SIEMENS

SIMATIC

Process Control System PCS 7 S7HMODM3 Modbus – Master – Communication package

Function Manual

08/2023

Legal information

Warning notice system

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

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

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

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

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

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

Qualified Personnel

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

Proper use of Siemens products

Note the following:

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

Trademarks

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

Disclaimer of Liability

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

Siemens AG
Digital Industries
Postfach 48 48
90026 NÜRNBERG
GERMANY

Ⓟ 09/2023 Subject to change

Table of contents

Security information 1

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

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

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

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

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

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

Introduction 2

2.1 General

2.1.1 License conditions

- For the usage of this library a plant license is mandatory.
- The blocks are valid in one coherent plant.

2.1.2 Hardware

- You can only use S7-41x (like S7-417 or S7410) CPUs with ET 200M and PROFIBUS DP and CP341 or ET 200SP HA with PROFINET and CM PtP.
- The release of the IM153-2 module has to be greater than V3.
- The release of the IM155-6-PN-HA IM153-2 module has to be greater than or equal V1.2.

2.1.3 Software

- The blocks are released for the following versions of PCS7:
  - PCS7 V7.1 SP2 to V8.2 SP1 (S7HMODM3 V4.9)
  - PCS7 V9.0 to V9.1 SP2 (S7HMODM3 V7.2)

  For the use with other PCS7 versions, please contact (mailto:function.blocks.industry@siemens.com).
- The usage with a S7-416 and plan-in-plan technique, is not approved

2.1.4 Application in the field with CP341

For the realization of a Modbus - Master communication via the CP 341, the communication processor will be connected via PROFIBUS DP to a S7 41x, respectively a S7-417 H for the redundant version.

For the realization of a Modbus - Master communication via the CM PtP, the communication processor will be connected via PROFINET IO to a S7 41x, respectively a S7-41x H for the redundant version.

As bus physics for the MODBUS link RS 232C, and RS 422/485 are supported.

A detailed description of required function blocks is available in chapter 5: "Detailed description of the function blocks".

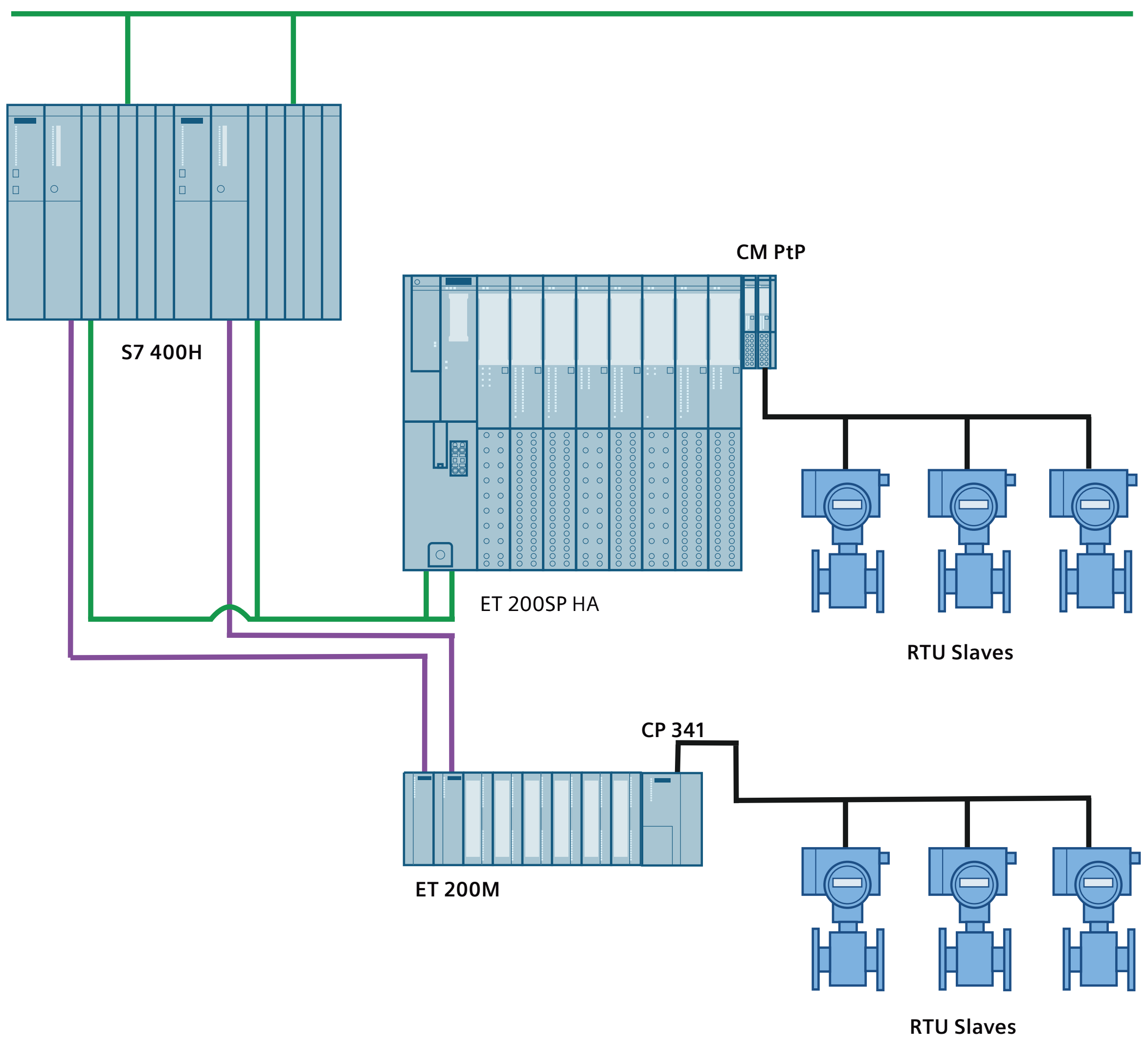
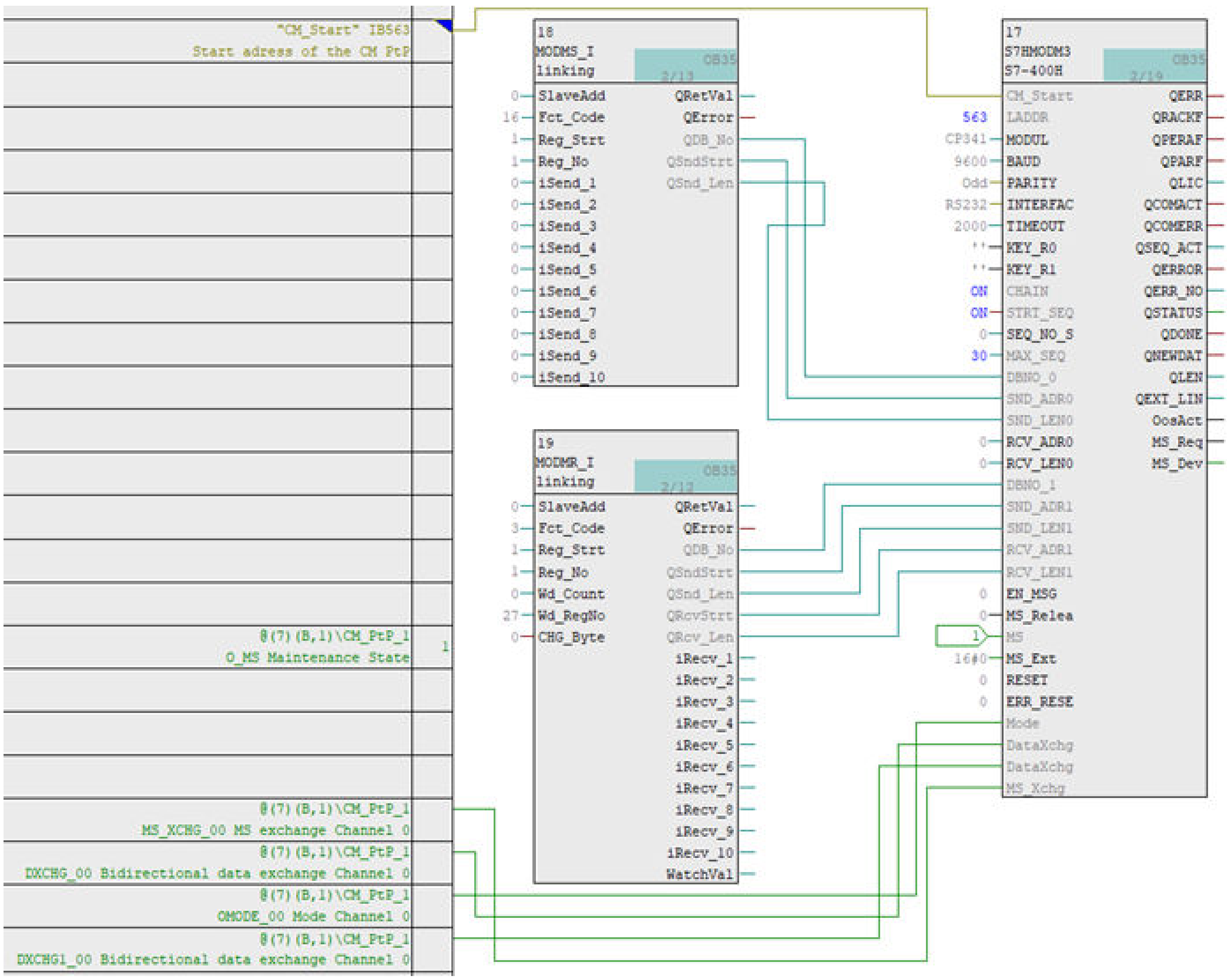


Figure 2-1 Link of a Modbus-Slave at a S7- or S7H-System

The communication block "S7HMODM3" (FB240) on the CPU41x handles up to 10 jobs (receive data or send data), sequential with up to 254 Bytes for each job. With attached blocks like "MODMR_I" (FB242) and "MODMS_I" (FB243), the addressing and interpretation of the partner data is done. Using Modbus, a maximum of 2040 coils (Bool = 1 Bit) or 127 registers (Integer = 16 Bit) can be transmitted.



2.1.5 Restrictions

A maximum number of one S7HMODM3 block, 9 EXSQ_MOD blocks and 100 read or write parameters are possible per CP. The number of possible parameters can be fewer, if large data packets are to be read or written. The execution time may also deeply vary according to the installation, the type of Modbus slaves, the settings, and the quantity of data to be exchanged. The total amount of CM PtPs/CP341 in one PN IO/Profibus DP subsystem must not exceed 8, due to available resources of the S7 400 PLC.

2.2 Security concept

The physical access to the CPU has to be restricted (mounting inside a locked cabinet). Unauthorized people shouldn’t get access to CPU or its connectors.

Both the overlay systems (like WinCC) and the underlayed systems (like IEDs) must implement protections and inter-locks to avoid damage of property or personal damage. These protections and interlocks must be done inside the single systems with the adequate configuration tool (like DIGSI).

Direct connection of the CPU to the internet has to be avoided. In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions constitute one element of such a concept. For more in-formation about industrial security, please visit (http://www.siemens.com/industrialsecurity).

To increase the plant security, Siemens provides several Industrial Security Services (https://new.siemens.com/global/en/products/services/industry/digital-industry-services/industrial-security-services.html). Please check which service is applicable for your project.

2.3 Qualified Personnel

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

2.4 Note of usage and misuse

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

2.5 Prevention of misconfiguration

A misconfiguration of configured tags or the hardware properties can lead to malfunctions, including:

- PLC not reachable
- Sporadic data loss (e.g. causes by wrong engineering parameters)
- Permanent data loss (e.g. by wrong configured tags)

To reduce this risk, a detailed signal loop test is highly recommended before commissioning.

2.6 Disclaimer

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

These blocks fulfil the functionality described in this document. Older releases of this document are invalid. Functionality which is not described or is explicit marked as "not supported" is not supported by this product or software release.

It is possible that superior products (like PCS 7 or STEP7) provide a system standard functionality. If this functionality isn't handled inside this document, the system behavior of this function blocks can differ from the superior system.
For more information about this topic, please contact (function.blocks.industry@siemens.com).

2.7 System hardening

To increase the security of the plant, we recommend a system hardening of this system and all other systems (like IEDs). Due to a large variety of needs/requirements, it is not possible to suggest a concrete hardening strategy.

For this device, please refer to "PCS 7 compendium f" and consider this with responsible and qualified personnel of your plant. System hardening requires a deep knowledge about the guidelines, environment and needs on site. For this reason, system hardening isn’t covered by the service agreement. Please note a wrong hardening configuration can impact the functionality.

2.8 Integrity of delivery

Please ensure the integrity of the delivered components, when the hardware is arrived and unpacked. To ensure the integrity, we seal the packaging of hardware components and the storage volumes with the following seal.

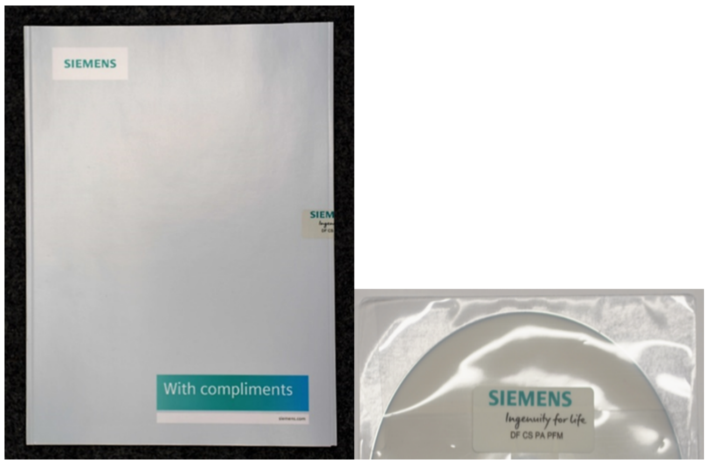


Figure 2-2 **Valid seal on software shipping package**

Valid seals

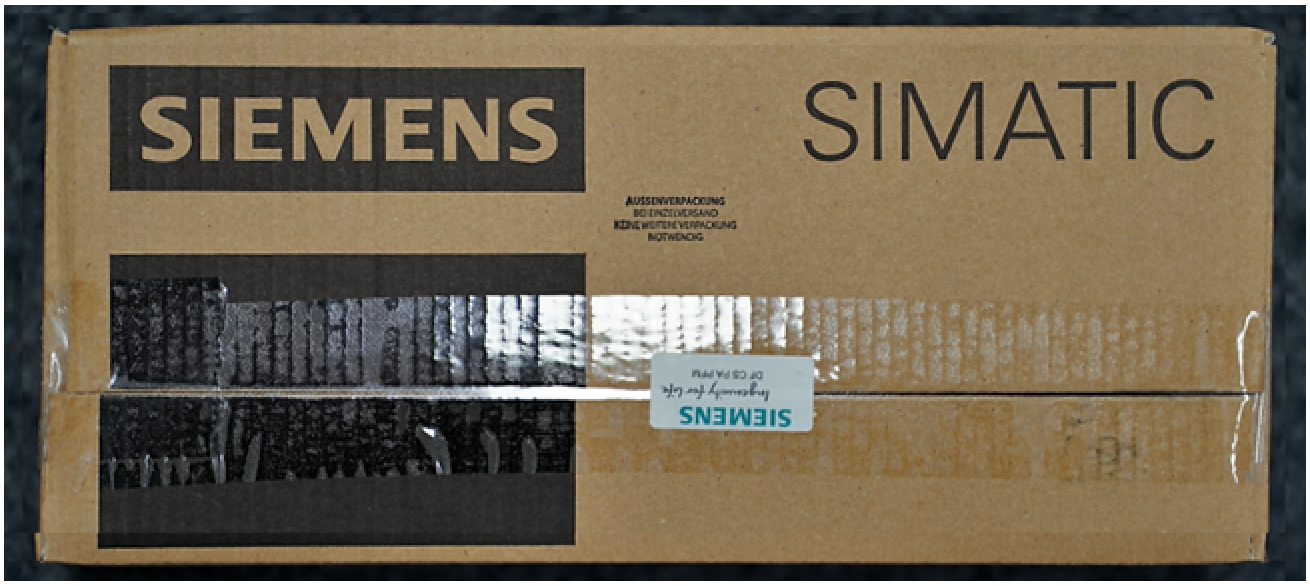


Figure 2-3 Valid seal on hardware shipping package

Invalid seals

Invalid seals can be identified by:

- A removed seal leaves a pattern on the underground. If the seal is placed again, that pattern is missing in background.
- The corners are damages from removing the seal.
- The seal was cut.

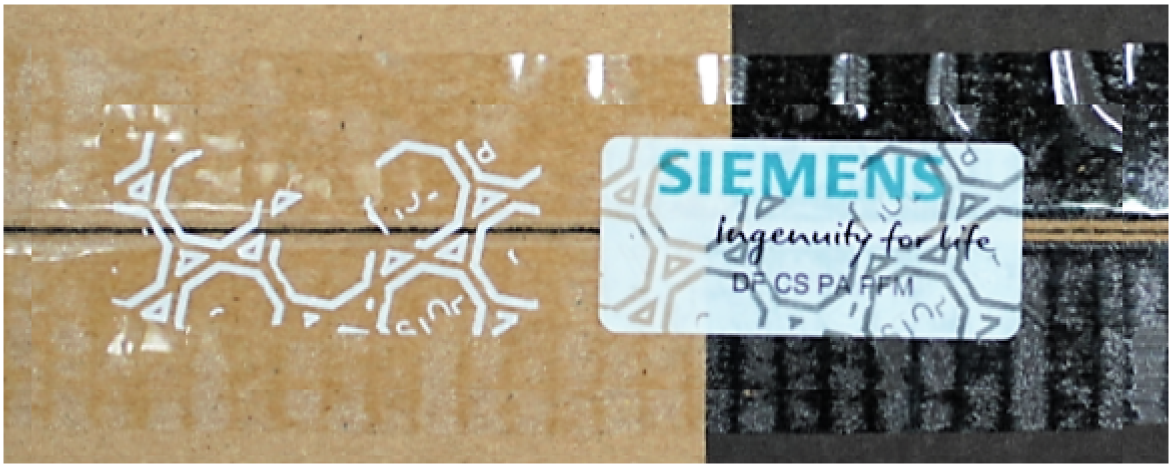


Figure 2-4 Invalid seal, seal was removed and replaced at a new position

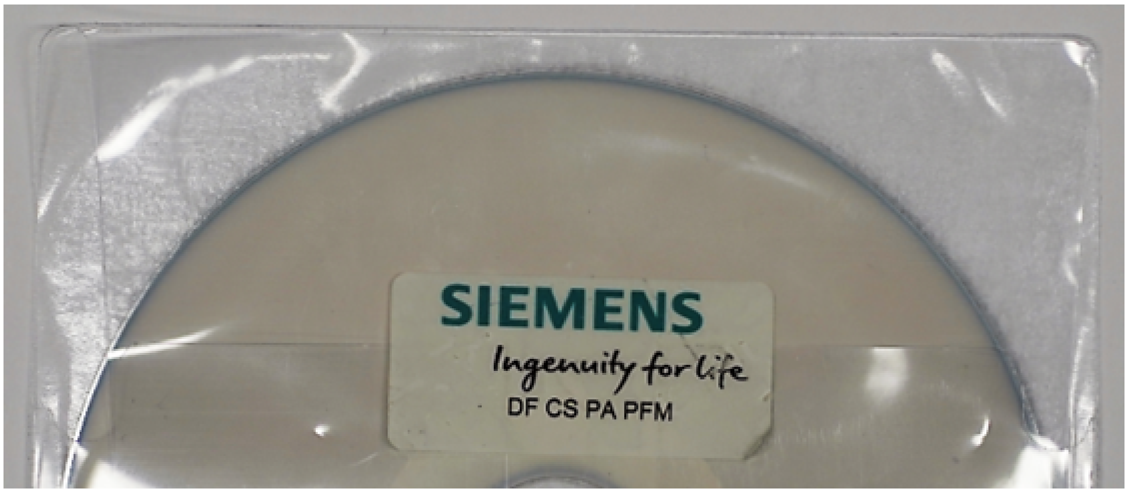


Figure 2-5 Invalid seal, seal was removed and replaced at the same position

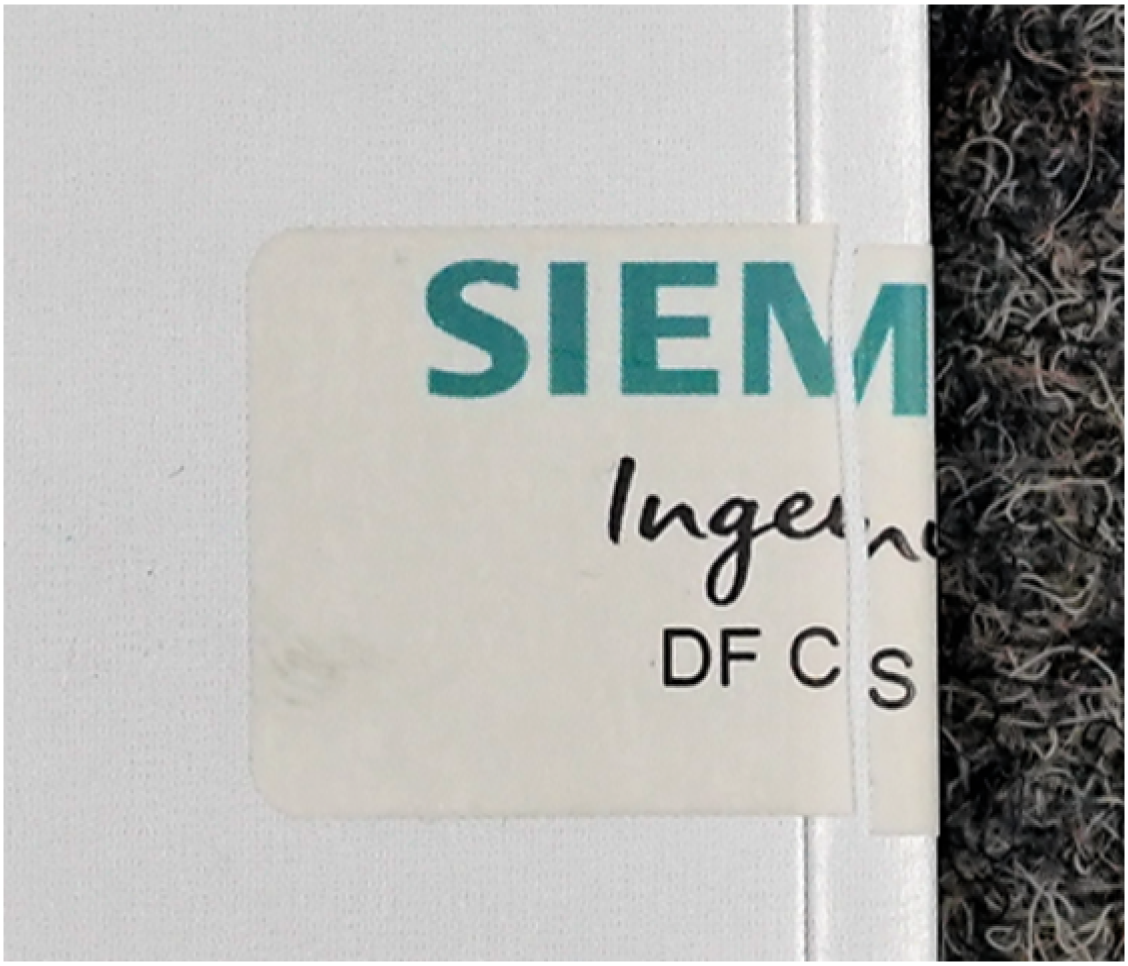


Figure 2-6 Invalid seal, seal was cut

Please ensure that all components contain a valid seal. If the seals are missing or broken, do not connect the device or data volume. In this case, please contact us!

2.9 Personal data disclaimer

Siemens observes the principles of data protection, in particular the principle of data minimization (privacy by design). For this reason, the product only processes / stores technical functional data (e.g. time stamps) and no personal data. If the user links this data with other data (e.g. shift plans) or stores personal data on the same medium (e.g. hard disk) and thus establishes a personal reference, the user must ensure compliance with data protection regulations.

