

SIEMENS

SIMATIC

S7 - OPEN MODBUS / TCP Communication via CP343-1 and CP443-1

Manual

SIEMENS

SIMATIC S7

OPEN MODBUS / TCP Communication via CP343-1 and 443-1

Manual

Preface, Table of Contents

Product Description **1**

Getting Started **2**

Commissioning **3**

Licensing **4**

FB MODBUSCP **5**

Diagnostics **6**

Application Samples **7**

Appendices

Literature

Glossary

Safety Precautions and Warnings

This manual contains warnings, which you should note for your own safety as well as for the prevention of damage to property. These warnings are indicated by means of a triangle and displayed as follows in accordance with the level of danger:



Danger

indicates that loss of life, severe personal injury or substantial damage **will** result if proper precautions are not taken.



Warning

indicates that loss of life, severe personal injury or substantial damage **can** result if proper precautions are not taken.



Caution

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

Note

call attention to information that is especially significant to the product, handling of the product or a specific part of this documentation.

Qualified Personnel

The equipment may be commissioned and put into operation by **qualified personnel** only. For the purpose of safety relevant warnings of this manual a qualified person is one who is authorized to commission, ground and tag devices, systems and circuits.

Use as prescribed

Please note the following:



Warning

This equipment must only be used in applications as prescribed in the catalogue and the technical description and in conjunction with equipment and components recommended and authorized by Siemens.

Successful and safe operation of this equipment is dependent upon proper transport, and storage, erection and installation as well as careful operation and maintenance.

SIMATIC[®] and SIMATIC NET[®] are registered trademarks of SIEMENS AG.

Trademarks

The other brand names in this manual may be trademarks use of which by third parties for their purposes may infringe the proprietors' rights.

Copyright © Siemens AG 2004-2009 All Rights Reserved

The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration of a utility model or design, are reserved.

Siemens AG
Industry Sector
Engineering and Construction
I IS IN E&C
P.O. Box 3240, D- 91050 Erlangen
IT4.Industry@siemens.com

Exclusion from Liability

We have checked the contents of this document with regard to conformity to the described hardware and software. Deviations, however, cannot be excluded; therefore we cannot accept prejudice for its complete conformity. The information in this document is checked regularly and necessary corrections are contained in subsequent issues. Any suggestions for improvement are gratefully received.

We reserve the right to make technical changes.

Preface

Purpose of the Manual

The information in this manual allows you to set up and put in operation the connection between the CP 343-1/CP 443-1 and a device that supports the Open MODBUS/TCP protocol.

Contents of the Manual

This manual describes the function of the function block **MODBUSCP** and its integration into the hardware and software of the communication processors CP 343-1/CP 443-1.

The manual contains the following topics:

- Production description / Mounting
- Getting Started
- Commissioning
- Parameterization
- Licensing
- Function block MODBUSCP
- Diagnostics
- Application samples

Scope of this Manual

This manual is valid for the following software:

Product	Identification number	From version
OPEN MODBUS / TCP	2XV9 450-1MB00	4.3
FB 108 "MODBUSCP"		1.3 / 2.2
FB 106 "MB_CPCLI"		1.2 / 2.2
FB 107 "MB_CPSRV"		1.2 / 2.1

Note

This manual contains the FB description, as it is valid at the time of publication.

How to Access the Information in this Manual

To enable you a quick access to selected information, the manual provides the following access aids:

- The next pages contain a complete table of contents.

Additional Sources of Information	All additional information concerning CP 343/CP 443 (mounting, commissioning etc.) can be found in the manual
	SIEMENS SIMATIC NET S7-CPs for Industrial Ethernet device manual C79000-G8900-C155
	SIEMENS SIMATIC NET S7-CPs for Industrial Ethernet device manual part B1 CP 343-1 / CP 343-1EX20 C79000-G8900-C158
	SIEMENS SIMATIC NET S7-CPs for Industrial Ethernet device manual part B4 CP 443-1 C79000-G8900-C152
	SIEMENS SIMATIC NET NCM S7 for Industrial Ethernet manual C79000-G8900-C129
	Additional information concerning STEP7 can be found in the following manuals:
	SIEMENS SIMATIC Software Base software for S7 and M7 STEP7 user manual C79000-G7000-C502-..
	SIEMENS SIMATIC Software System software for S7-300/400 System and standard functions Reference manual C79000-G7000-C503-02
Additional Questions	If you have further questions regarding the use of the FBs described in this manual, which are not answered in this document, please contact your Siemens partner who supplied you with this function block.
Terminology	This document uses the term CP or CP 343/CP 443. The descriptions only apply to communications processor CP 343-1/CP 443-1 .
Scope of Application	The function block described in this manual establishes a connection between the CP 343-1/CP 443-1 and third party MODBUS devices.

Table of Contents

1 Product Description.....	1-1
1.1 Field of Applications.....	1-1
1.2 Hardware and Software Prerequisites.....	1-2
2 Getting Started	2-1
3 Commissioning	3-1
3.1 Installing the Library on the STEP7 PG/-PC	3-1
3.2 Parameterization of the CP	3-2
3.3 Network Configuration	3-4
3.4 How to Select the Appropriate Modbus Block	3-9
3.5 Insertion of the Function Blocks into the Program.....	3-10
3.6 Multiple Connections via Port 502	3-12
3.7 Start_up Characteristics of CP343 / CP443	3-13
3.8 Changeover of a Previous Version to V4.x	3-14
4 Licensing	4-1
5 Function Block MODBUSCP	5-1
5.1 Functionality of the FB	5-1
5.2 Parameters of the Function Block MODBUSCP	5-4
5.3 Address Mapping.....	5-12
5.4 Data and Standard Functions used by the FB	5-15
5.5 Renaming of Standard Functions	5-17
6 Diagnostics	6-1
6.1 Diagnostics via the Display Elements of the CP	6-1
6.2 Diagnostic Messages of the FB MODBUSCP	6-2
6.3 Diagnostic Messages of FC5/FC6 and FC50/FC60.....	6-8
6.4 Diagnostic Messages of included FCs/SFCs	6-8
6.5 Diagnostic Messages of SFC24	6-8
7 Application Samples.....	7-1

7.1	Sample project STL	7-2
7.2	Sample project CFC	7-3

1 Product Description

1.1 Field of Applications

Placement in the System Environment

The driver described here is a software product for the Communications Processor CP343-1/CP443-1.

The CP343-1 can be used in the SIMATIC S7-300 automation system and can establish communication links to partner systems.

CP443-1 can be used in the SIMATIC S7-400 automation systems and can establish communication links to partner systems.

Function of the FBs

These function blocks enable a communication link between CP343-1 / CP443-1 and a device that supports the Open MODBUS/TCP protocol. The function codes 1, 2, 3, 4, 5, 6, 15 and 16 are provided.

Data transmission is carried out following the Client-Server principle.

The SIMATIC S7 can act as both client and server during the data transmission.

TCP/IP with CP343-1 / CP443-1

TCP/IP with CP343-1 and CP443-1 uses static connections. The TCP connection is not disconnected during error-free operation.

Network configuration of STEP7 enables only a **unique use of a specific port number**, when using TCP native stack of the CP.

However, with specific CP modules it is possible to use multiple connection via port 502 to different clients simultaneously.

In section 3.6 "Multiple connections via port 502" you can find technical details regarding this matter.

1.2 Hardware and Software Prerequisites

Usable Modules	You can find the latest hardware prerequisites on the internet: www.siemens.com/s7modbus .
Software Versions	The usage of the modbus blocks is possible with STEP7 Version 5.4 or higher. Withal the use of the blocks AG_LSEND/AG_LRECV V3.1 of the update of SIMATIC NET library (http://support.automation.siemens.com/WW/view/en/22172239) is required.
Memory requirements	<p>The FB MODBUSCP requires ca. 6k byte work and load memory.</p> <p>The FB MB_CPCLI requires ca.10k byte work and memory.</p> <p>The FB MB_CPSRV requires ca. 10 k byte work and load memory.</p> <p>The particular length of the blocks can be determined in the properties of the block in the Simatic Manager.</p>

2 Getting Started

Procedure

1. Install "OpenModbusTCP CP".
=> Section 3.1
2. Parameterize the connection parameters regarding your requirements (IP-address, port number, etc.).
=> Section 3.2 to 3.4
3. Insert the Modbus function blocks into your SIMATIC project.
=> Section 3.6
4. Call the Modbus block FB108 in the required OBs.
=> Section 5.1
5. Parameterize the Modbus block for initialization (client/server, DB areas, Modbus addresses) and runtime.
=> Section 5.2
6. Load the user program into the PLC and license the Modbus block for this CPU.
=> Section 4

3 Commissioning

General Information	<p>The configuration of the CP443-1/CP343-1 is possible via MPI or LAN/Industrial Ethernet.</p> <p>STEP7 is required with NCM S7 for Industrial Ethernet (shortly named "NCM IE").</p> <p>The information below about STEP7 and the communication link configuration refers to STEP7 Version 5.1 and NCM S7 Industrial Ethernet Version 5.1.</p> <p>In later versions, the sequences, names and directories might be different.</p>
Requirements	Knowledge of AWL and basic knowledge of STEP7 and PLC.

3.1 Installing the Library on the STEP7 PG-PC

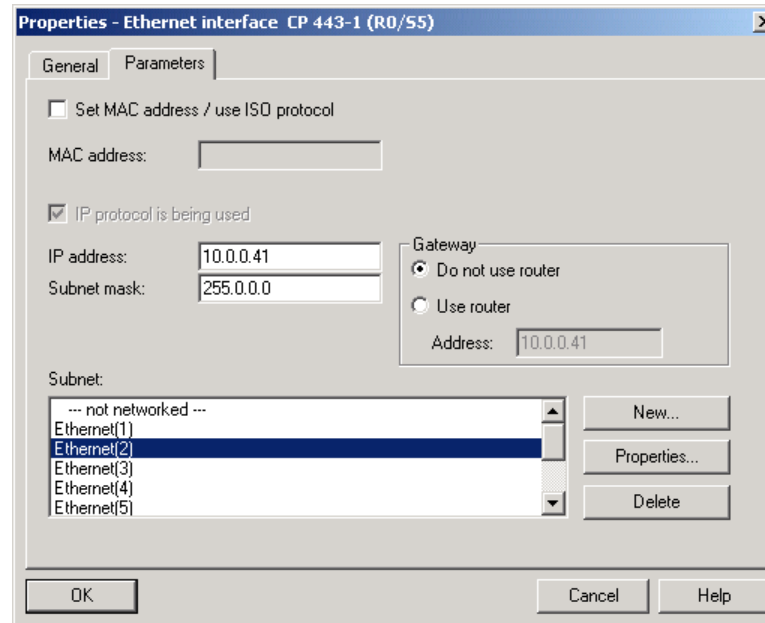
What We Provide You	<p>The attached CD contains a setup, which installs the library "Modbus_TCP_CP", 2 example projects and the manuals in English and German in the corresponding STEP7 directories.</p> <p>The manuals are available outside the setup additionally, to give you the opportunity to get all information regarding the software without installation.</p>
Requirements	To install, STEP7 must be installed. With older STEP7 versions the option NCM S7 for Industrial Ethernet is necessary additionally.
Installation	<p>Insert your Modbus CD into the CD-ROM drive and follow step-by-step the instructions that are automatically displayed by the installation program. If the installation program fails to automatically run, perform these steps:</p> <ol style="list-style-type: none"> 1. Using Windows Explorer, navigate to the CD-ROM drive and go to the directory setup and double-click Setup.exe file to start the installation procedure. 2. Follow step-by-step the instructions that are displayed by the installation program. <p>Now you can find</p> <ul style="list-style-type: none"> • the library in \Program Files\Siemens\Step7\S7libs, • 2 example projects in \Program Files\Siemens\Step7\Examples, • the manual in \Program Files\Siemens\Step7\S7manual\S7Comm, • the software registration form in \Program Files\Siemens\Step7\S7libs\Modbus_TCP_CP. <p>To access the Modbus library the first time, use the browse function of the open dialog for libraries.</p> <p>The manual can be accessed via short cut in \Program Files \Siemens \Documentation as well.</p>

3.2 Parameterization of the CP

Parameterization of the CP

If you have your stations connected with each other without a router, then they have to be within the same subnet.

In the field *Subnet* connect the CP with the Industrial Ethernet. In order to do that, select the entry with the name of your network. For newly created networks this is normally “**Ethernet(1)**”.



Save and Compile the parameterization.

