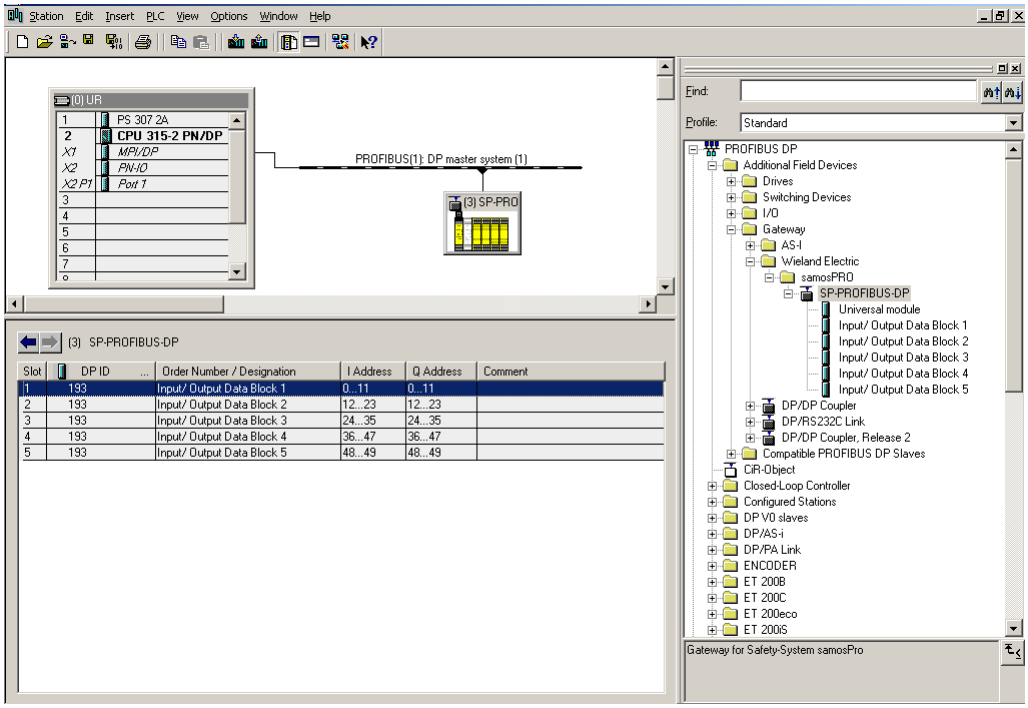


Fig. 40: PROFIBUS DP configuration example in Siemens SIMATIC Manager

NOTES

- Depending on the type of PLC used, further modules may be displayed (e.g.“universal module”). These modules are not needed and should be ignored.
- The data blocks 1-4 contain 12 bytes each, data block 5 contains 2 bytes.

The contents of the data blocks are freely selectable, but are preconfigured in the ***samos***®PLAN configuration software:

Tab. 57: Default content of input data block 1-5 of the SP-PROFIBUS-DP

	Data block 1	**Data block 2**	**Data block 3**	**Data block 4**	**Data block 5**
	Input data	Input data	Input data	Input data	Input data
Byte 0	Input values module 1	Output values module 1	Logic result 1	Not assigned	Not assigned
Byte 1	Input values module 2	Output values module 2	Logic result 2	Not assigned	Not assigned
Byte 2	Input values module 3	Output values module 3	Logic result 3	Not assigned	Not available
Byte 3	Input values module 4	Output values module 4	Logic result 4	Not assigned	
Byte 4	Input values module 5	Output values module 5	Not assigned	Not assigned	
Byte 5	Input values module 6	Output values module 6	Not assigned	Not assigned	
Byte 6	Input values module 7	Output values module 7	Not assigned	Not assigned	
Byte 7	Input values module 8	Output values module 8	Not assigned	Not assigned	
Byte 8	Input values module 9	Output values module 9	Not assigned	Not assigned	
Byte 9	Input values module 10	Output values module 10	Not assigned	Not assigned	
Byte 10	Input values module 11	Output values module 11	Not assigned	Not assigned	
Byte 11	Input values module 12	Output values module 12	Not assigned	Not assigned	
Length	12 bytes	12 bytes	12 bytes	12 bytes	2 bytes

For detailed information about the content of the process image please see section 3.2 “Data transmitted into the network” on page 13.

How to set the start address for the data blocks:

➡ Open the ***samos***®PLAN and load the hardware configuration including the PROFIBUS DP gateway. Ensure your project is offline.

➡ Click on the **Gateway** button above the main window and select the SP-PROFIBUS-DP to open the gateway configuration dialog.

➡ Click on **Gateway configuration** on the left hand menu. The following dialog appears:

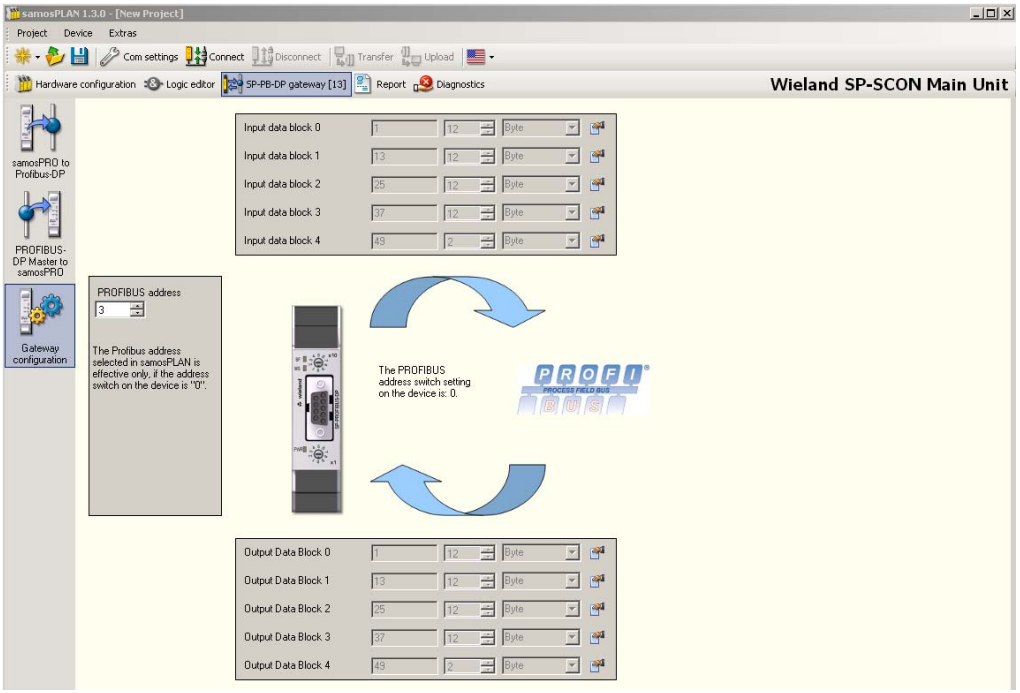


Fig. 41: Gateway configuration dialog for the SP-PROFIBUS-DP

➡ Click on the button to the right of the data block for which you want to change the start address. The following dialog appears:

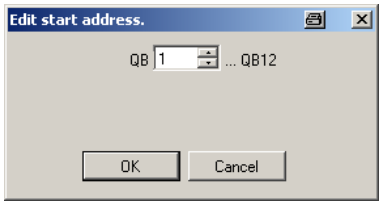


Fig. 42: Edit the data block start address

➡ Type in the desired new start address or use the arrows to change the address. The address set will be checked automatically for plausibility, i.e. it is not possible to configure data blocks with overlapping address ranges.

➡ Click **OK** to accept the new start address.

For further information on how to configure the process image, see chapter 7 “Layout and content of the process image” on page 109 and the ***samos***®PLAN operating instructions (Wieland part no. BA000518).

6.1.3 PROFIBUS configuration of the gateway – how the data is transferred

The following steps need to be taken in order to configure the communication between PLC and gateway.

NOTE

This document does not cover the creation of the PROFIBUS DP network or the rest of the automation system project in the network configuration tool. It is assumed the PROFIBUS project has already been set up in the configuration program, e.g. SIEMENS SIMATIC Manager. Examples refer to configurations performed with SIEMENS SIMATIC manager.

STEP 1 – Install the generic station description file (GSD file)

Before the SP-PROFIBUS-DP can be used as device in the network configuration tool, e.g. SIEMENS SIMATIC Manager, for the first time, the generic station description (GSD) of the gateway must be installed into the hardware catalogue of the tool.

➡ Download the GSD file and device icon from the product page of SP-PROFIBUS-DP (http://eshop.wieland-electric.com).

➡ Follow the instructions in the online help or in the user manual of the PROFINET network configuration tool for installing generic station description files.

Using the SIEMENS SIMATIC Manager – HW Config, the gateway then appears in the hardware catalogue under >>**PROFIBUS DP** > **Additional Field Devices** > **Gateway** > **Wieland** > ***samos***®PRO.

STEP 2 – Add the gateway to the project

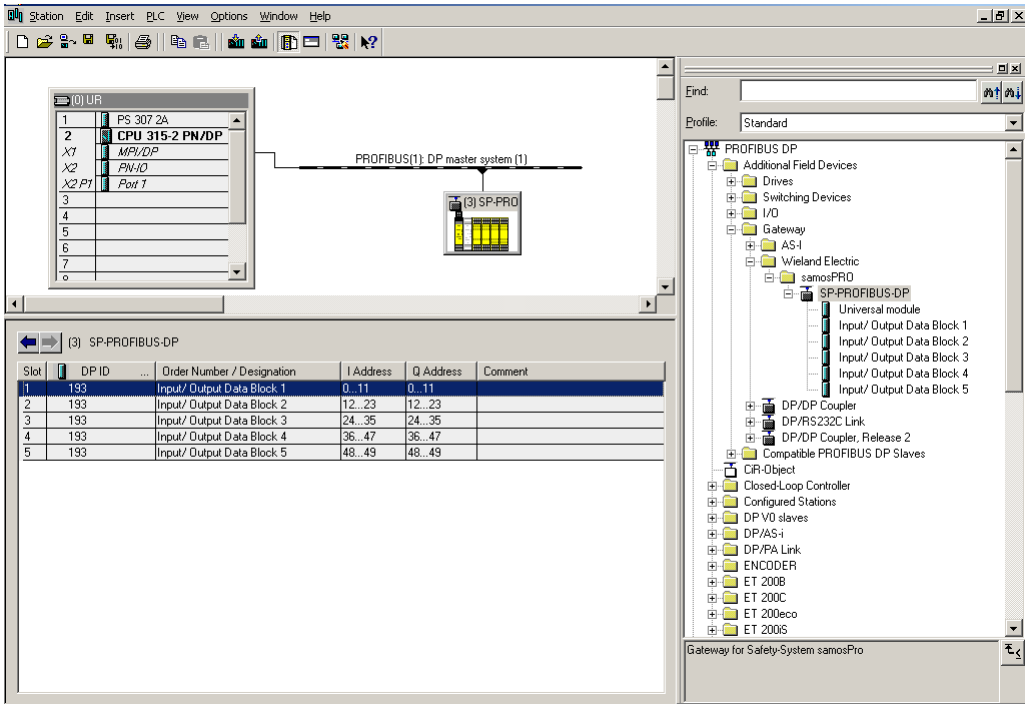
In order to have the ***samos***®PRO system data available in the PLC process image, the gateway must be added to the hardware configuration first. The procedure associated with this depends on the hardware configuration program of the PLC being used. On this topic, please also read the documentation for the corresponding program.

The example below shows how to add the gateway to a SIEMENS SIMATIC Manager project.

In the SIEMENS SIMATIC Hardware Manager, the gateway can be found in the hardware catalogue under >>**PROFIBUS DP** > **Additional Field Devices** > **Gateway** > **Wieland** > ***samos***®PRO.

➡ Drag & drop the device into the PROFIBUS network. Example:

Fig. 43: Gateway in the PROFIBUS HW Config



Diagnostics data SP-PROFIBUS-DP

With the SP-PROFIBUS-DP, diagnostics data are available via PROFIBUS standard DP-V0 diagnostics:

- Standard diagnostics (6 bytes)
- Device related diagnostics: Status messages or manufacturer specific messages

Each ***samos***®PRO module supports a unique module ID. Based on this ID the gateway determines the manufacturer specific diagnostics number. This way it is possible to retrieve module specific diagnostics texts from the GSD file. The content of the diagnostics message is shown in Tab. 58.

Octet	Content	Comment
7	0x09	Header
8	See **Tab. 59**	Module number
9	0	PROFIBUS module slot number. The PROFIBUS gateway supports five slots, which do not represent the physical slots, thus all messages shall be tied to slot 0 (gateway itself).
10 (Bit 0…2)	001 or 010	001 = error coming, 010 = error going
10 (Bit 3…7)	00000…11111	Alarm sequence number, will be incremented for each state change of octet 10 Bit 0…2 (error coming/going)
11	0 … 14	Position of the ***samos***®PRO module raising the diagnostic information. 0 = CPU 1 = 1st I/O module … 13 = 1st gateway 14 = 2nd gateway (relay modules are not counted)
12 … 15	Variable	4 bytes module specific diagnostics data. See Tab. 60.

Tab. 58: Content of the PROFIBUS diagnostics messages

The following table lists the module numbers for the ***samos***®PRO system.

Module number	Module
161	***samos***®PRO controller module (SP-SCON)
162	SP-SDI, SP-SDIO module
163	PROFIBUS gateway (SP-PROFIBUS-DP)
164	CANopen gateway (SP-CANopen)
165	DeviceNet gateway (SP-DeviceNet)
166	Modbus gateway (SP-EN-MOD)
167	Ethernet/IP gateway (SP-EN-IP)
168	Profinet IO gateway (SP-EN-PN)

*Tab. 59: **samos**® PRO module numbers*

The following table matches the module specific diagnostics data (as defined by the GSD) to the appropriate error message.