2.10 Updates

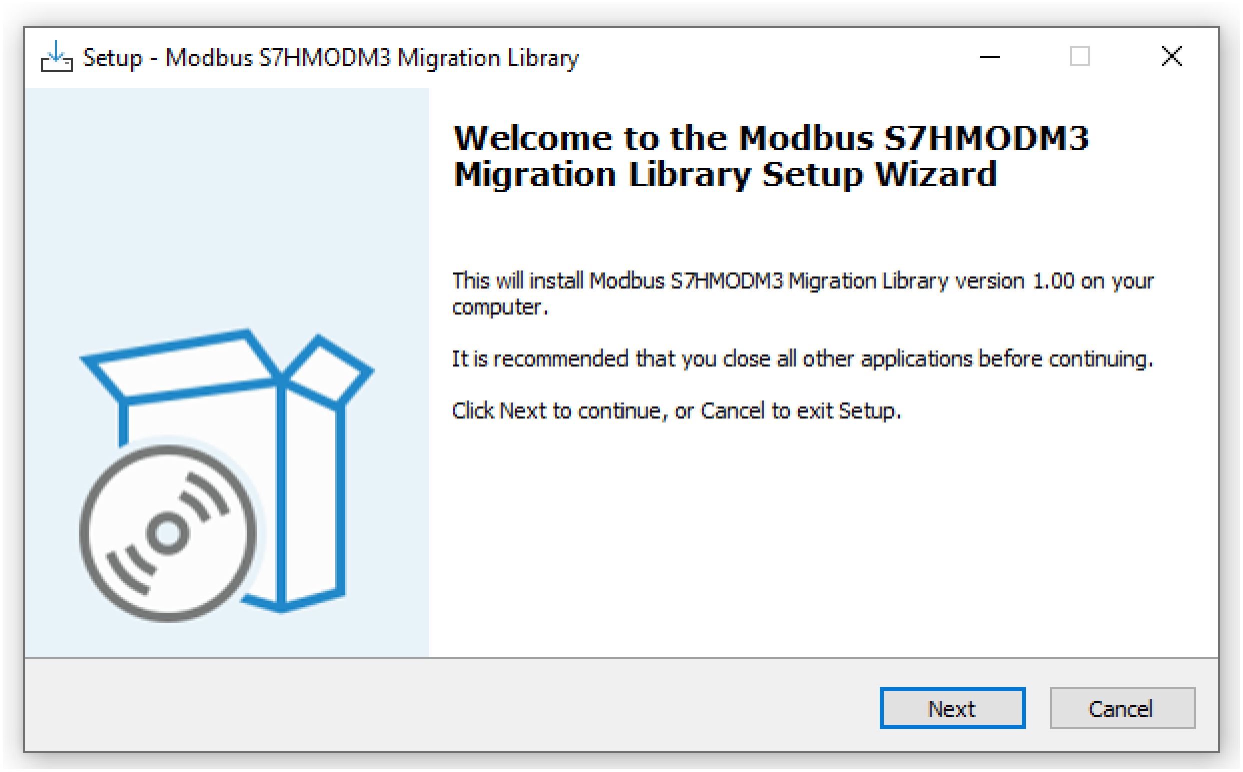
Updates are published on the Siemens SIOS webpage. Alternatively, updates can be requested on our hotline. Depending on update cause (feature extensions, bug fixing, closing security issues, ...), the installation of the update should be considered. The availability of security updates must be checked permanently, to ensure a proper and secure operation.

Installation / Configuration 3

3.1 Installation

The library is installed by the installation file.

The installation file will guide you through the following steps.


Setup - Modbus S7HMODM3 Migration Library

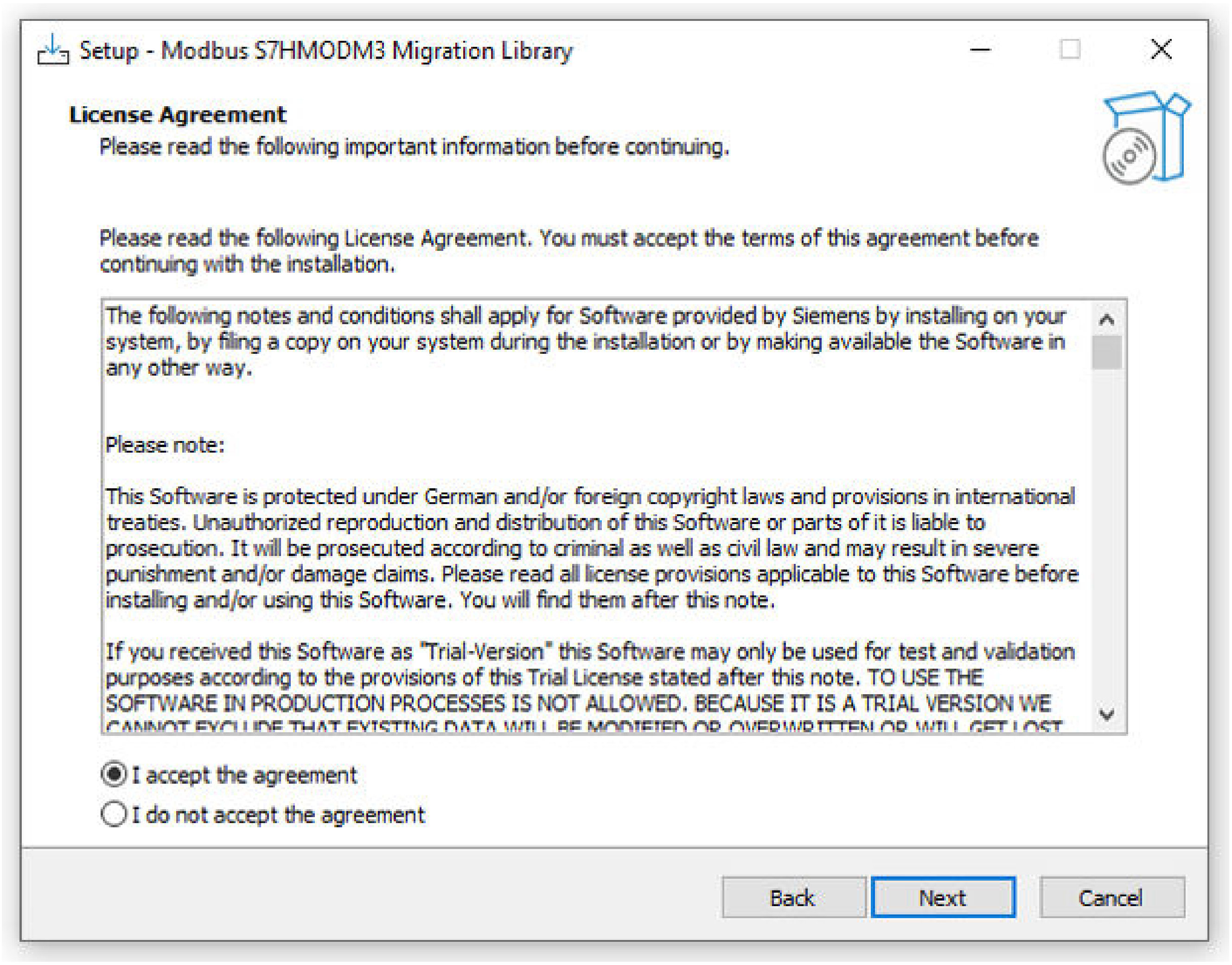
Welcome to the Modbus S7HMODM3 Migration Library Setup Wizard

This will install Modbus S7HMODM3 Migration Library version 1.00 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

Figure 3-1 Installation – Start screen



License Agreement

Please read the following important information before continuing.

Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

The following notes and conditions shall apply for Software provided by Siemens by installing on your system, by filing a copy on your system during the installation or by making available the Software in any other way.

Please note:

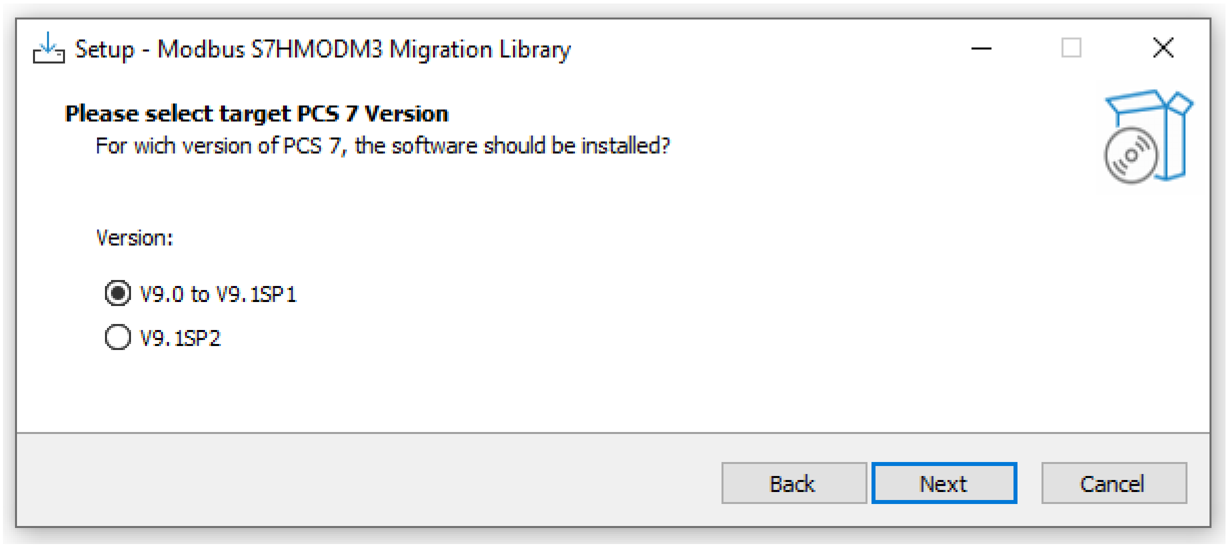
This Software is protected under German and/or foreign copyright laws and provisions in international treaties. Unauthorized reproduction and distribution of this Software or parts of it is liable to prosecution. It will be prosecuted according to criminal as well as civil law and may result in severe punishment and/or damage claims. Please read all license provisions applicable to this Software before installing and/or using this Software. You will find them after this note.

If you received this Software as "Trial-Version" this Software may only be used for test and validation purposes according to the provisions of this Trial License stated after this note. TO USE THE SOFTWARE IN PRODUCTION PROCESSES IS NOT ALLOWED. BECAUSE IT IS A TRIAL VERSION WE CANNOT EXCLUDE THAT EXISTING DATA WILL BE MODIFIED OR OVERWRITTEN OR WILL GET LOST

[✓ I accept the agreement]

I do not accept the agreement

Figure 3-2 Installation – License Agreement



Setup - Modbus S7HMODM3 Migration Library

Please select target PCS 7 Version

For wich version of PCS 7, the software should be installed?

Version:

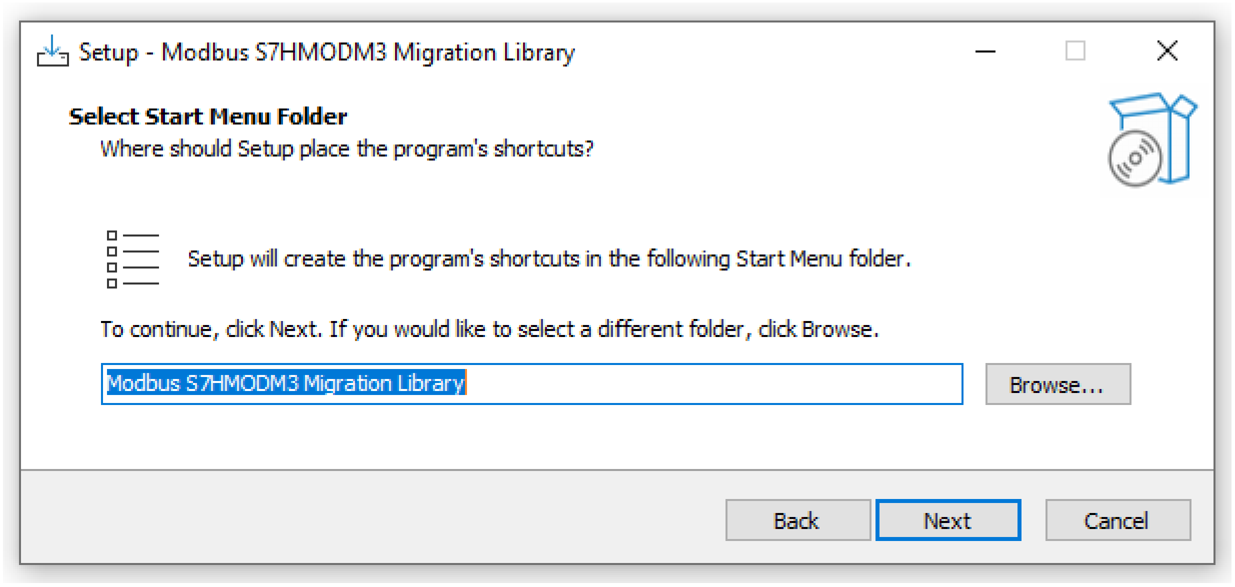
[✓ V9.0 to V9.1SP1]

V9.1SP2

Figure 3-3 Installation – Version of PCS 7

Note

For updating PCS 7 from a version earlier than PCS 7 V9.1SP2 to version V9.1SP2 or higher first you need to uninstall the existing version of Modbus Advanced CM PtP. This can be done with the uninstall function of windows.



Setup - Modbus S7HMODM3 Migration Library

Select Start Menu Folder

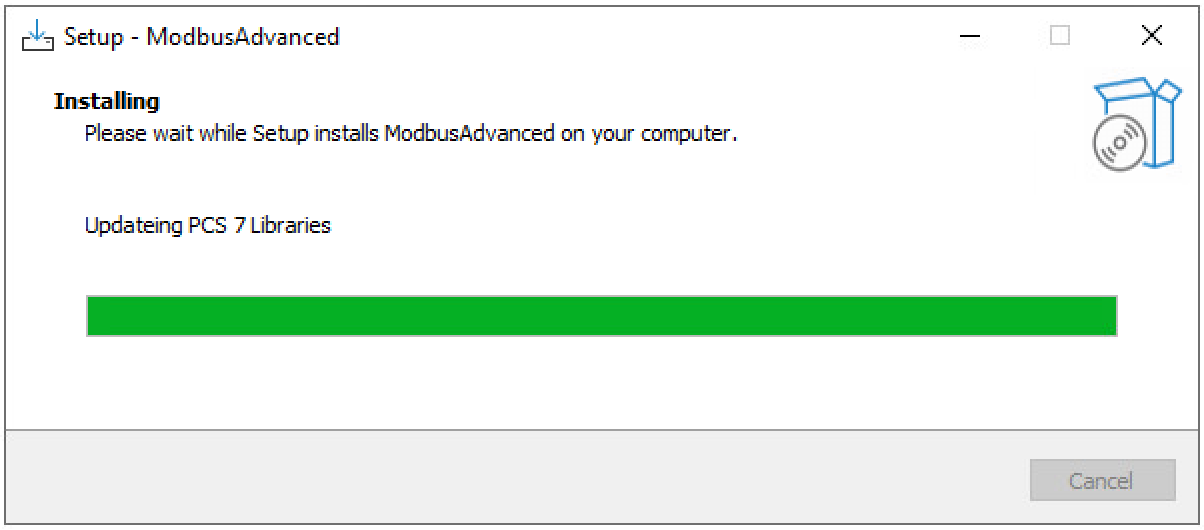
Where should Setup place the program's shortcuts?

Setup will create the program's shortcuts in the following Start Menu folder.

To continue, click Next. If you would like to select a different folder, click Browse.

Modbus S7HMODM3 Migration Library

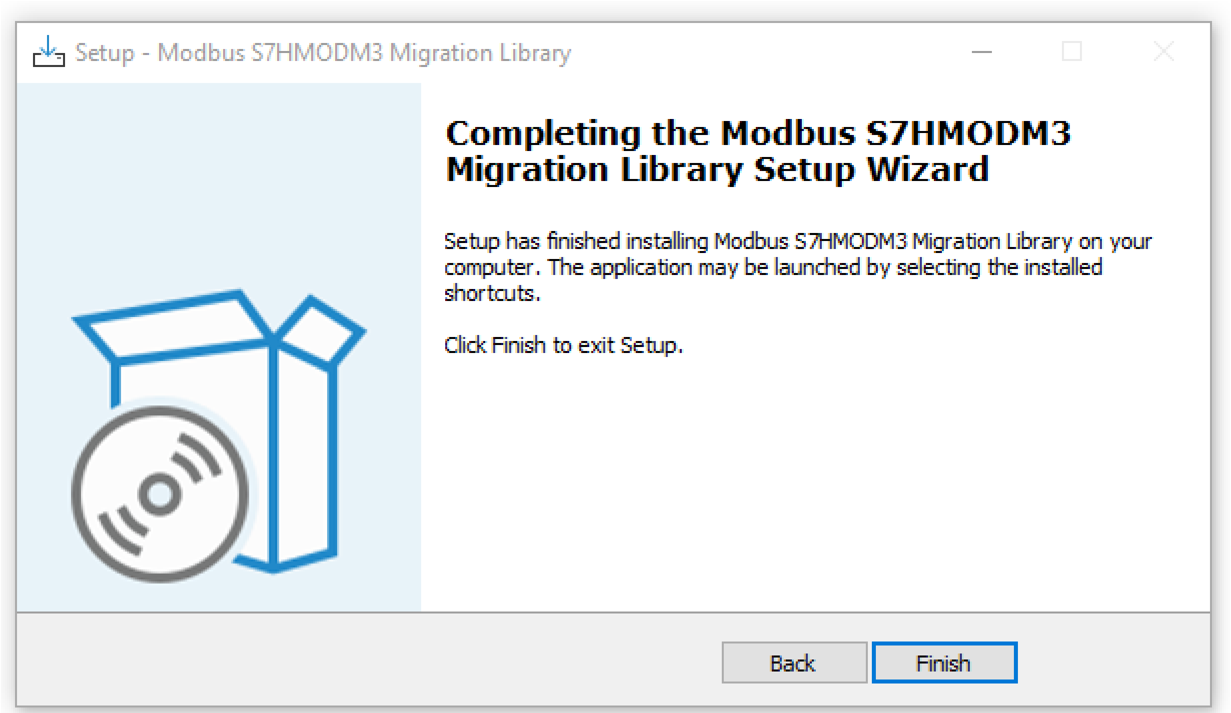
Figure 3-4 Installation - Path


Setup - ModbusAdvanced

Installing
Please wait while Setup installs ModbusAdvanced on your computer.

Updateing PCS 7 Libraries

Figure 3-5 Installation - Process

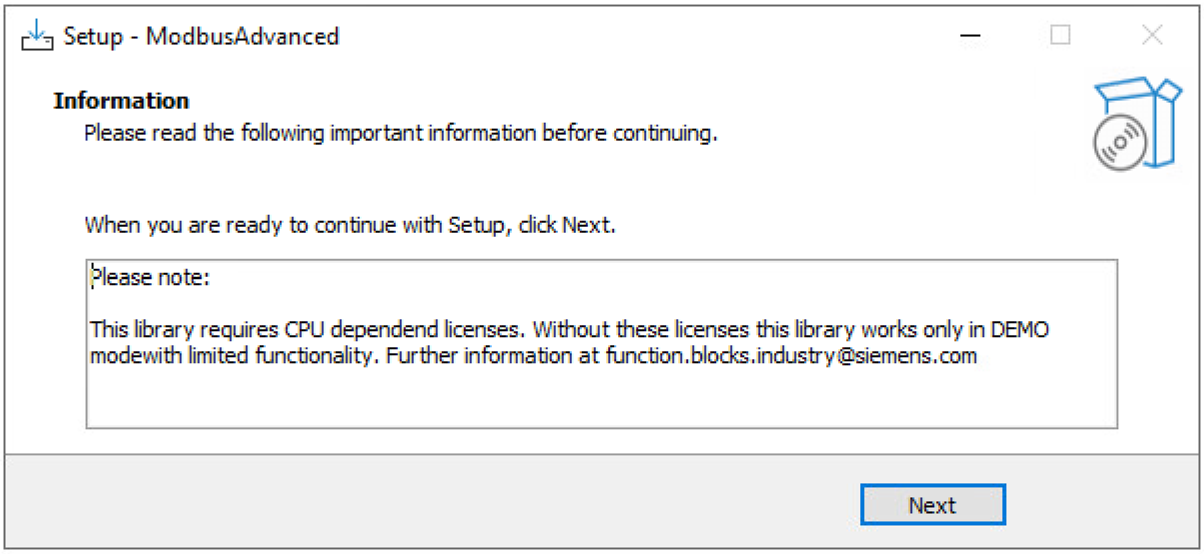

Setup - Modbus S7HMODM3 Migration Library

Completing the Modbus S7HMODM3 Migration Library Setup Wizard

Setup has finished installing Modbus S7HMODM3 Migration Library on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.

Figure 3-6 Installation successful



Information

Please read the following important information before continuing.

When you are ready to continue with Setup, click Next.

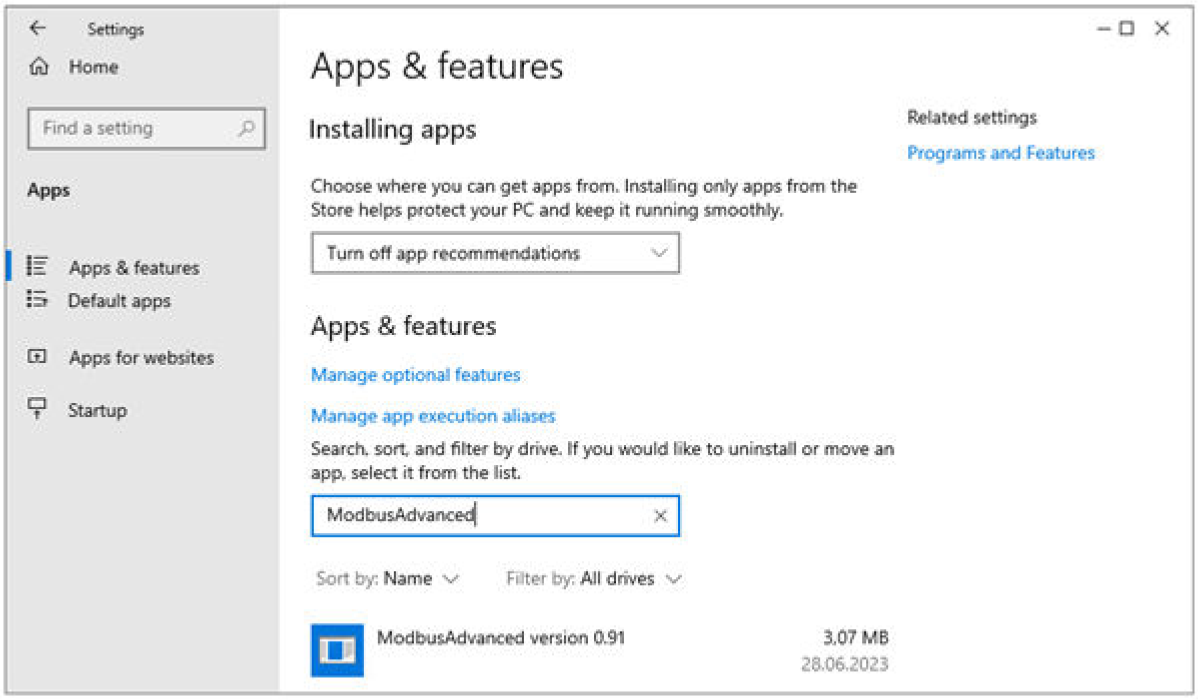
Please note:

This library requires CPU dependend licenses. Without these licenses this library works only in DEMO modewith limited functionality. Further information at function.blocks.industry@siemens.com

Figure 3-7 Installation – License information

3.2 Uninstallation

For uninstallation, please proceed as follow.



Apps & features

Installing apps

Choose where you can get apps from. Installing only apps from the Store helps protect your PC and keep it running smoothly.

Turn off app recommendations

Apps & features

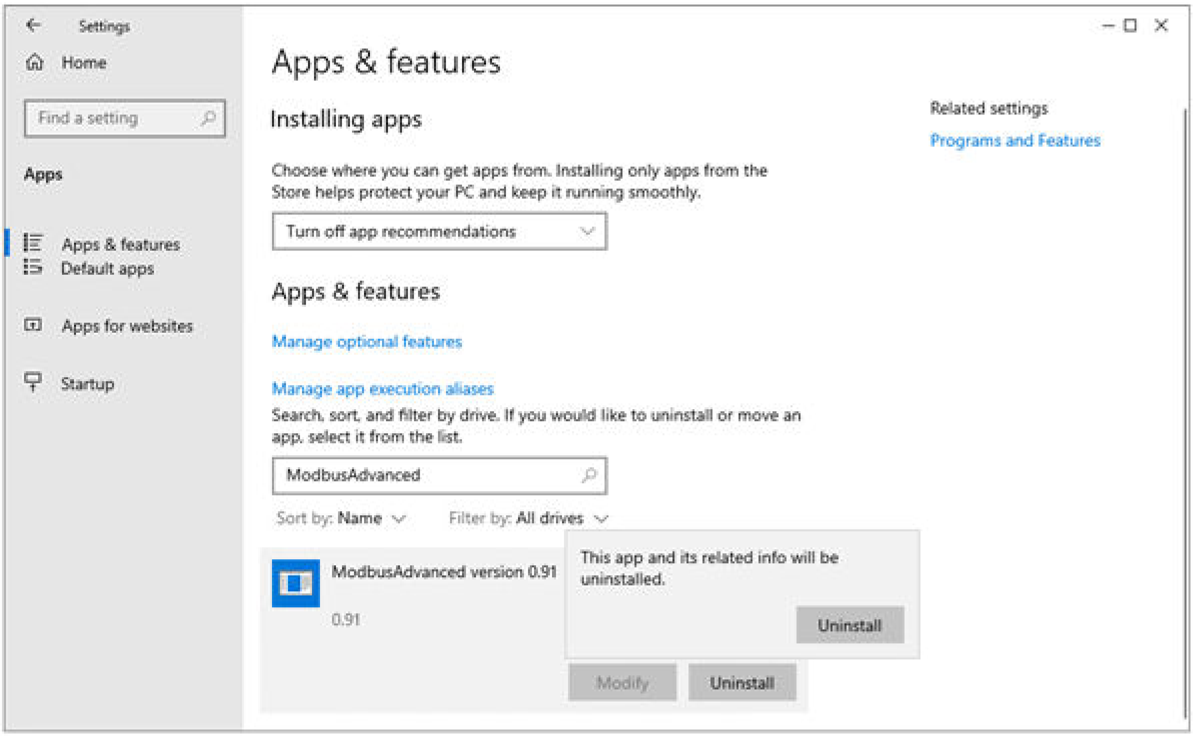
Search, sort, and filter by drive. If you would like to uninstall or move an app, select it from the list.