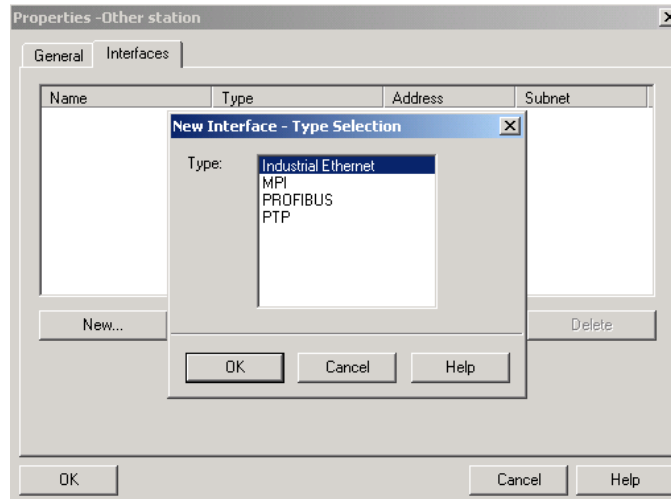
Parameterization of the Communication Partner

In the mode “**CP is client**” an “**other station**” is required for network configuration.

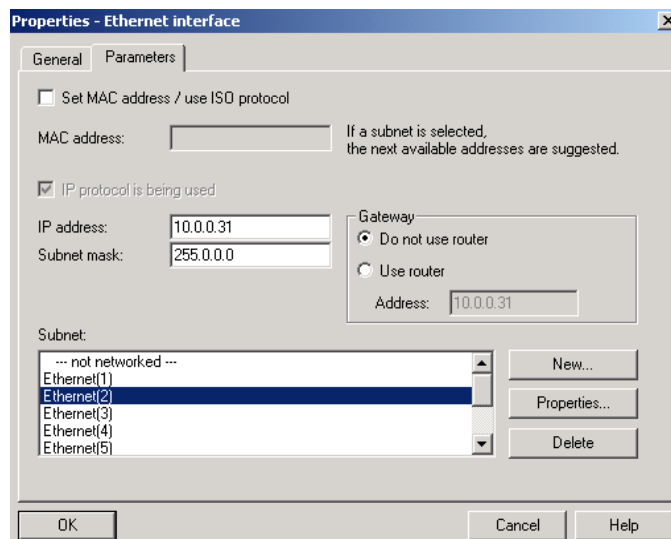
After you have inserted the communication partner’s station into your STEP7 project (as described in “Project components: Insert ➤ Other station”) you have to specify the object properties of the external station.

1) Properties – Other Station ➤ Interfaces

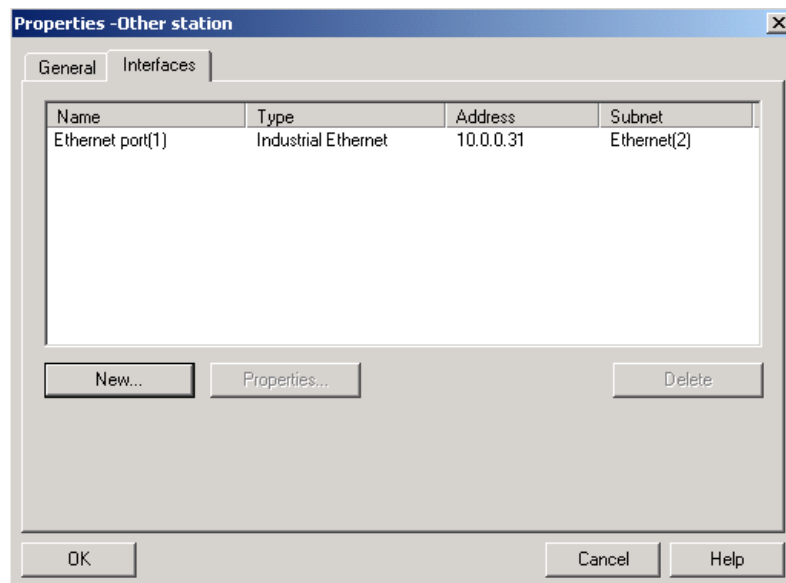
On the tab “**Interfaces**” click on “New”. In the upcoming selection, select “Industrial Ethernet” and click on “OK”.



This opens a dialog box “Properties – Ethernet Interface“. Enter an IP Address that is in the same subnet as the communication partner’s station. The subnet mask should be the same as the one of the partner’s station. Select the associated **subnet** that connects the CP interface with the communication partner’s interface.



Click on the “OK” button. This will bring you back to the tab “Interfaces”.



2) Properties – Other Station ➤ General

In tab “**General**” you do not have to make any settings.

3.3 Network Configuration

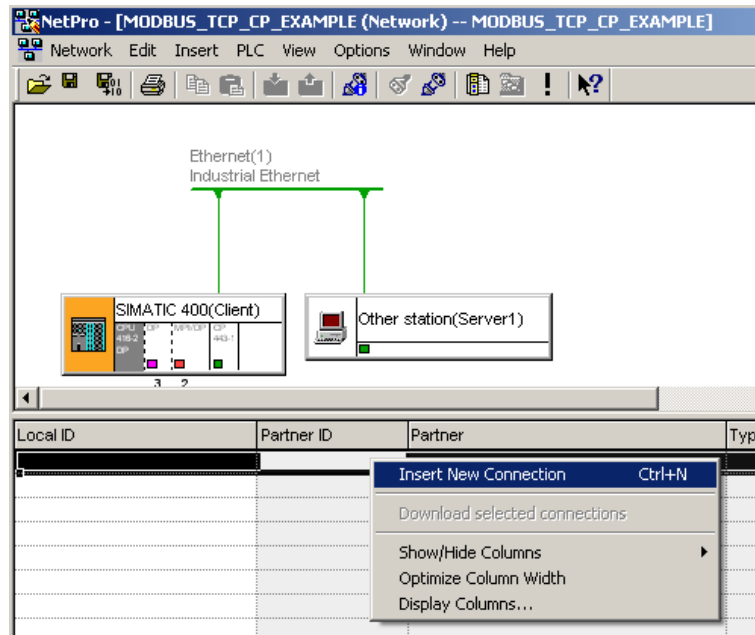
Communications Connection

The CP is the link for the Industrial Ethernet connection between the S7-CPU and the communication partner / bus. A connection configuration must be made for the connection of the interfaces to the communication partner / bus.

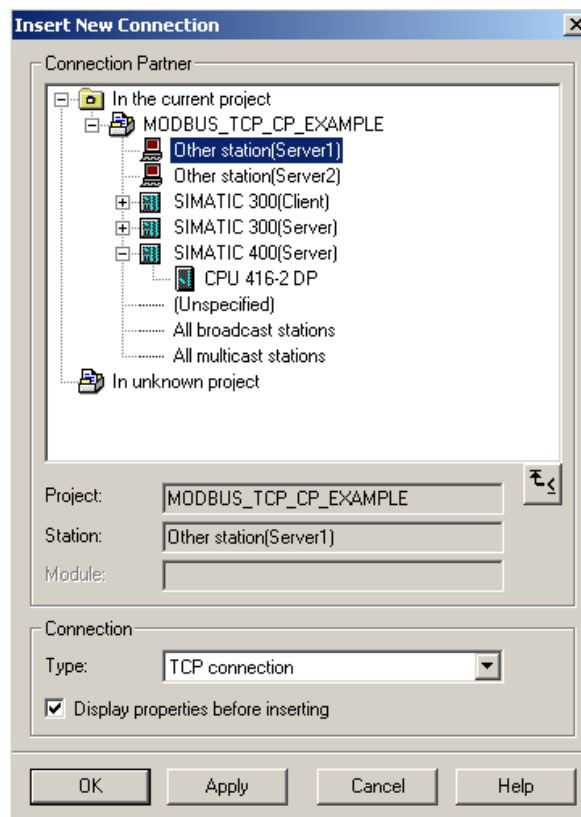
Configure Network in mode “CP is Client”

In the STEP7 project, select the CPU in your S7 300/S7 400-Station and open “Network configuration” by double clicking “**Connections**”. This opens the program “**NetPro**” with which your connections can be configured.

After selecting **Insert** > **New Connection...** the dialog box "Insert new connection" will come up.



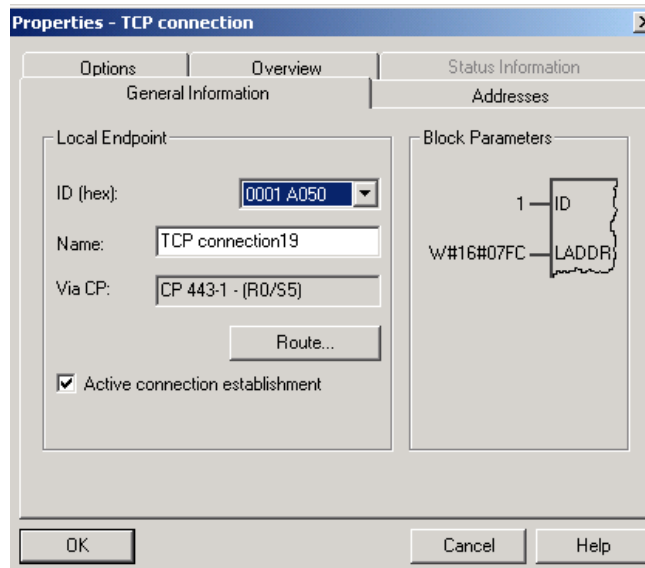
Select the connection partner (Other Station) for the new connection and use "TCP Connection" for the connection. Put a check mark on "Show properties dialog"



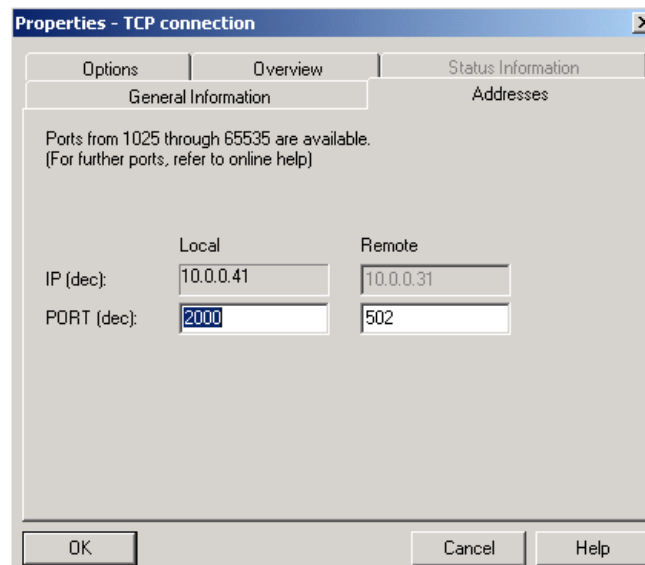
Click "OK". This will take you back to the dialog box "Properties TCP connection".

Object Properties of the Connection

- An **ID** is provided. You can change the ID if needed.
- Click on the button “Routing” and the configured connection will be shown.
- The MODBUS client must execute “Active connection establishment”.



- In the register “Addresses” the port numbers are defined.



Click on “OK” and the inputs are accepted.

Save the network configuration and close the program “NetPro”.

Please note that the connection ID (Local ID) has to be used when the FB is called in the user program.

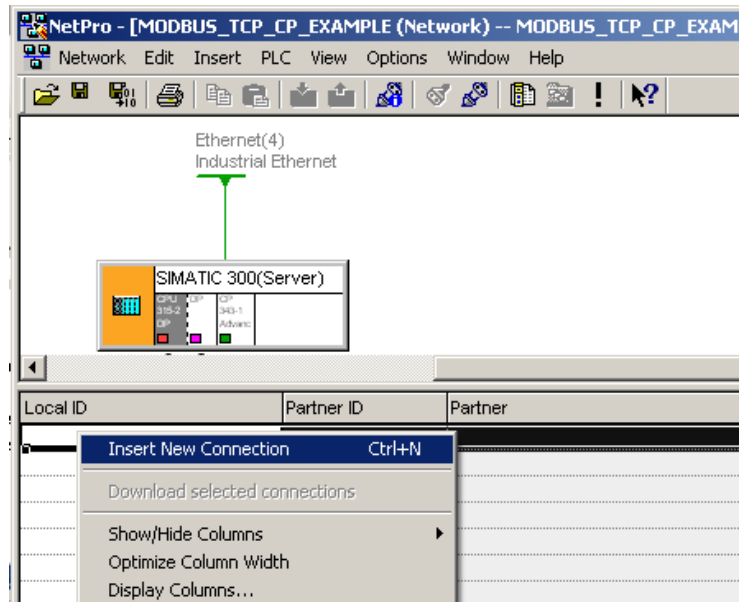
Selection of the Port Number

In a MODBUS communication a MODBUS server are normally addressed via port 502, whereas a MODBUS client uses a port from 2000.