Module number	Diagnostics bit (X_Unit_Diag_Bit)	Error origin	Error message
1	**0**	CPU	Reserved
	1		Module operating state is Critical Fault
	2		Power supply out of range or EFI communication failure
	3		Reserved
	4		Configuration status of a module within the system changed to invalid
	5		Power supply out of specified range
	6		EFI 1 communication failure
	7		EFI 2 communication failure
	8 … 31		Reserved
2	**0**	I/O module	Reserved
	1		Internal error: Internal tests failed or watchdog test failed or bad process data or self test failure
	2		Reserved
	3		Error history item existing: Access via configuration tool
	4		Configuration status changed to invalid
	5		Output power supply out of range
	6 … 7		Reserved
	8		Input 1-2 dual channel input evaluation: error detected
	9		Input 3-4 dual channel input evaluation: error detected
	10		Input 5-6 dual channel input evaluation: error detected
	11		Input 7-8 dual channel input evaluation: error detected
	12		Status output 1 fast shut off logic control time out
	13		Status output 2 fast shut off logic control time out
	14		Status output 3 fast shut off logic control time out

Tab. 60: PROFIBUS error messages

Module number	Diagnostics bit (X_Unit_Diag_Bit)	Error origin	Error message
	15		Status output 4 fast shut off logic control time out
	16		Input 1 external test signal failure. Check for stuck-at-high or cabling
	17		Input 2 external test signal failure. Check for stuck-at-high or cabling
	18		Input 3 external test signal failure. Check for stuck-at-high or cabling
	19		Input 4 external test signal failure. Check for stuck-at-high or cabling
	20		Input 5 external test signal failure. Check for stuck-at-high or cabling
	21		Input 6 external test signal failure. Check for stuck-at-high or cabling
	22		Input 7 external test signal failure. Check for stuck-at-high or cabling
	23		Input 8 external test signal failure. Check for stuck-at-high or cabling
	24		Output 1 stuck-at-high error
	25		Output 1 stuck-at-low error
	26		Output 2 stuck-at-high error
	27		Output 2 stuck-at-low error
	28		Output 3 stuck-at-high error
	29		Output 3 stuck-at-low error
	30		Output 4 stuck-at-high error
	31		Output 4 stuck-at-low error
3	**0**	PROFIBUS gateway	Reserved
	1		Internal error: Internal tests failed
	2		Reserved
	3		Reserved
	4		Configuration status changed to invalid
	5		Reserved
	6		Reserved
	7 ... 31		Reserved
4	**0**	CANopen gateway	Reserved
	1		Internal error: Internal tests failed
	2		Reserved
	3		Reserved
	4		Configuration status changed to invalid
	5		Reserved
	6		Reserved
	7 ... 31		Reserved
5	**0**	DeviceNet gateway	Reserved
	1		Internal error: Internal tests failed
	2		Reserved
	3		Reserved
	4		Configuration status changed to invalid
	5		Reserved
	6		Reserved
	7 ... 31		Reserved
6	**0**	Modbus gateway	Reserved
	1		Internal error: Internal tests failed
	2		Reserved
	3		Reserved
	4		Configuration status changed to invalid
	5		Reserved
	6		Reserved
	7 ... 31		Reserved
7	**0**	Ethernet/IP gateway	Reserved
	1		Internal error: Internal tests failed

Module number	Diagnostics bit (X_Unit_Diag_ Bit)	Error origin	Error message
	2		Reserved
	3		Reserved
	4		Configuration status changed to invalid
	5		Reserved
	6		Reserved
	7 ... 31		Reserved
8	**0**	ProfiNet gateway	Reserved
	1		Internal error: Internal tests failed
	2		Reserved
	3		Reserved
	4		Configuration status changed to invalid
	5		Reserved
	6		Reserved
	7 ... 31		Reserved
9	**0**	Other module	Reserved
	1		Internal error: Internal tests failed
	2		Reserved
	3		Reserved
	4		Configuration status changed to invalid
	5 ... 31		Reserved
12	**0**	Other module	Reserved
	1		Internal error: Internal tests failed
	2		Reserved
	3		Reserved
	4		Configuration status changed to invalid
	5 ... 31		Reserved

6.1.4 Diagnostics and troubleshooting

For information how to perform diagnostics on the ***samos***®PRO system please refer to the operating instructions for the ***samos***®PLAN software (Wieland part no. BA000518).

Tab. 61: Troubleshooting for the SP-PROFIBUS-DP

Error	Cause	Possible remedy
Symbol description: ○ – LED off, ● Green – LED lights up green, ✸ Red – LED flashes red		
The ***samos***®PLAN tool does not connect to the ***samos***®PRO gateway module	SP-PROFIBUS-DP has no power supply.	Establish the power supply. Check the communication settings in the ***samos***®PLAN.
SP-PROFIBUS-DP does not supply any data. LED PWR ● Green LED BF ○ Off LED MS ✸ Red (1 Hz)	Configuration required. Configuration download is not completed.	Configure the SP-PROFIBUS-DP and download the configuration to the device. Wait until the configuration download has been completed.
SP-PROFIBUS-DP does not supply any data. LED PWR ● Green LED BF ○ Off LED MS ● Green	No data set is activated.	Activate at least one data set.
SP-PROFIBUS-DP does not supply any data. LED PWR ● Green LED BF ○ Off/● Red LED MS ✸ Green (1 Hz)	SP-PROFIBUS-DP is in Idle mode	CPU/application is stopped. Start CPU (change into Run mode)
SP-PROFIBUS-DP does not supply any data. LED PWR ● Green LED BF ○ Off LED MS ● Green	PROFIBUS master is in stop mode	Set PROFIBUS master into Run mode
SP-PROFIBUS-DP functioned correctly after configuration, but suddenly no longer supplies data. LED PWR ● Green LED BF ● Red LED MS ✸ Red/Green	SP-PROFIBUS-DP PROFIBUS hardware address is changed. PROFIBUS cable is disconnected.	Check PROFIBUS address setting at hardware address Check PROFIBUS cable. Check PROFIBUS master.
SP-PROFIBUS-DP is in critical fault. LED PWR ● Green LED BF ● Red LED MS ✸ Red (2 Hz)	SP-PROFIBUS-DP internal device error CPU firmware version does not support ***samos***®PRO gateways.	Switch off the power supply of the ***samos***®PRO system and switch it on again. Check the diagnostics messages with ***samos***®PLAN. Use a CPU with the required firmware version (see section 2.2 “Intended use” on page 10). If the error remains, replace the gateway.
SP-PROFIBUS-DP/ ***samos***®PRO System is in critical fault LED PWR ● Red LED BF ○ Off LED MS ● Red	SP-PROFIBUS-DP is not plugged properly into the other ***samos***®PRO module. Module connecting plug is soiled or damaged. Other ***samos***®PRO module has internal critical error.	Plug the SP-PROFIBUS-DP in correctly. Clean the connecting socket/plug. Repower the system. Check the other ***samos***®PRO modules.

6.2 CANopen gateway

The following samos®PRO gateways can be used for CANopen:

- SP-CANopen

6.2.1 Interfaces and operation

Control and status indicators

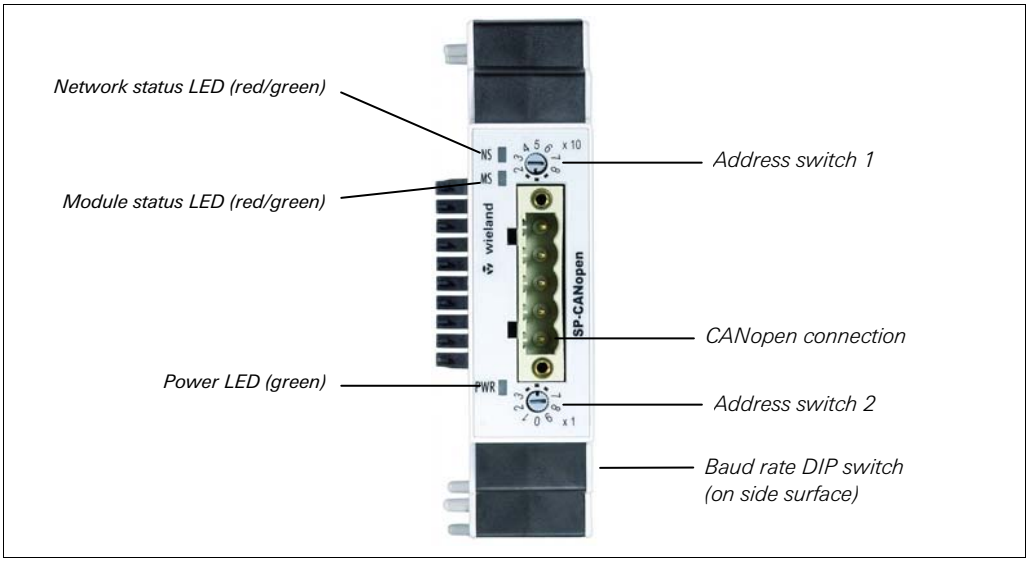


Fig. 44: Controls and status indicators SP-CANopen

LED		Meaning
Symbol description: ○ – LED off, ● Green – LED lights up green, ✷ Red – LED flashes red		
PWR Power	○	No power supply
	● Green	Operational, power supply switched on
	● Red	System error
NS (Network Status)	○	CANopen status: Stopped (except node guarding and heartbeat, if active)
	● Green	CANopen status: Operational (PDO + SDO data exchange)
	✷ Green	CANopen status: Pre-operational (SDO data exchange only)
	● Red	CAN busoff (hardware problem CAN physical layer) or Error passive
	✷ Red (1 Hz)	Node guarding failed (NMT master does not monitor slave anymore) or Heartbeat consumer failure
MS (Module Status)	○	Power up
	● Green	Executing, SBUS+ and PDO status: all “Good”
	✷ Green	Idle (cable not attached or node guarding failed)
	✷ Red/Green	Executing, SBUS+ and PDO status: any is “Bad”
	● Red	Critical fault, caused by emergency bit
	✷ Red (1 Hz)	Configuration required or in progress
	✷ Red (2 Hz)	Critical fault, caused by gateway itself

Tab. 62: Meaning of the status LEDs of the SP-CANopen

For diagnostics see section 6.2.14 “Diagnostics and troubleshooting” on page 107.

How to set the CANopen address via hardware address switches:

➡ Set the CANopen address using the hardware address switches on the device front. Then switch the samos®PRO system off and back on again.

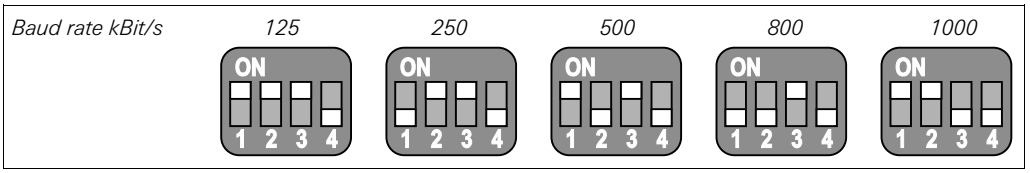
Tab. 63: Address switch SP-CANopen

Switch	Function
× 10	Address switch 1 10 position rotary switch for setting the module address (tens)
× 1	Address switch 2 10 position rotary switch for setting the module address (ones)

How to set the baud rate via hardware DIP switches

➡ Set the baud rate using the DIP switches on the device. Then switch the samos®PRO system off and back on again.

Fig. 45: DIP switch settings on the SP-CANopen



Tab. 64: DIP switch settings on the SP-CANopen

Baud rate in kBit	DIP 1	DIP 2	DIP 3	DIP 4
125	On	On	On	Off
250	Off	On	On	Off
500	On	Off	On	Off
800	Off	Off	On	Off
1000	On	On	Off	Off

NOTES

- All other DIP switch settings will set the Baud rate to 125 kBit.
- If the *address switches* on the device are set to “00”, the DIP switch settings are ignored and the baud rate setting in the samos®PLAN is used.

How to set the CANopen address and baud rate via software using the samos®PLAN:

➡ Set the two hardware address switches on the device front to “00”.

➡ Open the samos®PLAN and load the hardware configuration including the CANopen gateway. Ensure that your project is offline.

➡ Click on the **Gateways** button above the main window and select the SP-CANopen or double click the SP-CANopen in the hardware configuration view to open the gateway configuration dialog.

➡ Click on **Gateway configuration** on the left hand menu. The following dialog appears:

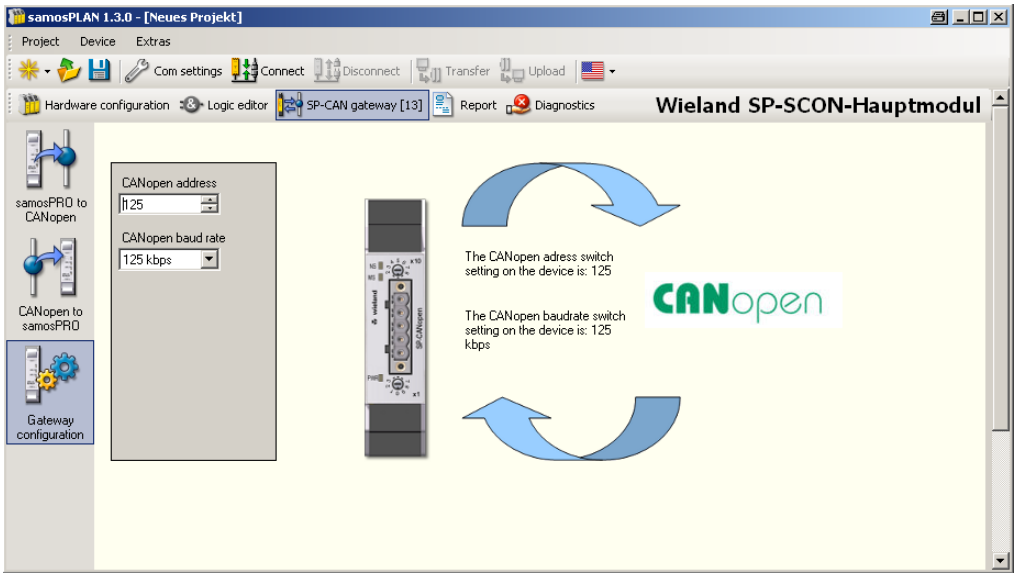


Fig. 46: Setting the CANopen address for the SP-CANopen

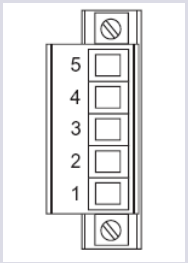
➡ Select the CANopen address in the **CANopen address** field.
➡ Select the baud rate in the **CANopen baud rate** field.
➡ Click **Connect** to go online and transfer the configuration to the samos®PRO system.

NOTES

- The address that can be set via the hardware address switch ranges from 1 … 99.
- The address that can be set via the samos®PLAN software ranges from 1 … 127.
- The CANopen master cannot overwrite the address.
- If the CANopen address and the baud rate are set using the samos®PLAN, the settings will become effective immediately after transferring the configuration (i.e. without switching off and switching on the samos®PRO system). Exception: If the system is in bus off state, a power cycle is required.

Plug assignment

The connection to the CANopen fieldbus is made using a 5 pin open style connector.



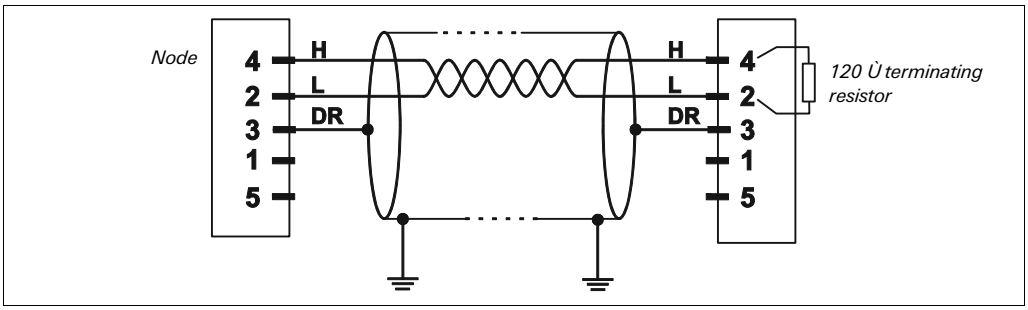
Pin	Description	
5	–	–
4	H CAN_H	CAN High
3	DR (CAN_SHLD)	Screen connection (optional)
2	L CAN_L	CAN Low
1	–	–

Fig. 47: Open style connector and assignment SP-CANopen

Bus cable

CANopen is based on a linear topology with screened, twisted pair 2 core cable and terminating resistors at both ends of the bus. The screen is connected to ground at both ends. The transmission rate, depending on the network length, is between 125 kBit/s and 1000 kBit/s. The possible network lengths are 20 m at 1000 kBit/s up to 500 m at 125 KBit/s.

Fig. 48: CANopen bus cable



NOTE It is not necessary to connect a voltage supply (pin 1/5) to the SP-CANopen.

The following maximum physical sizes are possible for the network:

Tab. 65: Maximum cable lengths SP-CANopen

Bitrate (kBit/s)	Max. cable length (m)
125	500
250	250
500	100
800	40
1000	20

EDS file

The device characteristics are described using the Electronic Data Sheet (EDS), which every standard bus configuration tool uses.

You will find the EDS file and device icon for PLC interfacing

- in the Internet on the SP-CANopen product page (http://eshop.wieland-electric.com).

6.2.2 CANopen configuration of the gateway — how the data are transferred

NOTE This document does not cover the creation of the CANopen network or the rest of the automation system project in the network configuration tool. It is assumed the CANopen project has already been set up in the configuration program, e.g. 3S Software CoDeSys 2.x. Examples refer to configurations performed with CoDeSys 2.3.