ModbusAdvanced

Sort by: Name Filter by: All drives

ModbusAdvanced version 0.91 3.07 MB 28.06.2023

Figure 3-8 Uninstallation – Selecting App

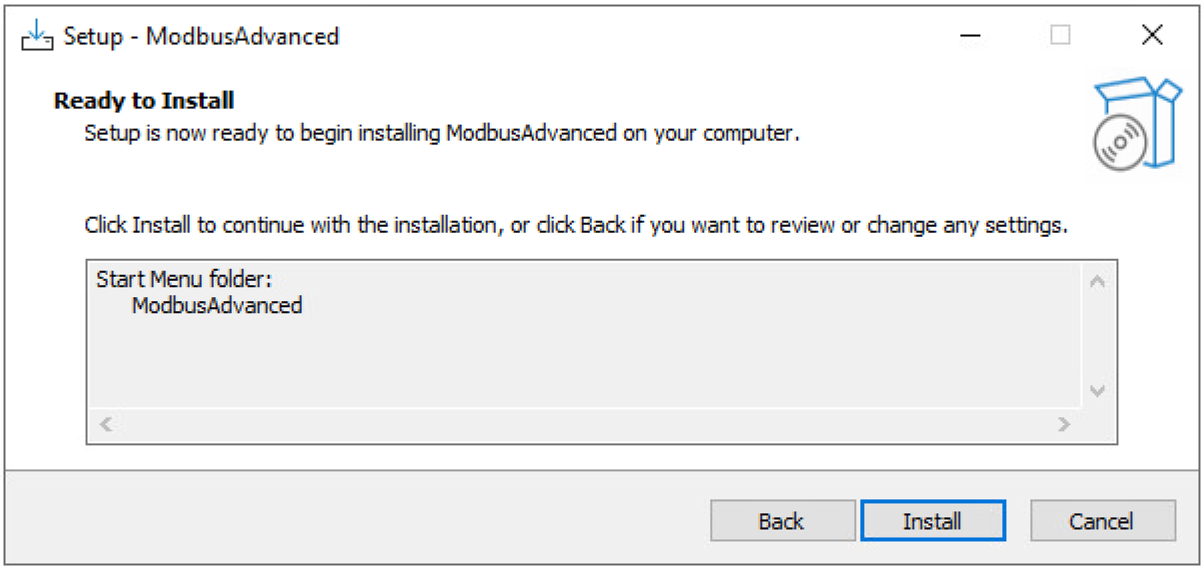


Choose where you can get apps from. Installing only apps from the Store helps protect your PC and keep it running smoothly.

Search, sort, and filter by drive. If you would like to uninstall or move an app, select it from the list.

This app and its related info will be uninstalled.

Figure 3-9 Uninstallation – Modify to uninstall



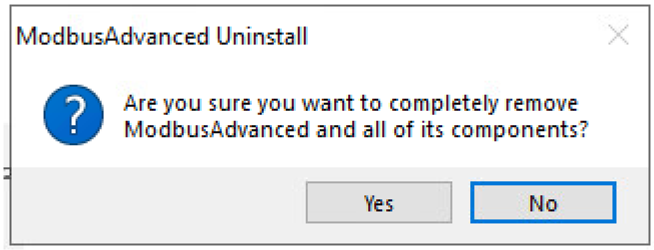
Ready to Install

Setup is now ready to begin installing ModbusAdvanced on your computer.

Click Install to continue with the installation, or click Back if you want to review or change any settings.

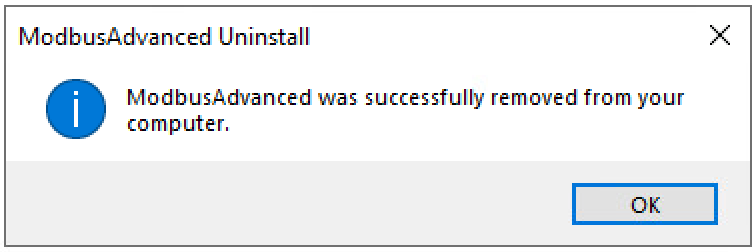
Start Menu folder:
ModbusAdvanced

Figure 3-10 Uninstallation – Proceeding uninstallation


ModbusAdvanced Uninstall

Are you sure you want to completely remove ModbusAdvanced and all of its components?

Figure 3-11 Uninstallation – Confirm uninstallation


ModbusAdvanced Uninstall

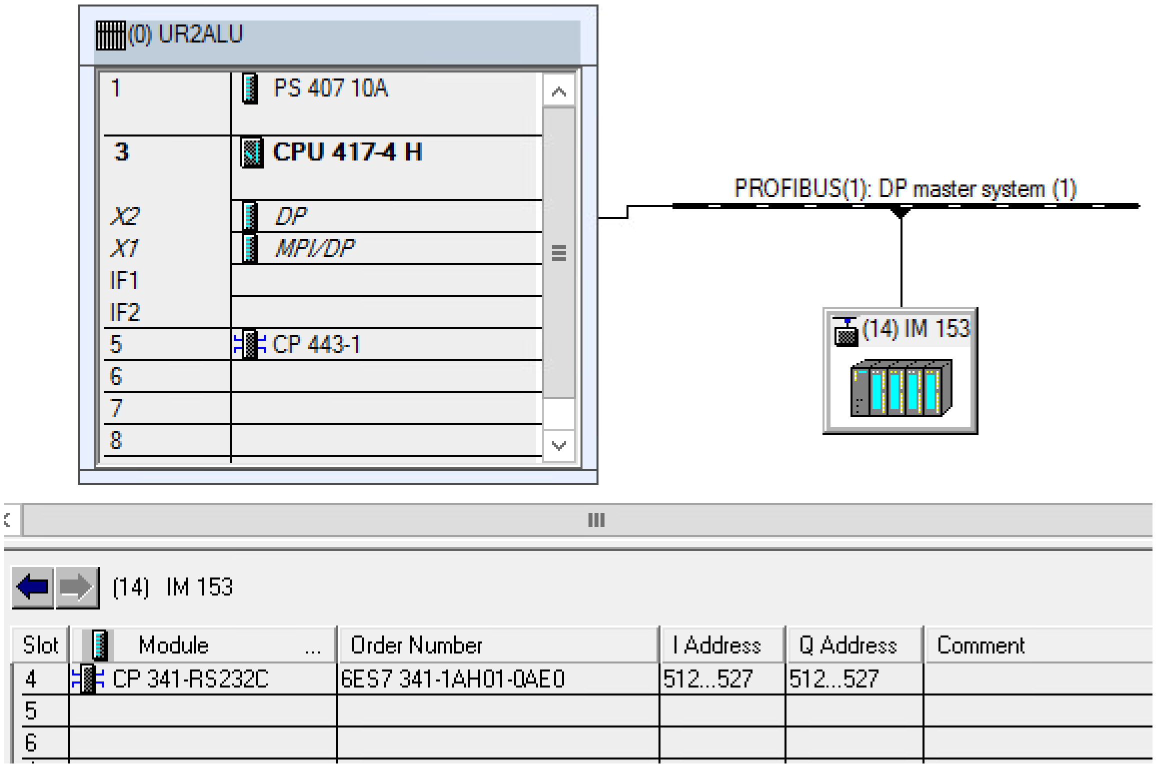
ModbusAdvanced was successfully removed from your computer.

Figure 3-12 Uninstallation – Feedback for successfully uninstallation

3.3 Hardware configure

- The actual HW-configuration must be loaded in the CPU of the S7-41x (H)
- The communication partner must be connected electrically correct.
- The Input and Output addresses must be the same.

Example CP341



(0) UR2ALU

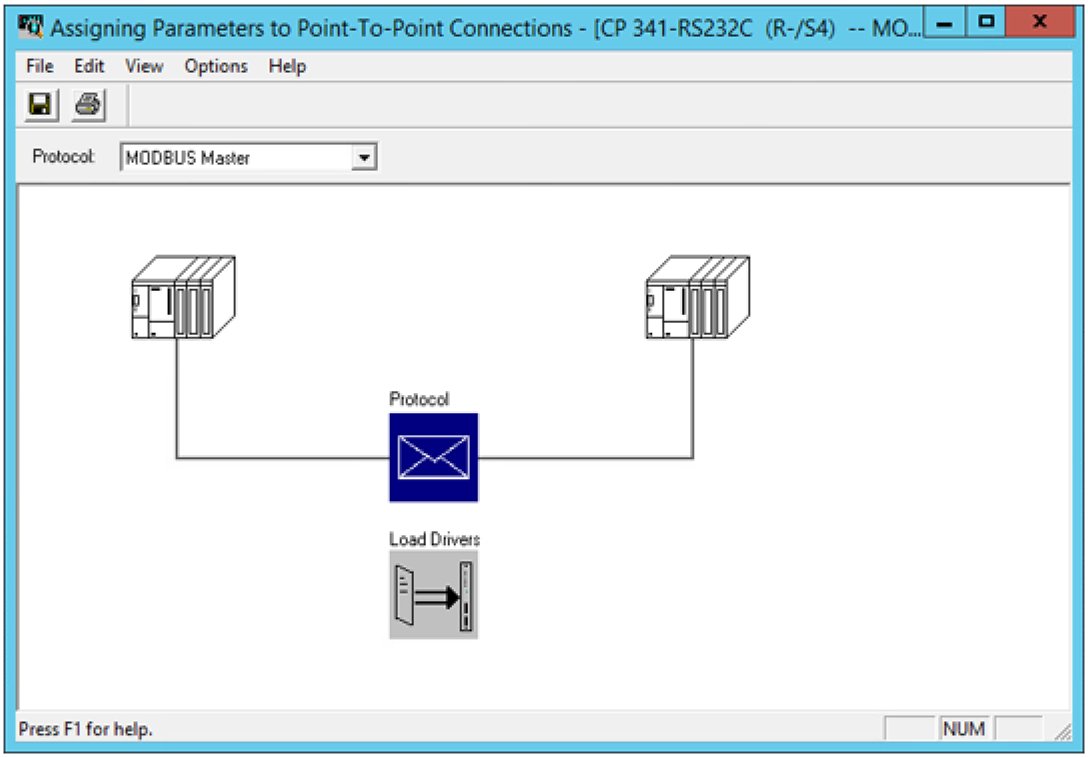
1	PS 407 10A
3	**CPU 417-4 H**
X2	*DP*
X1	*MPI/DP*
IF1	
IF2	
5	CP 443-1
6	
7	
8	

(14) IM 153

Slot	Module ...	Order Number	I Address	Q Address	Comment
4	CP 341-RS232C	6ES7 341-1AH01-0AE0	512...527	512...527	
5					
6					

Figure 3-13 Example of a HW configuration

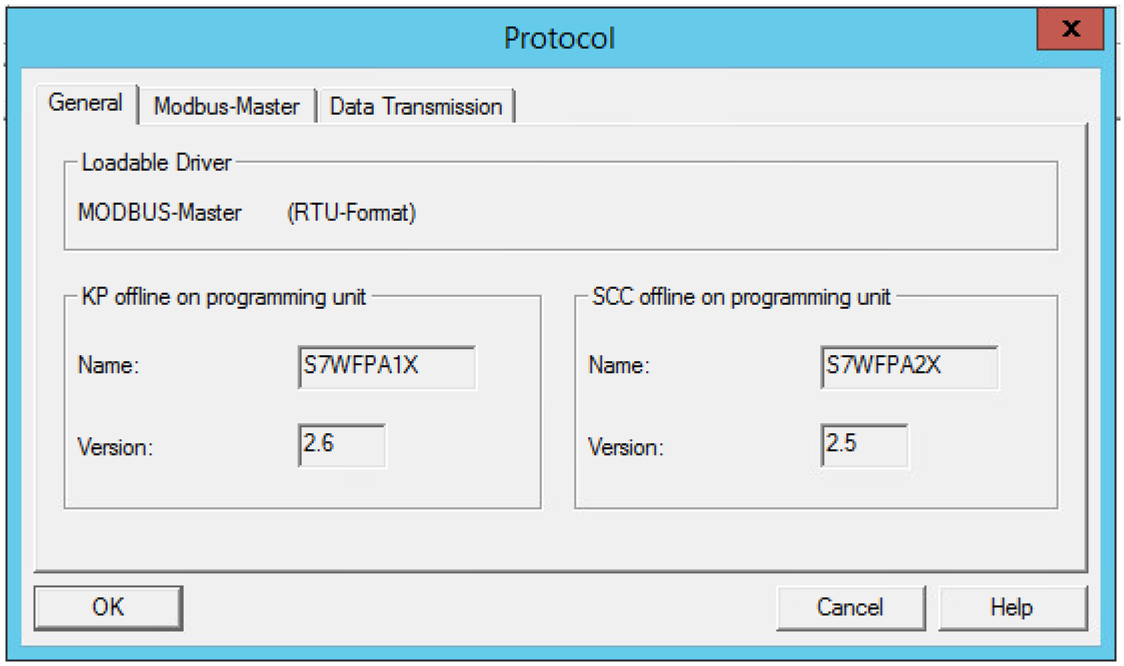
The bus physics are defined by the order number of the CP341.


Assigning Parameters to Point-To-Point Connections - [CP 341-RS232C (R-/S4) -- MO...
File Edit View Options Help
Protocol
Load Drivers
Press F1 for help.
NUM

Protocol: MODBUS Master

Figure 3-14 Parameterization – Starting mask for the CP341 as Modbus Master

- Configure the protocol with "MODBUS Master". You can download the driver at (https://support.industry.siemens.com/cs/document/27774018/loadable-driver-modbus-master-(rtu)).
- Open the protocol window in order to configure the Modbus communication. The settings are explained in the following paragraph.

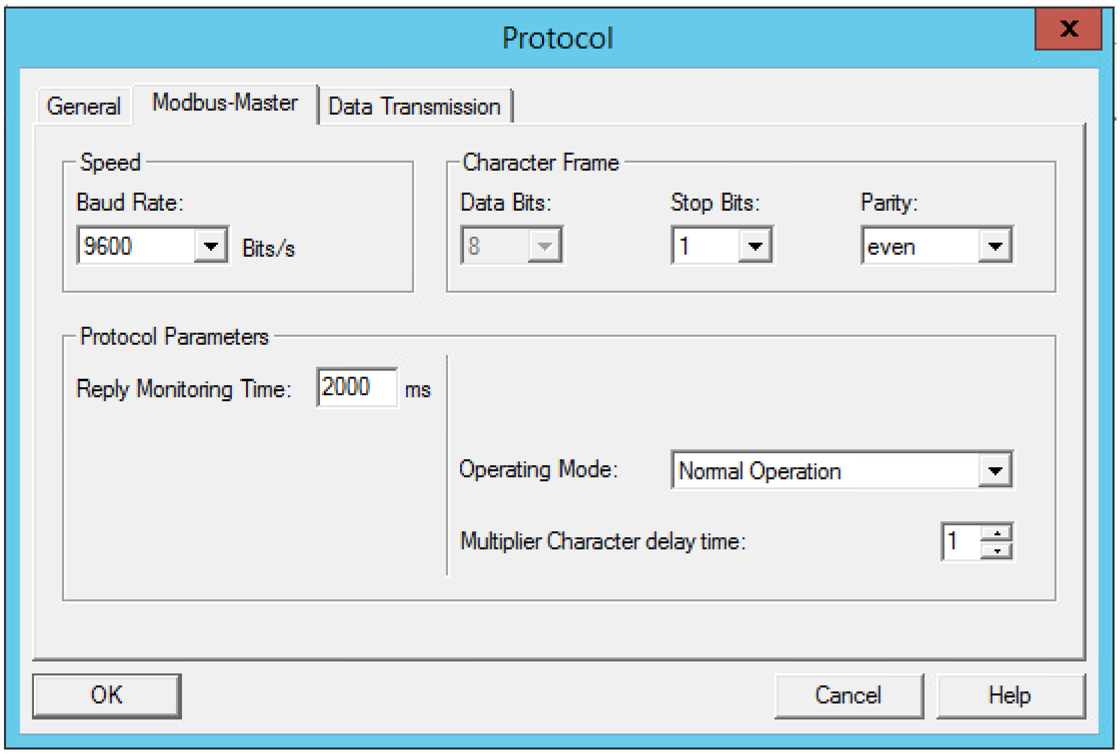


Loadable Driver
MODBUS-Master (RTU-Format)

KP offline on programming unit
Name: S7WFPA1X
Version: 2.6

SCC offline on programming unit
Name: S7WFPA2X
Version: 2.5

Figure 3-15 Protocol of the Loadable Driver



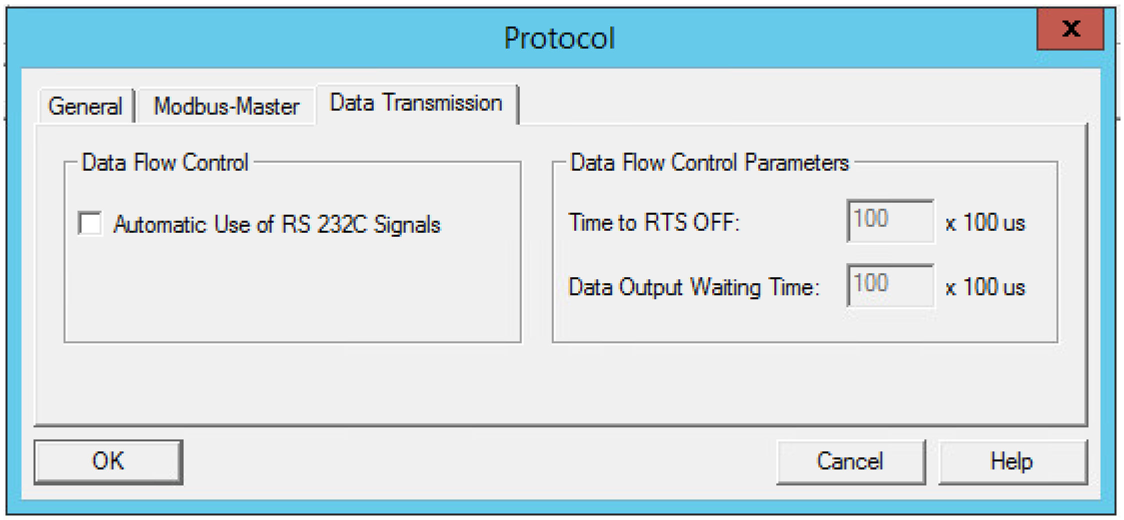
Speed
Baud Rate: 9600 Bits/s

Character Frame
Data Bits: 8
Stop Bits: 1
Parity: even

Protocol Parameters
Reply Monitoring Time: 2000 ms
Operating Mode: Normal Operation
Multiplier Character delay time: 1

Figure 3-16 Protocol settings for the Modbus Master

The specific protocol settings have to be coordinated with the Modbus partner.



Protocol

General | Modbus-Master | Data Transmission

Data Flow Control

☐ Automatic Use of RS 232C Signals

Data Flow Control Parameters

Time to RTS OFF: 100 x 100 us

Data Output Waiting Time: 100 x 100 us

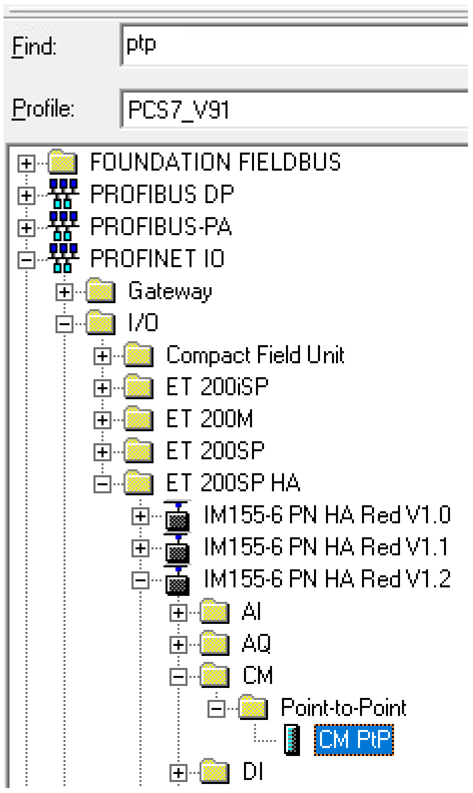
OK | Cancel | Help

Figure 3-17 Adjustment of a data flow parameter

- Load the drivers (only once).

Example CM PtP

In the SIMATIC "HW Config" the CM PtP is placed in PROFINET IO/ET 200SP HA/IM155-6 PN HA Red V1.2/CM/Point-to-Point/CM PtP



Find: ptp

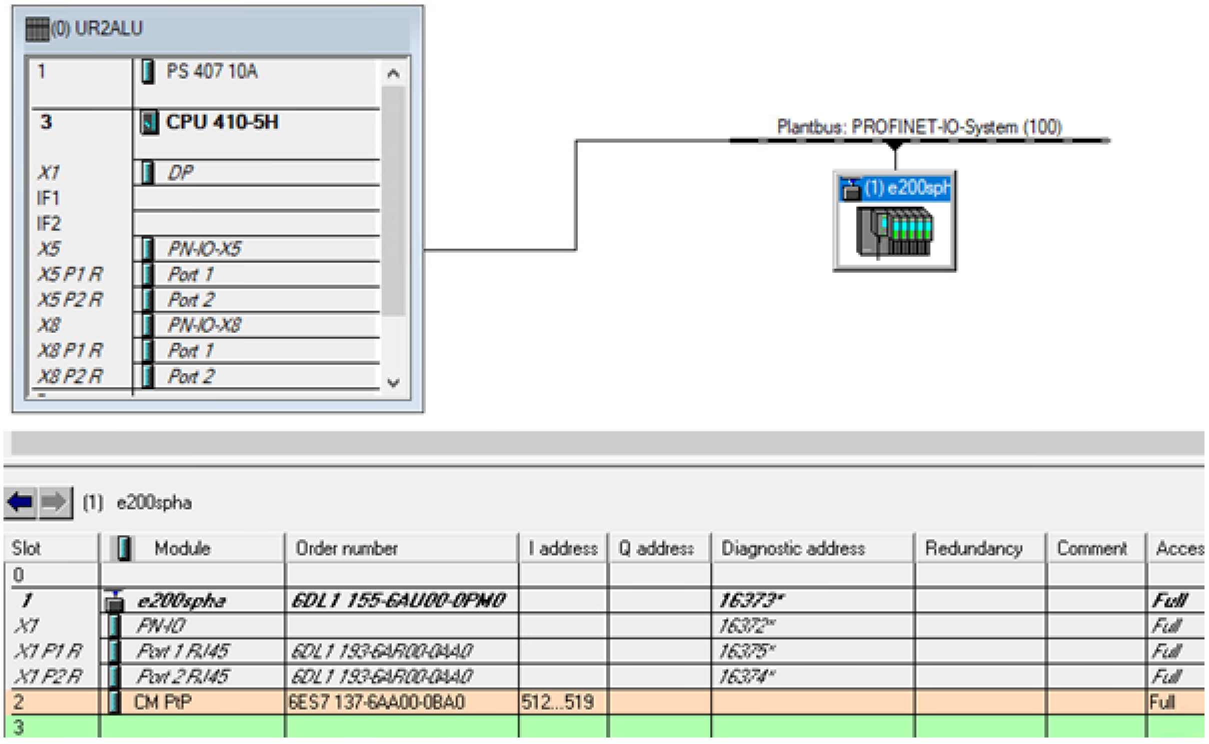
Profile: PCS7_V91

```
+ FOUNDATION FIELDBUS
+ PROFIBUS DP
+ PROFIBUS-PA
- PROFINET IO
   + Gateway
   - I/O
      + Compact Field Unit
      + ET 200iSP
      + ET 200M
      + ET 200SP
      - ET 200SP HA
         + IM155-6 PN HA Red V1.0
         + IM155-6 PN HA Red V1.1
         - IM155-6 PN HA Red V1.2
            + AI
            + AQ
            - CM
               - Point-to-Point
                    CM PtP
            + DI
```

Figure 3-18 HW config catalogue view of CM PtP

Note

If the CM PtP for ET 200SP HA is not available in the HW catalogue of SIMATIC "HW Config" it can be installed via hardware support package (HSP). After the integration via HSP the profile of the catalogue needs to be changed from “PCS7_Vxx” to “Standard”.



