Industrial-Automation

System *HIMatrix*

MODBUS Master/Slave

Manual

HIMA Paul Hildebrandt GmbH
Industrial Automation

Rev. 0.02

HI 800 003 HEA

Important Notes

All HIMA products mentioned in this manual are protected under the HIMA trademark. Unless not explicitly noted, this may apply for other referenced manufacturers and their respective products.

All technical statements and data in this manual have been written with great care and effective quality measures have been taken to ensure their validity; however this manual may contain flaws or typesetting errors.
For this reason, HIMA does not offer any warranties nor assume legal responsibility nor any liability for possible consequences of any errors in this manual. HIMA appreciates any correspondence noting potential errors.

Technical modifications reserved.

For more information see the documentation on CD-ROM and on our
web site www.hima.com .

More information can be requested from:

HIMA Paul Hildebrandt GmbH
Postfach 1261
68777 Brühl

Tel: +49(6202)709 0
Fax: +49(6202)709 107

e-mail: info@hima.com

Table of Contents Page

About this Manual

The objective of this manual is to familiarize the user with the menu functions and dialog boxes of ***ELOP II Factory's*** HIMA Modbus Master.
To set up HIMA Modbus Master, the user needs the programming tool ***ELOP II Factory***, which must be installed on a PC with Microsoft Windows NT© , Windows 2000© or Windows XP©.

The user should be familiar with the programming tool ***ELOP II Factory*** and the HIMA *HIMatri*x control devices. For self-study, HIMA recommends the manual ”First Steps ***ELOP II Factory***“ and the ***ELOP II Factory's*** Online Help. Additionally, HIMA offers client specific training.

This manual is organized in four parts:

- The first part explains the menu functions and the dialogs within ***ELOP II Factory*** used to configure the HIMA Modbus Master.
- The second part explains the menu functions and the dialogs within ***ELOP II Factory*** used to configure the HIMA Modbus Slave.
- The third part gives hints for diagnosis and troubleshooting.
- In the forth part, a configuration of the HIMA Modbus Master is described, which the customer can follow in step-by-step instructions.

For further information about Modbus, HIMA refer to the specifications:

- Modbus Application Protocol Specification
- Modbus over serial line specification and implementation guide V1.0

(See www.modbus.org)

We would like to wish You good speed in converting Your Modbus projects. For questions, please contact HIMA directly.

© HIMA Paul Hildebrandt GmbH
Postfach 1261
D - 68777 Brühl bei Mannheim

1 HIMA Modbus Master

Both, the serial interface (RS 485) and TCP/UDP (Ethernet) may be used to transmit data between the Modbus master and the Modbus slaves.
In addition the HIMA Modbus master can also work as a gateway (see chapter 1.10).

1.1 Equipment and System Requirements

HIMA ***ELOP II Factory***	From Version 5.2.0 on
HIMatrix control devices	F30, F35 und F60 from Hardware Revision:02 on
Operating system versions of *HIMatrix* control devices	- COM OS from Version 6.22 on - CPU OS from Version 4.50 on
HIMA Modbus Module	See data sheet of the respective *HIMatrix* control device in chapter „Connections for Field Bus Communication“.
License key	Each of the following Modbus Master functions must be activated by a License key. - Modbus Master (RS 485 communication) - Modbus Master TCP (Ethernet communication)

1.2 Modbus Master Properties

Modbus Master	Only one Modbus master per resource may be configured. One Modbus master may simultaneously - operate TCP/UDP slaves and - serial slaves on several serial buses, and - be used as gateway for another Modbus master.
Max. number of Modbus Slaves	One Modbus master can configure up to 247 slaves. - 122 Modbus slaves per serial interface[1]*) - 32 TCP/UDP slaves via TCP/IP connection - 32 TCP/UDP slaves via UDP/IP connection
Max. number of request telegram	Up to 988 request telegrams can be configured per Modbus Master.
Max. length of request telegram	Refer to the specifications “Modbus Application Protocol Specification” on www.modbus.org

Max. size of transmit data

8192 Bytes (up to COM OS V8.32)

16384 Bytes (From COM OS V8.32 on)

The actual size of the user data, which can be send must be calculated as follows:
The status byte of the master and the status byte of each slave must be subtracted from the max. size of the transmit data (8192 bytes or 16384 bytes).