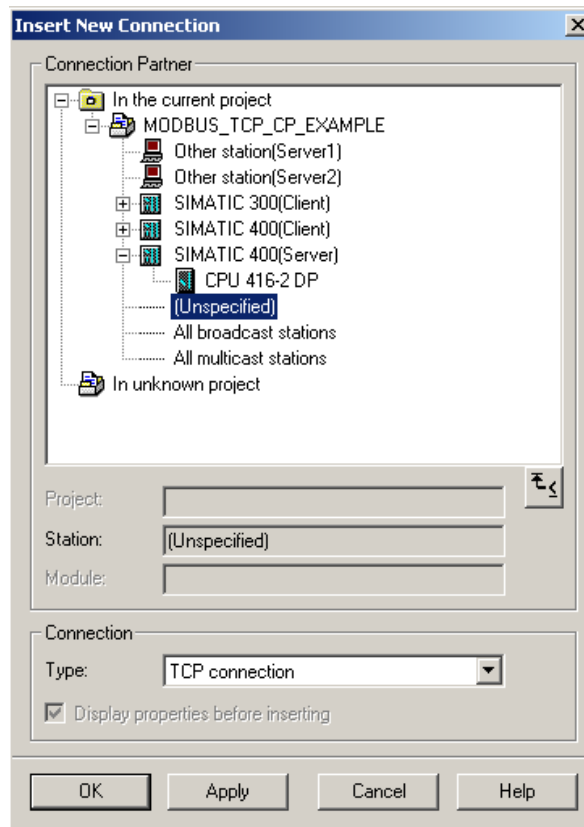
Unspecified Connection with "CP is server"

If you have got a communication with CP as MODBUS server, the communication is set up as "unspecified connection". The client must carry out active connection establishment.

After selecting **Insert > New Connection...** the dialog box "Insert new connection" will come up.

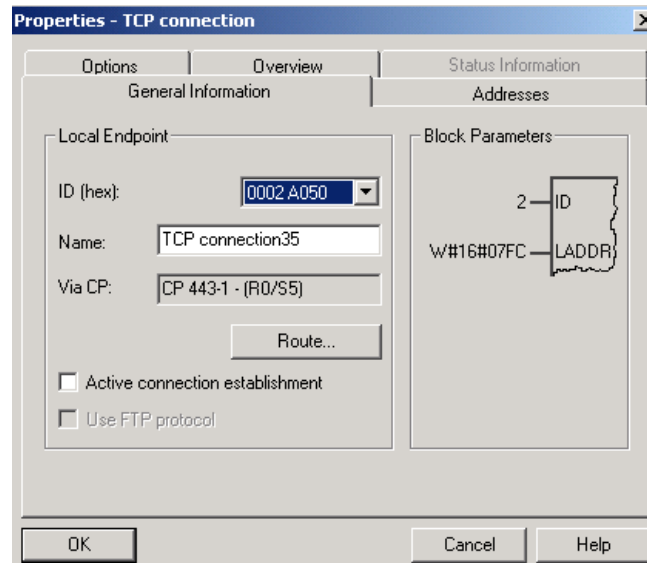


Select here "**unspecified**" instead of the communication partner and use "TCP Connection" for the connection. Put a check mark on "Show properties dialog"

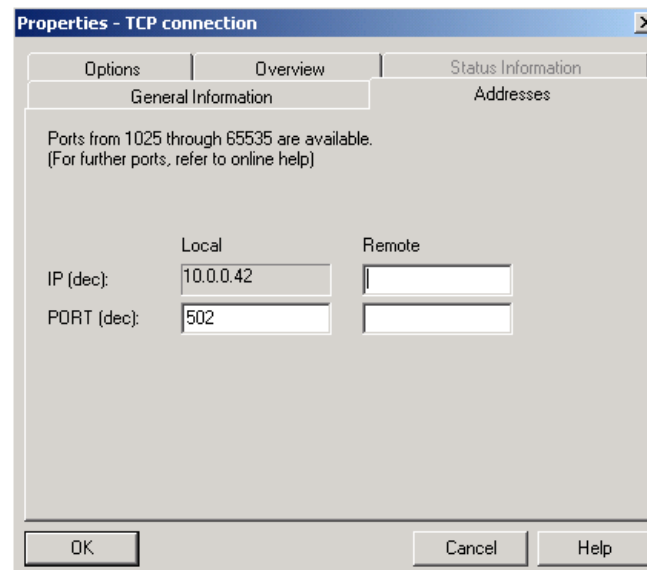


Click "OK". This will take you back to the dialog box "Properties TCP connection".

The check box "active connection establishment" must not be activated.



In the register "addresses" all information regarding the partner, "IP" and "PORT" are left blank.



Click on "OK" and the inputs are accepted.

3.4 How to Select the Appropriate Modbus Block

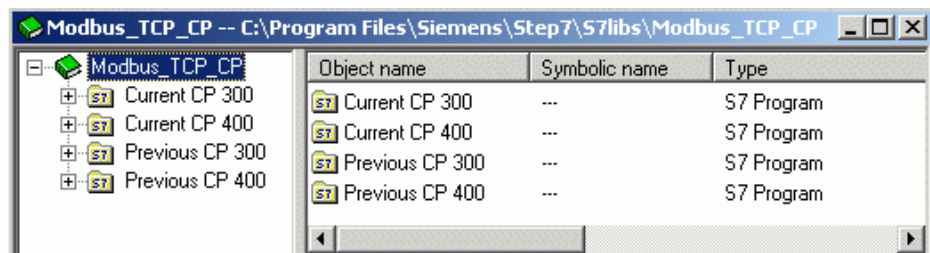
General Information

The block FC10 AG_CNTRL of the SIMATIC_NET library permits to terminate and reestablish an established connection. This block was also implemented in the Modbus blocks for a more effective use of the resources of CPU and CP. However, previous CPs or previous firmware releases do not support the use of AG_CNTRL.

Therefore the library contains directories for both alternatives, each for S7-300 CPs and S7-400 CPs:

- Current CP 300 CPs of S7-300 which support AG_CNTRL (**not for CPU315 and 317 HW release V1**)
- Current CP 400 CPs of S7-400 which support AG_CNTRL
- Previous CP 300 CPs of S7-300 which **do not** support AG_CNTRL
- Previous CP 400 CPs of S7-400 which **do not** support AG_CNTRL

Pay attention to the count of local data of the used CPU! The required local data for both alternatives are written in chapter 5.4.



Which CPs support AG_CNTRL?

Here you can find up-to-date information which CPs and which firmware releases support AG_CNTRL:
[Ethernet CPs and AG_CNTRL](#)

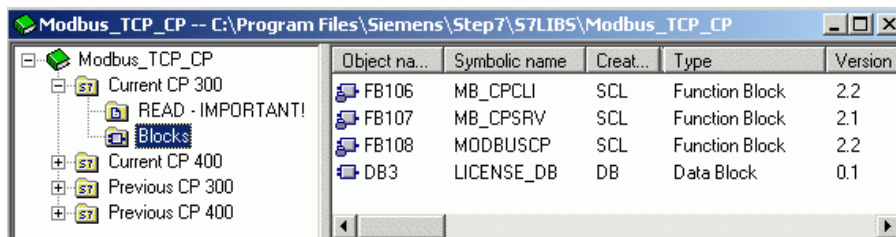
If the used CP is not mentioned here, the Modbus blocks without AG_CNTRL must be used.

3.5 Insertion of the Function Blocks into the Program

Content of MODBUS library

The following blocks are required for the Modbus communication:

- MODBUSCP
- MB_CPCLI
- MB_CPSRV



Insertion of the MODBUS FBs

In order to exchange data with MODBUS devices, you need the function blocks MODBUSCP, MB_CPCLI, MB_CPSRV and AG_CNTRL if applicable. To be able to insert this into your project you have to copy the function block from the library. For that, open the library “**Modbus_TCP_CP**” by selecting the menu item **File > Open**.

In the dialog box “**Open Project**” select the tab “**Libraries**”. Select the library “**Modbus_TCP_CP**” using the Browse-Button and click “**OK**”.

This opens the library. Open the folder “**CP 300**” or “**CP 400**” according to your Configuration by double clicking it. Select the function blocks and copy it via “**Edit > Copy**”. Then change back to your project .

In the Step7 project, in your opened S7 300/S7 400-Station, select the CPU. Double click “**S7-program**”, and then “**Blocks**”.

This will open the “**Blocks**” folder.

Select “**Edit > Paste**”. This will insert the block into your program.

**Insertion of the
Communication
Blocks**

The Modbus function blocks use the function blocks AG_SEND and AG_RECV in S7-300 or AG_LSEND and AG_LRECV in S7-400. You can find these communication function blocks in the library **"SIMATIC_NET_CP"** which are included in the software package **NCM S7 for Industrial Ethernet**.

CP 300

Copy the functions FC5 (AG_SEND) and FC6 (AG_RECV) from the folder **"CP 300"** and insert in your project.

CP 400

Copy the functions FC50 (AG_LSEND) and FC60 (AG_LRECV) from the folder **"CP 400"** and insert in your project.

Please note, that the following versions of the FCs are a prerequisite for the faultless function of the FB MODBUSCP:

S7-400:	AG_LSEND	V3.1 or higher
	AG_LRECV	V3.1 or higher
S7-300:	AG_SEND	V4.2 or higher
	AG_RECV	V4.7 or higher

3.6 Multiple Connections via Port 502

General	<p>Some CP modules are able to multiplex TCP connections. In doing so several Modbus clients can establish a connection via port 502. The CP acts as Modbus server.</p> <p>Here you can find information which CP and which firmware release support multiple connections via port 502: www.siemens.com/s7modbus</p>
Pre-requisites	<p>To use the functionality, the parameterization must be carried out as follows:</p> <ul style="list-style-type: none"> • CP is server • port 502 as local port • unspecified TCP connection in NETPRO • passive connection establishment <hr/> <p>Please note that only 1 connection is parameterized in NETPRO irrespective of the number of clients which address the CP as server.</p> <hr/>
Number of Connections	<p>The CP is capable to keep up to 8 connections to different clients. A ninth attempt for connection establishment will be rejected.</p>
Displaying the Status of the Connection	<p>The status of the connection is displayed even in NETPRO online as well as in special diagnosis of the CP.</p> <p>As only 1 connection is parameterized in NETPRO, the display shows the status of all TCP connections to the several clients.</p> <p>As long as no client has established a connection "Passive connection establishment in progress" is displayed. As soon as one client has established a connection "established" is displayed. It is not possible to check how many clients have actually established a connection to the CP.</p>
Characteristics of the Error Handling	<p>The FB MODBUSPN and the CP respectively must terminate and reestablish the connection in certain error situations. This action is carried out by the FB MODBUSPN. Thereby all existing connections via port 502 are terminated.</p>
Tip for your User Program	<p>When several connections are established via port 502, it is not possible to identify the client which has sent the recent request. If the clients use different UNIT numbers, the verification of the same in the user program allows a determination.</p>

3.7 Start_up Characteristics of CP343 / CP443

Introduction	The start up of the CP is divided into the following phases: <ul style="list-style-type: none">• Initialization (Power on of the CP)• Parameterization
Initialization	As soon as the CP is connected to power, the hardware self test runs. The firmware of the CP is set up for operation.
Parameterization	During parameterization the CP receives the device parameters that are assigned to its slot. The CP is now ready for operation.

3.8 Changeover of a Previous Version to V4.x

General Information for Changeover to V4.0

If a previous version of "ModbusTCP CP" is already used, some changes must be carried out in the S7 program. An update with the menu "file" -> "check and update access" is not possible because the interface of the block has changed.

Procedure for Changeover to V4.0

1. Copy the new Modbus blocks FB106, FB107, FB108 and DB3 into your project. You can rename the modbus blocks, if these block numbers are already used.
2. Open the blocks, in which the modbus blocks have been called so far (OB100 and cyclical OB).
3. Call the new block FB108 MODBUSCP directly after the prior MODBUS or MODB4 call with a new instance DB.
4. Transcribe the following parameters of MODBUS or MODB4 to MODBUSCP depending on the calling OB.
id, laddr, MONITOR, db_0 to db_8, ENQ_ENR, server_client, START_ADDRESS, LENGTH, WRITE_READ, INPUT_HOLDING, TI, UNIT, DONE_NDR, ERROR and STATUS.