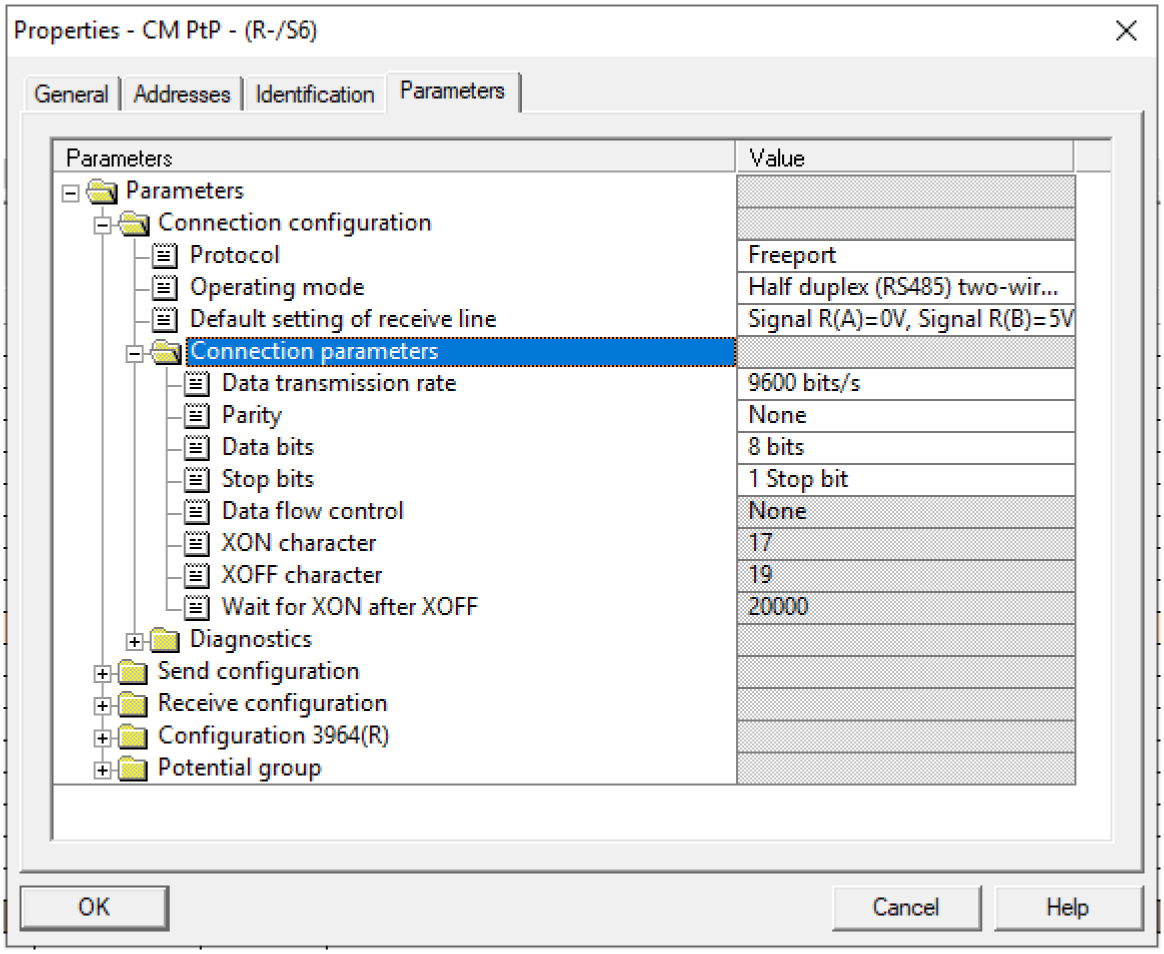
1	PS 407 10A
3	CPU 410-5H
X1	DP
IF1	
IF2	
X5	PN-IO-X5
X5 P1 R	Port 1
X5 P2 R	Port 2
X8	PN-IO-X8
X8 P1 R	Port 1
X8 P2 R	Port 2

(1) e200spha

Slot	Module	Order number	I address	Q address	Diagnostic address	Redundancy	Comment	Acces
0								
1	e200spha	6DL1 155-6AU00-0PM0			16373*			Full
X1	PN-IO				16372*			Full
X1 P1 R	Port 1 RJ45	6DL1 193-6AR00-0AA0			16375*			Full
X1 P2 R	Port 2 RJ45	6DL1 193-6AR00-0AA0			16374*			Full
2	CM PtP	6ES7 137-6AA00-0BA0	512...519					Full
3								

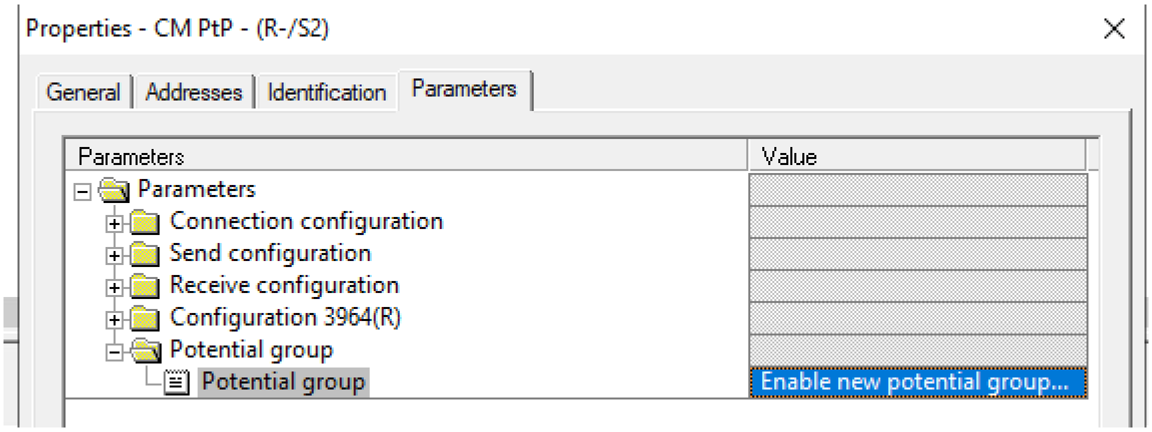
Figure 3-19 HW config ET 200SP HA with CM PtP

The necessary properties in the SIMATIC "HW Config" can be done in the properties of the CM PtP.



Parameters	Value
Parameters	
Connection configuration	
Protocol	Freeport
Operating mode	Half duplex (RS485) two-wir...
Default setting of receive line	Signal R(A)=0V, Signal R(B)=5V
Connection parameters	
Data transmission rate	9600 bits/s
Parity	None
Data bits	8 bits
Stop bits	1 Stop bit
Data flow control	None
XON character	17
XOFF character	19
Wait for XON after XOFF	20000
Diagnostics	
Send configuration	
Receive configuration	
Configuration 3964(R)	
Potential group	

Figure 3-20 HW config CM PtP example properties 1



Parameters	Value
Parameters	
Connection configuration	
Send configuration	
Receive configuration	
Configuration 3964(R)	
Potential group	
Potential group	Enable new potential group...

Figure 3-21 HW config CM PtP example properties 2

For the interconnection to the function block it is recommended to create a symbol for the first input byte of the CM PtP.

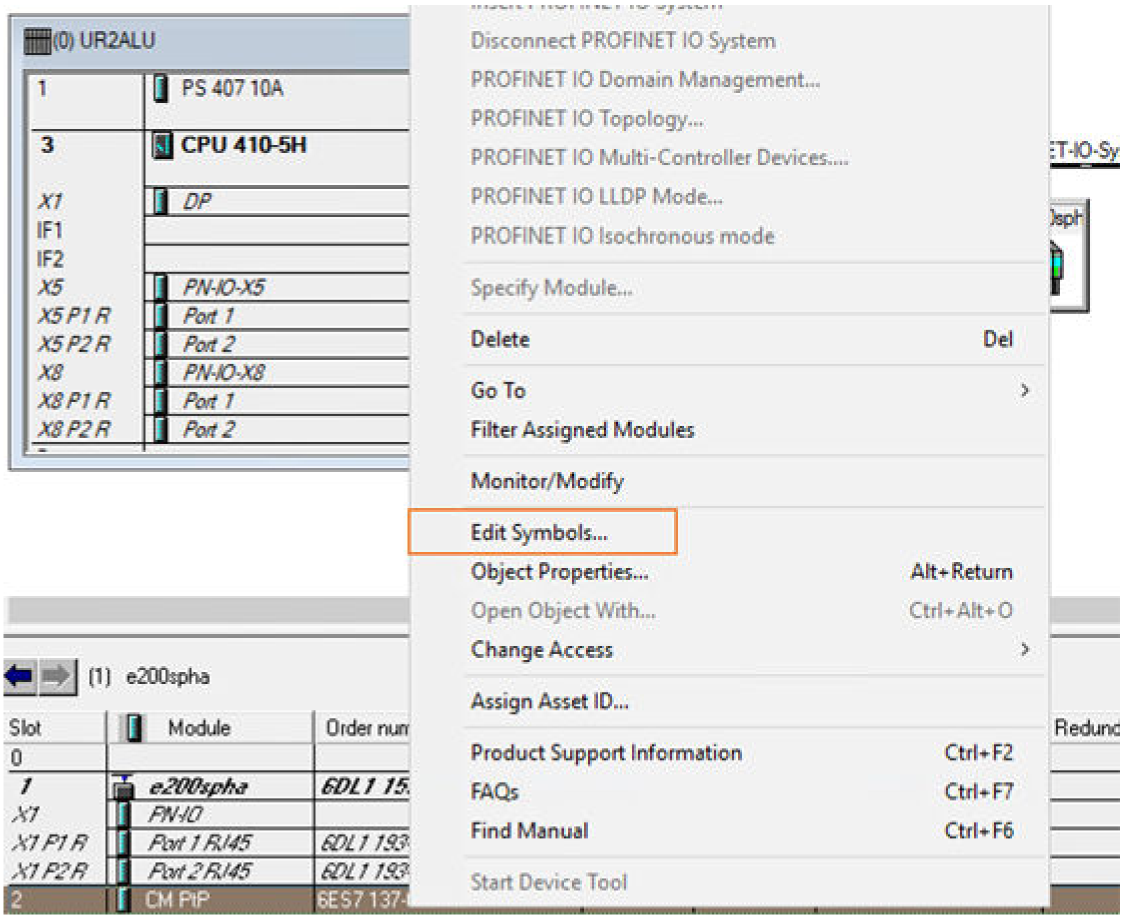
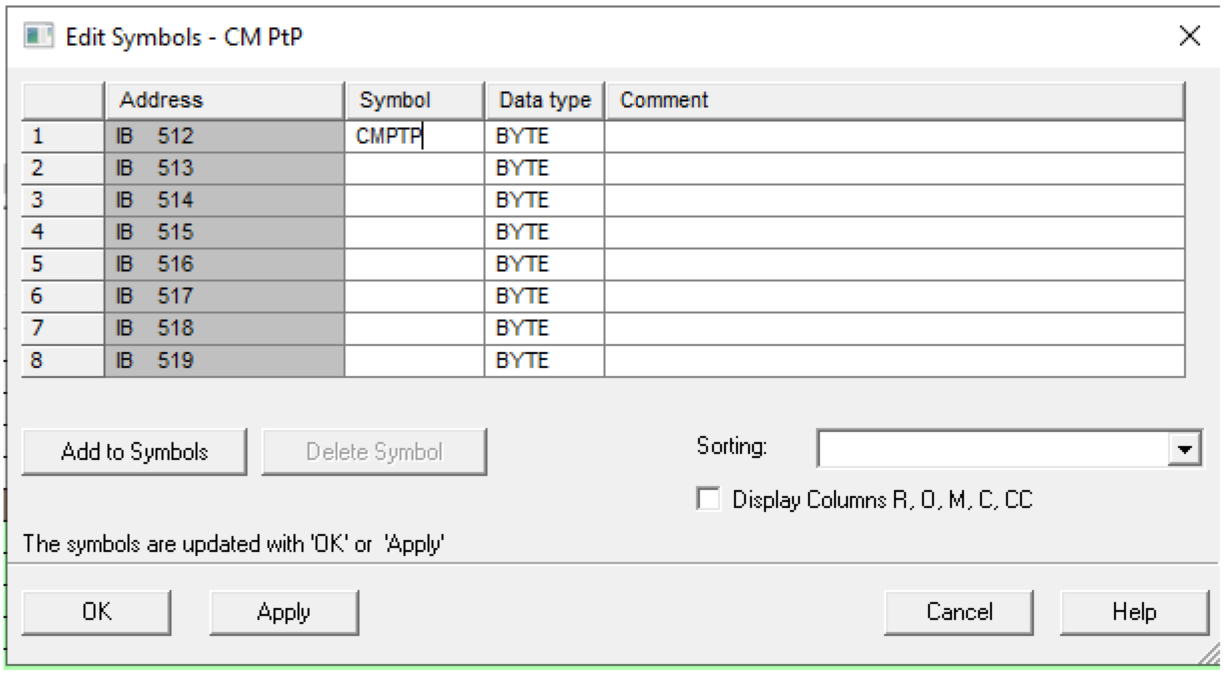


Figure 3-22 CM PtP Edit Symbols 1



Edit Symbols - CM PtP

	Address	Symbol	Data type	Comment
1	IB 512	CMPTP	BYTE	
2	IB 513		BYTE	
3	IB 514		BYTE	
4	IB 515		BYTE	
5	IB 516		BYTE	
6	IB 517		BYTE	
7	IB 518		BYTE	
8	IB 519		BYTE	

Sorting:

☐ Display Columns R, O, M, C, CC

The symbols are updated with 'OK' or 'Apply'

Figure 3-23 CM PtP Edit Symbols 2

CFC Programming 4

4.1 Principle

The Modbus communication is managed by one main function block: S7HMODM3. This main FB sends or receives the data via the CP341 resp. CM PtP by using the Modbus Point-to-Point protocol.

Each data sending or data receipt is managed by the main function block but is finalized by the following process blocks:

Process block	Task
MODMS_I	Send data coded over integer (data unit: 1 register)
MODMS_DI	Send data coded over double integer (data unit: 2 registers)
MODMS_R	Send data coded over real (data unit: 2 registers)
MODMR_I	Receive data coded over integer (data unit: 1 registers)
MODMR_DI	Receive data coded over double integer (data unit: 2 registers)
MODMR_R	Receive data coded over real (data unit: 2 registers)

The actions "Send data" or "Receive data" are named sequences or tasks. The S7HMODM3 processes each task sequentially. For each sequence, there is only one process FB (see list above) that will be used in order to process the data.

In the case of a large number of tasks or a large quantity of data to be sent or fetched, the following settings should be used in order to speed up this sequential process:

- The communication block S7HMODM3 (FB240) and extension block EXSQ_MOD shall be used in a quick cycle (i.e. 50msec: OB36), as it may take several cycles to execute one sequence.
- For the processing blocks (i.e. MODMR_R (FB242), MODMS_I (FB243) ...) a slower cycle i.e. 500msec (OB33) has to be selected.

Note

A maximum amount of one S7HMODM3 block, 9 EXSQ_MOD blocks and 100 read- or write-parameters are possible per CP. The number of possible parameters can be fewer, if many parameters with big data

4.2 Licensing

The S7HMODM3 Library requires an unique license key, for every automation system it is being used with.

The license key must be entered at the inputs "KEY_R0" and "KEY_R1" in case of a redundant S7 400, at the S7HMODM3 block.

In dynamic mode of the CFC editor, the status of the license can be observed at the output “QLIC”

Value	Text	Description
0	Lite	The Modbus library is running in demo mode. Only the first two sequences are executed.
1	Single	One AS has a valid license; full operation is possible.
2	OK	Both AS in a redundant setup are valid.

In case you are not sure which license key you need to enter, please contact our support team and provide the PLCs serial number and the libraries license number which is part of the delivery:

function.blocks.industry@siemens.com

The PLCs serial number can be identified in different ways:

- Printed on the PLCs housing
- Online diagnostic of the PLC
- Block outputs "QSNR_RX" at S7HMODM3

4.3 How to send data

To send data, the parameters QDB_No, QSndStrt and QSndLen of MODMS_I (FB243), MODMS_R or MODMS_DI have to be linked with the parameters DBNO_x, SND_ADRx and SND_LENx of S7HMODM3 (FB240).

The parameters DBNO_x, SND_ADRx and SND_LENx of S7HMODM3 transmit the information of the sequence x.

The data to be sent has to be entered via the inputs iSend_1 to iSend127 (to iSend63 for MODMS_R and MODMS_DI).

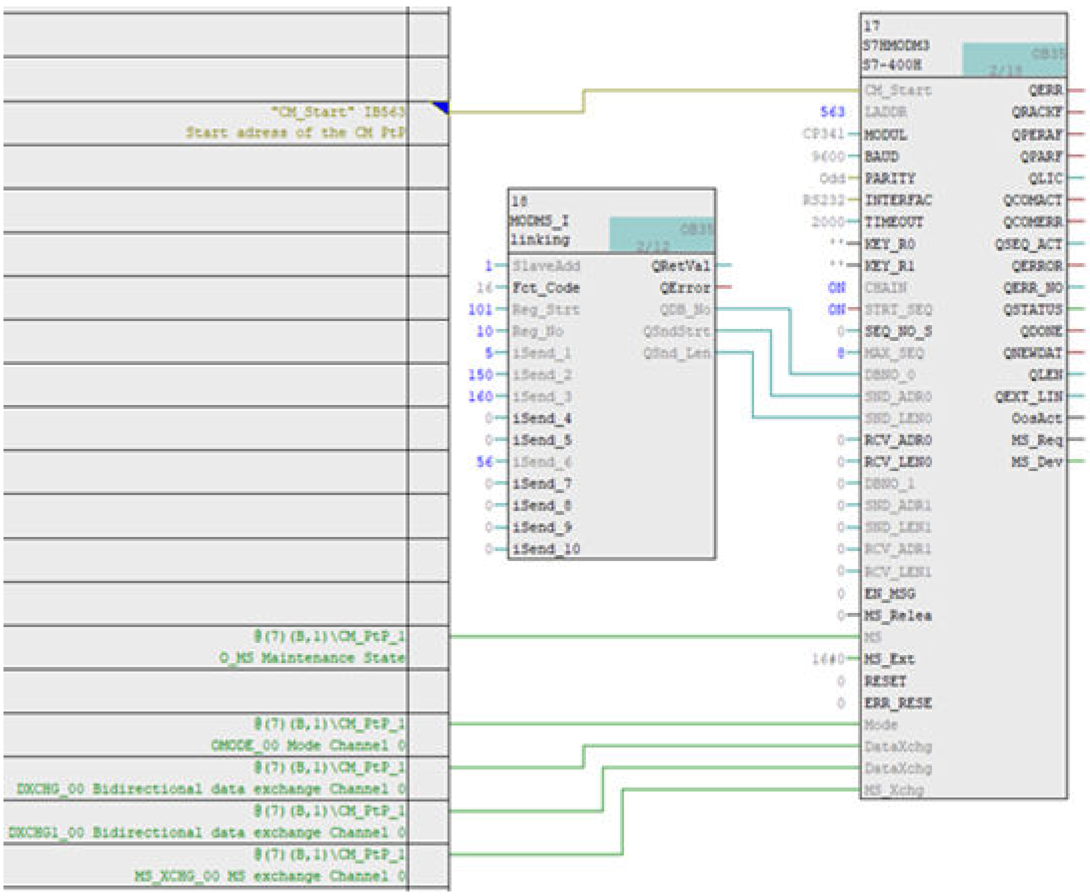


Figure 4-1 Example for data to be sent in CFC

Note

When writing coils with function code 15

The coils are stored in registers. Integer values are stored using 16 coils, Double Integer and Real values are both stored using 32 coils. Here, the parameter Reg_No refers to the number of coils and therefore needs to be set to 16 to write 1 integer value (32 for DI/R). This also means the coils 1 to 16 are written as an Integer value via iSend_1 (1 to 32 for diSend_1/rSend_1).

4.4 How to fetch data

In order to fetch data, the output parameters QDB_No, QSndStrt, QSndLen, QRcvStrt and QRcvLen of MODMR_I (FB242), MODMR_DI or MODMR_R have to be linked with the input parameters DBNO_x, SND_ADRx, SND_LENx, RCV_ADRx and RCV_LENx of the function block S7HMODM3 (FB240).

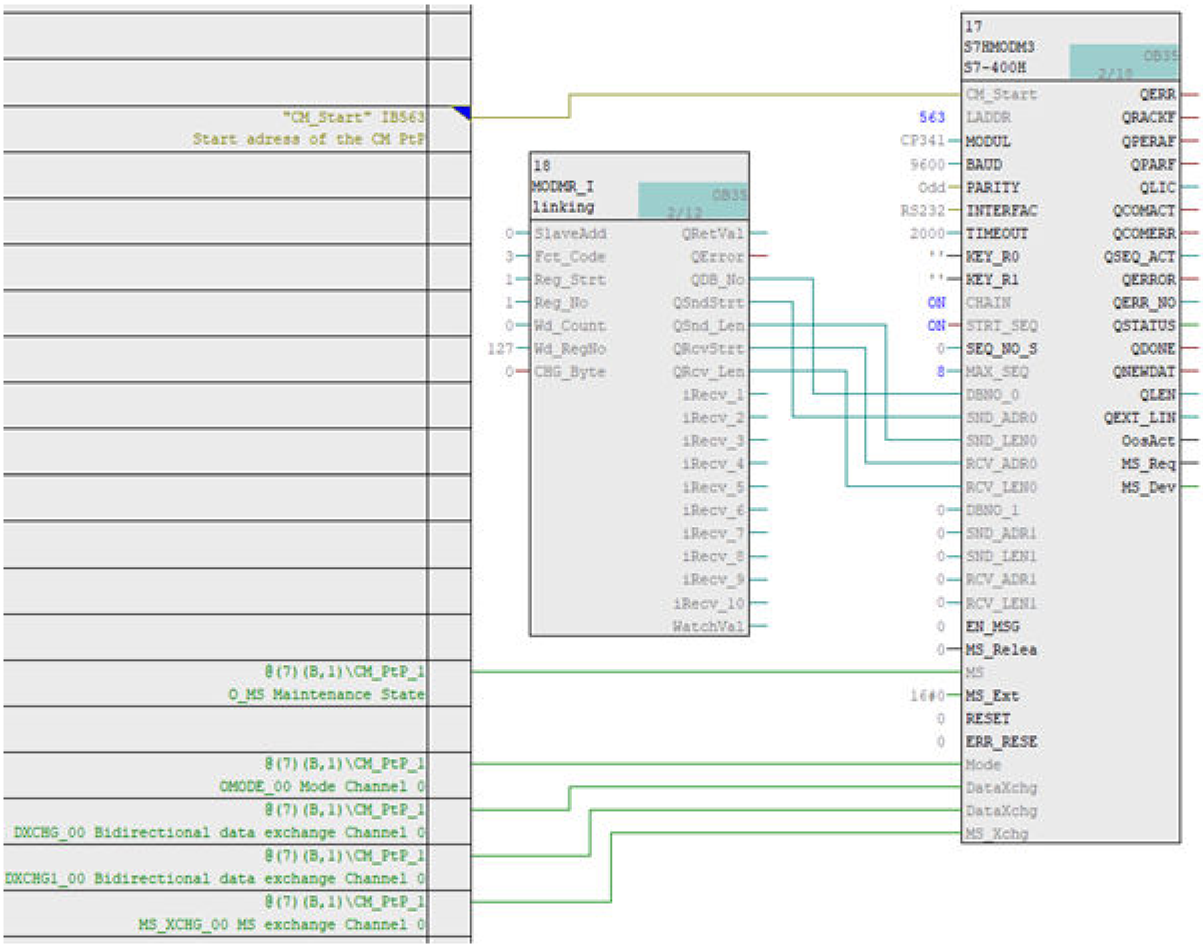


Figure 4-2 Example for data to be fetched in CFC

The received data is available at the outputs iRecv_1 to iRecv127 for further operation.

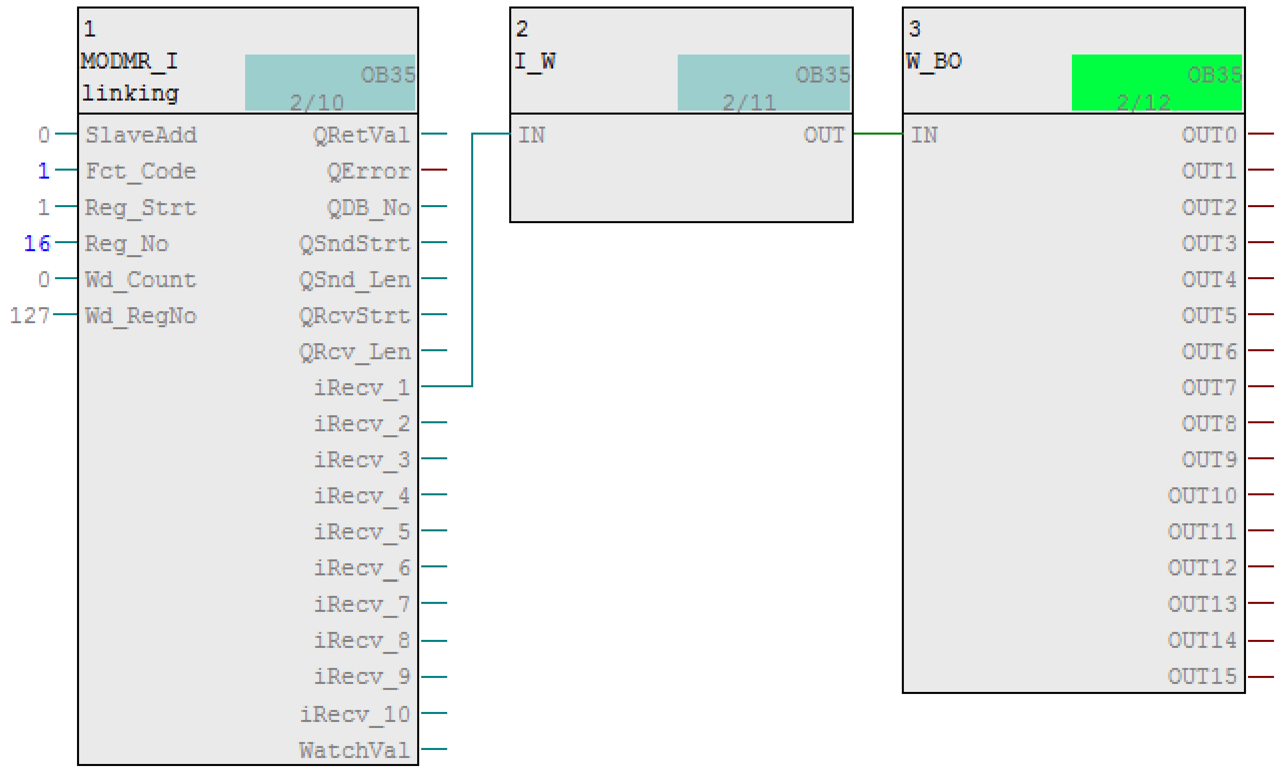


Figure 4-3 Example of MODMR_I configured to receive coils

Note

When fetching coils (function codes 1 and 2)

The coils are stored in registers. Integer values are stored using 16 coils, Double Integer and Real values are both stored using 32 coils. Here, the parameter Reg_No refers to the number of Coils and therefore needs to be set to 16 to read 1 integer value (32 for DI/R).