The following steps need to be taken in order to configure the communication between PLC and gateway.

Step 1: Install the electronic data sheet (EDS file)

Before the SP-CANopen can be used as device in the network configuration tool, e.g. CoDeSys 2.3, for the first time, the electronic data sheet (EDS file) of the gateway must be installed into the hardware catalogue of the tool.

➡ Download the EDS file and device icon from the product page of SP-CANopen (http://eshop.wieland-electric.com).

➡ Follow the instructions in the online help or in the user manual of the CANopen network configuration tool for installing EDS files.

Example – How to install the EDS file using CoDeSys 2.3:

➡ Open the **PLC Configuration** editor window.

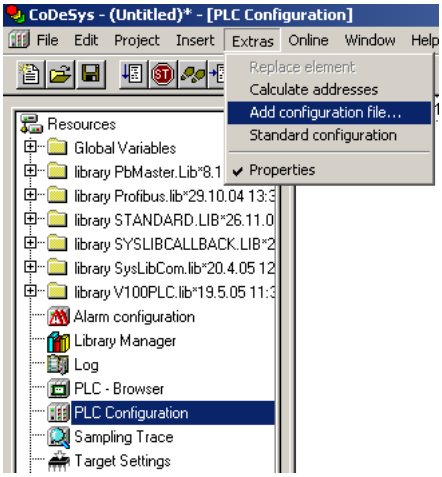


Fig. 49: CoDeSys PLC Configuration editor window

➡ From the **Extras** menu choose the command **Add configuration file...**. A file selection dialog opens.

➡ Select the EDS file for the SP-CANopen and click on the **Open** button.

Step 2: Add the gateway to the controller

In order to have the samos®PRO system data available in the PLC process image, the gateway must be added to the hardware configuration first. The procedure associated with this depends on the hardware configuration program of the PLC being used. On this topic, please also read the documentation for the corresponding program.

Example – How to add the SP-CANopen using CoDeSys 2.3:

➡ Open the **PLC Configuration** editor window and select the controller.

➡ Right click on the controller to open the context menu or open the **Insert** menu.

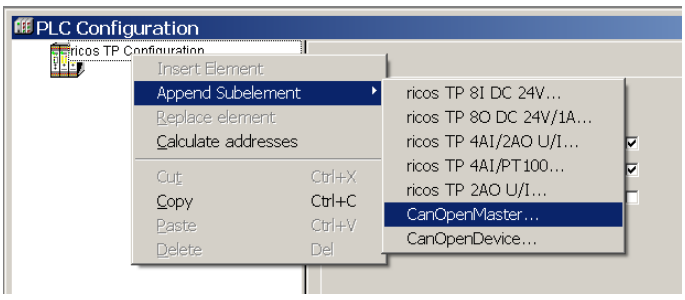


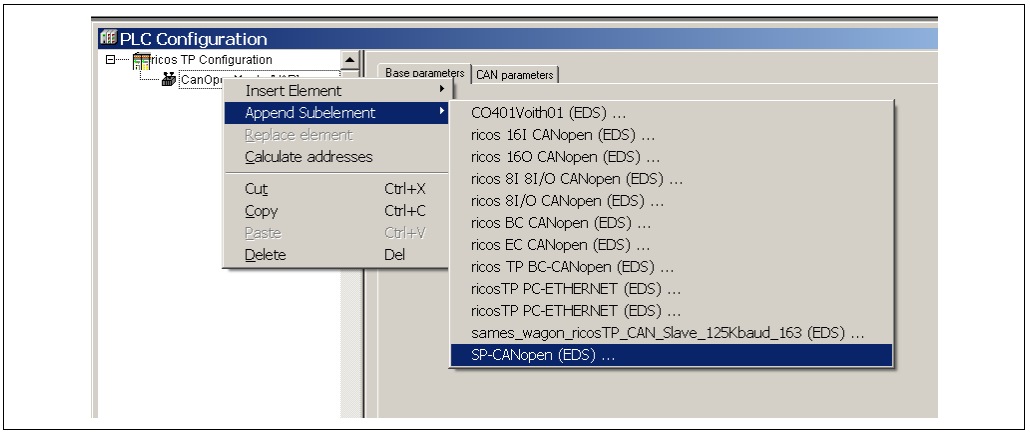
Fig. 50: Appending a CanMaster using CoDeSys 2.3

➡ In either menu, under **Append Subelement**, choose **CanMaster...**. A CanMaster is added to the controller.

➡ Now select the CanMaster.

➡ Right click on the CanMaster to open the context menu or open the **Insert** menu.

Fig. 51: Appending the SP-CANopen using CoDeSys 2.3



➡ In either menu, under **Append Subelement**, choose **SP-CANopen00000 (EDS) ...** to append the SP-CANopen to the CanMaster.

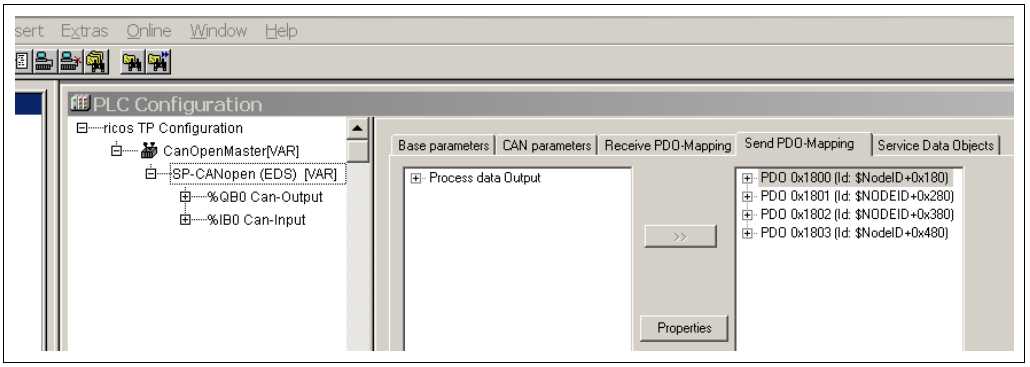
Step 3: Select and configure the process data objects (PDOs)

After adding the device to the automation network it is required to configure which of the process data objects will be used and how they will be transmitted.

Example – How to define the PDO transmission type using CoDeSys 2.3:

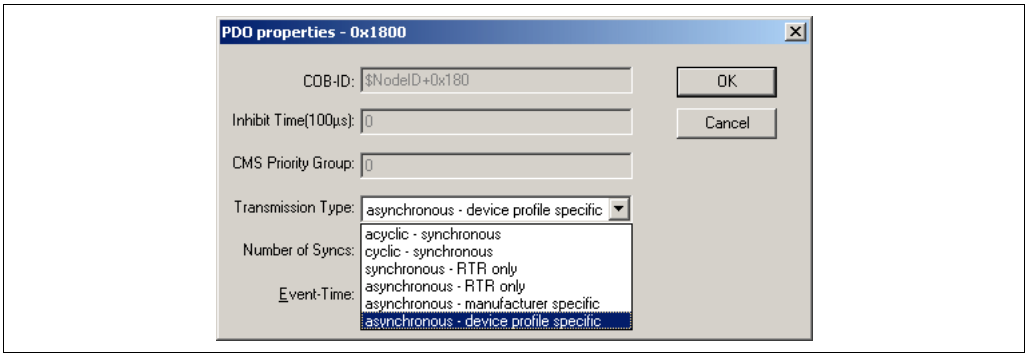
➡ In the **PLC Configuration** editor window select the SP-CANopen. Then click on the **Send PDO-Mapping** tab on the right side.

Fig. 52: PDO configuration using CoDeSys 2.3



➡ Select one of the displayed PDOs (e.g. PDO 1) and click on the **Properties** button. The **PDO properties** dialog opens.

Fig. 53: PDO properties dialog in CoDeSys 2.3



➡ Select the desired **Transmission Type** for the PDO from the selection list, enter the **Event-Time** in ms. and click **OK**. For more information please refer to section "TxPDO transmission types" on page 100 and to the manual for your CANopen system configuration software.

➡ Repeat this for the other Send-PDOs as well as for the Receive-PDOs.

6.2.3 CANopen configuration of the gateway — which data are transferred

Each CANopen device stores its data in *objects* that are listed in the *object dictionary*. *Service data objects* (SDOs) mainly contain the CANopen configuration data, while the process data are stored in *process data objects* (PDOs). *Communication objects* are used to read and write these SDOs and PDOs as well as to control the devices. A detailed description of the different objects will be given in the following sections.

Predefined connection set (PCS)

The predefined connection set provides a simple CAN identifier structure. The SP-CANopen gateway provides *communication objects* that can be addressed or sent using these CAN identifiers.
The PCS comprises 2 broadcast objects (NMT and SYNC) and a total of 12 peer-to-peer objects. Each of these objects has a unique 11 bit CAN identifier that consists of a function code and a device address. The device address for the broadcast objects is 0, for the other objects 1…127.

Tab. 66: CAN identifier structure

Bit number										
10	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Function code				Device address						

Tab. 67: PCS communication objects

Object	**CAN identifier**	**Meaning**
Broadcast objects		
NMT	00h	Network management
SYNC	80h	Sync message
Peer-to-peer objects		
EMERGENCY	081h…0FFh	Status message
TxPDO1	181h…1FFh	Send process data object 1
RxPDO1	201h…27Fh	Receive process data object 1
TxPDO2	281h…2FFh	Send process data object 2
RxPDO2	301h…37Fh	Receive process data object 2
TxPDO3	381h…3FFh	Send process data object 3
RxPDO3	401h…47Fh	Receive process data object 3
TxPDO4	481h…4FFh	Send process data object 4
RxPDO4	501h…57Fh	Receive process data object 4
TxSDO	581h…5FFh	Send service data object
RxSDO	601h…67Fh	Receive service data object
NMT-ErrorControl	701h…77Fh	Node guarding

Each object starts with its CAN identifier, followed by the RTR bit (Remote Transmission Request), followed by the Data Length Code (DLC), followed by 0 to 8 data bytes. The DLC (4 bits) indicates the number of data bytes.

6.2.4 NMT – network management

The NMT broadcast object is used to start, stop or initialise CANopen devices. For this purpose one device in the CANopen network must take over the role of the NMT master. Usually this is the PLC. All other devices are regarded as NMT slaves. NMT services are broadcast services, for which the slaves do not generate a reply.
All NMT objects start with the CAN-ID 00h.

Broadcast service for an NMT slave with address N:

Tab. 68: Network management for one NMT slave with address N

CAN-ID	DLC	DATA							
00h	2	OP	N						

Broadcast service for all NMT slaves:

Tab. 69: Network management for all NMT slaves

CAN-ID	DLC	DATA							
00h	2	OP	0						

OP	NMT operation	Explanation
80h	Change to “Pre-Operational”	After booting an NMT slave automatically enters the Pre-Operational state. In this state, communication is allowed via SDO, but not via PDO. The NMT slave can be changed from another state to this state.
01h	Change to “Operational”	The Operational state is reached from the Pre-Operational state. In this state communication via PDO is possible and the CANopen slave reacts to Sync commands. **Note:** On transition to NMT Operational state, each slave sends a TxPDO with the transmission type = 255, so that the NMT master is informed of the current input configuration.
02h	Change to “Prepared/Stopped”	Communication via SDO or PDO is not possible in this state, and there is also no reaction to SYNC commands.
81h	Change to “Reset Node”	Triggers a re-initialisation of the CANopen functionality in the NMT slave.
82h	Change to “Reset Communication”	Triggers a re-initialisation of the CANopen functionality in the NMT slave; the toggle bit for Node guarding is set to 0.

Example for resetting all communication:

The following NMT object (CAN-ID = 00h) has 2 data bytes (DLC = 2). Data byte 1 contains the “Reset communication” command (82h) while data byte 2 addresses this command to all devices in the CANopen network (address = 0):

Tab. 70: Example NMT object for resetting all communication

CAN-ID	DLC	DATA							
00h	2	82h	0						

6.2.5 SYNC

The SYNC command causes all TxPDOs for a CANopen slave to be sent. It is therefore possible to poll the slave using SYNC.

Tab. 71: Polling inputs using SYNC

CAN-ID	DLC	DATA							
80h	0								

The slave sends all input values when this message arrives. All TxPDOs are sent.
To ensure the slave automatically sends the actual input values on the receipt of a SYNC command, the transmission type for the related PDOs must be set to 1 (cyclic, synchronous). In addition, the operating mode of the device must be “Operational”.
It is possible to change the transmission type for the TxPDOs using the SDOs 1800…1803 (PDO communication parameter) and the sub-object 2. Allowed are the following types:

- acyclic/synchronous = 0
- cyclic/synchronous = 1…240
- acyclic by device profile = 255 (only for TxPDO 1…4, digital inputs)

6.2.6 Emergency

A CANopen slave with the address N sends an emergency message to inform the other devices about an error state.

CAN-ID	DLC	DATA							
80h + N	8	ErrL	ErrH	Err-Reg	M1	M2	M3	M4	M5
ErrL, ErrH		Emergency error code, 16 bit low byte/high byte 7001h … 7003h: Generic error							
Err-Reg		Error register, CANopen object SDO 1001h							
M1		Module number of the module causing the error (see Tab. 73)							
M2 … M5		4 bytes module specific status bits (see Tab. 73). Active bits are high (="1")							

Tab. 72: Emergency messages

The following table matches the module specific diagnostics data to the appropriate error message.