Max. size of user data (transmit)

= Max. size of the transmit data in Byte - 1 byte – (1 byte * number of slaves)

Max. size of receive data

8192 Bytes (up to COM OS V8.32)

16384 Bytes (From COM OS V8.32 on)

The actual size of the user data, which can be received must be calculated as follows:

The two status bytes of the master and the status byte of each slave must be subtracted from the max. size of the receive data (8192 bytes or 16384 bytes).

Max. size of user data (receive)

= Max. size of the receive data in byte - 2 byte – (1 byte * number of slaves)

Display format of the Modbus data

The *HIMatrix* controller is using the Big Endian Format for the data.

Example 32 Bit data (e.g. DWORD, DINT):

32 Bit data (hex)	0x12345678			
Memory offset	0	1	2	3
Big Endian (*HIMatrix*)	12	34	56	78
Middle Endian (H51q)	56	78	12	34
Little Endian	78	56	34	12

[1] According to the standard, three repeaters are permissible. Maximum 122 bus stations are possible per serial interface of a master.
The Modbus master can serve maximum 32 TCP/IP and 32 UDP/IP slaves at the same time.

Note

In addition to the Modbus protocol, other protocols (e.g. Profibus-DP, TCP S/R,…) can operate at the same time on a *HIMatrix* control device.

In sum, 16284 bytes of data can be transmitted and 16284 bytes of data can be received per *HIMatrix* control device.

The transmit and receive data can be arbitrarily separated between the protocols.

The maximum size of the transmit and receive data per protocol is depended by the COM OS (see above).

1.3 Modbus Functions

Signals may be written or read in both directions using the Modbus functions realized by HIMA. It is possible to write or read individual signals (BOOL/WORD) or several signals in sequence.

Signals may be read from a slave's import or export area using the read functions.

Signals may only be written in a slave's import area using the write functions.

For further information about Modbus, HIMA refer to the specifications:

- Modbus Application Protocol Specification
- Modbus over serial line specification and implementation guide V1.0

www.modbus.org

Element	Code	Type	Description
READ COILS	01	BOOL	Reads several signals (BOOL) from the slave's import or export[1)] area.
READ DISCRETE INPUTS	02	BOOL	Reads several signals (BOOL) from the slave's export area.
READ HOLDING REGISTERS	03	WORD	Reads several signals of any type from the slave's import or export[1)] area.
READ INPUT REGISTERS	04	WORD	Reads several signals of any type from the slave's export area.
READ WRITE HOLDING REGISTERS	23	WORD	Writes and reads several signals of any type in and from the slave's import area.
WRITE MULTIPLE COILS	15	BOOL	Writes several signals (BOOL) in the slave's import area.
WRITE MULTIPLE REGISTERS	16	WORD	Writes several signals of any type in the slave's import area.
WRITE SINGLE COIL	05	BOOL	Writes one single signal (BOOL) in the slave's area.
WRITE SINGLE REGISTER	06	WORD	Writes one single signal (WORD) in the slave's import area.

Table 1: Description of Modbus Functions

[1)] Export range can be selected only in HIMA Slaves

Note: For the function codes (1 and 3), the user must configure the import/export area in the HIMA Modbus slave from which the master should read the signals (refer to Description of the Slave).

For the *HIMatrix* control devices (COM OS' versions V.3 to V.6), the following read areas are available for function codes 1 and 3:

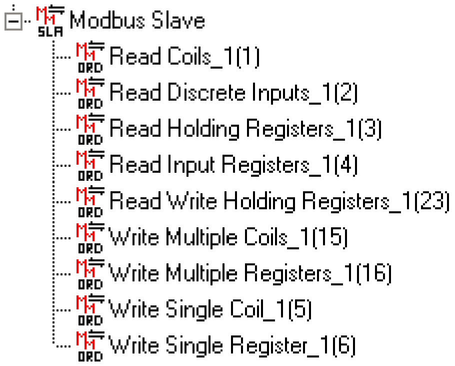
COM OS V.3:	Import area
COM OS V.4:	Export area (compatible with H51q)
COM OS V.6:	configurable

1.4 Overview of the Request Telegrams

In addition to the Modbus function, a request telegram of the Modbus master contains the start address of the read/write area.

Note	Before ELOP II Factory Version 6.42 the number of signals must be set in the „properties“ of the request telegrams.

The user may configure up to 988 request telegrams for each Modbus master.