This also means the coils 1 to 16 are stored in iRecv_1 (1 to 32 for diRecv_1/rRecv_1) and the user will have to separate the coils of these registers using function blocks I_W and W_BO (DI_DW/R_DW and DW_BO).

Upgrade from CP341 to CM PtP 5

5.1 Overview

This library was designed to switch the existing communication from CP341 to CM PtP with a minimum of modification.

Overview of the necessary steps:

- Electrical setup of an ET 200SP HA with CM PtP and change of wiring from CP341 to CM PtP.
- Build up an ET 200SP HA with CM PtP in the Hardware Configuration of PCS 7 transfer the parameter from CP341 to CM PtP.
- Installation and Block update if an older version of the library is still being used.
- Configuration of the basic communication parameters from the HW config at the existing Communication Block in the CFC Chart.
- Running the Driver Wizard for automatic call-up and automatic parameterization and connection of the di-agnostic blocks.

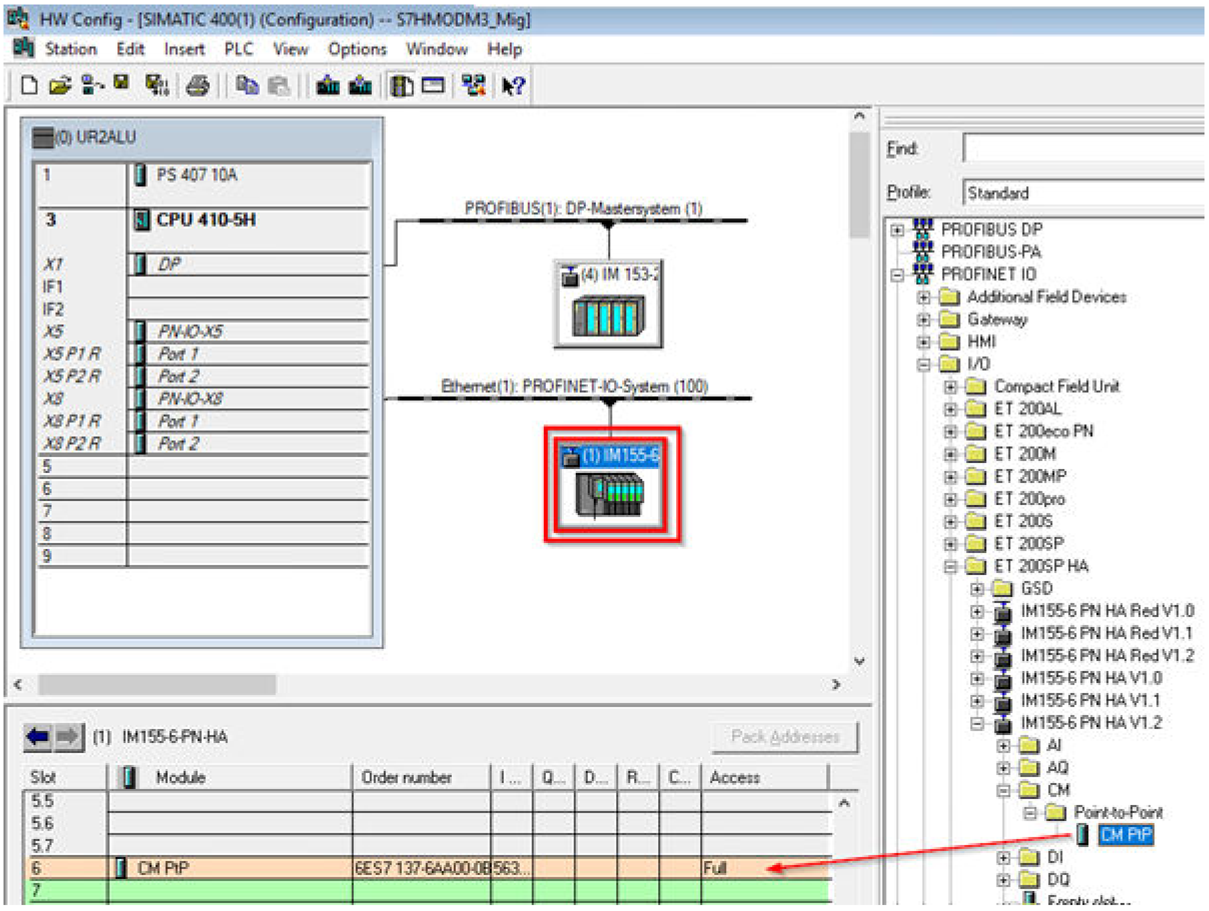
The following chapters describe the switch from CP341 with ET 200M PROFINET to CM PtP with ET 200SP HA with PROFINET as an example.

5.2 Electrical setup

Please follow the manual of the ET 200SP HA and the manual of the CM PtP.

5.3 Hardware Configuration

Setting up the ET 200SP HA with CM PtP.


HW Config - [SIMATIC 400(1) (Configuration) -- S7HMODM3_Mig]

(0) UR2ALU

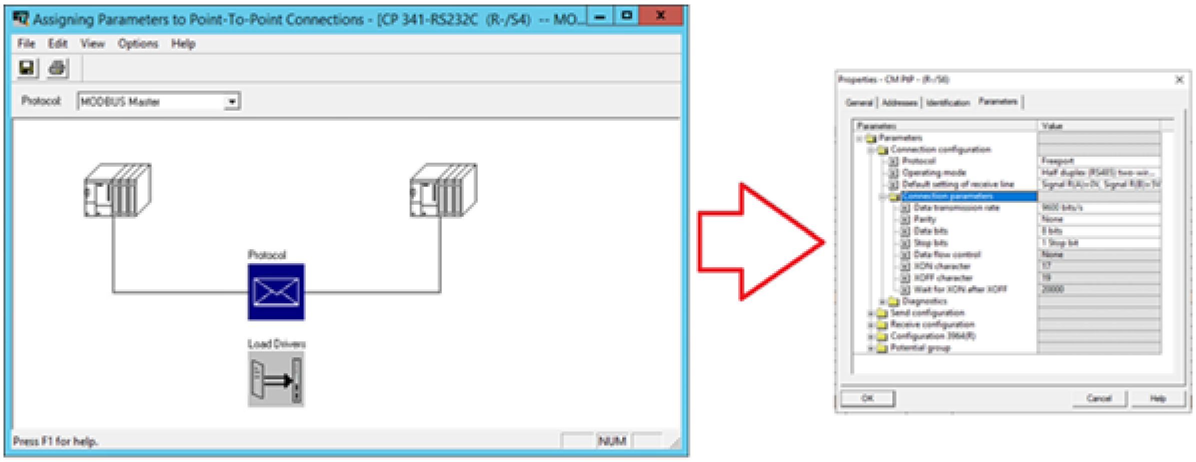
1	PS 407 10A
3	**CPU 410-5H**
X1	*DP*
IF1	
IF2	
X5	*PN-IO-X5*
X5 P1 R	*Port 1*
X5 P2 R	*Port 2*
X8	*PN-IO-X8*
X8 P1 R	*Port 1*
X8 P2 R	*Port 2*
5	
6	
7	
8	
9	

(1) IM155-6-PN-HA

Slot	Module	Order number	I ...	Q...	D...	R...	C...	Access
5.5								
5.6								
5.7								
6	CM PtP	6ES7 137-6AA00-0B563...						Full
7								

Figure 5-1 Hardware Configuration of CM PtP

Taking over the parameter from CP341 to CM PtP

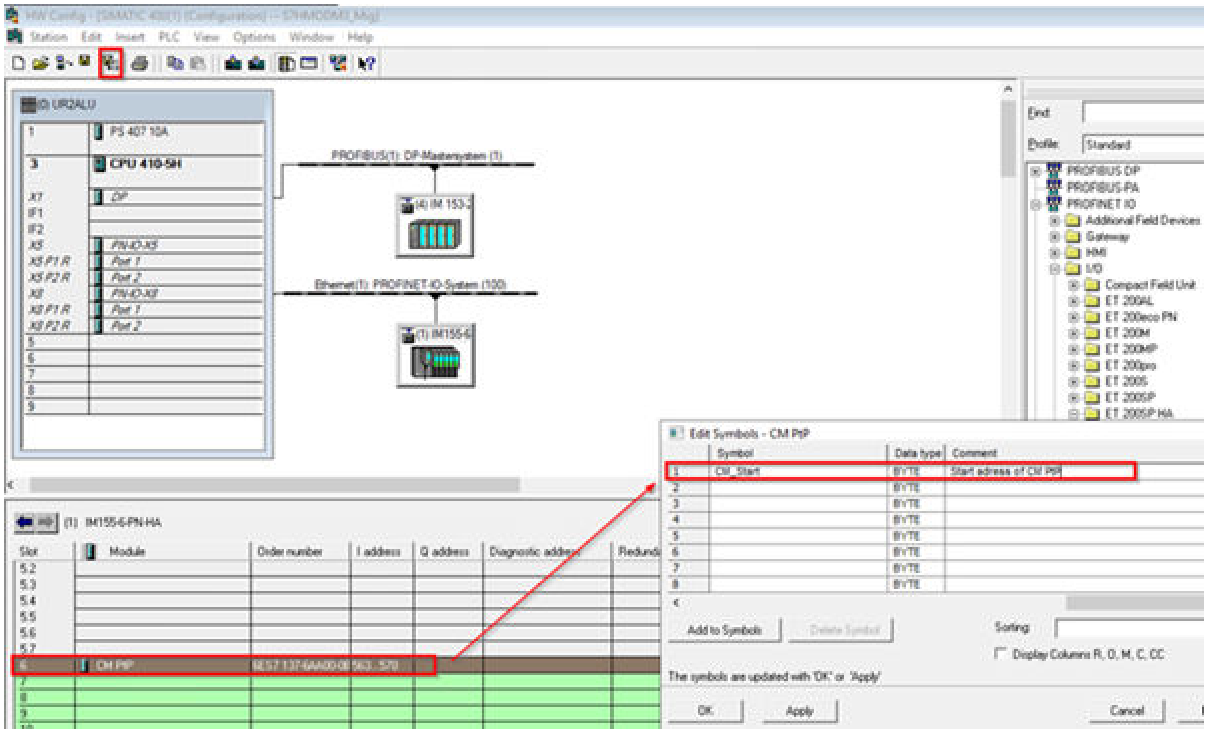

Assigning Parameters to Point-To-Point Connections - [CP 341-RS232C (R-/S4) -- MO...
File Edit View Options Help
Protocol
Load Drivers
Press F1 for help.
NUM
Properties - CM PtP - (R-/S8)
General | Addresses | Identification | Parameters
OK
Cancel
Help

Protocol: MODBUS Master

Parameters	Value
Parameters	
Connection configuration	
Protocol	Freeport
Operating mode	Half duplex (RS485) two-wir...
Default setting of receive line	Signal R(A)=2V, Signal R(B)=5V
Connection parameters	
Data transmission rate	9600 bits/s
Parity	None
Data bits	8 bits
Stop bits	1 Stop bit
Data flow control	None
XON character	17
XOFF character	19
Wait for XON after XOFF	20000
Diagnostics	
Send configuration	
Receive configuration	
Configuration 3964(R)	
Potential group	

Figure 5-2 Parametrization of CM PtP in the Hardware Configuration

Assigning a symbol for the start address of the CM PtP and save and compile.



	Symbol	Data type	Comment
1	CM_Start	BYTE	Start adress of CM PtP
2		BYTE	
3		BYTE	
4		BYTE	
5		BYTE	
6		BYTE	
7		BYTE	
8		BYTE	

Figure 5-3 Start Address of CM PtP in the Hardware Configuration

5.4 Installation and Block update

The library must be installed. Please follow the chapter Installation. Updating the Block Types of the project. In case that some blocks are missed after the Block update they can be copied manually from the library and the Utils library which is also installed by the installation wizard.

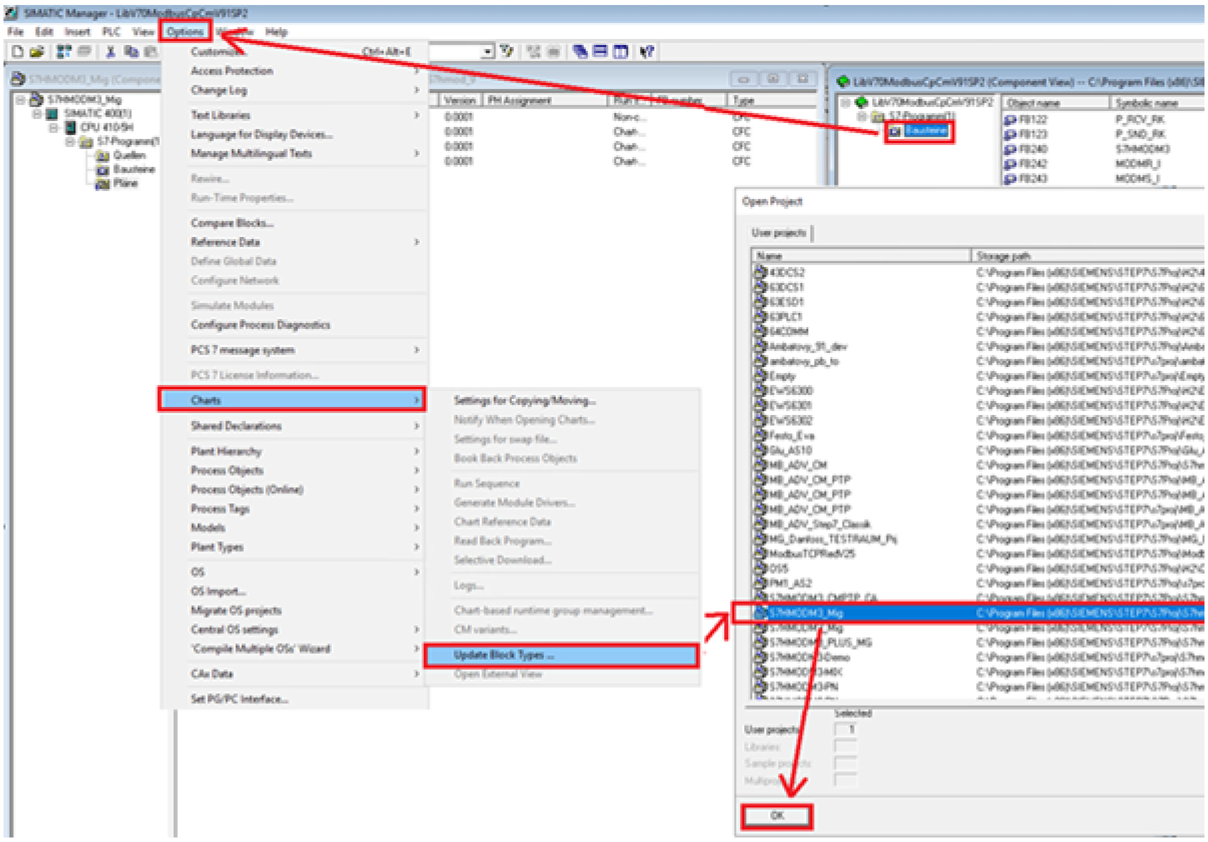
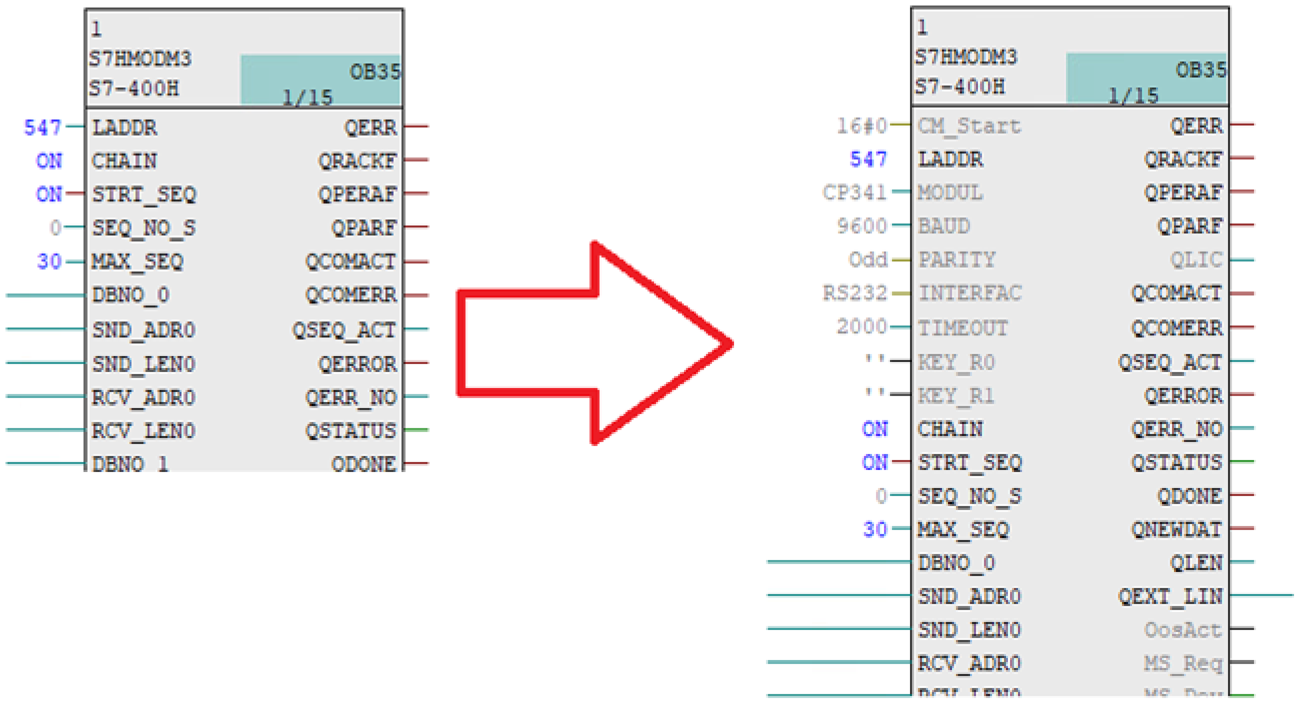


Figure 5-4 Block Update with the new library Version

5.5 Configuration of the basic communication parameter

After Block update the parameter for the communication are available and can be adjusted based on the hardware configuration. For using the library in demo mode, the parameter KEY_R0 and KEY_R1 can be left free. For the full functionality the license keys received must be used. With FEATURE.Bit30 = 1 the diagnostic behavior of CP341 can be kept. With FEATURE.Bit30 = 0 the new diagnostic behavior can be chosen.

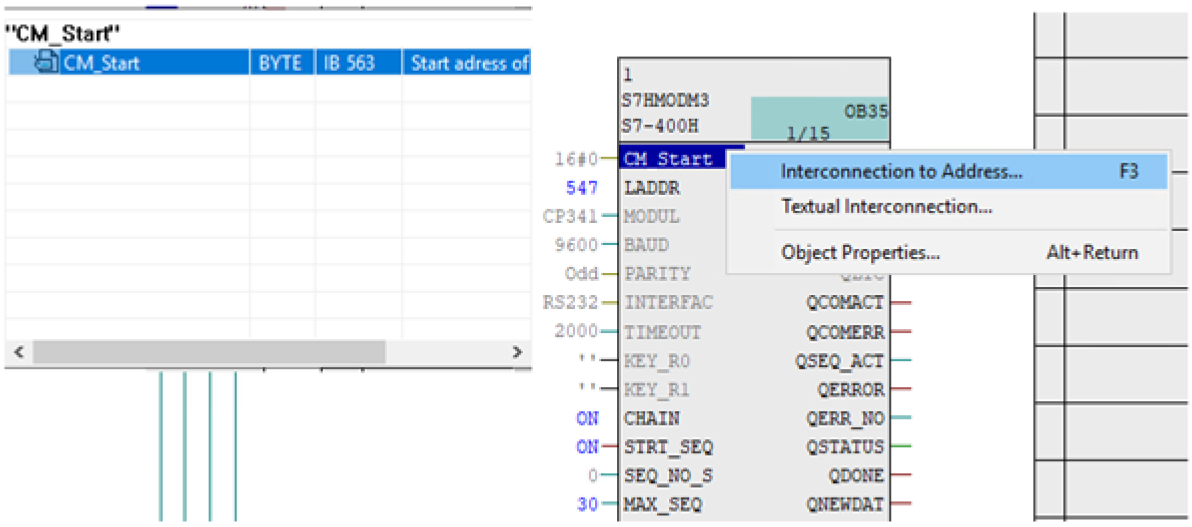

```
1
S7HMODM3                OB35
S7-400H                 1/15
547 — LADDR             QERR —
ON  — CHAIN           QRACKF —
ON  — STRT_SEQ        QPERAF —
0   — SEQ_NO_S         QPARF —
30  — MAX_SEQ        QCOMACT —
    — DBNO_0         QCOMERR —
    — SND_ADR0      QSEQ_ACT —
    — SND_LEN0        QERROR —
    — RCV_ADR0       QERR_NO —
    — RCV_LEN0       QSTATUS —
    — DBNO_1           QDONE —
```

```
1
S7HMODM3                  OB35
S7-400H                   1/15
16#0   — CM_Start         QERR —
547    — LADDR          QRACKF —
CP341  — MODUL          QPERAF —
9600   — BAUD            QPARF —
Odd    — PARITY           QLIC —
RS232  — INTERFAC      QCOMACT —
2000   — TIMEOUT       QCOMERR —
''     — KEY_R0       QSEQ_ACT —
''     — KEY_R1         QERROR —
ON     — CHAIN         QERR_NO —
ON     — STRT_SEQ      QSTATUS —
0      — SEQ_NO_S        QDONE —
30     — MAX_SEQ       QNEWDAT —
       — DBNO_0           QLEN —
       — SND_ADR0     QEXT_LIN —
       — SND_LEN0       OosAct —
       — RCV_ADR0       MS_Req —
```

Figure 5-5 Additional Parametrization of the new S7HMODM3 Block

5.6 Driver Wizard and Diagnostic concept

For using the driver wizard the Parameter CM_Start must be connected to the start address of the CM_PtP



"CM_Start"

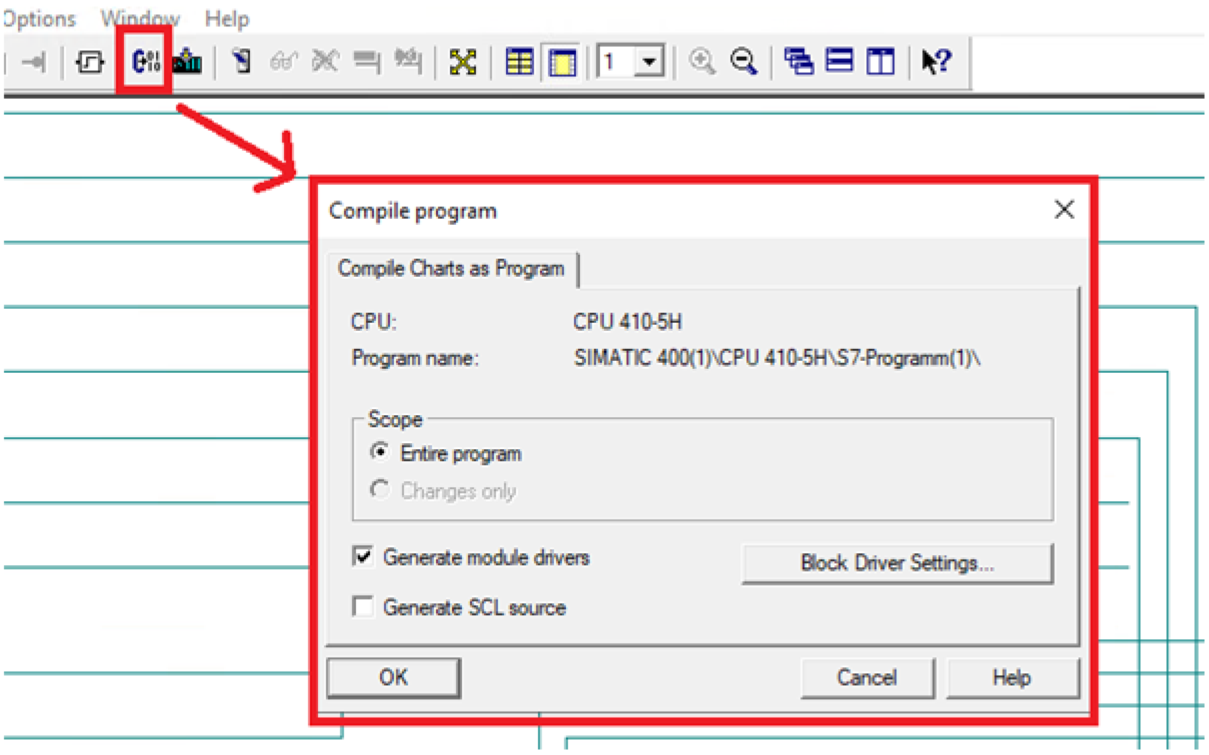
CM_Start	BYTE	IB 563	Start adress of

```
1
S7HMODM3                 OB35
S7-400H                  1/15
   16#0 — CM_Start
    547 — LADDR
  CP341 — MODUL
   9600 — BAUD
    Odd — PARITY
  RS232 — INTERFAC          QCOMACT —
   2000 — TIMEOUT           QCOMERR —
     '' — KEY_R0            QSEQ_ACT —
     '' — KEY_R1            QERROR —
     ON — CHAIN             QERR_NO —
     ON — STRT_SEQ          QSTATUS —
      0 — SEQ_NO_S          QDONE —
     30 — MAX_SEQ           QNEWDAT —
```

Interconnection to Address... F3
Textual Interconnection...
Object Properties... Alt+Return

Figure 5-6 Connecting the Start Address of CM PtP with the new S7HMODM3 Block

Compiling the Program with "Generate module drivers" for automatic call-up and automatic parameterization and connection of the diagnostic blocks



Compile program

Compile Charts as Program

CPU: CPU 410-5H
Program name: SIMATIC 400(1)\CPU 410-5H\S7-Programm(1)\

Scope
- [✓ Entire program]
- Changes only

- [✓] Generate module drivers
- [ ] Generate SCL source

Figure 5-7 Generating module drivers

5.7 Summary

With the design of the library, a conversion of the existing communication from CP341 to CM PtP can be done with a minimum of modification, thanks to the way the CM PtP support is integrated and the block update functionality.