Module number (M1)	Diagnostics bit (M2 … M5)	Emergency origin	Emergency message
01	**00**	CPU	Reserved
	01		Internal error: Internal tests failed
	02		Power supply out of range or EFI communication failure
	03		Reserved
	04		Configuration of a module within the system is incompatible or invalid
	05		Power supply out of specified range
	06		EFI1 communication failure
	07		EFI2 communication failure
	08 … 31		Reserved
02	**00**	SP-SDIO/SP-SDI	Reserved
	01		Internal error: Internal tests failed
	02		Reserved
	03		Reserved
	04		Configuration is incompatible or invalid
	05		Output power supply out of range
	06		Reserved
	07		Reserved
	08		Input 1-2 dual channel input evaluation: error detected
	09		Input 3-4 dual channel input evaluation: error detected
	10		Input 5-6 dual channel input evaluation: error detected
	11		Input 7-8 dual channel input evaluation: error detected
	12 … 15		Reserved
	16		Input 1 external test signal failure. Check for stuck-at-high or cabling
	17		Input 2 external test signal failure. Check for stuck-at-high or cabling
	18		Input 3 external test signal failure. Check for stuck-at-high or cabling
	19		Input 4 external test signal failure. Check for stuck-at-high or cabling
	20		Input 5 external test signal failure. Check for stuck-at-high or cabling
	21		Input 6 external test signal failure. Check for stuck-at-high or cabling

Tab. 73: CANopen emergency messages

Module number (M1)	Diagnostics bit (M2 … M5)	Emergency origin	Emergency message
	22		Input 7 external test signal failure. Check for stuck-at-high or cabling
	23		Input 8 external test signal failure. Check for stuck-at-high or cabling
	24		Output 1 stuck-at-high error
	25		Output 1 stuck-at-low error
	26		Output 2 stuck-at-high error
	27		Output 2 stuck-at-low error
	28		Output 3 stuck-at-high error
	29		Output 3 stuck-at-low error
	30		Output 4 stuck-at-high error
	31		Output 4 stuck-at-low error
03	**00**	PROFIBUS gateway	Reserved
	01		Internal error: Internal tests failed
	02		Reserved
	03		Reserved
	04		Configuration is incompatible or invalid
	05 … 31		Reserved
04	**00**	CANopen gateway	Reserved
	01		Internal error: Internal tests failed
	02		Reserved
	03		Reserved
	04		Configuration is incompatible or invalid
	05 … 31		Reserved
05	**00**	DeviceNet gateway	Reserved
	01		Internal error: Internal tests failed
	02		Reserved
	03		Reserved
	04		Configuration is incompatible or invalid
	05 … 31		Reserved
06	**00**	Modbus gateway	Reserved
	01		Internal error: Internal tests failed
	02		Reserved
	03		Reserved
	04		Configuration is incompatible or invalid
	05 … 31		Reserved
07	**00**	EtherNet/IP gateway	Reserved
	01		Internal error: Internal tests failed
	02		Reserved
	03		Reserved
	04		Configuration is incompatible or invalid
	05 … 31		Reserved
08	**00**	PROFINET gateway	Reserved
	01		Internal error: Internal tests failed
	02		Reserved
	03		Reserved
	04		Configuration is incompatible or invalid
	05 … 31		Reserved

Module number (M1)	Diagnostics bit (M2 … M5)	Emergency origin	Emergency message
09	**00**	Other gateway	Reserved
	01		Internal error: Internal tests failed
	02		Reserved
	03		Reserved
	04		Configuration is incompatible or invalid
	05 … 31		Reserved
0C	**00**	Other gateway	Reserved
	01		Internal error: Internal tests failed
	02		Reserved
	03		Reserved
	04		Configuration is incompatible or invalid
	05 … 31		Reserved
10	**00**	Other gateway	Reserved
	01		Internal error: Internal tests failed
	02		Reserved
	03		Reserved
	04		Configuration is incompatible or invalid
	05 … 31		Reserved
20 … 3F	**00**	Other module	Reserved
	01		Internal error: Internal tests failed
	02		Reserved
	03		Reserved
	04		Configuration is incompatible or invalid
	05 … 31		Reserved

NOTE

The diagnostics bit assignment for M2 to M5 is as follows:

Bit 0	Bit 1	…	Bit 7	Bit 8	…	Bit31
M5.0	M5.1	…	M5.7	M4.0	…	M2.7

6.2.7 Node guarding

An NMT master (e.g. a PLC with integrated CANopen master) uses the NMT error control object to detect a failure of an NMT slave with the address N. The NMT slave must reply within the node guarding time to the request from the NMT master. The node guarding time must be monitored by the NMT master.
The NMT master sends a CAN message with identifier <700h + node ID> and RTR bit (remote transmission request).
NMT master request:

CAN-ID	RTR	DLC	DATA							
700h + N	**1**	**0**								

Tab. 74: Request from NMT master:

The slave (e.g. the SP-CANopen) then sends a status byte Byte1 with the following content:

Slave response:

CAN-ID	DLC	DATA							
700h + N	**1**	Byte1							

Tab. 75: Response from slave:

Tab. 76: Remote transmission request

Bit	Meaning	
7	Toggle bit, changes value between two sequential requests	
6…0	NMT status	4 = Stopped 5 = Operational 127 = Pre-operational

Bootup

On bootup, the gateway sends a boot-up message with the CAN ID 700h+N, DLC = 1 and Byte 1 = 0.

Heartbeat producer

If the gateway is configured as heartbeat producer (i.e. if SDO 1017 contains a value for the producer heartbeat time, see Tab. 84 “Supported SDOs” on page 97), it sends a cyclic message with the CAN ID 700h+N, DLC = 1 and Byte 1 = 05h. The toggle bit (Bit 7) is always 0.

Heartbeat consumer

If the gateway is configured as heartbeat consumer (i.e. if SDO 1016.1 contains a value for the consumer heartbeat time, see Tab. 84 “Supported SDOs” on page 97), at least one node guarding message must be received within the configured consumer heartbeat time (typically from an NMT master).

6.2.8 PDO communication

Process data objects (PDOs) are the real time objects of the CANopen field bus. They are sent without protocol overhead, i.e. no confirmation is sent from the receiver.
The SP-CANopen provides four transmit process data objects (TxPDO) containing the operational data to be sent into the network and four receive process data objects (RxPDO) containing the operational data received from the network.
CANopen objects are addressed via 11-bit CAN identifiers. As a default, the used CAN identifier for each object derives from the object type and the configured CANopen device address. The CAN identifiers used for the PDOs can be changed using the SDOs 1400 to 1403 for the RxPDOs and SDOs 1800 to 1803 for the TxPDOs (“PDO linking”).

NOTES

Each process data object contains 8 bytes.

The contents of the process data objects are freely selectable, but are preconfigured in the samos®PLAN configuration software as follows:

Tab. 77: Default content of the transmit process data objects (TxPDOs) of the SP-CANopen

	PDO#1	PDO#2	PDO#3	PDO#4
	Input data set 1	**Input data set 2**	**Input data set 3**	**Input data set 4**
Byte 0	Logic result 0	Input values module 5	Output values module 1	Output values module 9
Byte 1	Logic result 1	Input values module 6	Output values module 2	Output values module 10
Byte 2	Logic result 2	Input values module 7	Output values module 3	Output values module 11
Byte 3	Logic result 3	Input values module 8	Output values module 4	Output values module 12
Byte 4	Input values module 1	Input values module 9	Output values module 5	Gateway direct output values 0
Byte 5	Input values module 2	Input values module 10	Output values module 6	Gateway direct output values 1
Byte 6	Input values module 3	Input values module 11	Output values module 7	Gateway direct output values 2
Byte 7	Input values module 4	Input values module 12	Output values module 8	Gateway direct output values 3

For detailed information about the content of the process image please see section 3.2 “Data transmitted into the network” on page 13.
For further information on how to configure the process image, see chapter 7 “Layout and content of the process image” on page 109 and the samos®PLAN operating instructions (Wieland Electric doc no. BA000518).

NOTES

- The process data can also be written and read using service data objects SDO 6000 and SDO 6200 (see section 6.2.9 “SDO communication” on page 96). The simple SDO access is recommended for diagnostic purposes. In normal operation the faster PDO communication should be used.
- After startup or after a configuration change has been performed (either via the CANopen master or via the samos®PLAN), the CANopen gateway’s MS LED flashes red/green until an initial transmit/receive PDO or SDO 6000/SDO 6200 data exchange on the CANopen network has happened.

TxPDO 1…4

A transmit PDO transports data from the CANopen gateway to a CANopen device.

Tab. 78: TxPDO 1…4

CAN ID	DLC	Data							
181-1FF	**8**	B1	B2	B3	B4	B5	B6	B7	B8
281-2FF	**8**	B9	B10	B11	B12	B13	B14	B15	B16
381-3FF	**8**	B17	B18	B19	B20	B21	B22	B23	B24
481-4FF	**8**	B25	B26	B27	B28	B29	B30	B31	B32

B1…B32: CAN telegram bytes as mapped into the network input data using the samos®PLAN software (see 7.3 “Customizing the operational data (***samos***®PRO to Network)” on page 110).
The gateway sends one or more TxPDOs if at least one of the following occurs:

- At least one input or output byte has changed its value and the transmission type for the TxPDO containing this byte has the value 255.
- At least one input or output byte has changed its value *and* the gateway receives a Sync command *and* at least one TxPDO has the transmission type 0.
- If the transmission type is n = 1…240, n Sync commands are required for the TxPDO to be sent.
- The transmission type for a TxPDO is 254 or 255 and the event timer (SDO 1800,5 for TxPDO1) has a value N > 0. In this case this TxPDO is sent every N ms.
- A TxPDO can also be polled via remote transmission request (RTR). This requires a CAN telegram to the gateway containing the CAN ID of the desired TxPDO with DLC = 0 and RTR = 1.

For all transmission methods the device operating mode must be set to “Operational” (see Tab. 69 “Network management for all NMT slaves” on page 90).

RxPDO 1…4

A receive PDO transports data from a CANopen device to the CANopen gateway.

Tab. 79: RxPDO 1…4

CAN ID	DLC	Data							
201-1FF	**8**	B1	B2	B3	B4	B5	B6	B7	B8
301-2FF	**8**	B9	B10	B11	B12	B13	B14	B15	B16
401-3FF	**8**	B17	B18	B19	B20	B21	B22	B23	B24
501-4FF	**8**	B25	B26	B27	B28	B29	B30	B31	B32

B1…B32: CAN telegram bytes as mapped into the gateway input data using the samos®PLAN software.

The transmission type 255 is pre-set for all RxPDOs. This means that the gateway forwards the received RxPDO data immediately to the controller module. This setting cannot be changed.

6.2.9 SDO communication

SDOs are Service Data Objects. They contain a wide variety of data. These include, among other items, data on the planning or input and output data. Unlike the PDO communication, each reception of an SDO is answered at protocol level, i.e. the receiving device sends a confirmation.
In this CANopen PCS implementation the following protocols are supported:

- SDO download expedited (write SDO)
- SDO upload expedited (read SDO)
- Upload SDO segment protocol (segmented reading of an SDO)

SDO download expedited (write SDO)

The client sends a request message to server N. The 16 bit index and the sub-index for the SDO to be written are coded in this message. In addition, the request also contains 4 data bytes with the data to be written.

Tab. 80: Write SDO

CAN ID	DLC	Data							
600h + N	**8**	23h	SDO_L	SDO_H	SUB	Byte 1	Byte 2	Byte 3	Byte 4

SDO_L = SDO index low byte
SDO_H = SDO index high byte
SUB = SDO sub-index
The server then replies with a confirmation message:

Tab. 81: Write SDO confirmation

CAN ID	DLC	Data							
580h + N	**8**	60h	SDO_L	SDO_H	SUB	Byte 1	Byte 2	Byte 3	Byte 4

Byte 1 to 4 in the write confirmation message contain zeros.

SDO upload expedited (Read SDO)

The client requests the contents of an SDO with a request message to server N. The 16 bit index and the sub-index for the SDO to be read are coded in this message. Byte 1 to 4 in the read request message contain zeros.

Tab. 82: Read SDO

CAN ID	DLC	Data							
600h + N	**8**	40h	SDO_L	SDO_H	SUB	Byte 1	Byte 2	Byte 3	Byte 4

The server will reply with the following message. Byte 1 to 4 contain the value of the requested object.

Tab. 83: Read SDO confirmation

CAN ID	DLC	Data							
580h + N	**8**	42h	SDO_L	SDO_H	SUB	Byte 1	Byte 2	Byte 3	Byte 4

CANopen data types UDINT and UINT

In order to transfer data types UDINT or UINT, the data must be in Intel format. E.g. the 32 bit value 12345678h must be transferred in the data bytes 5, 6, 7 and 8 in the following order: [5] = 78, [6] = 56, [7] = 34, [8] = 12.

NOTE

This applies also to the SDO index in data byte 2 and 3 which has the data type UINT. I.e. the low byte is transferred in data byte 2 and the high byte is transferred in data byte 3.

Example: The following messages are necessary to read the SDO 1003, 1 of the CANopen device with device address 2. The data type of the data to be read is UDINT.
The client sends:

CAN ID	DLC	Data							
602h	8	40h	03h	10h	01h	00h	00h	00h	00h

The server responds:

CAN ID	DLC	Data							
582h	8	42h	03h	10h	01h	08h	00h	50h	02h

The response data combine to the 32 bit word 02500008h.

6.2.10 SDO object directory

Every CANopen device manages its SDOs in an object directory. The complete object directory is formally described in an EDS file. Many CANopen tools can read this EDS file and as a result know the object characteristics of the CANopen device.
In the following table, all SDOs for the SP-CANopen gateway are shown.

Tab. 84: Supported SDOs

SDO #	Type
1000	Device type
1001	Error register
1003	Error list (error history)
1005	COB ID SYNC
1008	Device name
1009	Hardware version
100A	Software version
100C	Guard time
100D	Life time factor
1016	Consumer heartbeat time
1017	Producer heartbeat time
1018	Identity record
1027	Module list
1400…1403	RxPDO 1…4 communication parameters
1600…1603	RxPDO 1…4 mapping parameters
1800…1803	TxPDO 1…4 communication parameters
1A00…1A03	TxPDO 1…4 mapping parameters
3100	Module status bits
3200	Config CRC array
3300	Module type code array
6000	Process data input objects
6200	Process data output objects

You can find more detailed information on these SDOs in the CANopen draft standard DS 301 V4.02 (DSP 301 V4.1).

SDO 1001: Error register

The error register (SINT) contains an error bit indicating whether an error is present. If bit 0 is set to 1, a “generic error” has been detected.

SDO 1003: Error list (error history)

SDO 1003 is an array containing the last 10 error codes that have been reported by the gateway via emergency message. Array index 0 contains the number of error codes that have been recorded in SDO 1003.

A new error will be recorded in index 1, while older errors will be renumbered (increased by 1). The array index can be overwritten externally with a 0, which will clear the array completely.

NOTES

- Not all errors that are reported via emergency message will be recorded in SDO 1003, but only the errors listed in Tab. 73.
- Entries in SDO 1003 are UDINT and normally divided in 16 bit error code and 16 bit additional information. In case of an emergency, the module status diagnostics (4 byte) are entered here.

SDO 1005: COB ID SYNC

SDO 1005 contains the COB ID of the Sync object. As a default this value is 80h, but it can be changed.

NOTE When you change the COB ID of the Sync object, keep in mind that the new ID must not be assigned to another communication object already.

SDO 1008: Device name

SD0 1008 contains a device name (VISIBLE STRING).

NOTE This SDO can not be read using a simple “SDO upload expedited”. Instead, the “Upload SDO segment protocol” (client command specifier ccs = 3) must be used as described in the CANopen specification DS 301.

SDO 1009: Hardware version

SDO 1009 contains the current hardware version of the device (VISIBLE STRING).

NOTE This SDO can not be read using a simple “SDO upload expedited”. Instead, the “Upload SDO segment protocol” (client command specifier ccs = 3) must be used as described in the CANopen specification DS 301.

SDO 100A: Software version

SDO 100A contains the current software version of the device (VISIBLE STRING).

NOTE This SDO can not be read using a simple “SDO upload expedited”. Instead, the “Upload SDO segment protocol” (client command specifier ccs = 3) must be used as described in the CANopen specification DS 301.

SDO 100C: Guard time

The product of guard time (UINT) and life time factor (SINT) results in the life guarding time.
Life guarding time [ms] = guard time [ms] × life time factor
The master must send a node guarding message to the slave at least once during the life guarding time. If the life guarding time is exceeded (life-guarding error), the gateway generates a cable break error and sets all process data coming from the network to 0; the NS LED will start flashing Red.
Life guarding is activated in the slave by the first node guarding message if the set life guarding time is not 0. If after activation of the life guarding the guard time or the life time factor is set to 0, life guarding is deactivated.
See also section 6.2.11 “Guarding protocols” on page 101.

SDO 100D: Life time factor

SDO 100D contains the life time factor (SINT). See SDO 100C.

NOTE The life time factor must be either = 0 (disabled) or ≥ 1.5.

SDO 1016: Consumer heartbeat time

The gateway is configured as a heartbeat consumer if SDO 1016 contains a value higher than 0 for the consumer heartbeat time. The consumer heartbeat time is defined in ms. The NMT master must send at least one node guarding message to the slave within this time. If the consumer heartbeat time is exceeded (life-guarding error), the gateway gener-

ates a cable break error and sets all process data coming from the network to 0; the NS LED will start flashing Red.

SDO 1017: Producer heartbeat time

The gateway can also function as a heartbeat producer, i.e. send a heartbeat signal. This allows another device to detect whether the heartbeat producer (i.e. the gateway) is still working correctly.
The producer heartbeat time is defined in ms. For internal processing it is rounded to the next higher multiple of 4. If the heartbeat time is set to 0, the heartbeat signal is deactivated.
The heartbeat signal consists of a cyclic CAN message with the identifier 700h + device address.