Capital letters represent runtime parameters, small letters initialisation parameters.

5. Add suitable values to the new parameters:
REG_KEY, single_write, data_type_x, LICENSED, BUSY and STATUS_FUNC. It is not necessary to allocate the parameter IDENT_CODE.
You can find the description of the new parameters in the parameter list in section 5.2.
6. The output parameters are **displayed dynamically**, i.e. they are only available for **one PLC cycle**. They have to be copied to an additional memory area if you need to process them or to display the values in a VAT (STEP7 variable table).
Adapt your program correspondingly.
7. Insert the programming fault block OB121 in your program if this has not been included yet. This block is necessary for the licensing, see section "licensing" for further information.
8. Delete the previous Modbus call or comment it out. Load the program into the CPU and execute a restart of the CPU.

Notes to V4.3

Here you find information to V4.3: www.siemens.com/s7modbus, tab "Downloads".

4 Licensing


General

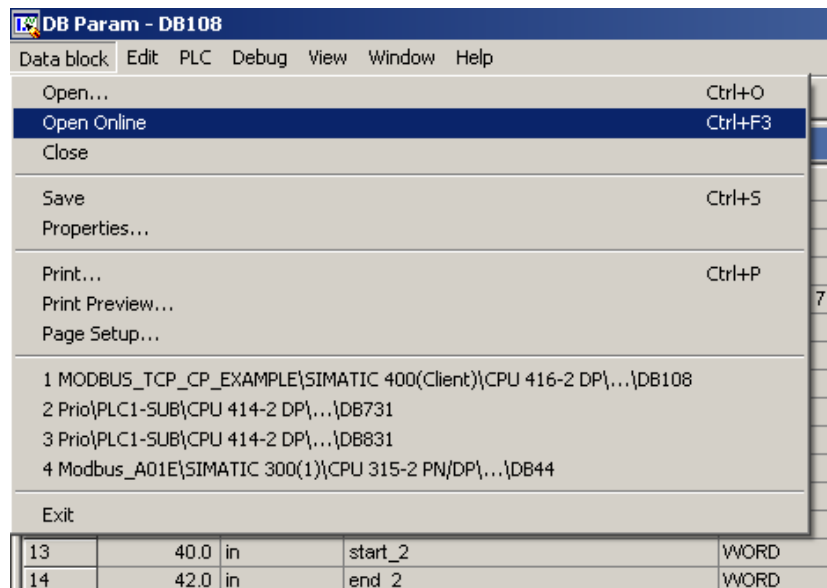
The block MODBUSCP must be licensed for each CPU individually. The licensing takes place in two steps: reading the IDENT_CODE and declaring the registration key REG_KEY. The OB121 must be available in the CPU.

Read the IDENT_CODE

To read the IDENT_CODE please proceed as follows:

1. Parameterize the block MODBUSCP in the cyclic OB (OB1 or cyclic interrupt OB) and in OB100 according to your requirements. Transfer the program to the PLC and turn it to RUN mode.
2. Open the instance DB of the block MODBUSCP. „Data block“ -> „Open Online“ to open the DB.

Monitoring the block via the button  is insufficient.



3. The output IDENT_CODE shows an 18 character string

Copy this string per copy/paste from the DB und and insert it in the form **IT4Industry SOFTWARE REGISTRATION FORM**. This form is stored in the library path

..\\Program Files\\Siemens\\Step7\\S7LIBS\\Modbus_TCP_CP during installation and is also available on the installation CD. Insert the License-No. of the product package into the form.

Address	Declar	Name	Type	Initial value	Actual value
34	82.0	in end_7	WORD	W#16#0	W#16#8FC
35	84.0	in data_type_8	BYTE	B#16#0	B#16#0
36	86.0	in db_8	WORD	W#16#0	W#16#0
37	88.0	in start_8	WORD	W#16#0	W#16#0
38	90.0	in end_8	WORD	W#16#0	W#16#0
39	92.0	in ENQ_ENR	BOOL	FALSE	FALSE
40	94.0	out LICENSED	BOOL	FALSE	TRUE
41	94.1	out BUSY	BOOL	FALSE	FALSE
42	94.2	out DONE_NDR	BOOL	FALSE	FALSE
43	94.3	out ERROR	BOOL	FALSE	FALSE
44	96.0	out STATUS	WORD	W#16#0	W#16#0
45	98.0	out STATUS_FUNC	STRING [8]	"	"
46	108.0	out IDENT_CODE	STRING [18]	"	FHJIFDBMHIDEFHDD2
47	128.0	in_out UNIT	BYTE	B#16#0	B#16#0
48	129.0	in_out DATA_TYPE	BYTE	B#16#0	B#16#0
49	130.0	in_out START_ADDRE...	WORD	W#16#0	W#16#0
50	132.0	in_out LENGTH	WORD	W#16#0	W#16#0
51	134.0	in_out TI	WORD	W#16#0	W#16#0
52	136.0	in_out WRITE_READ	BOOL	FALSE	FALSE

Please insert the IDENT-CODE here.
The manual contains information how to find out the IDENT-CODE.

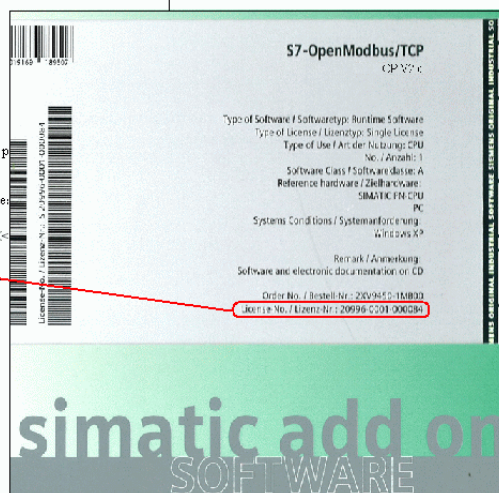
Bitte tragen Sie den IDENT-CODE hier ein.
Das Handbuch enthält Informationen, wie Sie den IDENT-CODE ermitteln.

>>> IDENT_CODE <<<

Please insert the License-No. here.
You find the License-No. on the package of the p...

Bitte tragen Sie die Lizenz-Nr. hier ein.
Sie finden die Lizenz-Nr. auf der Verpackung de...

>>> License-No / Lizenz-Nr <<<



4. Please send the form per e-mail to IT4Industry it4.industry@siemens.com.

Hereupon you will receive the registration key for your PLC.

Please note: No ident code can be displayed as long as an initialisation error exists. Evaluate the output STATUS and correct the parameterization.

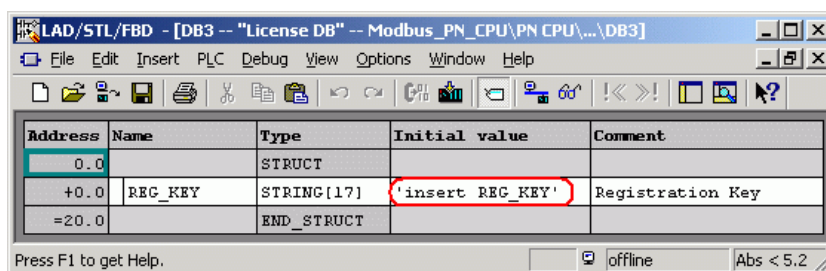
Declaration of the Registration Key REG_KEY

The registration key REG_KEY must be declared for each call of the block MODBUSCP.

The registration key REG_KEY should be stored in a global DB. Via this global DB all MODBUSCP blocks can receive the registration key (See also the following example).

Please proceed as follows to declare the registration key REG_KEY:

1. Copy the prepared license block DB3 of the library "Modbus_TCP_CP" into your project. If the DB number is already used in your project, rename the license DB.
2. Open the license DB and copy the 17 digit registration key you received from IT4industry to the column "Initial value".



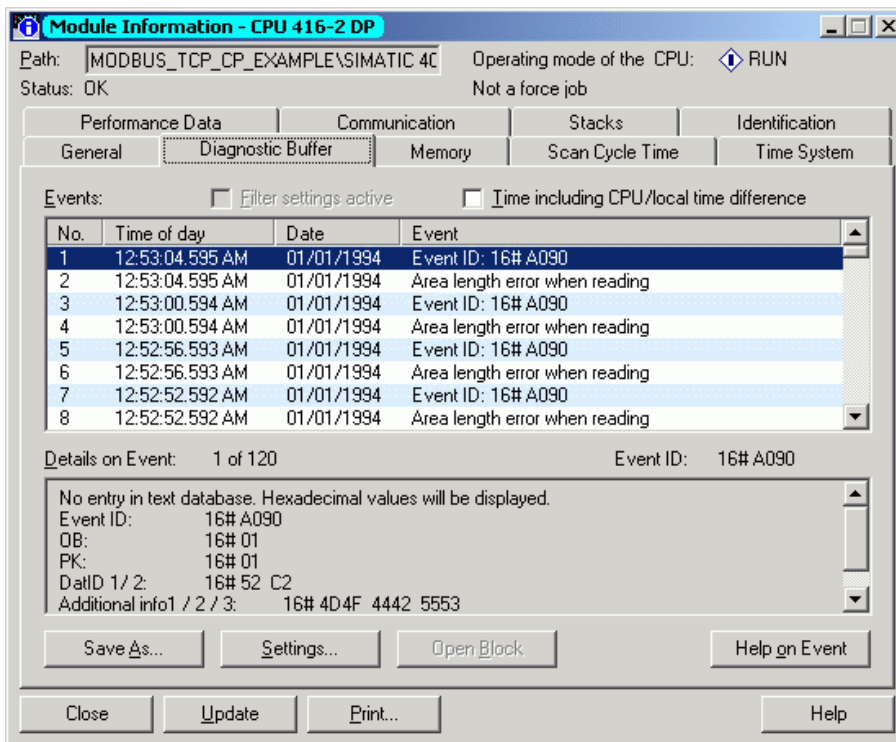
Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	REG_KEY	STRING[17]	'insert REG_KEY'	Registration Key
=20.0		END_STRUCT		

3. Declare the registration key in the data block as "initial value" to avoid a repeated insertion after reloading the PLC. Open the data block in the SIMATIC manager with the editor in the declaration view. Change over to the data view via the menu "View" -> "Data View". Choose in the menu "Edit > Initialize Data Block" – all values of the column "initial value" are copied to "actual values".
4. Assign the value "DB3.REG_KEY" to the parameter REG_KEY of the block MODBUSCP.
5. Transfer the changed blocks to the PLC. The registration key can be set at run time. A STOP -> RUN transition is not necessary.

The block is now licensed for this CPU.

Missing or Wrong Licensing

When the registration key is missing or a wrong one is detected, the SF LED (S7-300) or the INTF LED (S7-400) of the CPU is flashing. A cyclic error message regarding the missing license is displayed in the diagnostic buffer too. The error number of the missing license is W#16#A090.



Warning

The CPU will turn to STOP mode, if the OB121 is not available.

Modbus communication is carried out even with a missing or wrong registration key, but the output STATUS is set to W#16#A090 “no valid license”.

5 Function Block MODBUSCP

5.1 Functionality of the FB

General Information

The function block MODBUSCP allows a communication between a CP443-1 and CP 343-1 and a partner, which supports Open MODBUS/TCP protocol. The function codes 1, 2, 3, 4, 5, 6, 15 and 16 are supported.

Depending on the parameterization the FB can be operated both in client and in server mode. It is also possible to operate a CP as a client and as a server at the same time. In this case two NetPro connections and 2 FB calls are necessary. In the operating mode server the functionality multitasking, according to the MODBUS reference, is not implemented.

There is no limitation of the maximum number of parallel called modbus blocks on the part of the library. Though it depends on the CPU and CP how many AG functions can run simultaneously. The maximum number of AG calls can be taken from the manual of the CPU: "Technical Data" > "Communication". In the manual of the CP it's detailed, how many AG_SEND/AG_RECV or AG_LSEND/AG_LRECV can be processed by this CP simultaneously.

The block MODBUSCP calls the blocks MB_CPCLI (FB106) and MB_CPSRV (FB107) internally. The block MB_CPCLI comprises the function of Modbus client, the block MB_CPSRV executes the function of Modbus server.

The function block performs the following functions:

- Calls the standard functions for the data transfer between the CPU and the CP
- Generates MODBUS specific telegram header before send
- Verification of the MODBUS specific telegram header after receive
- Verification if the memory areas exist which are requested by the client
- Generate exception telegrams when failures occur (only when CP is in server mode)

Exception code	Meaning
1	The requested function code is not supported.
2	An access to a not existing or not permitted address was carried out.
3	An invalid length was indicated for this function code.