Detailed description of the function blocks 6

6.1 Overview

The data exchange between the S7-41x (H) CPU and the CP341 is implemented via PROFIBUS DP. A parameterization of the link is not required. The CP341 converts the telegrams by using the Modbus RTU format.

The library administrates the complete data traffic without any requirements of programming know-how.

6.2 Used blocks from PCS 7 Advanced Process Library(APL) and PCS 7 Basis Library

For the operating and diagnostic of the Cm PtP the following blocks of the PCS 7 Advanced Process Library (APL) and the PCS 7 Basis Library are used. The blocks are also available in the Modbus Advanced Utils Library (MB_ADV_CM_PTP_UtilsV912) which is also installed by the installer for Modbus_ADV_CM_PTP_VXX.

Table 6-1 **Used blocks from PCS 7 Advanced Process Library (APL)**

Block	FB-Number	Library
MOD_CM[1)]	FB221	PCS 7 Basis Library V91
Send_Config[2)]	FB1991	PCS_7_AP_Library_V91
Receive_Config[3)]	FB1992	PCS_7_AP_Library_V91
Send_P2P[4)]	FB1993	PCS_7_AP_Library_V91
Receive_P2P[5)]	FB1994	PCS_7_AP_Library_V91
Receive_Reset[6)]	FB1995	PCS_7_AP_Library_V91
Modbus_Comm_Load[7)]	FB1996	PCS_7_AP_Library_V91
Modbus_Master[8)]	FB1997	PCS_7_AP_Library_V91

For further information, please refer to the relevant documentation:

[1)] Process Control System PCS 7 Basis Library (V9.1 SP2), Chapter 6.36 (https://support.industry.siemens.com/cs/document/109812809)

[2)] Process Control System PCS 7 Advanced Process Library (V9.1 SP2), Chapter 21.11 (https://support.industry.siemens.com/cs/document/109812806)

[3] Process Control System PCS 7 Advanced Process Library (V9.1 SP2), Chapter 21.12 (https://support.industry.siemens.com/cs/document/109812806)

[4] Process Control System PCS 7 Advanced Process Library (V9.1 SP2), Chapter 21.13 (https://support.industry.siemens.com/cs/document/109812806)

[5] Process Control System PCS 7 Advanced Process Library (V9.1 SP2), Chapter 21.14 (https://support.industry.siemens.com/cs/document/109812806)

[6] Process Control System PCS 7 Advanced Process Library (V9.1 SP2), Chapter 21.15 (https://support.industry.siemens.com/cs/document/109812806)

[7] Process Control System PCS 7 Advanced Process Library (V9.1 SP2), Chapter 21.16 (https://support.industry.siemens.com/cs/document/109812806)

[8] Process Control System PCS 7 Advanced Process Library (V9.1 SP2), Chapter 21.17 (https://support.industry.siemens.com/cs/document/109812806)

6.3 Modbus-Master- Communication block: S7HMODM3

6.3.1 Type / Number / Symbol

S7HMODM3 (FB 240) V7.2

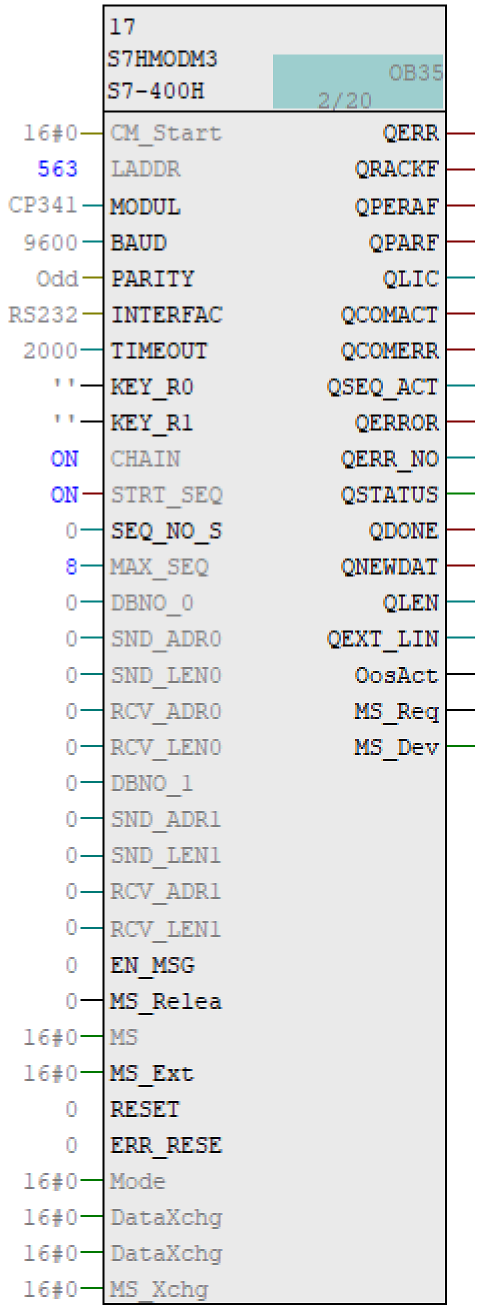


Figure 6-1 CFC-Function block S7HMODM3 (FB 240)

6.3.2 Application field

The function block manages the communication between a S7-41x / S7-417H CPU and a CP341. In case of using the CP341 the module is placed in an ET200M rack as Modbus-Master (HW-Dongle).

The first Input Byte of the CM PtP resp. CP341 must be connected to the CM_Start parameter.

The function block administrates up to 10 tasks (fetch / send data). Each task can have a number of up to 254 Bytes.

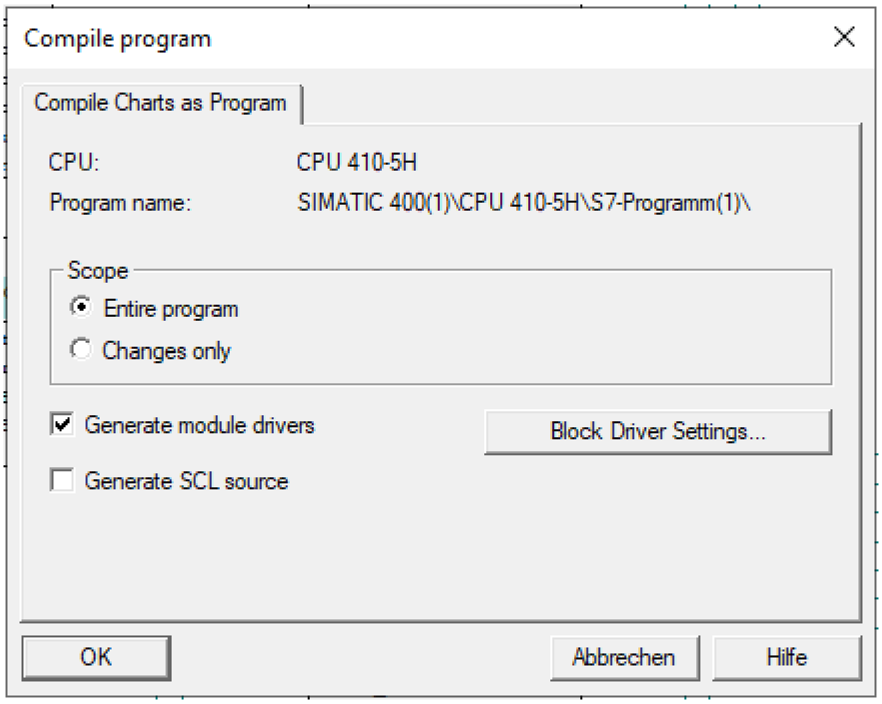
The starting task number [0...9] is adjusted with the input parameter SEQ_NO_S, the stop task number [0...9] with MAX_SEQ. The sequence is started with STRT_SEQ = 1.

If MAX_SEQ <= SEQ_NO_S, only the task with the number given over SEQ_NO_S will be handled.

The settings for the single tasks at the inputs DBNO_x, SND_ADRx, SND_LENx, RCV_ADRx and RCV_LENx are managed by the according process blocks (MODMS_I, MODMS_DI, MODMS_R, MODMR_I, MODMR_DI or MODMR_R).

Compiling CFC with "Generate module drivers"

For compiling the CFC charts with the option "Generate module drivers" the first Input Byte of the CP341/CM PtP must be connected to the CM_Start parameter.


Compile program

Compile Charts as Program

CPU: CPU 410-5H

Program name: SIMATIC 400(1)\CPU 410-5H\S7-Programm(1)\

Scope

[✓ Entire program]

Changes only

[✓] Generate module drivers

[ ] Generate SCL source

Figure 6-2 CFC Compiling with "Generate module drivers"

In case the option "Generate module drivers" is not used, the following parameter must be adjusted manually:

- Connections from the parameter MS, Mode, DataXchg, DataXchg1, MsgXchg of the S7HMODM3 to the MOD_CM block.
- Configuration from of the start address of the CM PtP at the LADDR parameter of MOD_CM block.
- License Key

Special case

In order to increase the maximum number of 10 tasks for one S7HMODM3 block, extension blocks "EXSQ_MOD" with 10 possible tasks each can be connected to the S7HMODM3.

In order to use this particular possibility, the parameter CHAIN must be set to 1.

More information on the use of the extension blocks can be found in chapter 6.3.

In this case, the maximum number of tasks supported by the S7HMODM3 is 100 (More tasks are possible, but this may strongly slow down the actualization time of the values).

6.3.3 Called organisation blocks (OBs)

- OB 35 (time cyclic interrupt) or any other cyclic OB (recommendation: OB 36)
- OB 70 (I/O redundancy error)
- OB 82 (Diagnostic interrupt)
- OB 83 (insert / remove module interrupt alarm)
- OB 85 (priority class error)
- OB 86 (rack failure)
- OB 100 (restart)
- OB 122 (I/O access error)

6.3.4 Alarm reaction

Alarms are enabled with EN_MSG = 1 (Default value = 0).

MSG_EVID contains the message-event-ID of the ALARM_8P FB in WINCC. This number is assigned automatically when the block is inserted in CFC. Shall the alarms be disabled, the EN_MSG should be set to 0. QMSG_ERR, QMSG_SUP, MGS_STAT and MSG_ACK are ALARM_8P specific parameter.

Depending on the FEATUREBIT.30 (1=internal diagnostic / 0 = external diagnostic) different alarms are enabled.

Alarm_8P:

Table 6-2 Alarm_8P of MB_ADV_COM (FB 2240)

Alarm Nb.	**FB parameter**	**Standard settings**	**Alarm class**	**Diagnostic**
SIG_1	QRACKF	1=Rack Failure Module	S	Internal
SIG_2	QPERAF	1=I/O Module Access Failure	S	Internal
SIG_3	QPARF	1=Parameter Assignment Error Module	S	Internal
SIG_4	QMODF	1=Module failure	S	Internal
SIG_5	QERR	1= RTU Communication failure	S	External
SIG_6	--	--	--	--
SIG_7	--	--	--	--
SIG_8	--	--	--	--

6.3.5 Error handling

Failures are identified by a specific failure number of the function block. The communication continues after the failure has been disappeared. The output parameter QRACKF and QPERAF identify rack failure and parameterization failure. In case of active communication QCOMACT is 1. Additional failures are identified with QERR_NO.

Table 6-3 Error code for FB240 („S7HMODM3")

QERR_NO	Description
0	Normal operation
1	NOP with sequence: (DBNO_x = 0) and (SND_LENx = 0)
2	MAX_SEQ > 9 (1)
3	SEQ_NO_S < 0
4	No valid task (Task number < 0 or > 9) 1)
5	MAX_SEQ not available 2)

1) 9 is the maximal sequence number if there is no function block EXSQ_MOD and if CHAIN = 0.

2) Look at figure below.

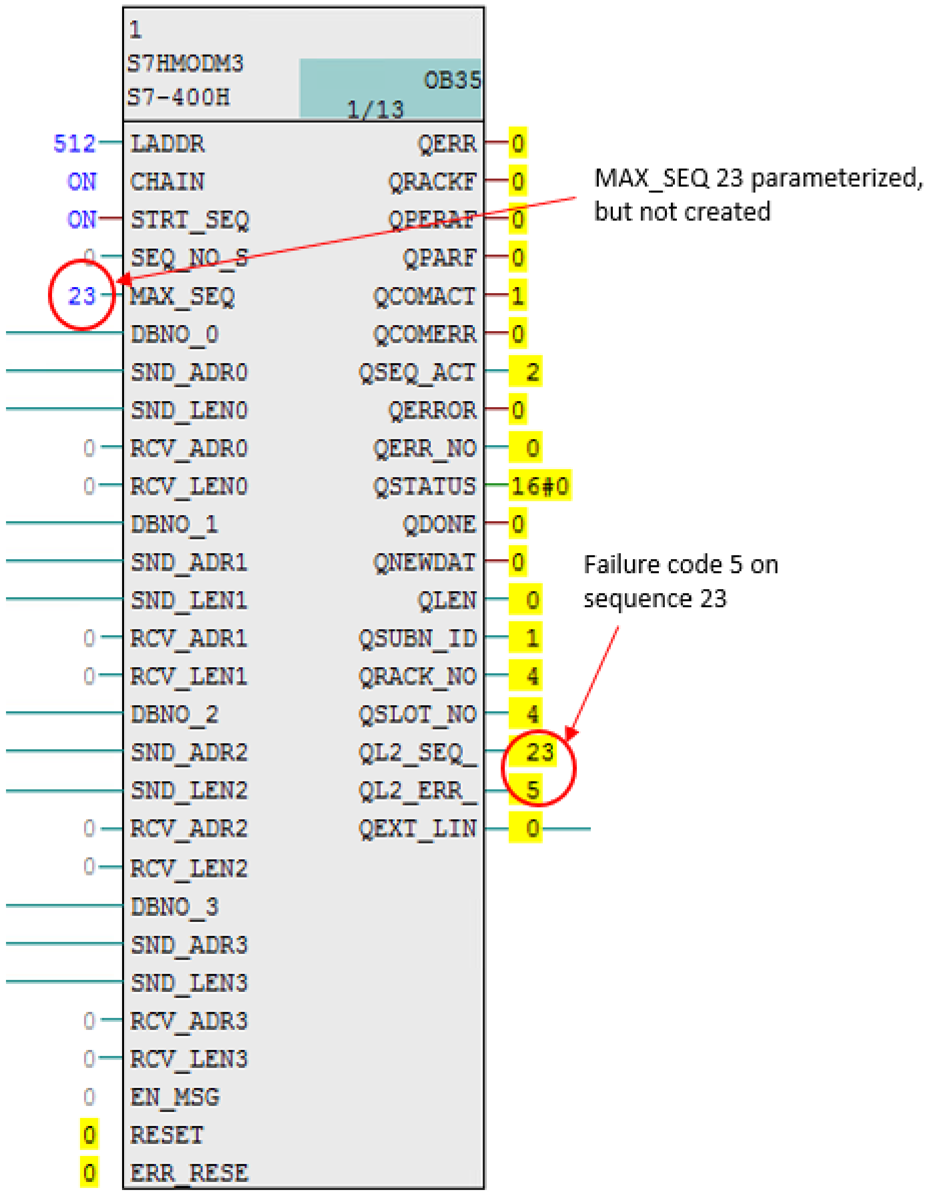


Figure 6-3 CFC- Function block FB 240 (S7HMODM3) with Error code 5

6.3.6 Parameters list

Input parameters

Parameter	Description	Type	Default value
LADDR	Logical I/O start address of CP; see HW config	INT	0
CHAIN	1 = chain with connected extension block(s)	BOOL	0

Parameter	Description	Type	Default value
SeqCycMax	Max CPU cycles allowed for executing a se-quence (>100)	INT	500
STRT_SEQ	Start sequence	BOOL	0
SEQ_NO_S	Start with sequence number	INT	0
MAX_SEQ	Maximal sequence number: 0...MAX_SEQ	INT	4
DBNO_0	Receive data block	INT	0
SND_ADR0	Start address for send or fetch request	INT	0
SND_LEN0	Send length or fetch request length	DINT	0
RCV_ADR0	Start address for receiving data	INT	0
RCV_LEN0	Receive length only used for check	DINT	0
DBNO_1	Receive data block	INT	0
SND_ADR1	Start address for send or fetch request	INT	0
SND_LEN1	Send length or fetch request length	DINT	0
RCV_ADR1	Start address for receiving data	INT	0
RCV_LEN1	Receive length only used for check	DINT	0
...		-	-
DBNO_9	Receive data block	INT	0
SND_ADR9	Start address for send or fetch request	INT	0
SND_LEN9	Send length or fetch request length	DINT	0
RCV_ADR9	Start address for receiving data	INT	0
RCV_LEN9	Receive length only used for check	DINT	0
MSG_EVID	Message ID for Alarm_8P	DWORD	2
EN_MSG	1 = Enable alarming	BOOL	0

Input / Output parameters

Parameter	Description	Type	Default value
RESET	1 = Reset of Modbus communication	BOOL	0
ERR_RESET	1 = Reset information about errors	BOOL	0

Output parameters

Parameter	Description	Type	Default value
QERR	1 = Error	BOOL	1
QREADY	1 = CP341-2 is ready	BOOL	0
QRACKF	1 = Rack Failure Module	BOOL	0
QMODF	1 = Module Removed/Out of Order	BOOL	0
QPERAF	1 = I/O Module Access Failure	BOOL	0
QPARF	1 = Parameter Assignment Error Module	BOOL	0
QRED_ERR	1 = Redundancy lost	BOOL	0
QCOMACT	1 = Communication is active	BOOL	0
QCOMERR	1 = Communication failure	BOOL	0

Parameter	Description	Type	Default value
QSEQ_ACT	Actual sequence number	INT	0
QSTATE	Internal state	INT	0
QERROR	Communication. error of P_x_RK (FB7, FB8)	BOOL	0
QERR_NO	General error number	INT	0
QERROR_R	Communication. error of P_RCV_RK (FB7)	BOOL	0
QERROR_S	Communication. error of P_SND_RK (FB8)	BOOL	0
QSTATUS	Detailed communication. status of P_x_RK (FB7, FB8)	WORD	0
QDONE	End without error P_SND_RK (FB8)	BOOL	0
QNEWDAT	New data received	BOOL	0
QLEN	Actual received length	INT	0
QMSG_ERR	ALARM_8P Error	BOOL	0
QMSG_SUP	1 = Message Suppression active	BOOL	0
MSG_STAT	ALARM_8P: STATUS Output	WORD	0
MSG_ACK	ALARM_8P: ACK_STATE Output	WORD	0
QSUBN_ID	First Subnet ID of CP341	INT	0
QRACK_NO	Rack number of CP341	INT	0
QSLOT_NO	Slot number of CP341	INT	0
QLSEQ_ERR	Last sequence number with error	INT	0
QLSTSTAT	Last detailed communic. status of P_x_RK (FB7, FB8)	WORD	0
QL2_SEQ_NO	Number of the latest sequence with QERR_NO <> 0	INT	0
QL2_ERR_NO	Latest value of QERR_NO (<>0)	INT	0
QTLIMIT	Counter for jobs exceeding execution time limit	INT	0
QEXT_LINK	Connection for an extension block	INT	0

6.4 Extension block for the Modbus communication: EXSQ_MOD

6.4.1 Type / Number / Symbol

EXSQ_MOD (FB 251)

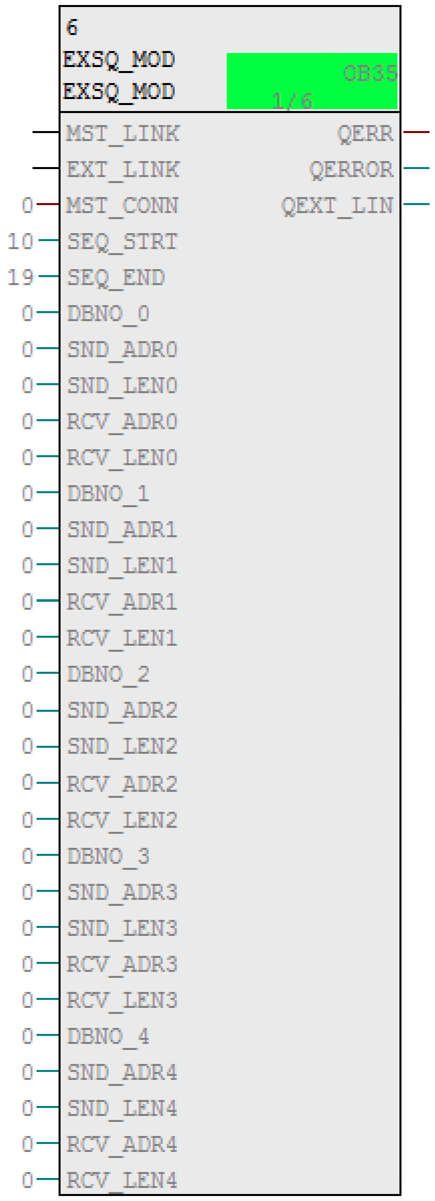


Figure 6-4 CFC-Function block EXSQ_MOD (FB 251)

6.4.2 Application field

If more than 10 tasks (i.e. "send data" or "fetch data") are needed, the main function block S7HMODM3 is not enough and a new FB EXSQ_MOD has to be used.

Up to 9 EXSQ_MOD blocks can be connected to one S7HMODM3 block. With one S7HMODM3 and the maximum number of 9 EXSQ_MOD blocks, a maximum amount of 100 tasks per CP are possible.

Each task has its own sequence number. With the input parameter SEQ_STRT ("Sequence start number") and SEQ_END ("Sequence end number") the user sets the range of numbers to be assigned to the tasks 0 to 9 of the function block EXSQ_MOD.

Example

If SEQ_STRT = 10 and SEQ_END = 15, the block will only administrate the tasks 10 to 15. The settings for the task 10 can then be configured on input parameters DBNO_0, SND_ADR0, SND_LEN0, RCV_ADR0 and RCV_LEN0.

Remarks

- The sequence number of the input parameter SEQ_STRT must be larger than 9 because the function block S7HMODM3 always administrates the tasks 0 to 9.
- The condition (SEQ_END - SEQ_STRT) =< 9 must always be fulfilled.

Mandatory links of the function block EXSQ_MOD

- S7HMODM3: the input parameter MST_LINK of the EXSQ_MOD must be connected to the output parameter QEXT_LINK of the S7HMODM3. The input parameter MST_CONN must be set to 1.
- Further EXSQ_MOD: the input parameter EXT_LINK of the EXSQ_MOD must be connected to the output parameter QEXT_LINK of the previous EXSQ_MOD. The input parameter MST_CONN must be set to 0.

6.4.3 Called organisation blocks (OBs)

OB 35 (time cyclic interrupt) or any other cycle OB.

Note

All the extension blocks EXSQ_MOD have to be placed in the same cyclic OB (for ex. OB 36) as the S7HMODM3.

All the extension blocks EXSQ_MOD have to be placed after the S7HMODM3 in this cyclic OB.