Modbus Slave
- Read Coils_1(1)
- Read Discrete Inputs_1(2)
- Read Holding Registers_1(3)
- Read Input Registers_1(4)
- Read Write Holding Registers_1(23)
- Write Multiple Coils_1(15)
- Write Multiple Registers_1(16)
- Write Single Coil_1(5)
- Write Single Register_1(6)

Figure 1: The request telegrams of a HIMA Modbus Master

1.5 Read Request Telegrams

To read signals, the Modbus master sends a “Read Request Telegram” to the Modbus slave, which replies with an answer telegram containing the signals required.

The following “Read Request Telegrams” are available:

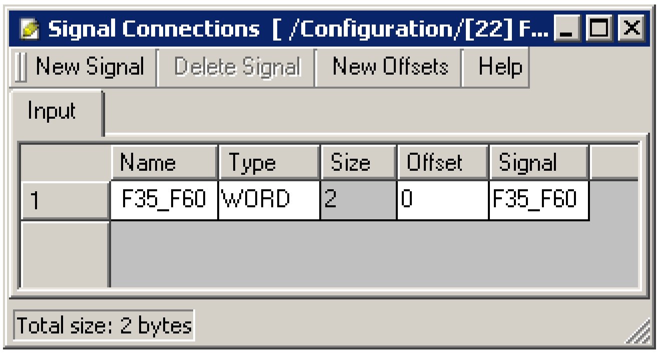
- Read Coils (01)
- Read Discrete Inputs (02)
- Read Holding Registers (03)
- Read Input Registers (04)

The context menu of a “Read Request Telegram” contains the following functions.

Read Request Telegram
Connect signals
Validate
New
Copy
Add
Delete
Print
Properties

1.5.1 Connect signals

In the dialog “Signal Connections” the signals are inserted, that the Modbus master requests from the Modbus slave.



	Name	Type	Size	Offset	Signal
1	F35_F60	WORD	2	0	F35_F60

Total size: 2 bytes

Figure 2: Dialog Box ”Signal Connections” of the “Read Request Telegram”

1.5.2 Validate

Prior to generating the code, the “Read Request Telegram” parameterisation may be tested. The user selects “Read Request Telegram” from the structure view and *Validate* from the context menu. The fault status field then displays faults and warnings, if any.

The validation function is always performed automatically prior to code generation. If a fault is detected during validation, code generation is aborted.

1.5.3 Copy

This function copies the “Read Request Telegram”, including its signals, to the clipboard.

1.5.4 Delete

This function deletes the selected “Read Request Telegram” from the project.

Important	”Delete“ deletes the “Read Request Telegram”. Make sure that the “Read Request Telegram” really has to be deleted.

1.5.5 Properties

Properties of the “Read Request Telegrams” (see 1.5.5.1 up to 1.5.5.4)

1.5.5.1 Request Telegram ”Read Coils (01)“

Element	Description
Type	Modbus function “Read Coils (01)”
Name	Any unique name for the Modbus function
Description	Description for the Modbus function
Start address of read area	0 to 65535 For *HIMatrix* Modbus Slaves 0 up to 8191 (up to COM-OS V8.32) 0 up to 16383 (from COM-OS V8.32 on)

Table 2: Request Telegram ”Read Coils (01)“

1.5.5.2 Request Telegram ”Read Discrete Inputs (02)“

Element	Description
Type	Modbus function “Read Discrete Inputs (02)”
Name	Any unique name for the Modbus function
Description	Description for the Modbus function
Start address of read area	0 to 65535 For *HIMatrix* Modbus Slaves 0 up to 8191 (up to COM-OS V8.32) 0 up to 16383 (from COM-OS V8.32 on)

Table 3: Request Telegram ”Read Discrete Inputs (02)“

1.5.5.3 Request Telegram ”Read Holding Registers (03)“

Element	Description
Type	Modbus function “Read Holding Registers (03)”
Name	Any unique name for the Modbus function
Description	Description for the Modbus function
Start address of read area	0 to 65535 For *HIMatrix* Modbus Slaves 0 up to 8190 (up to COM-OS V8.32) 0 up to 16382 (from COM-OS V8.32 on)

Table 4: Request Telegram ”Read Holding Registers (03)“

1.5.5.4 Request Telegram ”Read Input Registers (04)”

Element	Description
Type	Modbus function “Read Input Registers (04)”
Name	Any unique name for the Modbus function
Description	Description for the Modbus function
Start address of read area	0 to 65535 For *HIMatrix* Modbus Slaves 0 up to 8190 (up to COM-OS V8.32) 0 up to 16382 (from COM-OS V8.32 on)

Table 5: Request Telegram ”Read Input Registers (04)“

1.6 Read/Write Request Telegrams

For reading and writing the Modbus master sends a “Read/Write Request Telegram” to the Modbus slave.
Before this the Modbus master writes the signals into the Modbus slave’s defined import area.
After this, the Modbus master reads the defined read signals from the Modbus slave’s defined import area.