- Data transfer to and from the parameterized DB
- Monitoring the data reception with a time-out
- Verification of the registration key

Online-Help

In Simatic Manager an online help for function block MODBUSCP is provided. Mark the FB and press the key "F1". The online help is displayed; it contains the main information regarding the FB.

Call of the FB

The function block MODBUSCP has to be called in two OBs:

- in the start up OB100 and
- in a cyclic OB (OB1 or cyclic interrupt OB, e.g. OB35)

In doing so the same instance DB has to be used.

The subordinate blocks of the Modbus library MB_CPCLI and MB_CPSRV must not be called additionally in an organization block.

A coexistent call of FB MODBUSCP in OB1 and a cyclic interrupt OB, e.g. OB35 is not permissible.

The **OB121 must be available** in the CPU. You will find more information regarding this matter in **section "Licensing"**.

Start Up of the FB

The function block MODBUSCP should unconditionally be called once in OB100.

- The initialization parameters must be set according to the station configuration.
- The initialization parameters will be evaluated and copied into the instance DB.
- The runtime parameters are not evaluated during start up.

Cyclical Operation of the FB

In cyclical operation the FB MODBUSCP is called in OB1 or a cyclic interrupt OB.

- According to the runtime parameters, the functions of the function block are activated.
- While a request is running changes to the runtime parameters are ignored.
- In the cyclical operation initialization parameters are ignored.

**OB121
"Programming
Error"**

If the block has not been licensed yet, the OB121 is called.

**Warning**

The CPU will turn to STOP mode, if the OB121 is not available.

**Initiate Request
CP is Client**

A rising edge at the trigger input ENQ_ENR initiates a request. Depending on the input parameters UNIT, DATA_TYPE, START_ADDRESS LENGTH and WRITE_READ a MODBUS request telegram is generated and sent to the partner station via the TCP/IP connection. The client waits for the parameterized monitoring time MONITOR for a response from the server. When the monitoring time elapses (no response from the server) the activated request is terminated with an error. A new request can be initiated again.

After the receipt of the response telegram a validity check is carried out. If the result is positive, necessary actions will be taken and the request will be terminated without error. The output DONE_NDR is set. When an error is recognized during verification, the request is terminated with an error, the ERROR bit is set and an error number is returned in STATUS.

**Activation of the
Function Block
CP is Server**

With TRUE signal at the trigger input ENQ_ENR the FB is ready for receiving a request telegram from the client. The server remains passive.

The received telegram is verified. If the telegram is verified positive, the response telegram is sent. The completed transmission is reported to the user by setting the DONE_NDR bit.

At this point the completed function is indicated at the outputs UNIT, DATA_TYPE, START_ADDRESS, LENGTH and WRITE_READ.

A faulty request telegram causes an error message and the ERROR bit is set. The error number is returned in STATUS. The request of the client is not answered.

**Data Transfer
CPU – CP**

The data transfer between CP and CPU are done with the standard function blocks AG_SEND and AG_RECV respectively AG_LSEND and AG_LRECV.

At the activation of a MODBUS request by the user (CP is client) or at the receipt of a telegram from the client (CP is server) the standard blocks necessary for the CP are called by the FB in the right order and number.

At the receipt of a telegram, the first 6 Bytes are read with the function AG_RECV. This header contains the length of the rest of the telegram. A second call of the AG_RECV function follows with the rest of the telegram length. The verification of the received user data takes place after the complete receipt of the data.

**TCP/IP with
CP343-1 / CP443-1**

TCP/IP with CP 343-1 and CP443-1 uses static connections. The TCP connection cannot be disconnected while in run mode.

Given this system characteristic, telegrams might be lost under unfavorable conditions when the synchronization had been lost after an error.

**Termination of the
TCP connection by
the Communi-
cation Partner**

If the CP acts as server and the communication partner terminates the TCP connection, the next receive can be started not until an idle time of 1 second has elapsed.

If the CP acts as Client, the next send can be started not until an idle time of 150 milliseconds has elapsed. This delay, due to system characteristics, is executed by the function block.

5.2 Parameters of the Function Block MODBUSCP

Parameter	Decl.	Type	Description	Value range	Init
id	IN	WORD	Connection-ID as per configuration in NetPro	1 to 64	yes
laddr	IN	WORD	Input-Address of the CP in HW Config	CPU dependent	yes
MONITOR	IN	TIME	Monitoring Time: Wait for data from communication partner Shortest adjustable time is 20 ms	T#20ms to T#+24d20h31m23s647ms	no
REG_KEY	IN	STRING [17]	Registration key to activate the license	Character	no
server_client	IN	BOOL	CP/FB operates in server mode or client mode (server = slave; client = master)	TRUE FALSE	yes
single_write	IN	BOOL	Write 1 Coil/Register: Function code 5 and 6 are used respectively Function code 15 and 16 are used respectively	TRUE FALSE	yes
data_type_1	IN	BYTE	1st data area: data type Coils Inputs Holding Register Input Register	1 2 3 4	yes
db_1	IN	WORD	1st data area: data block number	1 to 65535 W#16#1 to W#16#FFFF	yes
start_1	IN	WORD	1st data area: first Modbus address in this DB	0 to 65535 W#16#0000 to W#16#FFFF	yes
end_1	IN	WORD	1st data area: last Modbus address in this DB	0 to 65535 W#16#0000 to W#16#FFFF	yes
data_type_2	IN	BYTE	2nd data area: data type (Coils, Inputs, Holding Register, Input Register), NULL if not used	0 to 4	yes
db_2	IN	WORD	2nd data area: data block number	1 to 65535 W#16#1 to W#16#FFFF	yes
start_2	IN	WORD	2nd data area: first Modbus address in this DB	0 to 65535 W#16#0000 to W#16#FFFF	yes
end_2	IN	WORD	2nd data area: last Modbus address in this DB	0 to 65535 W#16#0000 to W#16#FFFF	yes
data_type_3	IN	BYTE	3rd data area: data type (Coils, Inputs, Holding Register, Input Register), NULL if not used	0 to 4	yes
db_3	IN	WORD	3rd data area: data block number	1 to 65535 W#16#1 to W#16#FFFF	yes

Parameter	Decl.	Type	Description	Value range	Init
start_3	IN	WORD	3rd data area: first Modbus address in this DB	0 to 65535 W#16#0000 to W#16#FFFF	yes
end_3	IN	WORD	3rd data area: last Modbus address in this DB	0 to 65535 W#16#0000 to W#16#FFFF	yes
data_type_4	IN	BYTE	4th data area: data type (Coils, Inputs, Holding Register, Input Register), NULL if not used	0 to 4	yes
db_4	IN	WORD	4th data area: data block number	1 to 65535 W#16#1 to W#16#FFFF	yes
start_4	IN	WORD	4th data area: first Modbus address in this DB	0 to 65535 W#16#0000 to W#16#FFFF	yes
end_4	IN	WORD	4th data area: last Modbus address in this DB	0 to 65535 W#16#0000 to W#16#FFFF	yes
data_type_5	IN	BYTE	5th data area: data type (Coils, Inputs, Holding Register, Input Register), NULL if not used	0 to 4	yes
db_5	IN	WORD	5th data area: data block number	1 to 65535 W#16#1 to W#16#FFFF	yes
start_5	IN	WORD	5th data area: first Modbus address in this DB	0 to 65535 W#16#0000 to W#16#FFFF	yes
end_5	IN	WORD	5th data area: last Modbus address in this DB	0 to 65535 W#16#0000 to W#16#FFFF	yes
data_type_6	IN	BYTE	6th data area: data type (Coils, Inputs, Holding Register, Input Register), NULL if not used	0 to 4	yes
db_6	IN	WORD	6th data area: data block number	1 to 65535 W#16#1 to W#16#FFFF	yes
start_6	IN	WORD	6th data area: first Modbus address in this DB	0 to 65535 W#16#0000 to W#16#FFFF	yes
end_6	IN	WORD	6th data area: last Modbus address in this DB	0 to 65535 W#16#0000 to W#16#FFFF	yes
data_type_7	IN	BYTE	7th data area: data type (Coils, Inputs, Holding Register, Input Register), NULL if not used	0 to 4	yes
db_7	IN	WORD	7th data area: data block number	1 to 65535 W#16#1 to W#16#FFFF	yes
start_7	IN	WORD	7th data area: first Modbus address in this DB	0 to 65535 W#16#0000 to W#16#FFFF	yes

Parameter	Decl.	Type	Description	Value range	Init
end_7	IN	WORD	7th data area: last Modbus address in this DB	0 to 65535 W#16#0000 to W#16#FFFF	yes
data_type_8	IN	BYTE	8th data area: data type (Coils, Inputs, Holding Register, Input Register), NULL if not used	0 to 4	yes
db_8	IN	WORD	8th data area: data block number	1 to 65535 W#16#1 to W#16#FFFF	yes
start_8	IN	WORD	8th data area: first Modbus address in this DB	0 to 65535 W#16#0000 to W#16#FFFF	yes
end_8	IN	WORD	8th data area: last Modbus address in this DB	0 to 65535 W#16#0000 to W#16#FFFF	yes
ENQ_ENR	IN	BOOL	CP is client: Initiate request at TRUE signal CP is server: Ready to receive at TRUE signal	TRUE FALSE	no
LICENSED	OUT	BOOL	License state of the function block: Block is licensed Block is not licensed	TRUE FALSE	no
BUSY	OUT	BOOL	Operating state of the functions AG_SEND and AG_RECV Job processing No job processing active	TRUE FALSE	no
DONE_NDR	OUT	BOOL	CP is client: Active request finished without errors CP is server: Request from the client was executed and answered	TRUE/FALSE	no
ERROR	OUT	BOOL	An error has occurred No error has occurred	TRUE FALSE	no
STATUS	OUT	WORD	Error number	0 to FFFF	no
STATUS_FUNC	OUT	STRING[8]	Name of the function, which causes the error at STATUS	Character	no
IDENT_CODE	OUT	STRING[18]	Identification for licensing Please order your license with this identification string.	Character	no
UNIT	IN/ OUT	BYTE	Unit identification (INPUT if in CLIENT mode, OUTPUT if in SERVER mode)	0 to 255	no
DATA_TYPE	IN/ OUT	BYTE	Data type to be accessed: (INPUT if in CLIENT mode, OUTPUT if in SERVER mode) Coils Inputs Holding registers Input registers	1 2 3 4	no
START_ADDRESS	IN/ OUT	WORD	MODBUS start address (INPUT if in CLIENT mode, OUTPUT if in SERVER mode)	0 to 65535	no

Parameter	Decl.	Type	Description	Value range	Init
LENGTH	IN/ OUT	WORD	Number of registers to be processed (INPUT if in CLIENT mode, OUTPUT if in SERVER mode) <u>Coils</u> : reading function writing function <u>Inputs</u> : reading function <u>Holding Register</u> : reading function writing function <u>Input Register</u> : reading function	1 to 2000 1 to 1968 1 to 2000 1 to 125 1 to 123 1 to 125	no
WRITE_ READ	IN/ OUT	BOOL	INPUT if in CLIENT mode, OUTPUT if in SERVER mode: Write access Read access	TRUE FALSE	no

General Information

The parameters of the FB MODBUSCP can be divided into two groups:

- Initialization parameters
- Runtime parameters

Initialization parameters are evaluated only during the initial execution of the function block MODBUSCP and taken over into the instance DB. They are marked in the above table in the column "INIT" with "yes".

A modification of the initialization parameters during run mode has no impact. After a modification of these parameters (e. g. during the test phase), the instance DB must be initialized again via a STOP → RUN transition of the CPU.

Runtime parameters can be modified during the cyclical operation. In the mode "CP is client", it is not advisable to modify the input parameters while a request is active. Wait with the next request and the change of the parameters until the previous request ends with DONE_NDR or ERROR.

In the operation mode "CP is server", the output parameters may only be evaluated when DONE_NDR is TRUE.

The output parameters are **displayed dynamically**, i.e. they are only available for **one PLC cycle**. They have to be copied to an additional memory area if you need to process them or to display the values in a VAT (STEP7 variable table).

Range of Values

For the range of values of the different parameters, CPU specific restrictions must be taken into consideration.