NOTE

It is not possible to use heartbeat signals and life guarding messages at the same time because both functions use the same CAN identifier.

See also section 6.2.11 “Guarding protocols” on page 101.

SDO 1018: Identity record

This SDO contains basic information about the gateway.

Tab. 85: SDO 1018 contents

Subindex	Mapping	Format	Description
1	Vendor ID	UDINT	Unique identification of the manufacturer (e.g. Wieland Electric)
2	Product code	UDINT	Device variant
3	Revision number	UDINT	Software version of the device
4	Serial number	UDINT	Serial number of the device

SDO 1027: Module list

The module list contains the module type and the module ID of all samos®PRO modules in the system.

Tab. 86: SDO 1027 contents

Subindex	Module	Format
1	Main module (SP-SCON or SP-SCON-NET)	SINT
2...13	Extension modules (SP-SDIO or SP-SDI)	SINT
14, 15	Gateways	SINT

You can find the module types and module IDs in Tab. 73. The return value for free module slots is 0.

SDO 1400...1403: RxPDO communication parameters

Using SDO 1400 to 1403 the communication parameters for the RxPDOs 1 to 4 are configured. E.g. SD0 1400 defines the parameters for RxPDO 1 etc.

Tab. 87: SDO 1400...1403 contents

Subindex	Mapping	Format	Description
1	COB ID	UDINT	CAN identifier for this PDO, read-only
2	Receive mode	SINT	Fix 255 (asynchronous mode)

The receive mode (read-write) defines how the PDO shall be received. For the RxPDOs the receive mode is set to 255 (asynchronous mode). In this mode the data of a received RxPDO will be routed immediately to the outputs.

NOTE

If the receive mode is set to another value than 255, an error code is generated (abort code 0609 0030h, invalid parameter value).

SDO 1600...1603: RxPDO mapping parameters

This SDO can not be used since the RxPDO mapping is done using the samos®PLAN. See also Tab. 77 and Tab. 79.

SDO 1800…1803: TxPDO communication parameters

Using SDO 1800 to 1803 the communication parameters for the TxPDOs 1 to 4 are configured. E.g. SD0 1800 defines the parameters for TxPDO 1 etc.

Tab. 88: SDO 1800…1803 contents

Subindex	Mapping	Format	Description
1	COB ID	UDINT	CAN identifier for this PDO, read-only
2	Transmission type	SINT	Defines when the PDO is to be sent
5	Event timer	UINT	in ms

The transmission type is set to 255 (asynchronous mode, event triggered) as a default for all TxPDOs.
The event timer contains the rate in ms for the cyclic transmission of the TxPDO.

TxPDO transmission types

Tab. 89: TxPDO transmission types

TxPDO	Synchronous	Asynchronous	RTR
1, 2, 3, 4	0, 1…240	254, 255	253

NOTE

If the transmission type is set to an invalid value, an error code is generated (abort code 0609 0030h, invalid parameter value).

Synchronous: The synchronous transmission type 0 means that the TxPDO will be sent after the reception of a Sync message, but only if data has changed. The synchronous transmission types n = 1…240 define that the TxPDO will be sent after the reception of the n^{th} Sync message.
Asynchronous, event triggered on change of state: The asynchronous transmission type 255 (without configured event timer) means that the TxPDO will be sent each time if at least one input bit has changed that has been mapped on this PDO.
Asynchronous, event triggered on timer event: The asynchronous transmission type 254/255 (with configured event timer) defines that the TxPDO will be sent each time the event timer has expired. E.g. a value of 500 for the event timer means that the gateway will send the corresponding TxPDO each 500 ms.
RTR, on request: The transmission type 253 defines that the TxPDO can be polled via RTR (remote transmission request). This requires a CAN message with DLC = 0, RTR = 1 and the COB ID of the TxPDO to the gateway. The gateway will then respond with the requested TxPDO.

SDO 1A00…1A03: TxPDO mapping parameters

This SDO can not be used since the TxPDO mapping is done using the samos®PLAN. See also Tab. 77 and Tab. 78.

SDO 3100: Module status bits

SDO 3100 contains the module status bits from the samos®PRO system (see Tab. 73). Active bits are low (=“0”).

Tab. 90: SDO 3100 contents

SDO array	Data set parameter	Module	Size
3100,1	Module status module 0	CPU	UDINT
3100,2	Module status module 1	Extension	UDINT
…	…	…	…
3100,14	Module status module 13	Gateway	UDINT
3100,15	Module status module 14	Gateway	UDINT

NOTE

The module positions in the samos®PLAN are numbered from 0 to 14. Therefore the subindex for SDO 3100 is = module position + 1.

SDO 3100 is read-only.

SDO 3200: Config CRC array

SDO 3200 contains the system CRCs in UDINT format.

SDO 6000: Process data input objects

The 32 byte process input data can be written in SDO array 6000. These are the same data as in RxPDO 1-4 (see section 6.2.8 “PDO communication” on page 94). The mapping is as follows:

SDO 6000	RxPDO
6000,1	RxPDO 1, Byte 1
…	…
6000,8	RxPDO 1, Byte 8
6000,9-16	RxPDO 2, Byte 1-8
6000,17-24	RxPDO 3, Byte 1-8
6000,25-32	RxPDO 4, Byte 1-8

Tab. 91: Mapping table SDO 6000 – RxPDO 1-4

SDO 6000 is write-only.

SDO 6200: Process data output objects

The 32 byte process output data can be read from SDO array 6200. These are the same data as in TxPDO 1-4 (see section 6.2.8 “PDO communication” on page 94). The mapping is as follows:

SDO 6200	TxPDO
6200,1	TxPDO 1, Byte 1
…	…
6200,8	TxPDO 1, Byte 8
6200,9-16	TxPDO 2, Byte 1-8
6200,17-24	TxPDO 3, Byte 1-8
6200,25-32	TxPDO 4, Byte 1-8

Tab. 92: Mapping table SDO 6200 – TxPDO 1-4

SDO 6200 is read-only.

6.2.11 Guarding protocols

CANopen provides several possibilities for actively monitoring the correct function of the field bus interface (e.g. cable break detection).

Always use either node guarding or heartbeat!
According to the CIA CANopen specification DS 301 guarding is mandatory. Please activate either node guarding or heartbeat. If no guarding is configured the samos®PRO system can not detect an interruption of the CANopen communication, e.g. a broken network cable. In this case the CANopen gateway’s input and output data may be frozen.

Heartbeat

A heartbeat producer is a CANopen device that sends a cyclic heartbeat message. This enables all other CANopen devices to detect whether the heartbeat producer is still working correctly and what is its current status. Heartbeat messages are sent in a regular time interval, the producer heartbeat time, which can be configured using SDO 1017. The configured 16 bit value will be rounded to the next multiple of 4 ms.
A heartbeat consumer is a CANopen device that expects a cyclic node guarding message within a certain time interval, the consumer heartbeat time, which can be configured using SDO 1016. If the heartbeat consumer receives no node guarding message within the configured consumer heartbeat time, it sends a life guarding emergency message and sets the incoming process data to 0. Additionally the gateway generates an internal “cable break” error that can be processed by the controller module.

Node guarding

Node guarding is performed by an NMT master. This can be each CANopen device that is able to perform this function as a client. The NMT master sends a cyclic node guarding

message to the monitored device which must respond within a defined period that is monitored by the NMT master. If the monitored device does not answer within the node life time, the NMT master regards this as a malfunction of the device and takes the appropriate measures.

Life guarding

Life guarding is performed by the gateway itself. The node life time is calculated in the gateway from the values of SDO 100C (guard time) and SDO 100D (life time factor). If the gateway does not receive a node guarding message at least once within this node life time from an NMT master, the gateway generates an internal "cable break" error that can be processed by the controller module and the NS LED will start flashing √ Red.

NOTES

- Cable break detection is possible for the gateway either if life guarding is activated, i.e. if both objects SDO 100C and SDO 100D have a value other than 0. In this case life guarding starts when the first node guarding request is received from an NMT master and ends if the master sends a "Reset communication".
- Alternatively, cable break detection is possible if the gateway is configured as a heartbeat consumer. In this case, the cable break detection is performed by the gateway itself.
- Heartbeat (producer) works without node guarding. In this case, the gateway can not detect a field bus cable break.
- Heartbeat and node guarding/life guarding can not be used simultaneously.
- If the configuration is changed from activated life guarding to no life guarding or vice versa a complete power reset of the samos®PRO system is required to setup the CANopen network communication properly.

The following table gives an overview of the supported guarding protocols depending on the configuration of SDO 1016 and SDO 1017 (heartbeat), SDO 100C (guard time) and SDO 100D (life time factor).

Tab. 93: Guarding protocol overview and comparison

SDO 1016	SDO 1017	SDO 100C × 100D	Gateway heartbeat	Gateway life guarding	NMT master node guarding
0	**0**	**0**	Not allowed: Always use either node guarding or heartbeat!		
0	**0**	**> 0**	Deactivated	**Cable break detection**	**Required**
> 0	**0**	**0**	**Cyclic heartbeat (consumer)**	**Cable break detection**	**Possible for other slaves**
0	**> 0**	**0**	**Cyclic heartbeat (producer)**	Not possible	Not possible, but guarding as heartbeat consumer is possible
> 0	**> 0**	**0**	**Cyclic heartbeat (producer and consumer)**	Cable break detection	Not possible
> 0	**> 0**	**> 0**	**Not allowed**		

NOTE It is not useful to use heartbeat and life guarding at the same time.

6.2.12 Error objects

The SP-CANopen reports CAN specific errors (e.g. initialisation errors, cable break, CAN communication errors) as SBUS+ errors to the controller module.
Module specific errors as described in Tab. 73 are reported as extended diagnostics via the emergency object and SDO 1003.

Emergency object

The emergency producer (CANopen gateway) is triggered to send the emergency object to the emergency consumer (any CANopen device, normally the controller) by the occurence of CAN specific errors or an error condition as described in Tab. 73.
The emergency object is being sent as described in DS-301 (chapter 9.2.5):

Tab. 94: Emergency states and transitions

Emergency state before	Transition	Module specific alarms	Emergency state after
Error free	1	Incoming error	Error occured
Error occured	2	Outgoing error, other errors present	Error occured
Error occured	3	Incoming error, other errors present	Error occured
Error occured	4	All errors cleared	Error free

The gateway is in one of two emergency states, either *Error free* or *Error occured*. Emergency objects are sent depending on the transitions between these two emergency states. The error code in the emergency object shows the emergency state the gateway is in. See also Tab. 95.

Error objects overview

Tab. 95: Error objects overview

CAN specific errors	Error code SBUS+	Error type	Emergency error code Error register M1…M5	Error history SDO 10 03	Result/possible remedy
CAN data overrun CAN controller overrun in Rx Fifo	0x4501	Warning	0x8110 0x11 1, 0, 0, 0, 0	–	CAN messages have been lost. Bandwidth is limited. Check CAN settings, increase baud rate, reduce participants/traffic.
CAN error passive CAN controller is in error passive state	0x4503	Warning	0x8120 0x11 0, 0, 0, 0, 0	–	The gateway sends only recessive bits, i.e. it sets its own messages to invalid. The reason is either a gateway hardware error or external data transmission interferences. Check cabling.
CAN bus off CAN controller is in bus off state	0x4504	Warning	–	–	Massive transmission errors. CAN controller has disconnected from the bus. Possible hardware defect. Power cycle the samos®PRO system.
CAN Tx Fifo overflow CAN controller has no transmit ressources	0x4506	Warning	0x8110 0x11 2, 0, 0, 0, 0	–	CAN messages that were to be sent by the gateway have been lost. The number of events triggering the gateway to send CAN messages is too high for the set baud rate. Increase the baud rate or change the gateway configuration.
CAN init failed The CAN controller could not be initialised	0xC507	Critical	–	–	The CAN controller/transceiver is possibly defect. Replace the SP-CANopen with a new device.
CANopen life guarding CANopen life guarding has detected a cable break	0x4508	Warning	0x8130 0x11 0, 0, 0, 0, 0	–	Life guarding error message generated by the gateway: Either a failure of the node guarding or heartbeat NMT master has occurred or the CAN cable is broken. Check the CANopen master. Check cabling.

Module specific alarms	Error code SBUS+	Transition emergency state chart	Emergency error code Error register M1…M5	Error history SDO 10 03	
Gateway detected incoming error as per trigger conditions	–	1	0xFF01 0x81 M1 = Module index M2…M5 = Module diagnostics data	M2, M3, M4, M5	See Tab. 73.
Gateway detected outgoing error, other errors are present	–	2	0xFF02 0x81 M1 = Module index M2…M5 = Module diagnostics data	M2, M3, M4, M5	See Tab. 73.
Gateway detected incoming error, other errors are present	–	3	0xFF03 0x81 M1 = Module index M2…M5 = Module diagnostics data	M2, M3, M4, M5	See Tab. 73.
All errors cleared	–	4	0x0000 0x00 M1 = 0 M2…M5 = 0	–	

6.2.13 CANopen diagnostics examples

Example 1: SP-SDIO module on position 1, output Q4 stuck at high

The gateway sends an emergency message (see Tab. 72).

CAN-ID	DLC	DATA							
08C	8	03	FF	01	01	40	00	00	00

The gateway's CANopen address is 12 (= C Hex). The SP-SDIO module has the position 1 in the samos®PRO system.

08C: Identifier (80 + C)
8: Data length code: 8 bytes follow
03FF: Error code FF03: Device specific error
01: Error register 01 of SDO 1001H
01: Module index M1: Module in position 1
40: Module status bit 30 (bit 6 of byte M2) = 1: Output 4 stuck at high (see Tab. 73)

Reading the current error from SDO 3100:

PLC request:

CAN-ID	DLC	DATA							
60C	8	40	00	31	02	00	00	00	00

60C: Identifier (600 + C)
8: Data length code: 8 bytes follow
40: Expedited upload request
00 31: Index 3100
02: Subindex 02: Module in position 1 (see Tab. 90)
Gateway response:

CAN-ID	DLC	DATA							
58C	8	42	00	31	02	BF	FF	FF	FB

58C: Identifier (580 + C)
8: Data length code: 8 bytes follow
42: Upload response, data set size is not indicated
00 31: Index 3100
02: Subindex 02: Module in position 1 (see Tab. 90)
FB: Error byte M5, bit 2 = 0: External error
BF: Error byte M2, bit 30 = 0: Output 4 stuck-at-high error

Reading the error from the error history in SDO 1003:

PLC request:

CAN-ID	DLC	DATA							
60C	8	40	03	10	01	00	00	00	00

60C: Identifier (600 + C)
8: Data length code: 8 bytes follow
40: Expedited upload request
03 10: Index 1003
01: Subindex 01: Last error

Gateway response:

CAN-ID	DLC	DATA							
58C	8	42	03	10	01	40	00	00	00

58C: Identifier (580 + C)
8: Data length code: 8 bytes follow
42: Upload response, data set size is not indicated
03 10: Index 1003
01: Subindex 01: Last error
40: Module status bit 30 (bit 6 of byte M2 = 0: Output 4 stuck at high

Example 2: SP-SDI module dual channel input error on I1/I2

The gateway sends an emergency message (see Tab. 72).