Note: In the “Read/Write Request Telegram”, the functions Write and Read are independent of one another. They are just sent together in a common request telegram.

However, the “Read/Write Request Telegram” is often used in a way that the written signals of the Modbus master are read back. This action verifies that the transferred signals were written correctly.

The following “Read/Write Request Telegram” is available:
Read Write Holding Registers (23)

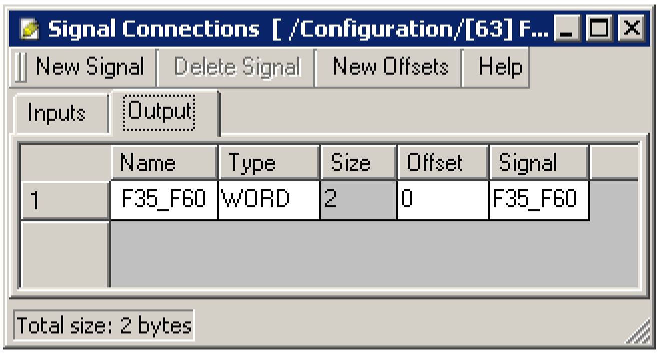
The context menu of a “Read/Write Request Telegram” contains the following functions.

Read/Write Request Telegram
Connect signals
Validate
New
Copy
Add
Delete
Print
Properties

1.6.1 Connect signals

In the dialog “Signal Connections”

- the signals transferred by the Modbus master to the Modbus slave, are entered into the tab ”Output“.

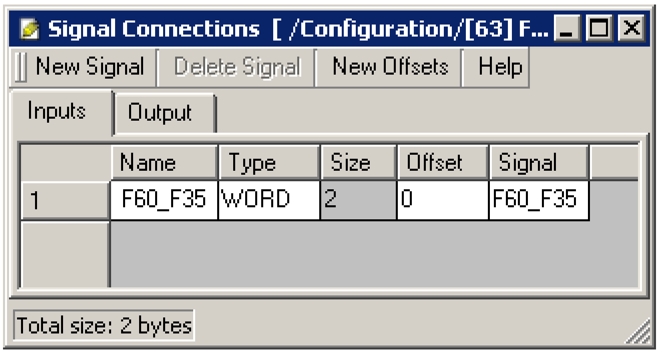


	Name	Type	Size	Offset	Signal
1	F35_F60	WORD	2	0	F35_F60

Total size: 2 bytes

Figure 3: Tab “Output” in dialog ”Signal Connections“ of the “Read/Write Request Telegram”

- the signals received by the Modbus master from the Modbus slave are entered in the tab ”Input“.



	Name	Type	Size	Offset	Signal
1	F60_F35	WORD	2	0	F60_F35

Total size: 2 bytes

Figure 4: Tab “Input” in dialog ”Signal Connections“ of the “Read/Write Request Telegram”

1.6.2 Validate

Prior to generating the code, the “Read/Write Request Telegram” parameterisation may be tested. The user selects “Read/Write Request Telegram” from the structure view and *Validate* from the context menu. The fault status field then displays faults and warnings, if any.

The validation function is always performed automatically prior to code generation. If a fault is detected during validation, code generation is aborted.

1.6.3 Copy

This function copies the “Read/Write Request Telegram”, including its signals, to the clipboard.

1.6.4 Delete

This function deletes the selected “Read/Write Request Telegram” from the project.

Important ”Delete“ deletes the “Read/Write Request Telegram”. Make sure that the “Read/Write Request Telegram” really has to be deleted.