- id** For each configured connection in STEP7 (NetPro) a connection ID is assigned. The connection ID identifies unambiguously the connection from the CPU via the CP to the communication partner. The number from the connection configuration has to be entered here. The range of values for this parameter is 1 to 64.
- laddr** The parameter LADDR is the base address of the CP from HW Config (input address). The configured value has to be entered here. The range of values for this parameter depends on the CPU. The parameters ID and LADDR can also be taken from the dialog box **“Properties of TCP Connection”**.
- MONITOR** The monitoring time MONITOR observes the data input from the communication partner. The monitoring time can be set in the format T#.... A monitoring time of approximately 1,5 second is recommended.
- In the operating mode **“CP is client”** MONITOR specifies the timeout for the receipt of the complete response telegram from the server. When the monitoring time elapses, the active request is cancelled with an error. The timer is started after the completed send of the request telegram and is stopped after the receipt of the complete data.
- In the operating mode **“CP is server”** the receipt of the second part of the telegram is monitored with the MONITOR time. When the time elapses an error is reported. The timer is started after the receipt of the MODBUS specific telegram header and is stopped after the receipt of the complete request telegram.
- REG_KEY** The block MODBUSCP must be licensed for each CPU individually to permit a correct program sequence. With the registration key REG_KEY the block MODBUSCP is licensed and the Modbus communication runs without any restraint. You can find further information in **section “Licensing”**.
- server_client** This parameter differentiates the client from the server mode. If the input is TRUE, then the operating mode is **“CP is server”**. If the input is FALSE, then the operating mode is **“CP is Client”**.
- single_write** In operating mode **“CP is client”** and single_write = TRUE write requests with length 1 are carried out with the function codes 5 and 6. With single_write = FALSE all write requests use the function codes 15 and 16.
- data_type_x** The parameter data_type_x defines the MODBUS data type which is to be mapped with this definition. If data_type_x set to 0, this data area is not used.

Identifier	Data type	Size
0	Area not used	
1	Coils	Bit
2	Inputs	Bit
3	Holding Register	Word
4	Input Register	Word

db_x The parameter db_x defines the number of the data block in which the consecutively defined MODBUS registers or bits are mapped. 0 cannot be used as DB number since it is reserved for system functions.

db_x
DB-Number 1 - 65535 (W#16#0001 to W#16#FFFF)

start_x Start_x specifies the first register or bit address, which is stored in the data element 0 of the DB. End_x defines the last MODBUS address.
end_x

When accessing registers, the number of the data element of the S7 DB in which the last register is mapped, can be calculated with the following formula:

$$\text{DBW number} = (\text{end_x} - \text{start_x}) * 2$$

When accessing coils or inputs, the number of the data element of the S7 DB in which the last bit is mapped, can be calculated with the following formula:

$$\text{DBB number} = (\text{end_x} - \text{start_x} + 7) / 8$$

The defined memory areas must not overlap. The parameter end_x must not be smaller than start_x. In case of an error occurring, the initialization of the FB is stopped with an error. When start_x is equal to end_x, one Modbus address (1 register or 1 bit) is allocated.

In section 5.3 you can find an example of the mapping of the MODBUS addresses to S7 memory areas.

ENQ_ENR Operating mode “**CP is client**“:
The data transfer is initiated with a TRUE signal at the input. The request telegram is generated with the values of the input parameters UNIT, DATA_TYPE, START_ADDRESS, LENGTH and WRITE_READ. A new request may only be initiated, when the previous one is ended with DONE or ERROR.

Operating mode “**CP is server**“:
The FB is activated with a TRUE signal at the input. Telegrams from the client can be received. With a FALSE signal at the input data is received from the CP and discarded.

LICENSED If this output is TRUE, then the Modbus block is licensed on this CPU. If the output is FALSE, none or a faulty license string was typed in. You can find further information in **section “Licensing”**.

BUSY If this output is TRUE, one of the functions AG_SEND or AG_RECV is running.

DONE_NDR	<p>In the operating mode “CP is client“ the activated request was executed without error. For a read function the response data from the server has already been entered into the DB. For a write function the response to the request telegram has been received from the server.</p> <p>In the operating mode “CP is server“ this output indicates a telegram exchange without errors. In the parameters UNIT, DATA_TYPE, START_ADDRESS, LENGTH and WRITE_READ the request parameters of the client are displayed. These outputs are only available and valid as long as DONE_NDR is TRUE.</p>
ERROR	<p>When this output is set, an error was recognized.</p> <p>In the operating mode “CP is client“ the activated request was ended with an error. The error number is displayed in the STATUS output.</p> <p>In the operating mode “CP is server“ an error is detected at a request telegram of the client or at sending a response telegram. The error number is displayed in the STATUS output.</p>
STATUS	<p>The STATUS output displays the error number when ERROR is TRUE. The error numbers are described in chapter 6.</p>
STATUS_FUNC	<p>This parameter shows the name of the function, which caused the error occurred.</p>
IDENT_CODE	<p>After start-up of the PLC this parameter displays the identification code, an 18 character string. With this IDENT_CODE you can order the registration key at IT4industry.</p> <p>You can find further information in section “Licensing”.</p>
UNIT	<p>In mode “CP is client“ the parameter UNIT is an input parameter. This input has to be set according to the requirements. The FB copies this value to the request telegram and verifies when the respond telegram is received.</p> <p>In mode “CP is server“ the parameter UNIT is an output parameter. The FB copies this value from the request telegram to the respond telegram. The output is set with the received value when the job is finished without error.</p>

DATA_TYPE

The parameter DATA_TYPE defines which Modbus data type is to be accessed with the current job. The following data types are available:

Coils	B#16#1
Inputs	B#16#2
Holding Register	B#16#3
Input Register	B#16#4

In the operating mode “**CP is client**”, DATA_TYPE is an input parameter. In the operating mode “**CP is server**”, DATA_TYPE is an output parameter.

The different data types are related directly to the used function codes.

Data type	DATA_TYPE	Function	Length	single_write	Function code
Coils	1	read	any	irrelevant	1
Coils	1	write	1	TRUE	5
Coils	1	write	1	FALSE	15
Coils	1	write	>1	irrelevant	15
Inputs	2	read	any	irrelevant	2
Holding Register	3	read	any	irrelevant	3
Holding Register	3	write	1	TRUE	6
Holding Register	3	write	1	FALSE	16
Holding Register	3	write	>1	irrelevant	16
Input Register	4	read	any	irrelevant	4

START_ADDRESS

The parameter START_ADDRESS specifies the first MODBUS address that is read or written.

In the operating mode “**CP is client**” it is an input parameter, in the operating mode “**CP is server**” it is an output parameter.

LENGTH

The parameter LENGTH specifies the number of MODBUS registers or bits that are read or written.

In the operating mode “**CP is client**” it is an input parameter. In the operating mode “**CP is server**” it is an output parameter.

For read functions, a maximum of 125 registers or 2000 bits is possible per telegram. For write functions a maximum of 123 registers or 1968 bits is possible.

All registers or bits have to be in the same DB per telegram.

WRITE_READ

This parameter defines if a read or write function should be carried out. If the value of the input/output is FALSE, it specifies the read mode. The value TRUE specifies the write mode.

Only Holding Registers and Coils can be written. Input Registers and Inputs can be read merely.

In the operating mode “**CP is client**“ it is an input parameter. In the operating mode “**CP is server**“ it is an output parameter.

5.3 Address Mapping

Interpretation of Modbus Addresses

MODBUS bases its data model on a series of tables, which have distinguishing characteristics. The distinction between these memory areas is done via the register address by some systems, e.g. MODICON PLCs. So a MODBUS message requesting the read of a holding register at offset 0 would return the value known to the application programmer as found in register 40001 (memory type 4xxxx, reference 0001).

One potential source of confusion is the varying interpretation of the register address in different manuals. Sometimes the register address means the address of the application layer, sometimes the address transferred.

The FB MODBUS uses the **Modbus address transferred** at its parameters start_x und START_ADDRESS. So it is possible to use register addresses from von 0000_H to FFFF_H with each function code.

**Example:
Parameterization
of the Memory
Areas**

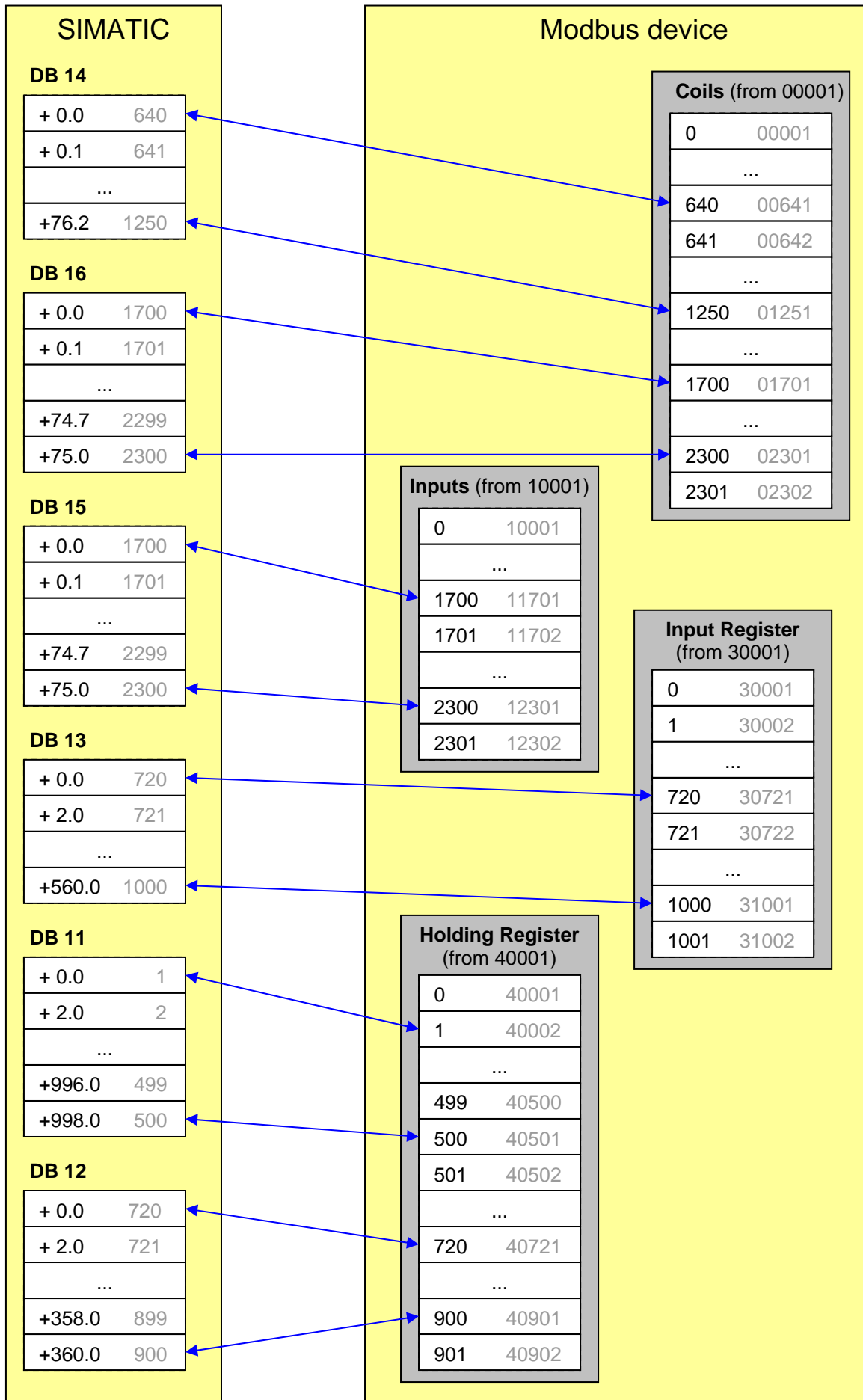
data_type_1 db_1 start_1 end_1	B#16#3 W#16#B W#16#1 W#16#1F4	Holding Register DB 11 Start address: 1 End address: 500
data_type_2 db_2 start_2 end_2	B#16#3 W#16#C W#16#2D0 W#16#384	Holding Register DB 12 Start address: 720 End address: 900
data_type_3 db_3 start_3 end_3	B#16#4 W#16#D W#16#2D0 W#16#3E8	Input Register DB 13 Start address: 720 End address: 1000
data_type_4 db_4 start_4 end_4	B#16#0 0 0 0	Not used 0 0 0
data_type_5 db_5 start_5 end_5	B#16#1 W#16#E W#16#280 W#16#4E2	Coils DB 14 Start address: 640 End address: 1250
data_type_6 db_6 start_6 end_6	B#16#2 W#16#F W#16#6A4 W#16#8FC	Inputs DB 15 Start address:1700 End address: 2300
data_type_7 db_7 start_7 end_7	B#16#1 W#16#10 W#16#6A4 W#16#8FC	Coils DB 16 Start address: 1700 End address: 2300
data_type_8 db_8 start_8 end_8	B#16#0 0 0 0	Not used 0 0 0

Address Mapping

The following diagram shows the comparison of the SIMATIC memory area and the Modbus typical register oriented storage arrangement. The diagram is based on the above parameterization.