6.4.4 Parameters list

Input parameters

Parameter	Description	Type	Default value
LADDR	Logical I/O start address of CP; see HW Config	INT	0
STRT_SEQ	Start sequence	BOOL	0
MST_LINK	Connection to the master block	ANY	0
EXT_LINK	Connection to an extension block	ANY	0
MST_CONN	1 = the block is connected to the master block	BOOL	0
SEQ_STRT	Start with sequence number	INT	10
SEQ_END	Maximal sequence number: 0...MAX_SEQ	INT	19
DBNO_0	Receive data block	INT	0
SND_ADR0	Start address for send or fetch request	INT	0
SND_LEN0	Send length or fetch request length	DINT	0
RCV_ADR0	Start address for receiving data	INT	0
RCV_LEN0	Receive length only used for check	DINT	0
...	...	...	-
DBNO_9	Receive data block	INT	0
SND_ADR9	Start address for send or fetch request	INT	0
SND_LEN9	Send length or fetch request length	DINT	0
RCV_ADR9	Start address for receiving data	INT	0
RCV_LEN9	Receive length only used for check	DINT	0

Output parameters

Parameter	Description	Type	Default value
QERR	1=Error	BOOL	-
QERROR	Intern error, process error	BOOL	-
QEXT_LINK	Connection for an extension block	BOOL	-

The following Errors from the S7HMODM3 are indicate on the QERROR output

External Error’s from the S7HMODM3 would be also indicating.:

Table 6-4 Error from (S7HMODM3)

Parameter	Description
QRACKF	Rack failure
QCOMERR	Communication Error
QPARF	Parameter Assignment Error
QMODF	Module Removed/Out of Order
QPERAF	Module Access Error

6.4.5 Example

The next picture is an example of Modbus communication based on the FB S7HMODM3 and two FB EXSQ_MOD. You will find this example project within the library as well.

Blocks connections and configurations

The FB S7HMODM3 uses the CP/CM having the I/O start address 512 (LADDR = 512).

The FB S7HMODM3 is going to execute the configured tasks (STRT_SEQ = ON).

The parameter MST_LINK of the FB EXSQ_MOD n°4 is connected to the parameter QEXT_LINK of the master block, the FB S7HMODM3. For this reason, the parameter MST_CONN of the FB EXSQ_MOD n°4 is set to 1 and the parameter CHAIN of the FB S7HMODM3 is set to ON.

The second FB EXSQ_MOD n°8 is connected to the FB EXSQ_MOD n°4 over QEXT_LINK and EXT_LINK. The parameter MST_CONN of the FB EXSQ_MOD n°8 is set to 0.

Tasks configuration

The 3 FB are configured to execute the tasks 0 to 20 (SEQ_NO_S = 0 and MAX_SEQ = 20 at S7HMODM3).

The S7HMODM3 always manages the tasks 0 to 9. The FB EXSQ_MOD n°4 is configured to manage the tasks 10 and 11 (SEQ_STRT = 10 and SEQ_END = 11) and the FB EXSQ_MOD n°8 manages the tasks 12 to 15 (SEQ_STRT = 12 and SEQ_END = 15).

It is not a problem if some tasks are configured but are not defined (i.e. the tasks 16 to 20) or if no send blocks or receive blocks are connected to fulfil the task (i.e. the tasks 2 to 9).

Nevertheless, this may slow down the Modbus communication because the FBs try to execute each configured task.

Status of the blocks

At the S7HMODM3, the user can observe that the block is executing the task 0 (SEQ_ACT = 20) and that the communication is not active (QCOM_ACT = 0) for this task.

The task execution is not completed (QDONE = 1).

One or several tasks have not been executed (QERROR = 1) because the task has been configured but not defined (MAX_SEQ > number of defined tasks) (QERR_NO = 5).

The latest error occurred for the task numbered 11 (QLSEQ_ERR = 11) and the error number was E43 (QLSTSTAT = E43). To know if errors still occur, overwrite the outputs QLSEQ_ERR and QLSTSTAT with 0.

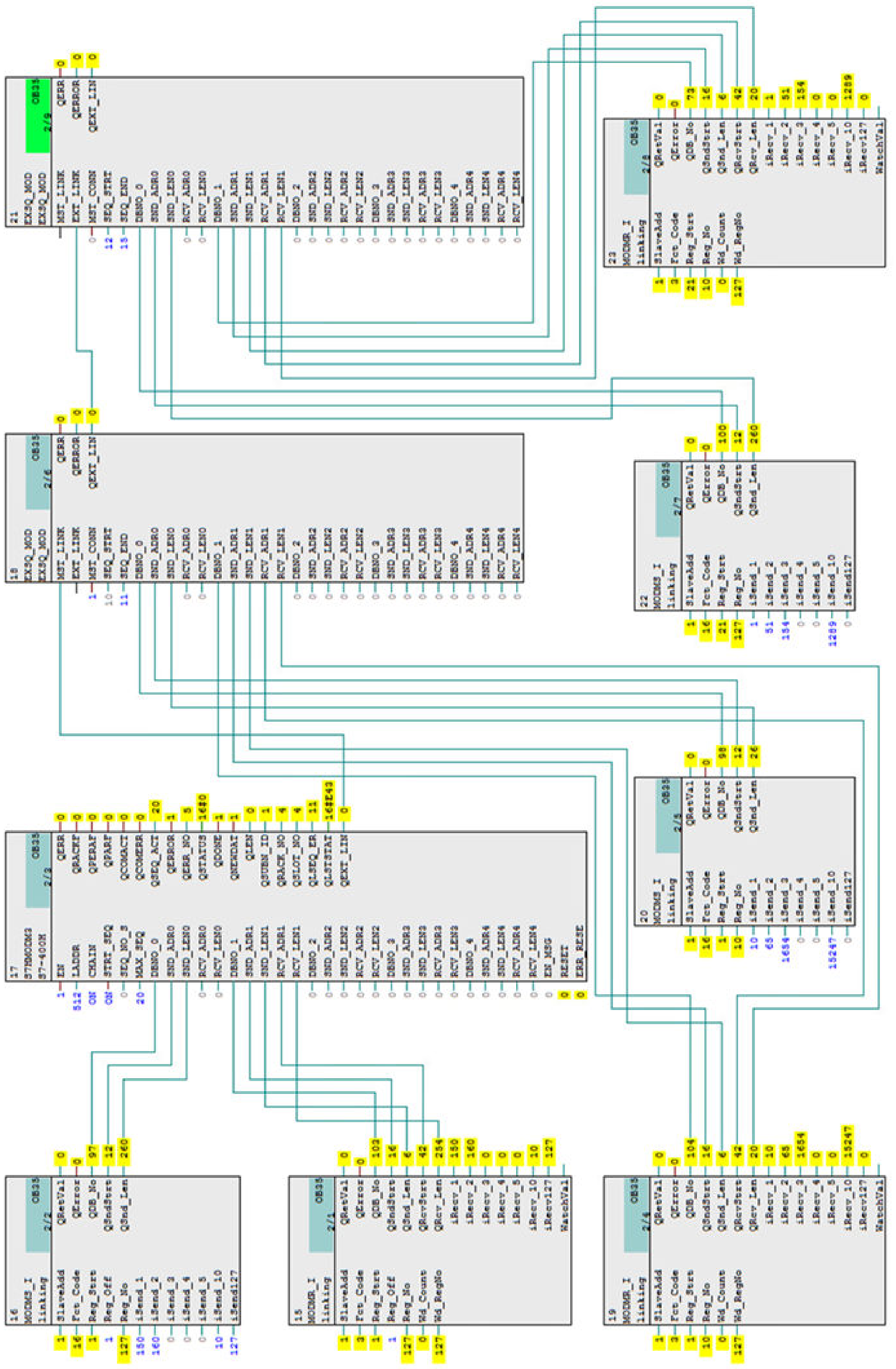


Figure 6-5 Example of Modbus communication with one S7HMODM3 and two EXSQ_MOD

Only one EXSQ_MOD configured with MST_CONN = 1 can be connected after only one S7HMOMD3. The setup shown below is incorrect.

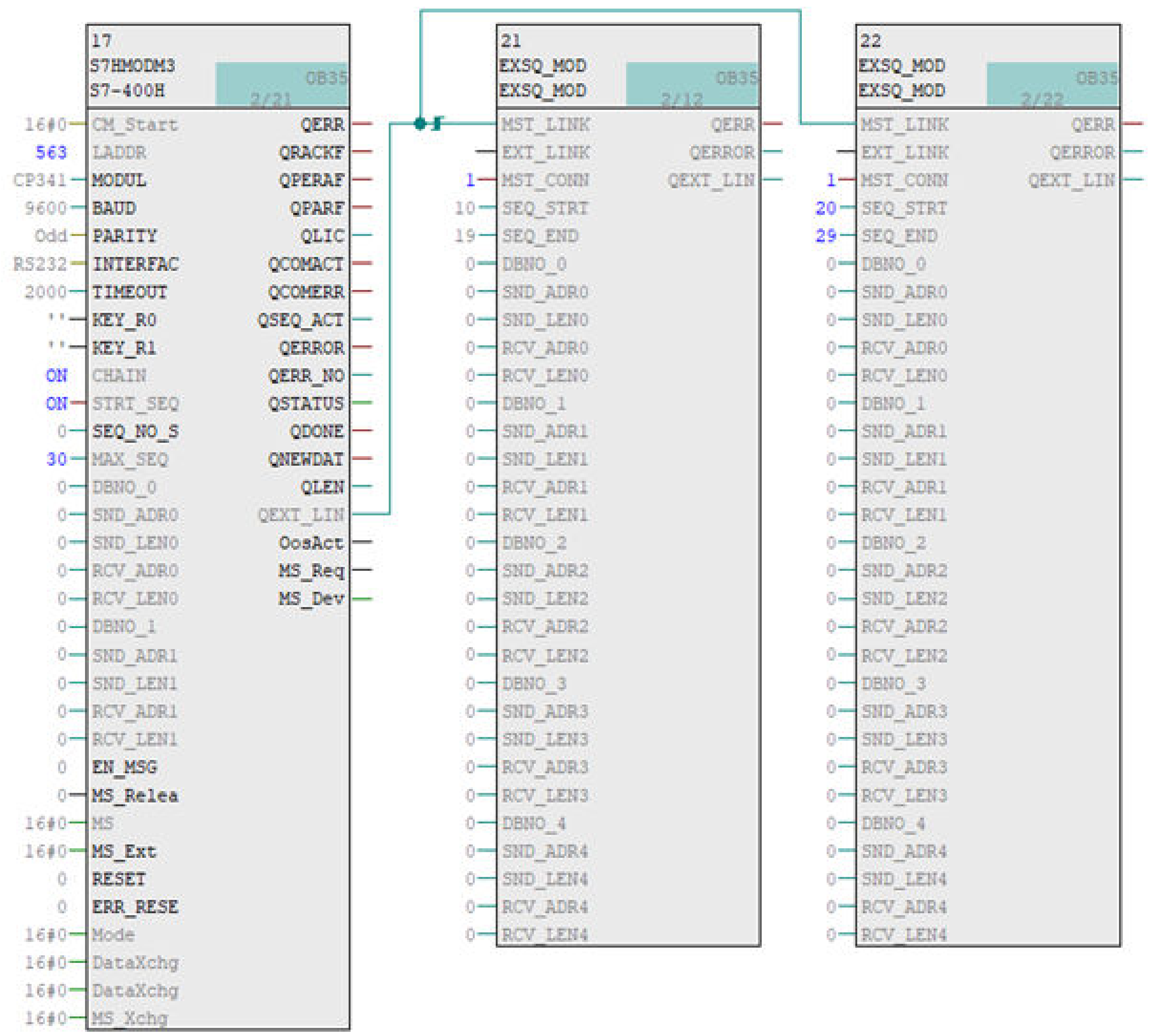


Figure 6-6 Incorrect setup of two EXSQ_MOD blocks to one S7HMOMD3 block

6.5 All process blocks

Concerning the blocks MODMR_I (FB242), MODMS_I (FB243), MODMR_R (FB244), MODMS_R (FB245), MODMR_DI (FB246), MODMS_DI (FB247), due to the programming of the block, there are two limits:

- The value of Reg_strt is limited to [1 ... 32767].
- The value of Reg_No is limited to [1... 127] registers or [1... 2032] coils.

In order to use the full range of starting addresses allowed by the Modbus protocol [0... 65535] (i.e. 0x0000 to 0xFFFF), the parameter Reg_Off with the range [0 ... 49999] may be used to extend the range. Indeed, the real starting address calculated and used by the block is Reg_Off + Reg_strt which belongs to the range [1... 82766].

The FB doesn’t check the validity of the starting address calculated with Reg_Off + Reg_strt.

Be sure that Reg_Off + Reg_strt is always <= 656635.

6.6 Modbus master receive block for integer values: MODMR_I

6.6.1 Type / Number / Symbol

MODMR_I (FB 242)

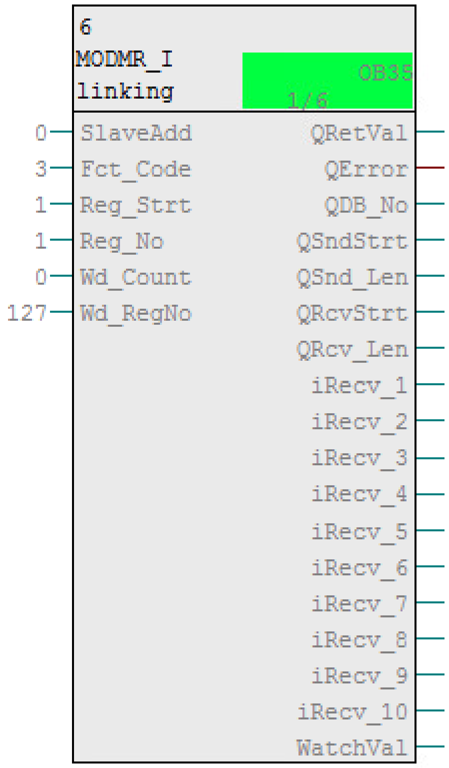


Figure 6-7 CCFC-Function block MODMR_I (FB 242)

6.6.2 Application field

The output parameters QDB_No, QSndStrt, QSndLen, QRcvStrt and QRcvLen of MODMR_I are linked with the input parameters DBNO_x, SND_ADRx, SND_LENx, RCV_ADRx and RCV_LENx of FB S7HMODM3 (FB240).

Each partner system (slave) on MODBUS has its own SlaveAdd. With function codes (Fct_code) the analogue and binary values are selected for reading input or output registers or status from the partner system. The meaning of the function codes is as follows:

Functions code	Task
1	Read of up to 2032 bits: Read Coils
2	Read of up to 2032 bits: Read Discrete Inputs

Functions code	Task
3	Read of up to 127 registers: Read Holding Registers
4	Read of up to 127 registers: Read Input Registers

The received values are available at the output iRecv_1 up to iRecv127.

The FB MODMR_I has to be configured differently according to the selected function code.

Function code 1 "Read Coils"

According to the Modbus protocol, the quantity of coils is limited to [1...2000] and the starting address to [1 ... 65535]. Here are the values supported by the block.

Parameter	Meaning	Values
Reg_Strt	Starting address	[1 ... 32767]
Reg_Off	Address offset	[0 ... 49999]
Reg_No	Quantity of coils	[1 ... 2032]

Function code 2 "Read Discrete Inputs"

According to the Modbus protocol, the quantity of inputs is limited to [1 ...2000] and the starting address to [1 ... 65535]. Here are the values supported by the block.

Parameter	Meaning	Values
Reg_Strt	Starting address	[1 ... 32767]
Reg_Off	Address offset	[0 ... 49999]
Reg_No	Quantity of registers	[1 ... 2032]

Function code 3 "Read Holding Registers"

According to the Modbus protocol, the quantity of holding registers is limited to [1...125] and the starting address to [1 ... 65535]. Here are the values supported by the block.

Parameter	Meaning	Values
Reg_Strt	Starting address	[1 ... 32767]
Reg_Off	Address offset	[0 ... 49999]
Reg_No	Quantity of registers	[1 ... 127]

Function code 4 "Read Input Registers"

According to the Modbus protocol, the quantity of inputs registers is limited to [1...125] and the starting address to [1... 65535]. Here are the values supported by the block.

Parameter	Meaning	Values
Reg_Strt	Starting address	[1 ... 32767]
Reg_Off	Address offset	[0 ... 49999]
Reg_No	Quantity of inputs	[1 ... 127]

The parameters CHG_Word and CHG_Byte are used to invert the low and high word and then the low and high byte of a received register.

According to the settings of CHG_Byte and CHG_Word, the register value given over the output parameter may be modified.

For example, let’s assume that the register receives 2 bytes A, B with A the high byte.

CHG_Word	X	X
CHG_Byte	0	1
Output	AB	BA

6.6.3 Watchdog function

This watchdog function can be activated with the parameter Wd_Count ≠ "0". The watchdog time is a result of (Wd_Count * time cycle of block MODMR_I). The parameter Wd_RegNo addresses the register or Bit that shall be controlled. The value of the controlled register sent by the partner system must be unequal to "0". On each call of the function block, the received value is checked and will be overwritten with "-1". If the actually received value is unequal to "0" the watchdog will be reset. If no value is received or if the value is "0", the output parameter QWdErr and QError are set to "1" and QRetVal is set to "10". To also use the controlled register, link it to WatchVal.

6.6.4 Alarm reaction

This function block has no integrated alarm function.

6.6.5 Called organisation blocks (OBs)

- OB 35 (time cyclic interrupt) or any other time cycle OB, or
- OB 1 (cyclic OB)

6.6.6 Error handling

This function block controls some alarm parameters such as limit and plausibility control as well as the watchdog control. Failures are always indicated with the output bit QError = "1" and QRetVal ≠ "0".

Table 6-5 Status codes for the FB 242 (MODMR_I)

QRetVal	Description
0	No error
1	Spare
2	SlaveAdd = 0

QRetVal	Description
3	Fct_Code < 1 or Fct_Code > 16
4	Reg_Strt <= 0
5	Fct_Code: 1, 2; Reg_No < 1 or Reg_No > 2032 (Bit_number)
6	Fct_Code: 3, 4; Reg_No < 1 or Reg_No > 127 (Reg_number)
7	Reg_Off > 50000 or Reg_Off < 0
10	Watchdog Error
11	Wd_RegNo out of range

6.6.7 Parameters list

Input parameters

Parameter	Description	Type	Default value
SlaveAdd	Slave address	INT	0
Fct_Code	Function code; 1, 2, 3, 4	INT	3
Reg_Strt	Register start address	INT	1
Reg_Off	Register start offset (0...49999)	DINT	0
Reg_No	Number of registers (Fc3, 4) or coils (Fc1, 2)	INT	1
Wd_Count	Watchdog Counter	INT	0
Wd_RegNo	Register number used for watchdog	INT	127
SlaveAdd_1	Slave address	BYTE	16#00
Fct_Code_1	Function code 1,2,3,4	BYTE	16#03
Reg_Strt_1	Register start address	INT	1
Reg_No_1	Number of registers	INT	1
CHG_Word	1=Change High- and Low-Word	BOOL	0
CHG_Byte	1=Change Hight- and Low-Byte	BOOL	0

Output parameters

Parameter	Description	Type	Default value
QRetVal	General return value	INT	-
QError	Error detect	BOOL	-
QDB_No	Instance data block number	INT	-
QSndStrt	Start address of send data	INT	-
QSnd_Len	Send length = (6 Byte Add-Header)	DINT	-
QRcvStrt	Start address of receive data	INT	-
QRcv_Len	Receive length = (Reg_No * 2)	DINT	-
QWd_Err	Watchdog error	BOOL	-
iRecv_1	1. int value; 2 Byte	INT	-
...	...	INT	-

Parameter	Description	Type	Default value
iRecv127	127. int value; 2 Byte	INT	-
WatchVal	Substitute int value because of watchdog; 2 Byte	INT	-

6.7 Modbus Master send block for integer values: MODMS_I

6.7.1 Type / Number / Symbol

MODMS_I (FB 243)

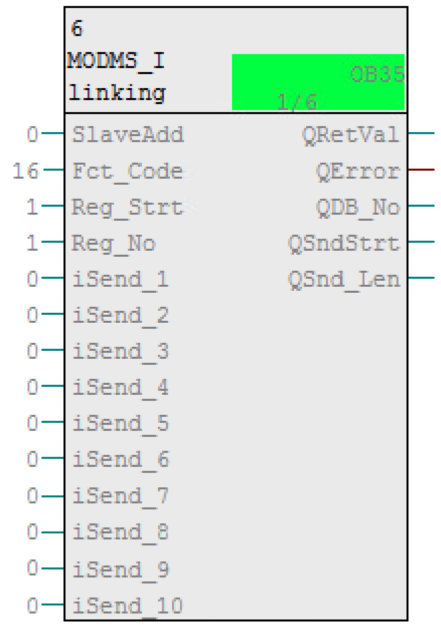


Figure 6-8 CFC-FB MODMS_I (FB 243)

6.7.2 Application field

The output parameter QDB_No, QSndStrt and QSndLen of MODMS_I are linked with the input parameters DBNO_x, SND_ADRx and SND_LENx of FB S7HMODM3 (FB240).

Each partner system (slave) on MODBUS has its own SlaveAdd. With function codes (Fct_code) the analogue and binary values are selected for writing the input or output registers or status to the partner system. The meaning of the function codes is as follows:

Functions code	Task
5	Writing of single bit: Write Single Coil
6	Writing of single register: Write single Register

Functions code	Task
15	Writing of up to 2032 bits: Write Multiple Coils
16	Writing of up to 127 registers: Write Multiple Registers

The parameter Reg_Strt addresses the first register for writing data. With function codes 5 and 15 Reg_Strt means the bit start address, for function codes 6 and 16 Reg_Strt means the register-start address. The register offset Reg_Off has a default value of 0. It will be added to the parameter Reg_Strt and can be adjusted in the range of [0...49999]. The maximum amount of registers to write (Reg_No) is 127 (function code 16), with function code 15 on Reg_No the number of bits given (max 2032).

The values to send have to be set or linked to the input parameter iSend_1 up to iSend127.

The FB MODMS_I has to be configured differently according to the selected function code.

Function code 5 "Write Single Coil"

Parameter	Meaning	Values
Reg_Strt	Coil/bit address	[0 to 32767]
Reg_Off	Address offset	[0 ... 49999]
Reg_No	Coil value	-256 (ON), 0 (OFF)

Function code 6 "Write Single Register"

Parameter	Meaning	Values
Reg_Strt	Register address	[0 to 32767]
Reg_Off	Address offset	[0 ... 49999]
Reg_No	Register value	[-32768 ... 32767]

Function code 15 "Write Multiple Coils"

Parameter	Meaning	Values
Reg_Strt	Coils start address	[0 to 32767]
Reg_Off	Address offset	[0 ... 49999]
Reg_No	Quantity of coils	[0 to 2032]

Function code 16 "Write Multiple Registers"

Parameter	Meaning	Values
Reg_Strt	Register start address	[0 to 32767]
Reg_Off	Address offset	[0 ... 49999]
Reg_No	Quantity of registers	[0 to 127]

6.7.3 Called organisation blocks (OBs)

- OB 35 (time cyclic interrupt) or any other time cycle OB, or
- OB 1 (cyclic OB)

6.7.4 Alarm reaction

This function block has no integrated alarm function.

6.7.5 Error handling

This function block controls some parameter on limits or plausibility. Failures are always indicated with QError = "1" and QRetVal ≠ "0".

Table 6-6 Status code for the FB 243 (MODMS_I)

QRetVal	Description
0	Normal Function
1	Spare
2	SlaveAdd = 0
3	Fct_Code < 1 or Fct_Code > 16
4	Reg_Strt <= 0 or Reg_Strt > 65536
5	Fct_Code: 15; Reg_No < 0 or Reg_No > 2032 (Bit_number)
6	Fct_Code: 16; Reg_No < 0 or Reg_No > 127 (Reg_number)
7	Reg_Off > 50000 or Reg_Off < 0
8	Fct_Code: 5; Reg_No <> -256 or Reg_No <> 0
9	Fct_Code: 6; Reg_No < -32768 or Reg_No > 32767

6.7.6 Parameters list

Input parameters

Parameter	Description	Type	Default value
SlaveAdd	Slave address	INT	0
Fct_Code	Function code; 5, 6, 15, 16	INT	16
Reg_Strt	Register start address	INT	1
Reg_Off	Register start offset (0...49999)	DINT	0
Reg_No	Number of registers; Fct_Code=6: Reg_Val	INT	1
SlaveAdd_1	Slave address	BYTE	16#00
Fct_Code_1	Function code 6,15,16	BYTE	16#10

Parameter	Description	Type	Default value
Reg_Strt_1	Register start address	INT	1
Reg_No_1	Number of registers; Fct_Code=6: Reg_VAL	INT	1

Output parameters

Parameter	Description	Type	Default value
iSend_1	1. int value; 2 Byte	INT	-
...	...	INT	-
iSend127	127. int value; 2 Byte	INT	-
QRetVal	General return value	INT	-
QError	Error detect	BOOL	-
QDB_No	Instance data block number	INT	-
QSndStrt	Start address of send data	INT	-
QSnd_Len	Send length= (Reg_No * 2) + (6 Byte Add-Header)	DINT	-

6.8 Modbus Master receive block for real values: MODMR_R

6.8.1 Type / Number / Symbol

MODMR_R (FB 244)

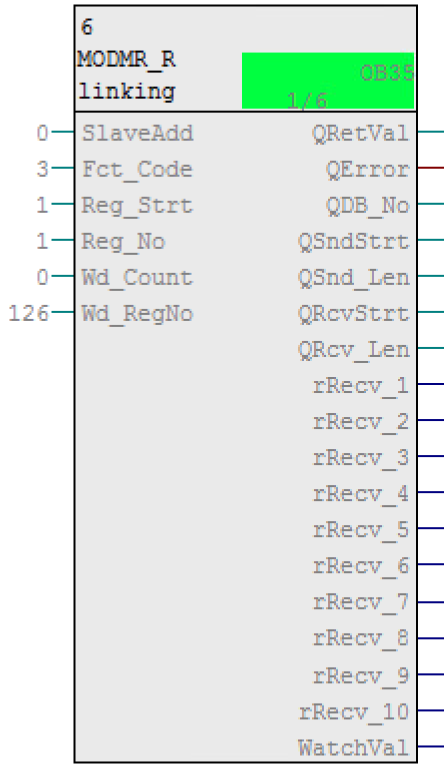


Figure 6-9 CFC- Function block MODMR_R (FB 244)

6.8.2 Application field

The output parameters QDB_No, QSndStrt, QSndLen, QRcvStrt and QRcvLen of MODMR_R are linked with the input parameters DBNO_x, SND_ADRx, SND_LENx, RCV_ADRx and RCV_LENx of FB S7HMODM3 (FB240).