1.6.5 Properties

Properties of the “Read/Write Request Telegram”

1.6.5.1 Request Telegram ”Read Write Holding Registers (23)“

Element	**Description**
Type	Modbus function “Read Write Holding Registers (23)”
Name	Any unique name for the Modbus function
Description	Description for the Modbus function
Start address of read area	0 to 65535 For *HIMatrix* Modbus Slaves 0 up to 8190 (up to COM-OS V8.32) 0 up to 16382 (from COM-OS V8.32 on)
Start address of write area	0 to 65535 For *HIMatrix* Modbus Slaves 0 up to 8190 (up to COM-OS V8.32) 0 up to 16382 (from COM-OS V8.32 on)

Table 6: Request Telegram ”Read Write Holding Registers (23)“

1.7 Write Request Telegrams

For writing signals, the Modbus master sends a “Write Request telegram” to the Modbus slave. The Modbus slave writes the received signals into its import area.

The following “Write Request telegrams” are available:

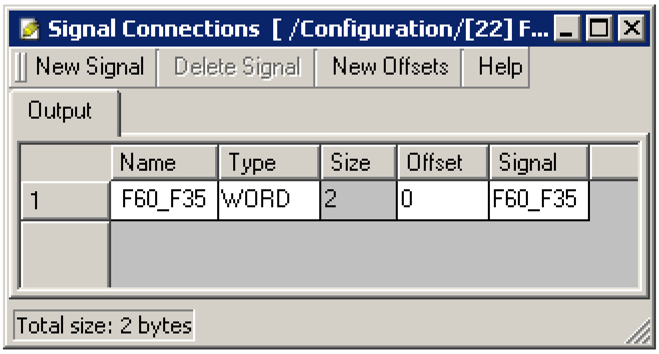
- Write Multiple Coils (15)
- Write Multiple Registers (16)
- Write Single Coil (05)
- Write Single Register (06)

The context menu of a “Write Request telegrams” contains the following functions.

Write Request Telegram
Connect signals
Validate
New
Copy
Add
Delete
Print
Properties

1.7.1 Connect signals

In the dialog “Signal Connections” of a “Write Request telegrams” the signals must be entered, that the Modbus master writes into the Modbus slave.



	Name	Type	Size	Offset	Signal
1	F60_F35	WORD	2	0	F60_F35

Total size: 2 bytes

Figure 5: Dialog ”Signal Connections“ of the of the “Write Request Telegram”

1.7.2 Validate

Prior to generating the code, the “Write Request telegrams” parameterisation may be tested. The user selects “Write Request telegrams” from the structure view and *Validate* from the context menu. The fault status field then displays faults and warnings, if any.

The validation function is always performed automatically prior to code generating. If a fault is detected during validation, code generation is aborted.

1.7.3 Copy

This function copies the “Write Request telegrams”, including its signals, to the clipboard.

1.7.4 Delete

This function deletes the selected “Write Request telegrams” from the project.

Important ”Delete“ deletes the “Write Request telegram”. Make sure that the “Write Request telegrams” really has to be deleted.

1.7.5 Properties

Properties of the “Write Request telegrams” (see 1.7.5.1 up to 1.7.5.4).

1.7.5.1 Request Telegram “Write Multiple Coils (15)“

Element	Description
Type	Modbus function ”Write Multiple Coils (15)”
Name	Any unique name for the Modbus function
Description	Description for the Modbus function
Start address of write area	0 to 65535 For *HIMatrix* Modbus Slaves 0 up to 8191 (up to COM-OS V8.32) 0 up to 16383 (from COM-OS V8.32 on)

Table 7: Request Telegram ”Write Multiple Coils (15)“

1.7.5.2 Request Telegram ”Write Multiple Registers (16)“

Element	Description
Type	Modbus function ”Write Multiple Registers (16)”
Name	Any unique name for the Modbus function
Description	Description for the Modbus function
Start address of write area	0 to 65535 For *HIMatrix* Modbus Slaves 0 up to 8190 (up to COM-OS V8.32) 0 up to 16382 (from COM-OS V8.32 on)

Table 8: Request Telegram ”Write Multiple Registers (16)“

1.7.5.3 Request Telegram “Write Single Coil (05)”

Element	Description
Type	Modbus function ”Write Single Coil (05)”
Name	Any unique name for the Modbus function
Description	Description for the Modbus function
Start address of write area	0 to 65535 For *HIMatrix* Modbus Slaves 0 up to 8191 (up to COM-OS V8.32) 0 up to 16383 (from COM-OS V8.32 on)

Table 9: Request Telegram ”Write Single Coil (05)“

1.7.5.4 Request Telegram “Write Single Register (06)”

Element	Description
Type	Modbus function ”Write Single Register (06)”
Name	Any unique name for the Modbus function
Description	Description for the Modbus function
Start address of write area	0 to 65535 For *HIMatrix* Modbus Slaves 0 up to 8190 (up to COM-OS V8.32) 0 up to 16382 (from COM-OS V8.32 on)

Table 10: Request Telegram ”Write Single Register (06)“

1.8 HIMA Modbus Master

The context menu of the Modbus master contains the following functions.