CAN-ID	DLC	DATA							
08C	8	03	FF	01	0C	00	00	01	00

The gateway's CANopen address is 12 (= C Hex). The SP-SDI module has position 12 in the samos®PRO system.
08C: Identifier (80 + C)
8: Data length code: 8 bytes follow
03FF: Error code FF03: Device specific error
01: Error register 01 of SDO 1001H
0C: Module index M1: Module in position 12 (C Hex)
01: Module status bit 8 (bit 0 of byte M4) = 1: Input 1-2 dual channel input evaluation: error detected (see Tab. 73)

Reading the current error using SDO 3100:

PLC request:

CAN-ID	DLC	DATA							
60C	**8**	40	00	31	0D	00	00	00	00

60C: Identifier (600 + C)
8: Data length code: 8 bytes follow
40: Expedited upload request
00 31: Index 3100
0D: Subindex 0D = Module in position 12 (module position = subindex - 1, see also Tab. 90)

Gateway response:

CAN-ID	DLC	DATA							
58C	**8**	42	00	31	0D	FF	FF	FE	FB

58C: Identifier (580 + C)
8: Data length code: 8 bytes follow
42: Upload response, data set size is not indicated
00 31: Index 3100
0D: Subindex 0D: Module in position 12 (see Tab. 90)
FB: Error byte M5, bit 2 = 0: External error
FE: Error byte M4, bit 0 = 0: Input 1-2 dual channel input evaluation: error detected (see Tab. 73)

Reading the error from the error history in SDO 1003:

PLC request:

CAN-ID	DLC	DATA							
60C	**8**	40	03	10	01	00	00	00	00

60C: Identifier (600 + C)
8: Data length code: 8 bytes follow
40: Expedited upload request
03 10: Index 1003
01: Subindex 01: Last error

Gateway response:

CAN-ID	DLC	DATA							
58C	**8**	42	03	10	01	00	00	01	00

58C: Identifier (580 + C)
8: Data length code: 8 bytes follow
42: Upload response, data set size is not indicated
03 10: Index 1003
01: Subindex 01: Last error
01: Module status bit 8 (bit 0 of byte M4) = 0: Input 1-2 dual channel input evaluation: error detected

6.2.14 Diagnostics and troubleshooting

For information on how to perform diagnostics on the samos®PRO system please refer to the operating instructions for the samos®PLAN software (Wieland Electric doc. no. BA000518).

Tab. 96: Troubleshooting for the SP-CANopen

Error	Possible cause	Possible remedy
Symbol description: ○ – LED off, ● Green – LED lights up green, ✸ Red – LED flashes red		
SP-CANopen does not supply any data LED PWR ● Green LED NS ○ Off LED MS ✸ Red (1 Hz)	Configuration required, node guard or heartbeat message have not been sent. Configuration download is not completed.	Configure the SP-CANopen and download the configuration to the device. Wait until the configuration download has been completed.
SP-CANopen does not supply any data LED PWR ● Green LED NS ● Green LED MS ✸ Red (1 Hz)	Configuration download is not completed.	Wait until the configuration download has been completed.
SP-CANopen does not supply any data LED PWR ● Green LED NS ● Green LED MS ✸ Red/Green	No PDO transfer since power up.	Start PDO transfer. Transfer PDO via SDO 6000 or 6200.
SP-CANopen does not supply any data LED PWR ● Green LED NS ✸ Green LED MS ✸ Red/Green	No PDO Transfer since power up. False baud rate (CAN transceiver could be in passive error). False Node ID/CANopen address. CAN cable has been disconnected.	Start PDO transfer. Transfer PDO via SDO 6000 or 6200. Check and correct the baud rate. Check and correct the address. Check CANopen cabling.
SP-CANopen does not supply PDO data LED PWR ● Green LED NS ○ Off/● Red/ ● Green LED MS ✸ Green (1 Hz)	SP-CANopen is in Idle mode. Node guard or heartbeat messages will be sent. samos®PRO configuration is not verified and CPU module is stopped.	CPU/application is stopped. Start CPU (change into Run mode). Verify the configuration with samos®PLAN and start the CPU module.
SP-CANopen does not supply PDO data LED PWR ● Green LED NS ● Green LED MS ○ Off	Supply voltage too low.	Check supply voltage.
SP-CANopen does not supply any data LED PWR ● Red LED NS ● Red LED MS ● Red	Supply voltage dip.	Check supply voltage. Reset samos®PRO system
SP-CANopen does not supply any data LED PWR ● Green LED NS ✸ Green (1 Hz) LED MS ✸ Green (1 Hz)	False Node ID/CANopen address. False baud rate (CAN transceiver could be in passive error), SP-CANopen is in Idle mode.	Check and correct the address. Check and correct the baud rate.
SP-CANopen does not supply any data LED PWR ● Green LED NS ● Red LED MS ✸ Red/Green	False baud rate and SP-CANopen transceiver is in bus off state (hardware problem at CAN physical layer). CAN cable has been disconnected.	Check and correct the baud rate. Check CANopen cabling. Reset samos®PRO system.
SP-CANopen does not supply any data LED PWR ● Green LED NS ✸ Green (1 Hz) LED MS ● Green	CANopen master is in stop or preoperational mode. During initialization of the bus system, another slave could not be initialized. CANopen state of SP-CANopen is preoperational. False Node ID/CANopen address.	Set CANopen master into Run mode (CANopen state Operational). Check whether all slaves on the bus are “On”. Check CANopen cabling. Check whether CAN master starts automatically. Check and correct the CANopen address.

Error	Possible cause	Possible remedy
Symbol description: ○ – LED off, ● Green – LED lights up green, ✸ Red – LED flashes red		
SP-CANopen does not supply any data LED PWR ● Green LED NS ● Red LED MS ● Green	SP-CANopen transceiver is in error passive. CAN cable has been disconnected.	Check CANopen cabling. Check the diagnostics messages with samos®PLAN. Reset samos®PRO system.
SP-CANopen does not supply any data LED PWR ● Green LED NS ✸ Red (1 Hz) LED MS ✸ Red/Green	Node guarding or heartbeat consumer failure. Guarding configuration has been changed.	Check CANopen cabling. Check life guarding time (life time factor 1) Check heartbeat consumer time (should be 1,5 × heartbeat producer time) Check the diagnostics messages with samos®PLAN. Reset samos®PRO system.
SP-CANopen is in Critical Fault mode LED PWR ● Green LED NS ● Red LED MS ✸ Red (2 Hz)	SP-CANopen internal device error. CPU firmware version does not support samos®PRO gateways.	Switch off the power supply of the samos®PRO system and switch it on again. Check the diagnostics messages with samos®PLAN. Use a CPU with the required firmware version (see section 2.2 “Intended use” on page 10). If the error remains, replace the gateway.
SP-CANopen/samos®PRO system is in Critical Fault mode LED PWR ● Red LED NS ○ Off LED MS ● Red	SP-CANopen is not plugged properly into the other samos®PRO module. Module connecting plug is soiled or damaged. Other samos®PRO module has internal critical error.	Plug the SP-CANopen in correctly. Clean the connecting socket/plug. Repower the system. Check the other samos®PRO modules.

7 Layout and content of the process image

7.1 Routing

The process image transmitted by the ***samos***®PRO gateways into the network consists of the operational data (e.g. logic results, input and output states) and the diagnostics data (e.g. module status, CRCs). These data are organised in 4 data sets.

Data set #	Content	Size	Can be customized
1	Operational data	50 bytes	Yes
2	CRCs	32 bytes	No
3	Status and diagnostics	60 bytes	No
4	Reserved	60 bytes	No

Tab. 97: Content of the data sets 1-4

The operational data in data set 1 may comprise up to maximally 50 bytes, independent of the network protocol used. These 50 bytes are organized into one or several data blocks, dependent of the network protocol. For detailed information about the modularisation of the data sent into the network please see Tab. 98 and read the chapter on the related gateway.

The content of data set 1 can be freely customized with a granularity of 1 byte but is pre-configured in the delivery status (see section 7.2 “Default settings for the operational data” on page 109 and section 7.3 “Customizing the operational data ” on page 110).

The diagnostics data in data sets 2-4 depend on the network protocol used and are described in the chapter on the related gateway.

7.2 Default settings for the operational data

In the delivery status, the operational data are pre-configured. Depending on the gateway used, these data are subdivided in several data blocks.

The following table gives an overview which bytes are assigned to the default configuration and how the data are modularised for the different gateways.

	EtherNet/IP, Modbus TCP, Ethernet TCP/IP		Profinet IO, PROFIBUS DP	
Byte	**Default assignment**	**Input data set**	**Default assignment**	**Input data block**
0	Logic result 0	#1 (50 bytes)	Module 1 input	#1 (12 bytes)
1	Logic result 1		Module 2 input	
2	Logic result 2		Module 3 input	
3	Logic result 3		Module 4 input	
4	Module 1 input		Module 5 input	
5	Module 2 input		Module 6 input	
6	Module 3 input		Module 7 input	
7	Module 4 input		Module 8 input	
8	Module 5 input		Module 9 input	
9	Module 6 input		Module 10 input	
10	Module 7 input		Module 11 input	
11	Module 8 input		Module 12 input	

Tab. 98: Default configuration for the operational data transmitted into the network

	EtherNet/IP, Modbus TCP, Ethernet TCP/IP		Profinet IO, PROFIBUS DP	
Byte	**Default assignment**	**Input data set**	**Default assignment**	**Input data block**
12	Module 9 input		Module 1 output	#2 (12 bytes)
13	Module 10 input		Module 2 output	
14	Module 11 input		Module 3 output	
15	Module 12 input		Module 4 output	
16	Module 1 output		Module 5 output	
17	Module 2 output		Module 6 output	
18	Module 3 output		Module 7 output	
19	Module 4 output		Module 8 output	
20	Module 5 output		Module 9 output	
21	Module 6 output		Module 10 output	
22	Module 7 output		Module 11 output	
23	Module 8 output		Module 12 output	
24	Module 9 output		Logic result 0	#3 (12 bytes)
25	Module 10 output		Logic result 1	
26	Module 11 output		Logic result 2	
27	Module 12 output		Logic result 3	
28-35	Not assigned		Not assigned	
36-47	Not assigned		Not assigned	#4 (12 bytes)
48-49	Not assigned		Not assigned	#5 (2 bytes)

The default byte assignment can be freely customised as will be described in the following section.

7.3 Customizing the operational data (*samos*®PRO to Network)

This section outlines briefly how you can customize the operational data that the ***samos***®PRO gateway transmits to the network. You will find more detailed information in the ***samos***®PLAN software operating instructions (Wieland part no. BA000518).

In the delivery status, the data routing configuration of the ***samos***®PRO gateways is shown in the gateway configuration dialog.

➡ Click on the **Gateway** button above the main window and select the respective gateway to open the gateway configuration dialog.
➡ Click on the ***samos***®PRO **to Network** tab on the left hand menu to display the routing configuration dialog.

The default setting is as follows (example for Modbus TCP):

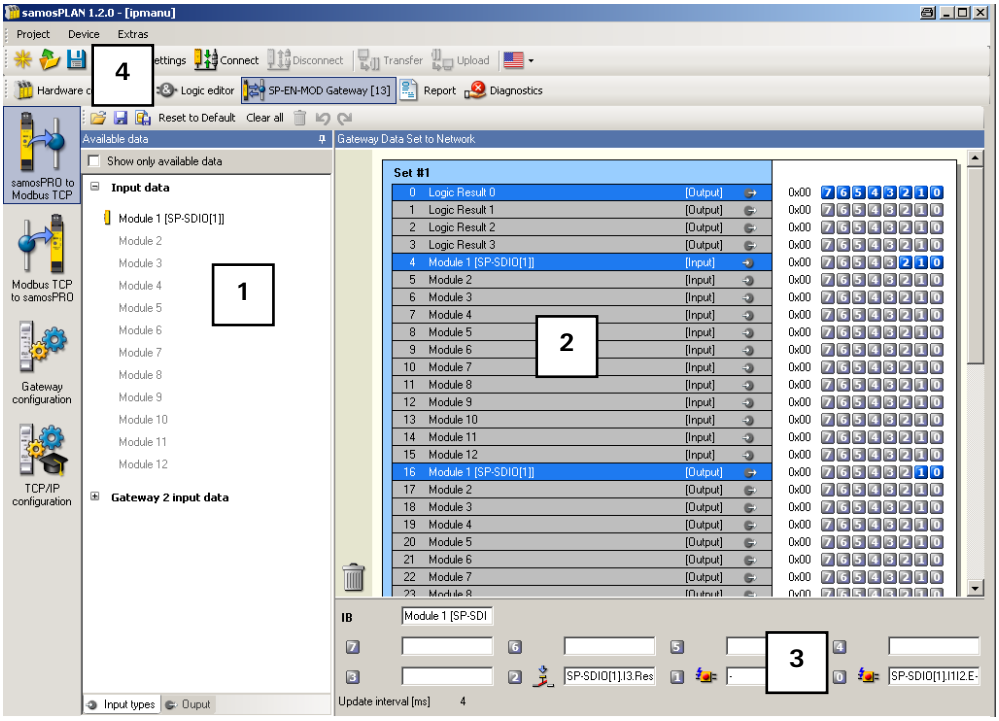


Fig. 54: Default configuration for the operational data transmitted into the network

Basically this dialog is divided into three areas: **Available data** [1], **Gateway data** [2] and **Tag Names** [3]. The upper left corner of the dialog holds the toolbar [4].

7.3.1 The toolbar

Fig. 55: Toolbar for the routing configuration

The toolbar contains buttons for the following actions (from left to right):

- The **Load user configuration** and **Save user configuration** buttons allow you to load and/or save a configuration in XML format. If you load a configuration, all previously made changes that have not been saved will be lost. You can not undo this action.
- With the **Import** and **Export** buttons you can import and export a configuration including the tag names used as a CSV (comma separated values) file or in a network specific file format, e.g. SIEMENS .seq for PROFIBUS or Profinet. This allows you to import and use the assigned tag names in a PLC program.

NOTE

The **Import** button is only available for the *Network to gateway* routing configuration.

- **Reset to Default** restores the default routing configuration. You will be prompted for confirmation. If you click **Yes**, all previously made changes that have not been saved will be lost. You can not undo this action.
- **Clear all** clears the configuration, i.e. deletes all assigned bytes in the **Gateway Data** area. You will be prompted for confirmation.
- **Delete routing** deletes the selected byte from the **Gateway Data** area.
- The **Undo** and **Redo** buttons allow you to undo or redo changes you made to your configuration.

7.3.2 Available data area

This area offers all sources from which data may be routed into the network. It is divided in two views holding the available **Input types** and **Output** data. You can switch between these views using the file cards at the bottom.

- The **Input types** view contains the input values for the connected ***samos***®PRO modules. If your ***samos***®PRO system contains a second gateway, the input data of this gateway (i.e. data received from the network the second gateway is connected to) will be available here as well.
- The **Output** view offers the output values for the connected ***samos***®PRO modules as well as the **Logic results** from the logic editor.

All sources supported by the current configuration are displayed in black:

- connected ***samos***®PRO modules
- configured logic results[8)]
- gateway input data available from another gateway in the system

Sources currently not configured will be displayed in grey. Activating the **Show only available data** checkbox in the upper left corner hides the unused sources from the view.

Sources that offer "live" data are marked with a little icon next to the text.

How to add a data byte to the routing table:

➧ Drag and drop an element (i.e. byte) from the **Available data** area to a free slot in the **Gateway Data** area. If the desired position is not free, you will have to clear it first by deleting or moving the byte currently assigned to it.

NOTE It is possible to use the same byte several times in the routing table.

7.3.3 Gateway Data area

This area contains the routing table. It shows the current content of the ***samos***®PRO gateway's input data modules, Bytes and bits highlighted blue will hold "live" data from the system since the hardware configuration does support the source. Bytes highlighted grey actually do not have data associated with them since the hardware configuration does not support the sources.

How to delete a data byte from the routing table:

➧ Drag and drop the byte you want to delete to the trashcan icon in the bottom left corner of the **Gateway Data** area.

Or:

➧ Select the byte you want to delete by clicking it with the left mouse button. Then, click on the **Delete routing** button in the toolbar.

Or:

➧ Call up the context menu by clicking the respective byte with the right mouse button. In the context menu, select the **Delete routing** command.