Each partner system (slave) on MODBUS has its own SlaveAdd. With function codes (Fct_code) the analogue and binary values are selected for reading input or output registers or status from the partner system. The meaning of the function codes is as follows:

Functions code	Task
1	Read of up to 2016 bits: Read Coils
2	Read of up to 2016 bits: Read Discrete Inputs

Functions code	Task
3	Read of up to 63 registers (Real value): Read Holding Registers
4	Read of up to 63 registers (Real value): Read Input Registers

The input parameter Reg_Strt defines the first address of data to be read. The default setting of register offset Reg_Off is "0". This parameter is added to Reg_Strt and can be adjusted accordingly [0...49999]. The maximum amount of registers Reg_No that can be fetched is 63 REAL values (126 Registers; function code 3 and 4). With function code 1 and 2 Reg_No gives the maximum number of Bits (max. 2016)

The received values are available at the output rRecv_1 up to rRecv_63.

The FB MODMR_R has to be configured differently according to the selected function code.

Function code 1 "Read Coils"

According to the Modbus protocol, the quantity of coils is limited to [1 ...2000] and the starting address to [1 ... 65535].

Parameter	Meaning	Values
Reg_Strt	Starting address	[1 ... 32767]
Reg_Off	Address offset	[0 ... 49999]
Reg_No	Quantity of coils	[1 ... 2016]

Function code 2 "Read Discrete Inputs"

According to the Modbus protocol, the quantity of inputs is limited to [1 ...2000] and the starting address to [1 ... 65535].

Parameter	Meaning	Values
Reg_Strt	Starting address	[1 ... 32767]
Reg_Off	Address offset	[0 ... 49999]
Reg_No	Quantity of inputs	[1 ... 2016]

Function code 3 "Read Holding Registers"

According to the Modbus protocol, the quantity of holding registers is limited to [1 ...125] and the starting address to [1 ... 65535].

Parameter	Meaning	Values
Reg_Strt	Starting address	[1 ... 32767]
Reg_Off	Address offset	[0 ... 49999]
Reg_No	Quantity of registers	[1 ... 126]

Function code 4 "Read Input Registers" Principle

According to the Modbus protocol, the quantity of inputs registers is limited to [1 ...125] and the starting address to [1 ... 65535].

Parameter	Meaning	Values
Reg_Strt	Starting address	[1 ... 32767]
Reg_Off	Address offset	[0 ... 49999]
Reg_No	Quantity of registers	[1 ... 126]

Remark for the function code 3 and 4

Each real value given over one parameter rRecv_x is stored over 2 registers. For example, if the values for rRecv_1 to rRecv_10 have to be fetched (i.e. 20 registers), Reg_No must be set to 20.

The parameters CHG_Word and CHG_Byte are used to invert the low and high word and then the low and high byte of a received register.

According to the settings of CHG_BYTE and CHG_WORD, the register value given over the output parameter may be modified.

For example, let’s assume that the register receives 4 bytes A, B, C and D with A the high byte.

CHG_Word	0	0	1	1
CHG_Byte	0	1	0	1
Output	ABCD	BADC	CDAB	DCBA

6.8.3 Called organisation blocks (OBs)

- OB 35 (time cyclic interrupt) or any other time cycle OB, or
- OB 1 (cyclic OB)

6.8.4 Watchdog function

This watchdog function can be activated with the parameter Wd_Count ≠ "0". The watchdog time is a result of (Wd_Count * time cycle of block MODMR_R). The parameter Wd_RegNo addresses the register or Bit that shall be controlled. The value of the controlled register sent by the partner system must be unequal to "0". On each call of the function block the received value is checked and will be overwritten with "-1". If the actually received value is unequal to "0" the watchdog will be reset, if no value is received or the value is "0" the output parameter QWdErr and QError are set to "1" and QRetVal is set to "10". To also use the controlled register, link it to WatchVal.

6.8.5 Alarm reaction

This function block has no integrated alarm function.

6.8.6 Error handling

This function block controls some alarm parameter such as limit and plausibility control as well as the watchdog control. Failures are always indicated with the output bit QError = "1" and QRetVal ≠ "0".

Table 6-7 Status code for the FB 244 (MODMR_R)

QRetVal	Description
0	No error
1	Spare
2	SlaveAdd = 0
3	Fct_Code <> 1,2,3 or 4
4	Reg_Strt <= 0
5	Fct_Code: 1, 2; Reg_No < 1 or Reg_No > 2016 (Bit_number)
6	Fct_Code: 3, 4; Reg_No < 1 or Reg_No > 126 (Reg_number)
7	Reg_Off > 50000 or Reg_Off < 0
10	Watchdog Error
11	Wd_RegNo out of range

6.8.7 Parameters list

Input parameters

Parameter	Description	Type	Default value
SlaveAdd	Slave address	INT	0
Fct_Code	Function code; 1, 2, 3, 4	INT	3
Reg_Strt	Register start address	INT	1
Reg_Off	Register start offset (0...49999)	DINT	0
Reg_No	Number of registers (Fc3, 4) or coils (Fc1, 2)	INT	1
Wd_Count	Watchdog Counter	INT	0
Wd_RegNo	Register number used for watchdog	INT	127
SlaveAdd_1	Slave address	BYTE	16#00
Fct_Code_1	Function code 1,2,3,4	BYTE	16#03
Reg_Strt_1	Register start address	INT	1
Reg_No_1	Number of registers	INT	1
CHG_Word	1=Change High- and Low-Word	BOOL	0
CHG_Byte	1=Change Hight- and Low-Byte	BOOL	0

Output Parameters

Parameter	Description	Type	Default value
QRetVal	General return value	INT	-
QError	Error detect	BOOL	-
QDB_No	Instance data block number	INT	-
QSndStrt	Start address of send data	INT	-
QSnd_Len	Send length = (6 Byte Add-Header)	DINT	-
QRcvStrt	Start address of receive data	INT	-
QRcv_Len	Receive length = (Reg_No * 4)	DINT	-
QWd_Err	Watchdog error	BOOL	-
rRecv_1	1. real value; 4 Byte	REAL	-
...	...	REAL	-
rRecv_63	63. real value; 4 Byte	REAL	-
WatchVal	Substitute int value because of watchdog; 2 Byte	INT	-

6.9 Modbus Master send block for real values: MODMS_R

6.9.1 Type / Number / Symbol

MODMS_R (FB 245)

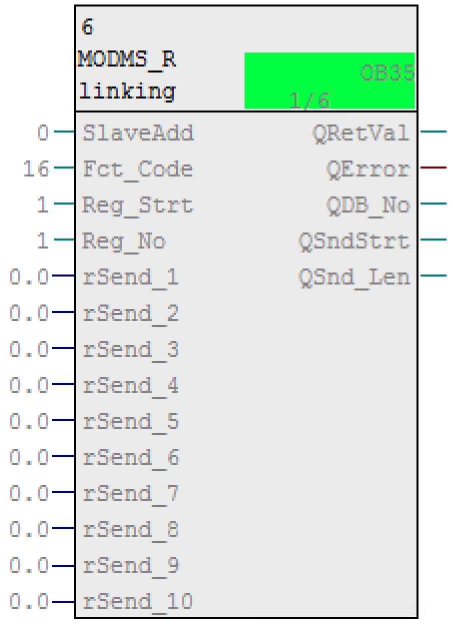


Figure 6-10 CFC-FB MODMS_R (FB 245)

6.9.2 Application field

The output parameters QDB_No, QSndStrt and QSndLen of MODMS_R are linked with the input parameters DBNO_x, SND_ADRx and SND_LENx of FB S7HMODM3 (FB240).

Each partner system (slave) on MODBUS has its own SlaveAdd. With function codes (Fct_code) the analogue and binary values are selected for writing the input or output registers or status to the partner system. The meaning of the function codes is as follows:

Functions code	Task
5	Writing of single bit: Write Single Coil
6	Writing of single register: Write Single Register
15	Writing of up to 2016 bits: Write Multiple Coils
16	Writing of up to 63 REAL values (126 Registers)
	Write Multiple Registers

The parameter Reg_Strt addresses the first register for writing data. Reg_Strt means the bit-start-address with function code 5 and 15 and register-start address with function codes 6 and 16. The register offset Reg_Off has a de-fault value of "0". It will be added to the parameter Reg_Strt and can be adjusted in the range of [0...49999]. The maximum amount writing registers Reg_No is 126 (Function code 16); with function code 15 this parameter gives the number of bits (max. 2016).

The values to5 send have to be set or linked to the input parameter rSend_1 up to rSend_63.

The FB MODMS_R has to be configured differently according to the selected function code.

Function code 5 "Write Single Coil"

Parameter	Meaning	Values
Reg_Strt	Coil/bit address	[0 to 32767]
Reg_Off	Address offset	[0 ... 49999]
Reg_No	Coil value	-256 (ON), 0 (OFF)

Function code 6 "Write Single Register" Introduction

Parameter	Meaning	Values
Reg_Strt	Register address	[0 to 32767]
Reg_Off	Address offset	[0 ... 49999]
Reg_No	Register value as INT	[-32768 ... 32767]

Function code 15 "Write Multiple Coils"

Parameter	Meaning	Values
Reg_Strt	Coils start address	[0 to 32767]
Reg_Off	Address offset	[0 ... 49999]
Reg_No	Quantity of coils	[0 to 2016]

Function code 16 "Write Multiple Registers"

Parameter	Meaning	Values
Reg_Strt	Register start address	[0 to 32767]
Reg_Off	Address offset	[0 ... 49999]
Reg_No	Quantity of registers	[0 to 126]

6.9.3 Called organisation blocks (OBs)

- OB 35 (time cyclic interrupt) or any other time cycle OB, or
- OB 35 (time cyclic interrupt) or any other time cycle OB, or

6.9.4 Alarm reaction

This function block has no integrated alarm function.

6.9.5 Error handling

This function block controls some parameter on limits or plausibility. Failures are always indicated with QError = "1" and QRetVal ≠ "0".

Table 6-8 Status code for the FB 245 (MODMS_R)

QRetVal	Description
0	No error
1	Spare
2	SlaveAdd = 0
3	Fct_Code <> 1,2,3 or 4
4	Reg_Strt <= 0
5	Fct_Code: 15; Reg_No < 1 or Reg_No > 2016 (Bit_number)
6	Fct_Code: 16; Reg_No < 1 or Reg_No > 126 (Reg_number)
7	Reg_Off > 50000 or Reg_Off < 0
8	Fct_Code: 5; Reg_No <> -256 or Reg_No <> 0
9	Fct_Code: 6; Reg_No < -32768 or Reg_No > 32767

6.9.6 Parameters list

Input parameters

Parameter	Description	Type	Default value
SlaveAdd	Slave address	INT	0
Fct_Code	Function code; 5, 6, 15, 16	INT	16
Reg_Strt	Register start address	INT	1
Reg_Off	Register start offset (0...49999)	DINT	0
Reg_No	Number of registers; Fct_Code=6: Reg_Val	INT	1
SlaveAdd_1	Slave address	BYTE	16#00
Fct_Code_1	Function code 6,15,16	BYTE	16#10
Reg_Strt_1	Register start address	INT	1
Reg_No_1	Number of registers; Fct_Code=6: Reg_VAL	INT	1

Output parameters

Parameter	Description	Type	Default value
rSend_1	1. real value; 4 Byte	REAL	-
...	...	REAL	-
rSend_63	63. real value; 4 Byte	REAL	-
QRetVal	General return value	INT	-
QError	Error detect	BOOL	-
QDB_No	Instance data block number	INT	-
QSndStrt	Start address of send data	INT	-
QSnd_Len	Send length= (Reg_No * 4) + (6 Byte Add-Header)	DINT	-

6.10 Modbus Master receive block for double integer values: MODMR_DI

6.10.1 Type / Number / Symbol

MODMR_DI (FB 246)

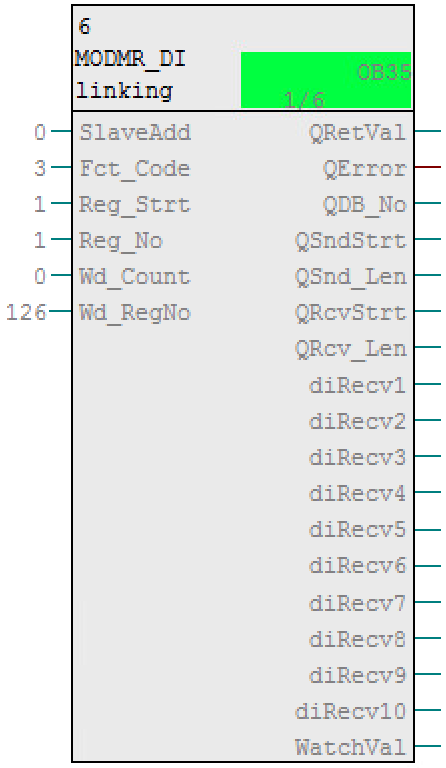


Figure 6-11 CFC- Function block MODMR_DI (FB 246)

6.10.2 Application field

The output parameters QDB_No, QSndStrt, QSndLen, QRcvStrt and QRcvLen of MODMR_DI are linked with the input parameters DBNO_x, SND_ADRx, SND_LENx, RCV_ADRx and RCV_LENx of the FB S7HMODM3 (FB240).

Each partner system (slave) on MODBUS has its own SlaveAdd. With function codes (Fct_code) the analogue and binary values are selected for reading input or output registers or status from the partner system. The meaning of the function codes is as follows:

Functions code	Task
1	Read of up to 2016 bits: Read coils
2	Read of up to 2016 bits: Read discrete inputs

Functions code	Task
3	Read of up to 63 double integer values (126 Registers) Read holding registers
4	Read of up to 63 double integer values (126 Registers) Read Input Registers

Over Reg_Strt is given the address of the first register (or coil) to fetch. The register offset Reg_Off is as default "0". It is added to Reg_Strt and can be adjusted accordingly [0...49999]. The maximum amount of registers Reg_No to be fetched is 63 Double Integer (126 Register; function code 3 and 4). With function code 1 and 2 Reg_No gives the number of bits (max. 2016).

The received values are available at the outputs diRecv1 to diRecv63 for further data treatment.

The FB MODMR_DI has to be configured differently according to the selected function code.

Function code 1 "Read Coils"

According to the Modbus protocol, the quantity of coils is limited to [1 ...2000] and the starting address to [1 ... 65535].

Parameter	Meaning	Values
Reg_Strt	Starting address	[1 ... 32767]
Reg_Off	Address offset	[0 ... 49999]
Reg_No	Quantity of coils	[1 ... 2016]

Function code 2 "Read Discrete Inputs"

According to the Modbus protocol, the quantity of inputs is limited to [1 ...2000] and the starting address to [1 ... 65535].

Parameter	Meaning	Values
Reg_Strt	Starting address	[1 ... 32767]
Reg_Off	Address offset	[0 ... 49999]
Reg_No	Quantity of inputs	[1 ... 2016]

Function code 3 "Read Holding Registers"

According to the Modbus protocol, the quantity of holding registers is limited to [1 ...125] and the starting address to [1 ... 65535].

Parameter	Meaning	Values
Reg_Strt	Starting address	[1 ... 32767]
Reg_Off	Address offset	[0 ... 49999]
Reg_No	Quantity of registers	[1 ... 126]

Function code 4 "Read Input Registers"

According to the Modbus protocol, the quantity of inputs registers is limited to [1 ...125] and the starting address to [1 ... 65535].

Parameter	Meaning	Values
Reg_Strt	Starting address	[1 ... 32767]
Reg_Off	Address offset	[0 ... 49999]
Reg_No	Quantity of registers	[1 ... 126]

Remark for the function code 3 and 4

Each real value given over one parameter rRecv_x is stored over 2 registers. For example, if the values for rRecv_1 to rRecv_10 have to be fetched (i.e. 20 registers), Reg_No must be set to 20.

The parameters CHG_Word and CHG_Byte are used to invert the low and high word and then the low and high byte of a received register.

According to the settings of CHG_BYTE and CHG_WORD, the register value given over the output parameter may be modified.

For example, let’s assume that the register receives 4 bytes A, B, C and D. A is the high byte.

CHG_Word	0	0	1	1
CHG_Byte	0	1	0	1
Output	ABCD	BADC	CDAB	DCBA

6.10.3 Called organisation blocks (OBs)

- OB 35 (time cyclic interrupt) or any other time cycle OB, or
- OB 1 (cyclic OB)

6.10.4 Watchdog function

This watchdog function can be activated with the parameter Wd_Count ≠ "0". The watchdog time is a result of (Wd_Count * time cycle of block MODMR_I). The parameter Wd_RegNo addresses the register or Bit that shall be controlled. The value of the controlled register sent by the partner system must be unequal to "0". On each call of the function block the received value is checked and will be overwritten with "-1". If the actually received value is unequal to "0" the watchdog will be reset, if no value is received or the value is "0" the output parameter QWdErr and QError are set to "1" and QRetVal is set to "10". To also use the controlled register, link it to WatchVal.

6.10.5 Alarm reaction

This function block has no integrated alarm function.

6.10.6 Error handling

This function block controls some alarm parameter – such as limit and plausibility control as well as the watchdog control. Failures are always indicated with the output bit QError = "1" and QRetVal ≠ "0".

Table 6-9 Status code for the FB 245 (MODMS_R)

QRetVal	Description
0	No error
1	Spare
2	SlaveAdd = 0
3	Fct_Code <> 1, 2, 3 or 4
4	Reg_Strt <= 0
5	Fct_Code: 1, 2; Reg_No < 1 or Reg_No > 2016 (Bit_number)
6	Fct_Code: 3, 4; Reg_No < 1 or Reg_No > 126 (Reg_number)
7	Reg_Off > 50000 or Reg_Off < 0
10	Watchdog Error
11	Wd_RegNo out of range

6.10.7 Parameters list

Input parameters

Parameter	Description	Type	Default value
SlaveAdd	Slave address	INT	0
Fct_Code	Function code; 1, 2, 3, 4	INT	3
Reg_Strt	Register start address	INT	1
Reg_Off	Register start offset (0...49999)	DINT	0
Reg_No	Number of registers (Fc3, 4) or coils (Fc1, 2)	INT	1
Wd_Count	Watchdog Counter	INT	0
Wd_RegNo	Register number used for watchdog	INT	127
SlaveAdd_1	Slave address	BYTE	16#00
Fct_Code_1	Function code 1,2,3,4	BYTE	16#03
Reg_Strt_1	Register start address	WORD	16#001
Reg_No_1	Number of registers	INT	1
CHG_Word	1=Change High- and Low-Word	BOOL	0
CHG_Byte	1=Change Hight- and Low-Byte	BOOL	0

Output parameters

Parameter	Description	Type	Default value
QRetVal	General return value	INT	-
QError	Error detect	BOOL	-
QDB_No	Instance data block number	INT	-
QSndStrt	Start address of send data	INT	-
QSnd_Len	Send length = (6 Byte Add-Header)	DINT	-
QRcvStrt	Start address of receive data	INT	-
QRcv_Len	Receive length = (Reg_No * 4)	DINT	-
QWd_Err	Watchdog error	BOOL	-
diRecv_1	1. double int value; 4 Byte	DINT	-
...	...	DINT	-
diRecv63	63. double int value; 4 Byte	DINT	-
WatchVal	Substitute int value because of watchdog; 2 Byte	INT	-

6.11 Modbus Master send block for double integer values: MODMS_DI

6.11.1 Type / Number / Symbol

MODMS_DI (FB 247)

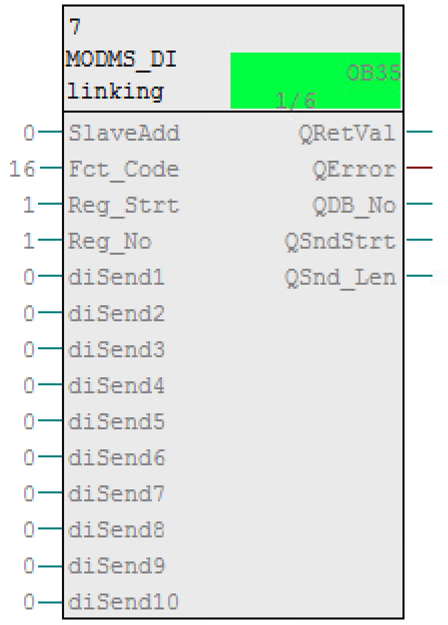


Figure 6-12 CFC MODMS DI (FB 247)

6.11.2 Application field

The output parameters QDB_No, QSndStrt and QSndLen of "MODMS_DI" (FB243) are linked with the input parameters DBNO_x, SND_ADRx and SND_LENx of FB "S7HMODM3"(FB240).

Each partner system (slave) on MODBUS has its own SlaveAdd. With function codes (Fct_code) the analogue and binary values are selected for writing the input or output registers or status to the partner system. The meaning of the function codes is as follows

Functions code	Task
5	Writing of single bit: force single coil
6	Writing of single registers: preset single register (Reg_No = register value)
15	Writing of up to 2016 bits: force multiple coils
16	Writing of up to 63 REAL values (126 Register: preset multiple registers)

The parameter Reg_Strt addresses the first register for writing data. Reg_Strt means the register-start address with function codes 6 and 16. The register offset Reg_Off has a default value of "0". It will be added to the parameter Reg_Strt and can be adjusted in the range of [0...49999]. The maximum amount of register Reg_No for writing data is 63 (Function code 16). For function code 15 Reg_Off gives the number of Bits (max. 2016).

The values to send have to be set or linked to the input parameter diSend_1 up to diSend_63.

The FB MODMS_DI has to be configured differently according to the selected function code.

Function code 5 "Write Single Coil"

Parameter	Meaning	Values
Reg_Strt	Coil/bit address	[0 to 32767]
Reg_Off	Address offset	[0 ... 49999]
Reg_No	Coil value	-256 (ON), 0 (OFF)

Function code 6 "Write Single Register"

Parameter	Meaning	Values
Reg_Strt	Register address	[0 to 32767]
Reg_Off	Address offset	[0 ... 49999]
Reg_No	Register value as INT	[-32768 ... 32767]

Function code 15 "Write Multiple Coils"

Parameter	Meaning	Values
Reg_Strt	Coils start address	[0 to 32767]
Reg_Off	Address offset	[0 ... 49999]
Reg_No	Quantity of coils	[0 to 2016]

Function code 16 "Write Multiple Registers"

Parameter	Meaning	Values
Reg_Strt	Register start address	[0 to 32767]
Reg_Off	Address offset	[0 ... 49999]
Reg_No	Quantity of registers	[0 to 216]

6.11.3 Called organisation blocks (OBs)

- OB 35 (time cyclic interrupt) or any other time cycle OB, or[
- OB 1 (cyclic OB)

6.11.4 Alarm reaction

This function block has no integrated alarm function.

6.11.5 Error handling

This function block controls some parameter on limits or plausibility. Failures are always indicated with QError = "1" and QRetVal ≠ "0".

Table 6-10 Status code for the FB 247 MODMS_DI

QRetVal	Description
0	No error
1	Spare
2	SlaveAdd = 0
3	Fct_Code < 1 or Fct_Code > 16
4	Reg_Strt <= 0
5	Fct_Code: 15; Reg_No < 0 or Reg_No > 2016 (Bit_number)
6	Fct_Code: 16; Reg_No < 0 or Reg_No > 126 (Reg_number)
7	Reg_Off > 50000 or Reg_Off < 0

6.11.6 Parameters list

Input parameters

Parameter	Description	Type	Default value
SlaveAdd	Slave address	INT	0
Fct_Code	Function code; 5, 6, 15, 16	INT	16

Parameter	Description	Type	Default value
Reg_Strt	Register start address	INT	1
Reg_Off	Register start offset (0...49999)	DINT	0
Reg_No	Number of registers; Fct_Code=6: Reg_Val	INT	1
SlaveAdd_1	Slave address	BYTE	16#00
Fct_Code_1	Function code 6,15,16	BYTE	16#10
Reg_Strt_1	Register start address	INT	1
Reg_No_1	Number of registers; Fct_Code=6: Reg_VAL	INT	1

Output parameters

Parameter	Description	Type	Default value
diSend_1	1. double int value; 4 Byte	DINT	-
...	...	DINT	-
diSend63	63. double int value; 4 Byte	DINT	-
QRetVal	General return value	INT	-
QError	Error detect	BOOL	-
QDB_No	Instance data block number	INT	-
QSndStrt	Start address of send data	INT	-
QSnd_Len	Send length= (Reg_No * 4) + (6 Byte Add-Header)	DINT	-