Modbus Master
Connect signals
Validate
New
Import
Export
Copy
Add
Delete
Print
Properties

1.8.1 Connect signals

Three status signals are available in the dialog box ”Signal Connections“:

- Bus faults,
- Master status and
- Master status control.

These signals enable the user to assess the status of the Modbus master in the user program and to control the Modbus master.

Inputs	Description
Bus Fault	Not used
Master Status	The master status displays the current protocol status: up to COM OS version 8.0 0: OPERATE 1: OFFLINE from COM OS version 8.0 1: OPERATE 0: OFFLINE

Table 11: Dialog Box ” Signal Connections” Associated with the Modbus Master

Outputs	Description
Master Status Control	Stops or starts the user program’s Modbus master. 0: START 1: STOP

Table 12: Dialog Box ”Signal Connections“ Associated with the Modbus Master

1.8.2 Validate

Prior to generating the code, the master and slave's parameterisation may be tested. The user selects "Modbus master" from the structure view and *Validate* from the context menu. The fault status field then displays faults and warnings, if any.

The validation function is always performed automatically prior to code generating. If a fault is detected during validation, code generation is aborted.

1.8.3 New

Using *New*, the user can add to the Modbus master:

- A ”Modbus gateway“ or a
- ”Serial Modbus“.

1.8.4 Import

In the menu function "Import" of the context menu are the submenus:

- Connections
- Slaves
- Orders of Slaves
- Signals of Slaves

By this submenus the configured *.CSV files for the configuration of the Modbus Master can be imported.

1.8.5 Export

In the menu function "Export" of the context menu are the submenus:

- Connections
- Slaves
- Orders of Slaves
- Signals of Slaves

By this submenus the configuration of the Modbus Master can be exported in *.CSV files.

1.8.6 Copy

This function copies the Modbus Master, including its configuration, to the clipboard.

1.8.7 Delete

This function deletes the selected Modbus master from the project.

Important	”Delete“ deletes the Modbus master. Make sure that the Modbus master really has to be deleted. Archive the project before using ”Delete“.

1.8.8 Properties

1.8.8.1 Tab „General“

The Modbus master name and description are assigned using the function *Properties*. The user may set the parameters In the corresponding window to specify whether the Modbus master should also function as a TCP and/or a UDP gateway.

Element	Description
Type	Modbus Master
Name	Any unique name for the Modbus master.
Description	Any unique description for the Modbus master.
Enable TCP Gateway	If the TCP Modbus gateway is enabled, at least a Modbus RS-485 Interface must be configured.
TCP server port	Standard: 502 Additional TCP/UDP ports may also be configured. The user must take these TCP/UDP ports from the ICANN (Internet Corporation for Assigned Names and Numbers) description of the TCP/UDP port list.
Maximum number of TCP connections as server	Maximum number of TCP connections opened simultaneously and functioning as a server. Value range: 1 to 32 Default value: 5
Enable UDP gateway	If the UDP Modbus gateway is enabled, at least a Modbus RS-485 interface must be configured.
UDP Port	Standard: 502 Other UDP ports may also be configured. The user must take these UDP ports from the ICANN (Internet Corporation for Assigned Names and Numbers) description of the UDP port list.
Maximum length of queue	Length of the gateway queue for other masters’ request telegrams, which have not been answered yet. It is only considered if the gateway is enabled. Value range: 1 to 20 Default value: 3

Table 13: Properties of the Modbus Master

1.8.8.2 Tab „COM/CPU“

The default values for the parameters provide the fastest possible data exchange of the Modbus data between the COM processor (COM) and the CPU processor (CPU) in the *HIMatrix* controller.
These parameters should only be changed if a reduction of the COM and/or CPU load for an application is necessary and if the process does permit it.

⚠ **The modification of the parameters are only recommended for the experienced programmer.**

An increase of the refresh rate of the COM and the CPU means also that the real refresh rate of the Modbus data will be increased. The time requirements of the plant must be verified.

Please regard also the parameter "Master Slave Data Exchange [ms]“ (see 1.9.4.6 and 1.10.4.6), which determines the refresh rate of the Modbus data from/to the Modbus slave.
This one can be increased according to the COM/CPU refresh rate.