The Modbus addresses printed in black refer to the Data Link Layer, the ones printed in grey refer to the Application Layer.

The SIMATIC addresses printed in black are the offset in the DB. You can find additionally the Modbus register numbers printed in grey.



5.4 Data and Standard Functions used by the FB

Instance DB

The function block MODBUSCP stores its data in an instance DB. This instance DB is created by STEP7 at the first call of the FB.

The instance data block contains parameters of type Input, Output, Input/Output as well as static variables that it needs for its execution. These variables are non-volatile and keep its validity between FB calls. The variables control the internal process flow of the FB.

Required memory of the instance-DBs:

Instance-DB	work memory	load memory
MODBUSCP	ca. 1kByte	ca. 2kByte

Local Variables

The FB requires 80 Bytes of local variables. Additionally the subordinate FBs require local variables: FBs MB_CPCLI (30 byte) or MB_CPSRV (30 byte). V1.x: AG_SEND/AG_RECV or AG_LSEND/AG_LRECV require, up to 58 bytes, depending on the used function block. That gives a maximum of 168 bytes of local data for a FB MODBUSCP V1.x-call.

V2.x: The function AG_CNTRL requires 178 bytes local data. That gives a maximum of 288 bytes of local data for a FB MODBUSCP V2.x-call.

Timers

The function block does not use any timer.

Flags

The function block does not use any flag.

Standard FCs for Data Transfer

The function blocks MB_CPCLI and MB_CPSRV respectively called by FB MODBUSCP use the blocks AG_SEND/AG_RECV (S7-300) and AG_LSEND/AG_LRECV (S7-400) from the SIMATIC_NET library for the data transfer between CPU and CP.

The blocks MODCLI and MODSRV version 2.x use the block AG_CNTRL additionally to reset and reestablish the connection in case of error.

The following versions of the FCs are tested with FB MODBUSCP and released for the communication:

S7-300:	FC5 "AG_SEND" version 4.2 FC6 "AG_RECV" version 4.7 FC10 "AG_CNTRL" version 1.0
S7-400	FC50 "AG_LSEND" version 3.1 FC60 "AG_LRECV" version 3.1 FC10 "AG_CNTRL" version 1.0

**MODBUSCP:
SFCs for
Miscellaneous
Functions**

The FB MODBUSCP uses the following SFBs/SFCs from the standard library:

- SFB4 „TON“
- SFC6 „RD_SINFO“
- SFC20 „BLKMOV“
- SFC51 „RDSYST“
- SFC52 „WR_USMSG“

**MB_CPCLI and
MB_CPSRV:
SFCs for
Miscellaneous
Functions**

The FBs MB_CPCLI and MB_CPSRV use the following SFCs from the standard library:

- SFC20 „BLKMOV“
- SFC24 „TEST_DB“
- SFB4 „TON“

5.5 Renaming of Standard Functions

Inducement

Whether you have already used the numbers of standard functions in your project or the block number is reserved for a different application, you may rename the internal called functions FC5/FC50 or FC6/FC60, FC10 or the blocks MODBUSCP, MB_CPCLI and MB_CPSRV.

It is not possible to rename the system functions SFC6, SFC20, SFC24, SFC51 and SFC52 as well as the system function block SFB4.

Behavior

A set of rules concerning the function block numbering have to be considered when rewiring function blocks in SIMATIC STEP 7 Manager.

If you want to rewire the blocks of the Modbus library, the following sequence is required:

1. FC50 AG_LSEND
FC60 AG_LRECV
2. FB106 MB_CPCLI
FB107 MB_CPSRV
3. FB108 MODBUSCP

It is not necessary to rewire all blocks. Even if you want to rewire only some of the blocks, you must follow the mentioned sequence.

Renaming

To rename the blocks proceed as described subsequently:

1. Get the information about the used operand by clicking "Extras > Reference data > Display".
2. Set the operand priority in the object properties of the block folder to "Absolute value".
3. Call the function "Extras > Re-wire" in the SIMATIC Manager, in order to re-wire the operands into free areas.
4. To be able to keep on using the symbolism diagnostics tools, add the modifications in the symbolism table supplementary.

The modifications can be verified by clicking "Extras > Reference data > Display".

6 Diagnostics

Diagnostic Function	The diagnostic functions of the CP 343/ CP 443 allow you fast failure localization. The following diagnostic features are available:
	<ul style="list-style-type: none"> • Diagnostics via the display elements of the CP • Diagnostics via the STATUS output of the function block MODBUSCP.
Display Elements (LED)	The display elements inform you about the operating mode or about the failure conditions of the CP. The display elements give you an overview of internal failures, external failures and interface specific failures.
STATUS Output of the FB MODBUSCP	For error diagnostics, the function block MODBUSCP has got a STATUS output. By reading the STATUS output you get a general indication of failures that have occurred during the communication. The STATUS parameter can be evaluated in the user program.
	The output STATUS_FUNC shows the name of the function, which has caused the error.

6.1 Diagnostics via the Display Elements of the CP

Display Functions	The display elements of the CP give you information on the module status. There are two types of display functions:
	<ul style="list-style-type: none"> • Group Error Displays <ul style="list-style-type: none"> - INTF Internal failure - EXTf External failure
	<ul style="list-style-type: none"> • Special Displays <ul style="list-style-type: none"> CP 343-1: <ul style="list-style-type: none"> - RX/TX A telegram is being transmitted via the interface. CP 443-1: <ul style="list-style-type: none"> - TXD A telegram is being sent via the interface. - RXD A telegram is being received via the interface.
	A detailed description of the display elements can be found in the device manual of the CP.

6.2 Diagnostic Messages of the FB MODBUSCP

Messages at the STATUS Output of the FB

The block MODBUSCP has got 2 status outputs: STATUS and STATUS_FUNC. STATUS displays all error numbers. STATUS_FUNC shows the name of the function, which caused the error. STATUS is valid when ERROR is TRUE. Below you can find a list of FB-specific error messages.

Error Messages of the called SFCs and FCs

The FB MODBUSCP, MB_CPCLI and MB_CPSRV use the standard functions SFC6, SFC20, SFC24, SFC51, SFC52, FC5 and FC6, or FC50 and FC60. The error messages of these blocks are passed on to STATUS without any changes.

In the diagnostics buffer or in the online help of SIMATIC Manager you will find further details on these error messages, as well as in the SIMATIC STEP7 NCM S7 Industrial Ethernet Manual.

Error messages of FB MODBUSCP		
STATUS (Hex)	Event text	Remedy
A002	The parameter end_x is less than start_x.	Correct the parameterization of start_x and end_x
A003	The DB, to which MODBUS addresses shall be mapped, is too short. Minimum length: registers: $(end_x - start_x + 1) * 2$ bits: $(end_x - start_x) / 8 + 1$ Other possible reasons: <ul style="list-style-type: none"> Wrong initialization parameter (CP is client). Wrong address area in the request telegram of the client (CP is server). The CP sends an exception telegram. 	Extend the DB. CP is Client: Correct the parameters START_ADDRESS or LENGTH CP is Server: Modify the request of the client.
A004	Applies only with CP is client: An invalid combination of DATA_TYPE and WRITE_READ is given.	Correct the parameters. Only data type 1 or 3 can be written.
A005	CP is client: An invalid value for the parameter LENGTH is given. CP is Server: The number of registers/bits in the request telegram is invalid. The CP sends an exception telegram. <u>Range of values:</u> Read coils/inputs: 1 to 2000 Write coils: 1 to 1968 Read registers: 1 to 125 Write holding registers: 1 to 123	CP is Client: Correct the parameter LENGTH. CP is Server: Modify the number of registers/bits in the request telegram.

Error messages FB MODBUSCP		
STATUS (Hex)	Event text	Remedy
A006	The given range of data defined with DATA_TYPE, START_ADDRESS and LENGTH does not exist in data_type_1 to data_type_8. CP is server: The CP sends an exception telegram.	CP is Client: Correct the parameter's combination DATA_TYPE, START_ADDRESS and LENGTH. CP is Server: Modify the request of the client or correct the parameterization of data_type_x.
A007	CP is client: An invalid monitoring time MONITOR is parameterized. A value > 20ms is required.	Correct the parameterization.
A008	Monitoring time MONITOR elapsed when AG_RECV waits for receipt. E.g. Partner is not ready. <u>MODBUSCP V2.x:</u> All connections via Port 502 are terminated and reestablished. <u>MODBUSCP V1.x:</u> As an after effect the loss of synchronization can occur, which leads to a loss of telegrams.	Verify error messages at the communication partner. Check if the communication partner needs a special unit identifier.
A009	CP is client: The received transaction identifier TI is not equal to the sent one. <u>MODBUSCP V2.x:</u> All connections via Port 502 are terminated and reestablished.	Verify the data of the communication partner with the help of a telegram trace.
A00A	CP is client: The received UNIT is not equal to the sent one. <u>MODBUSCP V2.x:</u> All connections via Port 502 are terminated and reestablished.	
A00B	CP is client: Received function code is not equal to the sent one. CP is server: An invalid function code was received. The CP sends an exception telegram. <u>MODBUSCP V2.x:</u> All connections via Port 502 are terminated and reestablished. <u>MODBUSCP V1.x:</u> As an after effect the loss of synchronization can occur, which leads to a loss of telegrams.	CP is client: Verify the data of the communication partner with the help of a telegram trace. CP is server: The FB MODBUSCP supports the function codes 1, 2, 3, 4, 5, 6, 15 and 16.
A00C	The received byte count does not match the number of registers/bits. CP is server: The CP sends an exception telegram. <u>MODBUSCP V2.x:</u> All connections via Port 502 are terminated and reestablished.	Verify the data of the communication partner with the help of a telegram trace.
A00D	Only when CP is client: The register/bit address or the number of registers/bits in the response telegram is not equal to the one in the request telegram.	

Error messages the FB MODBUSCP		
STATUS (Hex)	Event text	Remedy
A00E	The length indicated in the MODBUS specific telegram header does not match the number of registers/bits or the byte count in the telegram. The FB receives all data and ignores them. <u>MODBUSCP V2.x:</u> All connections via Port 502 are terminated and reestablished. <u>MODBUSCP V1.x:</u> As an after effect a loss of synchronization might occur.	Verify the data of the communication partner with the help of a telegram trace.
A00F	A protocol identifier $\neq 0$ was received. <u>MODBUSCP V2.x:</u> All connections via Port 502 are terminated and reestablished. <u>MODBUSCP V1.x:</u> As an after effect a loss of synchronization might occur.	
A010	In the parameterized area db_1 to db_8 a DB number is used twice.	Correct the parameterization of db_x.
A011	An invalid value for DATA_TYPE is given (Value range: 1 to 4).	Correct the parameters.
A012	The parameterized areas data_type_1 and data_type_2 overlap.	Correct the parameterization. The data areas must not contain any overlapping register areas.
A013	The parameterized areas data_type_1 and data_type_3 overlap.	
A014	The parameterized areas data_type_1 and data_type_4 overlap.	
A015	The parameterized areas data_type_1 and data_type_5 overlap.	
A016	The parameterized areas data_type_1 and data_type_6 overlap.	
A017	The parameterized areas data_type_1 and data_type_7 overlap.	
A018	The parameterized areas data_type_1 and data_type_8 overlap.	
A019	0 is assigned to one of the parameters db_x while the according data_type_x is $\neq 0$. DB 0 can't be used; it is reserved for system functions.	
A01A	Wrong length in header: Range of values: 3 to 253 bytes. <u>MODBUSCP V2.x:</u> All connections via Port 502 are terminated and reestablished.	Verify the data of the communication partner with the help of a telegram trace.
A01B	CP is server and function code 5: An invalid value for coils was received. The CP sends an exception telegram.	