How to move a data byte to another place in the routing table:

➧ Drag and drop the byte you want to move to the desired position. If the desired position is not free, you will have to clear it first by deleting or moving the byte currently assigned to it.

[8)] In the default configuration, only the first logic result byte (Logic Result 0) is active and available. You can activate more logic result output bits in the logic editor (see the Samos(r)PLAN operating instructions, Wieland part no. 8012479).

7.3.4 Tag names area

This area shows the tag names associated with each bit of the byte currently selected in the **Available Data** or the **Gateway Data** area. You can enter these tag names in the logic editor and in the hardware configuration dialog (e.g. for extension modules).

In the **Tag Names** area of the ***samos***®PRO to Network configuration dialog, it is not possible to edit the tag names.

7.3.5 Gateway direct output values

It is possible to write values directly from the logic editor into a gateway. In the default process image, four bytes are reserved for these gateway direct output values which can be found in the logic editor in the **Outputs** file card.
In order to use gateway direct output values, a CPU with firmware V2.00.0 or higher is required.

Note

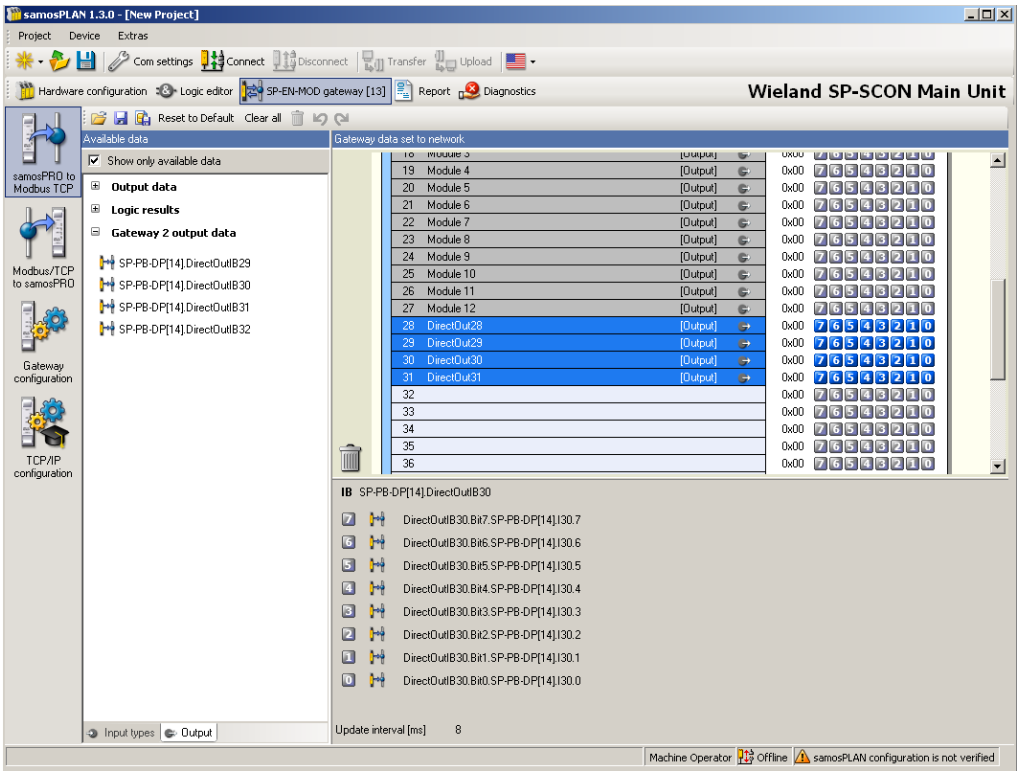


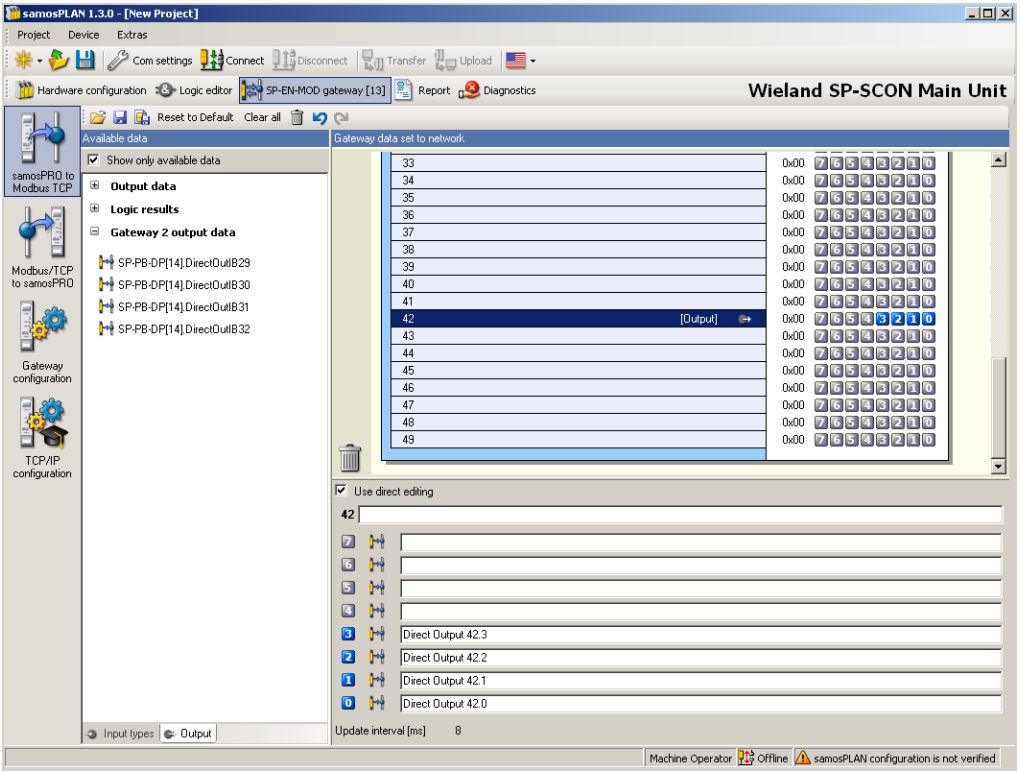
Fig. 56: Gateway direct output values in the default process image

If required, you can configure any byte as gateway direct output value. In order to do this, you need to assign tag names for the bits you want to use.

How to define additional gateway direct output values:

- ➡ Click on a free byte in the Gateway data area to select it.
- ➡ Activate the Use direct editing checkbox in the upper left corner of the Tag names area. You will now be able to edit the tag names for this byte.
- ➡ If desired, enter a tag name for the selected byte.
- ➡ Enter tag names for the individual bits of the selected byte that you want to use as gateway direct output values.

Fig. 57: Defining additional gateway direct output values in the process image



All bits of the selected byte with a tag name will now appear in the logic editor in the **Outputs** file card.

Note

You can edit the predefined gateway direct output values in the same way.

7.3.6 Output data configuration (Network to *samos*®PRO)

In order to enable incoming data bits:

➧ Click on **Network to *samos*®PRO** on the left hand menu. The following dialog appears:

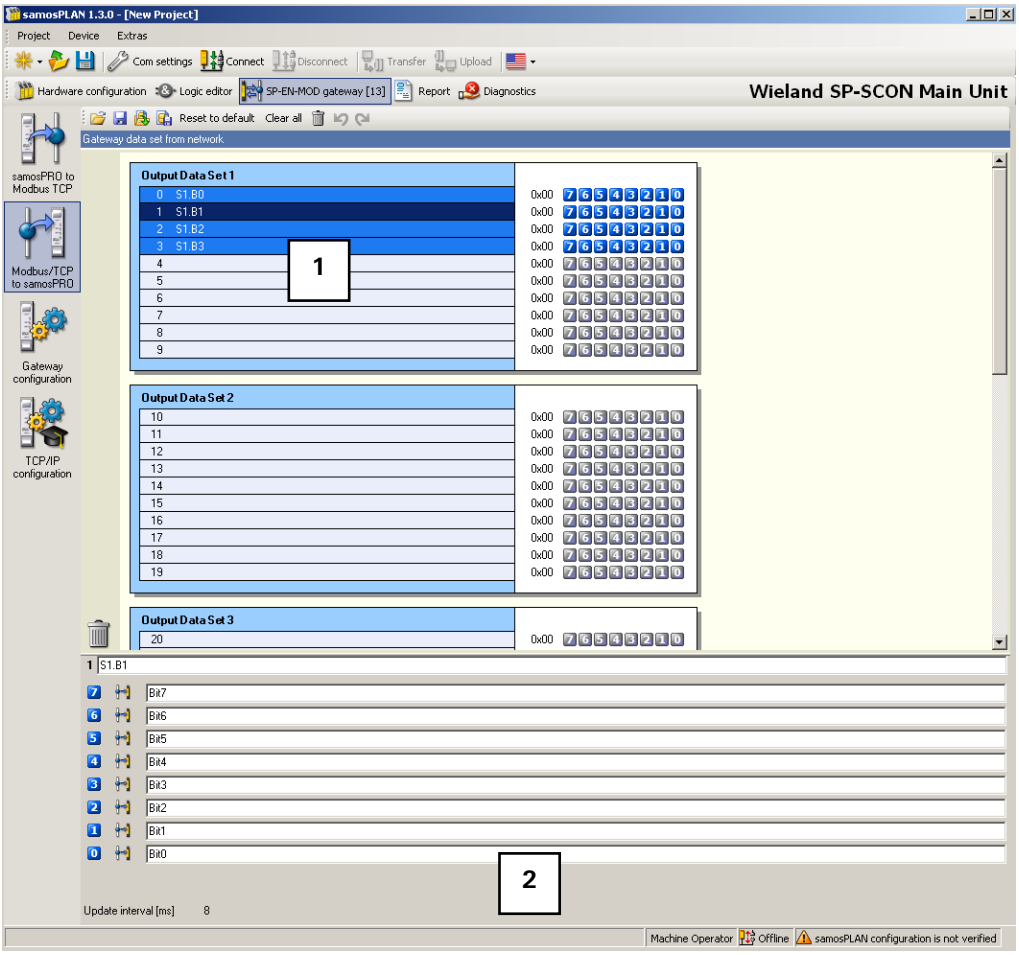


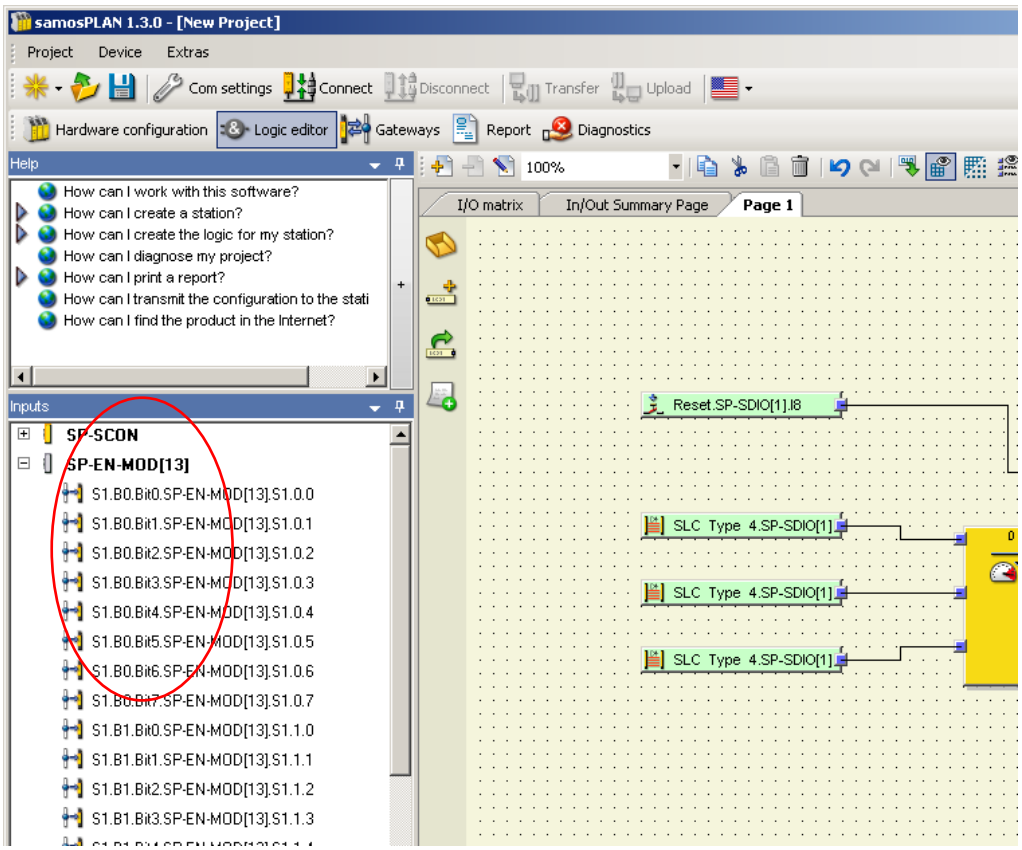
*Fig. 58: Network to **samos**® PRO dialog of the SP-EN-MOD*

Basically this dialog is divided into two areas: **Gateway data** [1] and **Tag Names** [2]:
The **Gateway data** area shows the current configuration of the output modules,
The **Tag Names** area shows the tag names associated to the byte selected in the **Gateway data** area.

➧ Select a byte in the **Gateway data** area.
➧ For each bit of the selected byte that you wish to use, enter a tag name in the **Tag Names** area.

Each bit you enter a tag name for here will be available within the logic editor or for the process image of a second gateway:

Fig. 59: Tag names of incoming bits in the logic editor dialog of the SP-SCON



7.3.7 Saving and loading a configuration

Using the buttons **Load user configuration** and **Save user configuration** you can save and load your configuration in XML format. If you load a configuration, all previously made changes that have not been saved will be lost. You can not undo this action.

7.3.8 Importing and exporting a configuration

With the **Import** and **Export** buttons you can import and export a configuration including the tag names used as a CSV (comma separated values) file or in a network specific file format, e.g. SIEMENS .seq for PROFIBUS or Profinet. This allows you to import and use tag names you have assigned in the ***samos***®PRO project in the PLC program and vice versa.

If you import a configuration, all previously made changes that have not been saved will be lost. You can not undo this action.

NOTE The **Import** button is only available for the *Network to gateway* routing configuration.

7.4 Monitoring the operational data online

When the ***samos***®PRO system is online and running, you can monitor the operational data online in the gateway configuration window.

➥ Click on the **Gateway** button above the main window and select the respective gateway to open the gateway configuration dialog.

➡ Click on the ***samos***®PRO **to Network** or the **Network to *samos***®PRO tab on the left hand menu to display the routing view for the input or output data you want to monitor.

For both directions, ***samos***®PRO **to Network** as well as **Network to *samos***®PRO, inactive bits are displayed grey while active bits are highlighted green:

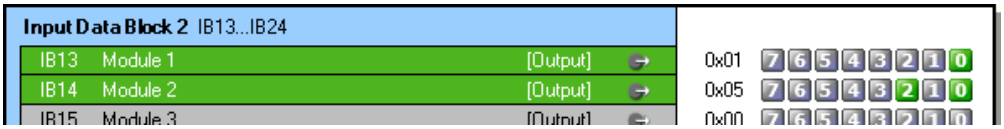


Fig. 60: Active and inactive bits in the online process image

In the ***samos***®PRO **to Network** view, bits that are inactive due to an error are displayed red. This could be the case e.g. for the outputs of an SP-SDIO module if the power supply of this module is faulty:

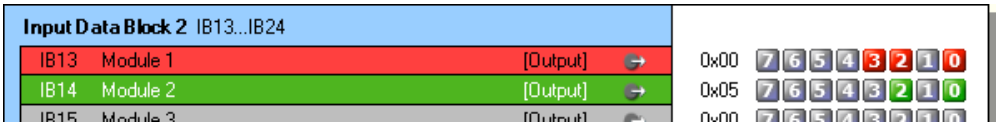


Fig. 61: Inactive Network input bits as a result of an error.