Element	Description
Refresh Rate [ms]	Refresh rate in milliseconds with which the data of the protocol are exchanged between COM and CPU. If the "Refresh Rate“ zero or smaller than the cycle time of the controller then the data exchange is as fast as possible. Range of value: 0 to (2^{31}-1) Default value: 0
In one cycle	Activated: Transfer of the whole data of the protocol from the CPU to the COM within a cycle time of the CPU. Deactivated: Transfer of the whole data of the protocol from the CPU to the COM distributed over several CPU cycles of each 900 byte per data direction. Thereby the cycle time of the controller can be reduced. Default value: activated

Table 14: Properties of the Modbus Master

1.9 Ethernet Slaves

The TCP/UDP slaves are created in the directory of the Modbus master ”Ethernet Slaves“.

The Modbus master communicates with its TCP/UDP slaves via TCP/IP or UDP/IP. Up to 32 TCP/IP und 32 UDP/IP slaves may be configured in the Modbus master.

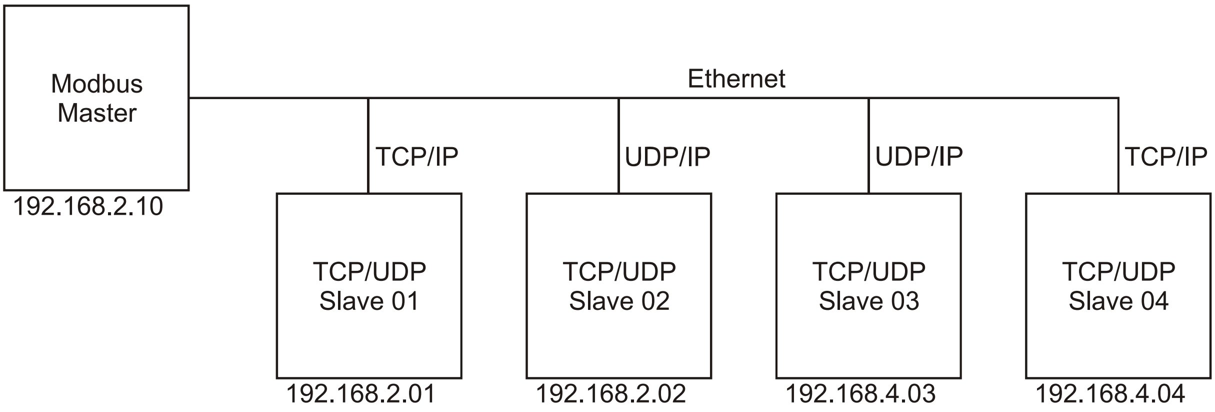


Figure 6: Modbus Communication via TCP/IP

Note: If the TCP/UDP slaves and the Modbus master are located in different subnets, the routing table must contain the corresponding user-defined routes.

Regard the ***ELOP II Factory*** online help to define the "Routing table".

The context menu ”Ethernet Slaves“ contains the following functions:

Ethernet Slave
New
Copy
Add
Delete
Print
Properties

1.9.1 New

New adds a new TCP/UDP Slave to the ”Ethernet Slaves“ directory.

1.9.2 Add

Add adds a TCP/UDP slave copied from another configuration.

1.9.3 Properties

Properties opens the dialog box ”Properties“. The ”Ethernet Slaves“ directory’s name can now be changed.

1.9.4 TCP/UDP Slave

The TCP/UDP slave's context menu contains the following functions.

TCP/UDP Slave
Connect signals
Validate
New
Import
Export
Copy
Add
Delete
Print
Properties

1.9.4.1 Connect signals

Two status signals are available in the dialog box "Signal Connections" accessible from the TCP/UDP slave's context menu:

- Slave Status and
- Slave Activation Control

These signals enable the user to assess the TCP/UDP Slave status in the user program and to control the TCP/UDP Slave master.

Inputs	Description
Slave Status	TCP/UDP slave's connection status: up to COM OS version 8.0 0: Connected 1: Disabled 2: Not connected from COM OS version 8.0 0: Not connected 1: Disabled 2: Connected

Table 15: Dialog Box "Signal Connections" Associated with the TCP/UDP Slave

Outputs	Description
Slave Activation Control	The user program enables or disables the TCP/UDP slave using this function. 0: Enable 1: Disable

Table 16: Dialog Box ”Signal Connections“ Associated with the TCP/UDP Slave

1.9.4.2 Validate

Prior to generating the code, the TCP/UDP slave’s parameterisation may be tested. The user selects the TCP/UDP slave from the structure view and *Validate* from the context menu. The fault status field then displays faults and warnings, if any.