Error messages of FB MODBUSCP		
STATUS (Hex)	Event text	Remedy
A01E	The CP has received invalid data which could not be assigned. <u>MODBUSCP V2.x:</u> All connections via Port 502 are terminated and reestablished. <u>MODBUSCP V1.x</u> The CP has lost synchronization and needs data from the communication partner to finish the active AG_RECV.	Check the error message of the communication partner and verify the data with a telegram trace if needed.
A01F	The FB MODBUSCP has turned to an invalid state.	Please contact the product support.
A023	The parameterized areas data_type_2 and data_type_3 overlap.	Correct the parameterization. The data areas must not contain any overlapping register areas.
A024	The parameterized areas data_type_2 and data_type_4 overlap.	
A025	The parameterized areas data_type_2 and data_type_5 overlap.	
A026	The parameterized areas data_type_2 and data_type_6 overlap.	
A027	The parameterized areas data_type_2 and data_type_7 overlap.	
A028	The parameterized areas data_type_2 and data_type_8 overlap.	
A034	The parameterized areas data_type_3 and data_type_4 overlap.	
A035	The parameterized areas data_type_3 and data_type_5 overlap.	
A036	The parameterized areas data_type_3 and data_type_6 overlap.	
A037	The parameterized areas data_type_3 and data_type_7 overlap.	
A038	The parameterized areas data_type_3 and data_type_8 overlap.	
A045	The parameterized areas data_type_4 and data_type_5 overlap.	
A046	The parameterized areas data_type_4 and data_type_6 overlap.	
A047	The parameterized areas data_type_4 and data_type_7 overlap.	
A048	The parameterized areas data_type_4 and data_type_8 overlap.	
A056	The parameterized areas data_type_5 and data_type_6 overlap.	
A057	The parameterized areas data_type_5 and data_type_7 overlap.	
A058	The parameterized areas data_type_5 and data_type_8 overlap.	

Error messages of FB MODBUSCP		
STATUS (Hex)	Event text	Remedy
A067	The parameterized areas data_type_6 and data_type_7 overlap.	Correct the parameterization. The data areas must not contain any overlapping register areas.
A068	The parameterized areas data_type_6 and data_type_8 overlap.	
A078	The parameterized areas data_type_7 and data_type_8 overlap.	
A07A	An invalid value id is parameterized. Range of values is 1 to 64.	Correct the parameterization of the input id.
A07C	An invalid value data_type_x was given. The value range is 0 to 4.	Correct the parameterization of data_type_x.
A07D	Parameter data_type_1 is not defined. The parameter area _1 is the default area and must be defined.	Correct the parameterization of data_type_1.
A07E	The DB number of db_x is identical to the number of the instance DB.	Correct the parameterization of db_x.
A080	Different instance DBs were used for the call of MODBUSCP in OB100 and the cyclic OB.	MODBUSCP must be called with the identical instance DB in OB100 and the cyclic OB.
A081	Only if CP is client and function code 5: The received coil status is not equal to the sent one.	Verify the data of the communication partner with the help of a telegram trace.
A082	Only if CP is client and function code 6: The received register value is not equal to the sent one.	Verify the data of the communication partner with the help of a telegram trace.
A083	Only if CP is client: A request was initiated prior to the completion of the previous one. The request is not executed.	Wait with the initiation of a new request until the previous one was finished either with DONE/NDR = TRUE or ERROR = TRUE.
A085	An error occurred during the license handling due to an invalid write access.	Verify the project if there is any invalid write access to the license DB. The structure of REG_KEY must not be changed. Please contact the Product Support if necessary.
A090	The block MODBUSCP is not licensed for this CPU. This is a status information. The bit ERROR is not set. The Modbus communication runs without a license as well.	Read the identification string IDENT_CODE for this CPU and order the registration key at IT4industry. See also section "Licensing".

Error messages of FB MODBUSCP		
STATUS (Hex)	Event text	Remedy
A091	An exception telegram with exception code 1 was received (only if CP is client)	The communication partner does not support the requested function.
A092	An exception telegram with exception code 2 was received (only if CP is client) An attempt to an invalid or non existing address at the communication partner was made.	Correct LENGTH or START_ADDRESS at the call of the FB.
A093	An exception telegram with exception code 3 was received (only if CP is client)	The communication partner can't process the received request (e.g. doesn't support the requested length).
A094	An exception telegram with exception code 4 was received (only if CP is client)	The communication partner is in a state, in which the received request can't be processed.
A095	An exception telegram with an unknown exception code was received (only if CP is client).	Check the error message of the communication partner and verify the data with a telegram trace if needed.

6.3 Diagnostic Messages of FC5/FC6 and FC50/FC60

Diagnostic messages of the included blocks AG_SEND/AG_RECV (FC5/FC6) and AG_LSEND/AG_LRECV (FC50/FC60) at the output STATUS		
STATUS (Hex)	Event text	Remedy
7xxx	For detailed information please refer to the online help of SIMATIC Manager.	See online help (SIMATIC manager -> mark block -> key F1 -> Ethernet -> see also -> code evaluation)
8xxx	For detailed information please refer to the online help of SIMATIC Manager.	See online help (SIMATIC manager -> mark block -> key F1 -> Ethernet -> see also -> code evaluation)

6.4 Diagnostic Messages of included FCs/SFCs

Diagnostic messages of the included blocks RD_SINFO (SFC6), BLKMOV (SFC20) , SFC51 (RDSYSST), SFC52 (WR_USMSG) and AG_CNTRL at the output STATUS		
STATUS (Hex)	Event text	Remedy
7xxx	For detailed information please refer to the online help of SIMATIC Manager.	See online help (SIMATIC manager -> mark block -> key F1)
8xxx	For detailed information please refer to the online help of SIMATIC Manager.	See online help (SIMATIC manager -> mark block -> key F1)

6.5 Diagnostic Messages of SFC24

Error messages of SFC24		
STATUS (Hex)	Event text	Remedy
80A1	DB Number = 0 or too large for the CPU	Choose a valid DB number.
80B1	The DB does not exist in the CPU.	All data blocks that are specified in db_x must be created and copied into the CPU.
80B2	DB UNLINKED	DB must not be created as UNLINKED.

7 Application Samples

General Information

With the installation 2 sample projects are installed in \Program Files\Siemens\Step7\Examples.

- sample project “MB_TCP_CP” written in STL and
- sample project “MB_TCP_CP_CFC” written in CFC

The S7 programs are for information purposes only and are not to be understood as a solution for a customer specific installation configuration.

Programming Example

The programming examples consist of the blocks:

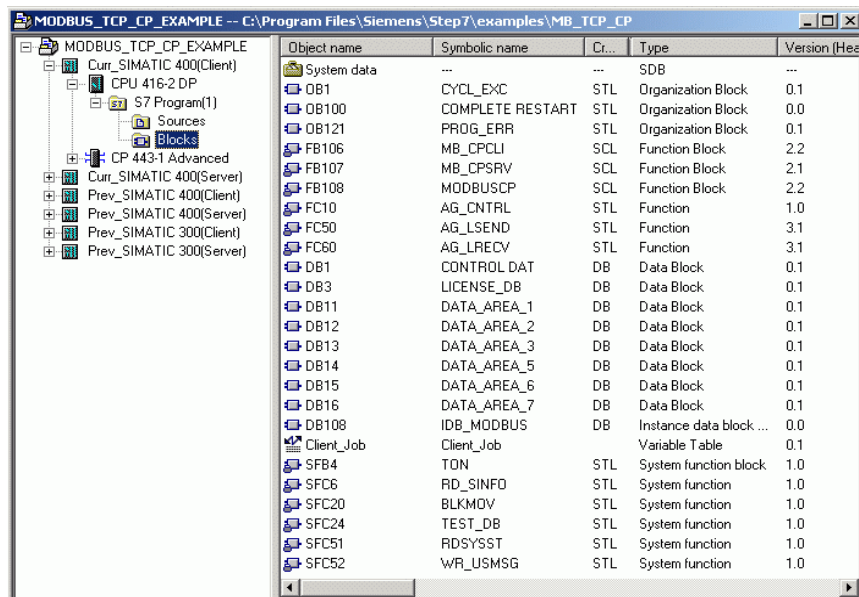
- Start-Up OB100
- Programming error OB121
- Cyclic program processing OB1 and OB35 respectively with call of FB 108
- Global DB for job trigger (e.g. via a variable table) and for licensing
- Data blocks for register values

7.1 Sample project STL

General Information

The sample project written in STL includes several Simatic stations for the different applications.

- Simatic Station is S7-300 or S7-400
- Simatic Station is client or server
- Simatic Station supports or doesn't support AG_CNTRL



Used Blocks

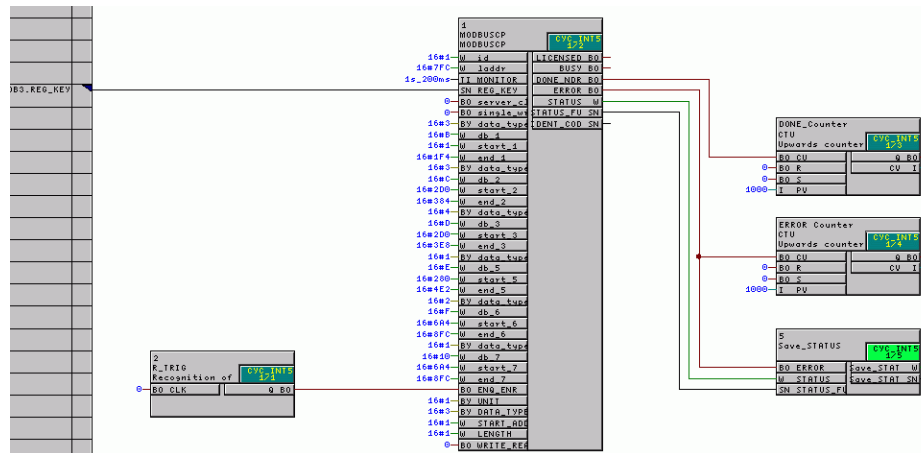
The subsequent listed blocks are used in the provided example project for S7 stations with FB MODBUSCP.

Block	Symbol	Comment
OB 1	CYCL_EXC	Cyclic program processing
OB 100	COMPLETE RESTART	Start-Up OB for Re-Start
OB 121	PROG_ERR	Programming error OB
FB 108	MODBUSCP	User block FB MODBUSCP
FB 106	MB_CPCLI	Internally called FB MB_CPCLI
FB 107	MB_CPSRV	Internally called FB MB_CPSRV
DB 1	CONTROL_DAT	Work-DB CONTROL DAT for FB MODBUSCP
DB 3	LICENSE_DB	License DB for FB MODBUSCP
DB 11	DATA_AREA_1	Value DB for area 1
DB 12	DATA_AREA_2	Value DB for area 2
DB 13	DATA_AREA_3	Value DB for area 3
DB 14	DATA_AREA_5	Value DB for area 5
DB 15	DATA_AREA_6	Value DB for area 6
DB 16	DATA_AREA_7	Value DB for area 7
DB 108	IDB_MODBUS	Instance DB for FB MODBUSCP

7.2 Sample project CFC

General Information

The sample project written in CFC includes two Simatic S7-400 stations, one is parameterized as client, the other one as server.



Used Blocks

The subsequent listed blocks are used in the provided example project with FB MODBUSCP. The system functions, system function blocks and the blocks which are generated by CFC are not listed in the following table.

Block	Symbol	Comment
OB 35	CYC_INT5	Cyclic program processing
OB 100	COMPLETE RESTART	Start-Up OB for Re-Start
OB 121	PROG_ERR	Programming error OB
FB 8	R_TRIG	Recognition of positive edge
FB 24	CTU	Upwards counter
FB 99	Save_STATUS	FB to save the error number and the name of the according function
FB 106	MB_CPCLI	Internally called FB MB_CPCLI
FB 107	MB_CPSRV	Internally called FB MB_CPSRV
FB 108	MODBUSCP	User block FB MODBUSCP
FC 500	AG_LSEND	FC to send data
FC 600	AG_LRECV	FC to receive data
DB1	MEMORY_DB	Help DB in server mode
DB 3	LICENSE_DB	License DB for FB MODBUSCP
DB 11	DATA_AREA_1	Value DB for area 1
DB 12	DATA_AREA_2	Value DB for area 2
DB 13	DATA_AREA_3	Value DB for area 3
DB 14	DATA_AREA_5	Value DB for area 5
DB 15	DATA_AREA_6	Value DB for area 6
DB 16	DATA_AREA_7	Value DB for area 7

A Literature

MODBUS IDA

MODBUS APPLICATION PROTOCOL SPECIFICATION
V1.1b, December 28, 2006

<http://www.modbus-IDA.org>

Customer Support

Siemens AG
Industry Sector
I IS IN OC
Werner-von-Siemens-Str. 60
91052 Erlangen
Tel: ++49 9131 7-46111
Fax: ++49 9131 7-44757
Mail: it4.industry@siemens.com

<http://www.siemens.com/s7modbus>

Siemens Aktiengesellschaft

Subject to change without prior notice.

Stand: 06/2011