In the **Network to *samos***®PRO view, bits that have no tag name assigned (so that they can not be used in the logic editor) but which are included in the process image that the ***samos***®PRO gateway receives from the PLC, are highlighted yellow:

Fig. 62: Network output bits with no tag names assigned in the online process image

NOTE

The ***samos***®PRO gateways always reflect the actual physical status of the inputs and outputs on the connected modules and devices. This means that even when Force mode is active and inputs that are physically **Low** are forced **High** (or vice versa) the actual physical status of these inputs will be transmitted to the PLC instead of the (virtual) forced status. If, however, as a result of forcing of one or several inputs, one or several outputs change their status, the changed status of these outputs will be transmitted to the PLC since the actual physical status of the outputs on the devices has changed.

8 Technical specifications

8.1 PROFIBUS DP gateways

Interface	Min.	Typ.	Max.
Fieldbus	PROFIBUS-DP-V0		
Interface level	RS-485		
Connector technology	D-Sub socket 9-pin		
Slave address (set via rotary switches)	0		99
Slave address (set via ***samos***®PLAN) [9]	3		125
Baud rate (auto-matic adjustment)			12 MBaud
Baud rate (kBits/s with standard cable)			Max. cable length
9.6/19.2/93.75			1200 m
187.5			1000 m
500			400 m
1,500			200 m
12,000			100 m

Cable parameters see chapter 6.1 “PROFIBUS DP gateway” from page 72.

8.2 CANopen gateways

Interface	Minimum	Typical	Maximum
Fieldbus	CANopen DS-301		
Interface level	RS-485		
Connector technology	5 pin open style connector		
Slave address (set via rotary switches)	0		99
Slave address (set via ***samos***®PLAN) [10]	1		127
Baud rate (kBit/s with standard cable)			Max. cable length
125			500 m
250			250 m
500			100 m
800			40 m
1000			20 m

Cable parameters see chapter 6.2 “CANopen gateway” from page 83.

[9] In order to set the slave address via software, the hardware setting for the address must be “0”.
[10] In order to set the slave address via software, the hardware setting for the address must be “00”.

8.3 EtherNet/IP, PROFINET IO, Modbus TCP gateways

Interface	
Fieldbus	EtherNet/IP, PROFINET IO, Modbus TCP
Integrated switch	3-port layer-2 managed switch with Auto-MDI-X for automatic detection of crossed Ethernet cable
Connection technique	RJ45 socket
Transfer rate	10 Mbit/sec (10Base-T) or 100 Mbit/sec (100Base-TX), autosen-sing
Update rate (heartbeat rate)	Configurable from 40 … 65535 ms
Change of state (COS) update rate	10 ms
Addressing facto- ry setting	IP: 192.168.250.250 Subnet mask: 255.255.0.0 Default gateway: 0.0.0.0
MAC address	Printed on type label, example: 00:06:77:02:00:A7

8.4 Supply circuit specifications

These technical specifications apply to all gateways.

Supply circuit (e.g. via SBUS+)	**Min.**	**Typ.**	**Max.**
Supply voltage	16.8 V DC	24 V DC	30 V DC
Power consumption			2.4 W

8.5 General specifications

These technical specifications apply to all gateways.

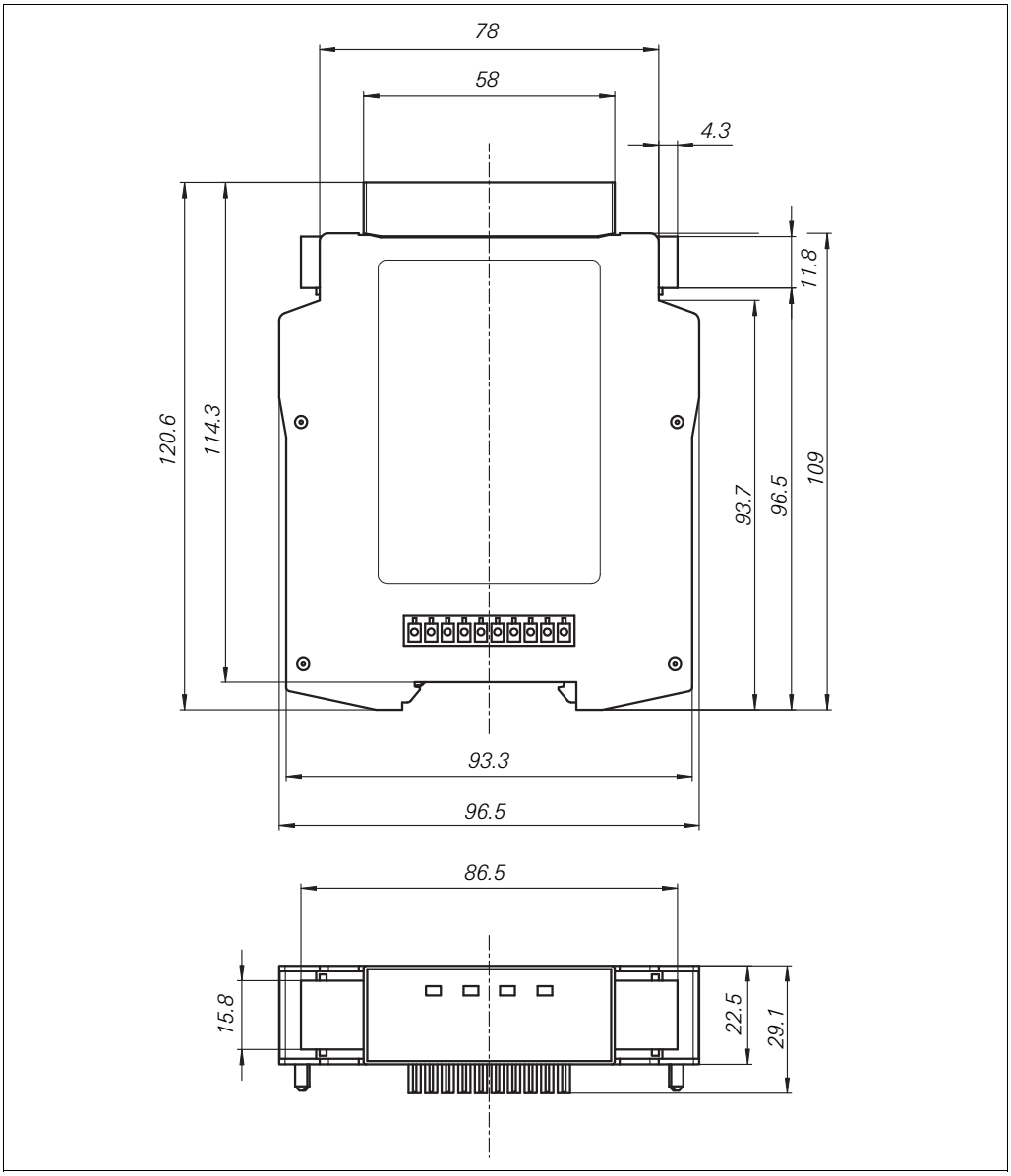
Terminals	
Fieldbus	See technical specifications of the gateways
SBUS+	10-pin connector for internal safety bus (plug)
Climatic conditions	
Ambient operating temperature T_A	–25 to +55 °C
Storage temperature	–25 to +70 °C
Relative humidity	10% to 95%, non-condensing
Climatic conditions (EN 61131-2) Air pressure in operation	 860 to 1060 hPa
Mechanical strength	
Sinusoidal vibration (EN 60068-2-6) Frequency range Amplitude Acceleration Number of cycles	 5 to 150 Hz 3.5 mm (5 to < 9 Hz) 1 g (9 to 150 Hz) 10 per axis (on 3 axes)
Vibration wideband noise (EN 60068-2-64) Frequency range Acceleration	 10 to 500 Hz 5 g
Half-sinusoidal shocks (EN 60068-2-27) Acceleration Duration	 15 g 11 ms
Electrical safety	See SP-SCON

Enclosure rating (EN 60529)	IP 20
Protection class	III
Electromagnetic compatibility	EN 61000-6-2/EN 55011 Class A
Mechanical and assembly	
Housing material	Polycarbonate
Housing type	Device for control cabinet installation
Housing enclosure rating/terminals	IP 20/IP 40
Colour Gateways	 Light grey
Weight	0.16 kg
SBUS+ connection (internal bus) Number of poles Gateways	 10 1 connector left and 1 connector right
Mounting rail	Mounting rail acc. to EN 60715

8.6 Dimensional drawings

8.6.1 SP-EN-IP, SP-EN-MOD and SP-EN-PN

Fig. 63: Dimensional drawing SP-EN-IP, SP-EN-MOD and SP-EN-PN (mm)



8.6.2 SP-CANopen and SP PROFIBUS-DP

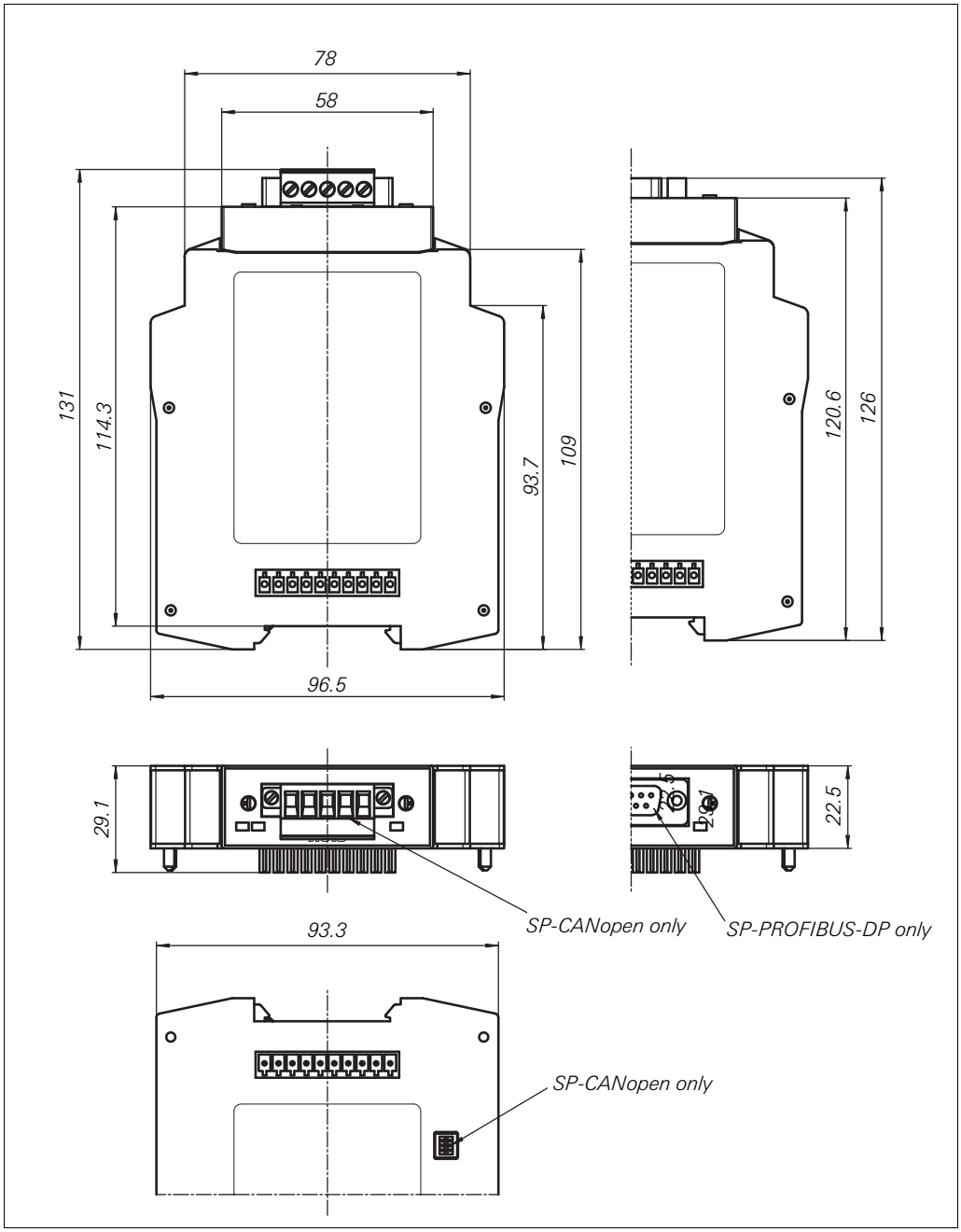


Fig. 64: Dimensional drawing SP-CANopen and SP-PROFIBUS-DP (mm)

9 Ordering information

9.1 Available modules and accessories

Tab. 99: Part numbers of samos PRO modules and accessories

Device type	Part	Part no.
SP-SCON-P1-K	Main module	R1.190.0010.0
SP-SCON-NET-P1-K	Main module, network capable Two-tier spring terminals, pluggable	R1.190.0020.0
SP-CANopen	CANopen gateway	R1.190.0210.0
SP-PROFIBUS-DP	PROFIBUS-DP gateway	R1.190.0190.0
SP-DeviceNet	DeviceNet gateway	R1.190.0230.0
SP-EN-MOD	Modbus/TCP gateway	R1.190.0130.0
SP-EN-PN	ProfiNet gateway	R1.190.0140.0
SP-EN-IP	EtherNet/IP gateway	R1.190.0150.0
SP-SDIO84-P1-K-A	Input/output extension 8 inputs/4 outputs Two-tier screw terminals, pluggable	R1.190.0030.0
SP-SDIO84-P1-K-C	Input/output extension 8 inputs/4 outputs Two-tier spring terminals, pluggable	R1.190.0040.0
SP-SDI8-P1-K-A	Input extension 8 inputs Two-tier screw terminals, pluggable	R1.190.0050.0
SP-SDI8-P1-K-C	Input extension 8 inputs Two-tier spring terminals, pluggable	R1.190.0060.0
SA-OR-S1-4RK-A	Output module 2 NO contacts and 1 24-V DC alarm signal Two-tier screw terminals, pluggable	R1.180.0080.0
SA-OR-S2-2RK-A	Output module 4 NO contacts and 2 24-V DC alarm signals Two-tier screw terminals, pluggable	R1.180.0320.0
SA-OR-S1-4RK-C	Output module 2 NO contacts and 1 24-V DC alarm signal Two-tier spring terminals, pluggable	R1.180.0430.0
SA-OR-S2-2RK-C	Output module 4 NO contacts and 2 24-V DC alarm signals Two-tier spring terminals, pluggable	R1.180.0440.0
SP-PLAN	Programming software and manual	R1.190.0070.0
SP-MEMORY	System plug	R1.190.0080.0
SP-CABLE1	Configuration cable 2 m, M8, Sub D	R1.190.0090.0
SP-PRO-STARTER-SET	Set, consisting of one SP-SCON, SP-SDIO, SP-MEMORY, SP-PLAN, SP-CABLE1	R1.190.0100.0
SP-CABLE3	CAN-cable 2 x 2 x 0,34 mm2, with shield	00.102.5202.0
SP-FILTER1	Terminal block with filter	R1.190.0260.0
SP-FILTER2	Terminal block with filter	R1.190.0270.0
WKFN 2,5 E/35 GO-URL	***fasis*** terminal block with diodes	56.703.8755.9
APFN 2,5 E/35	Terminal plate for WKFN 2,5 E/35	07.312.7355.0

10 Annex

10.1 Index of tables

Annex

10.2 Index of illustrations

Annex

Annex