The validation function is always performed automatically prior to code generating. If a fault is detected during validation, the code generation is aborted.

1.9.4.3 New

Using *New*, the user can add a new request telegram to the TCP/UDP slave.

1.9.4.4 Import

In the menu function “Import” of the context menu are the submenus:
-Orders of Slaves
-Signals of Slaves
By this submenus the configured *.CSV files for the configuration of the TCP/UDP slave can be imported.

1.9.4.5 Export

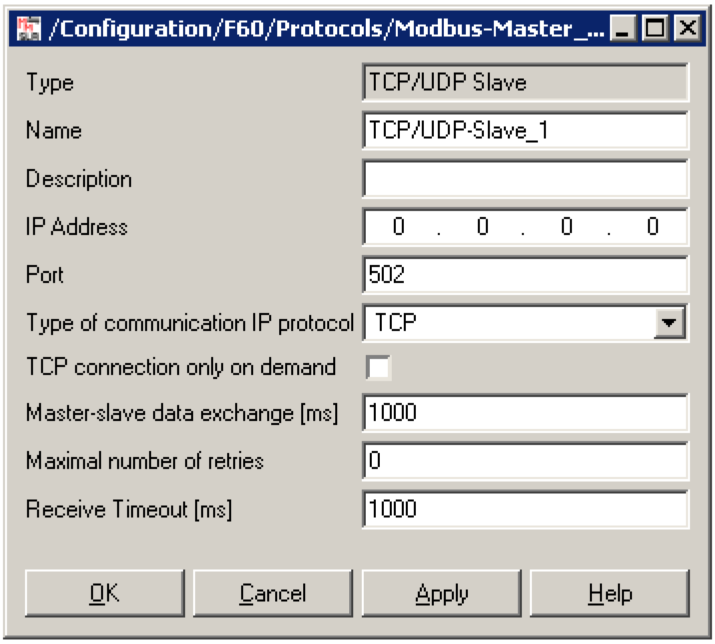
In the menu function “Export” of the context menu are the submenus:
-Orders of Slaves
-Signals of Slaves
By this submenus the configuration of the TCP/UDP slave can be exported in *.CSV files.

1.9.4.6 Properties

Using *Properties*, the user can set up the parameters for the TCP/IP or UDP/IP communication with the TCP/UDP slave.

Element	Description
Type	TCP/UDP slave
Name	Any unique name for the TCP/UDP slave
Description	Any unique description for the TCP/UDP slave
IP Address	TCP/UDP slave's IP address
Port	Standard: 502 Additional TCP/UDP ports may also be configured. The user must take these TCP/UDP ports from the ICANN (Internet Corporation for Assigned Names and Numbers) description of the TCP/UDP port list.
Type of communication IP Protocol	TCP or UDP Default value: TCP
TCP connection only on demand	If the transport protocol's type is TCP, one can set here whether the connection to this slave should be automatically removed after each data exchange: TRUE: The connection should be removed, FALSE The connection is not removed. Default value: FALSE
Master-slave data exchange [ms]	Updating rate for this slave 1 to (2^{31}-1). If the number of retries set did not help to reach the slave, the updating rate is set four times higher. See also Refresh Rate [ms] between CPU/COM 1.8.8.2.
Maximal number of retries	Number of send attempts when the slave does not answer. The number of retries can be freely set (0 to 65535). A number from 1 to 8 is recommended.
Receive Timeout [ms]	Receive Timeout [ms] for the slave. After this time a new send attempt is undertaken.

Table 17: TCP/UDP Slave's Properties



Type	TCP/UDP Slave
Name	TCP/UDP-Slave_1
Description	
IP Address	0 . 0 . 0 . 0
Port	502
Type of communication IP protocol	TCP
TCP connection only on demand	□
Master-slave data exchange [ms]	1000
Maximal number of retries	0
Receive Timeout [ms]	1000

Figure 7: Dialog Box ”Properties“ Associated with the TCP/UDP Slave

1.10 Modbus Gateway

If the gateway function has been configured, a Modbus master can additionally function as TCP and/or UDP gateway.

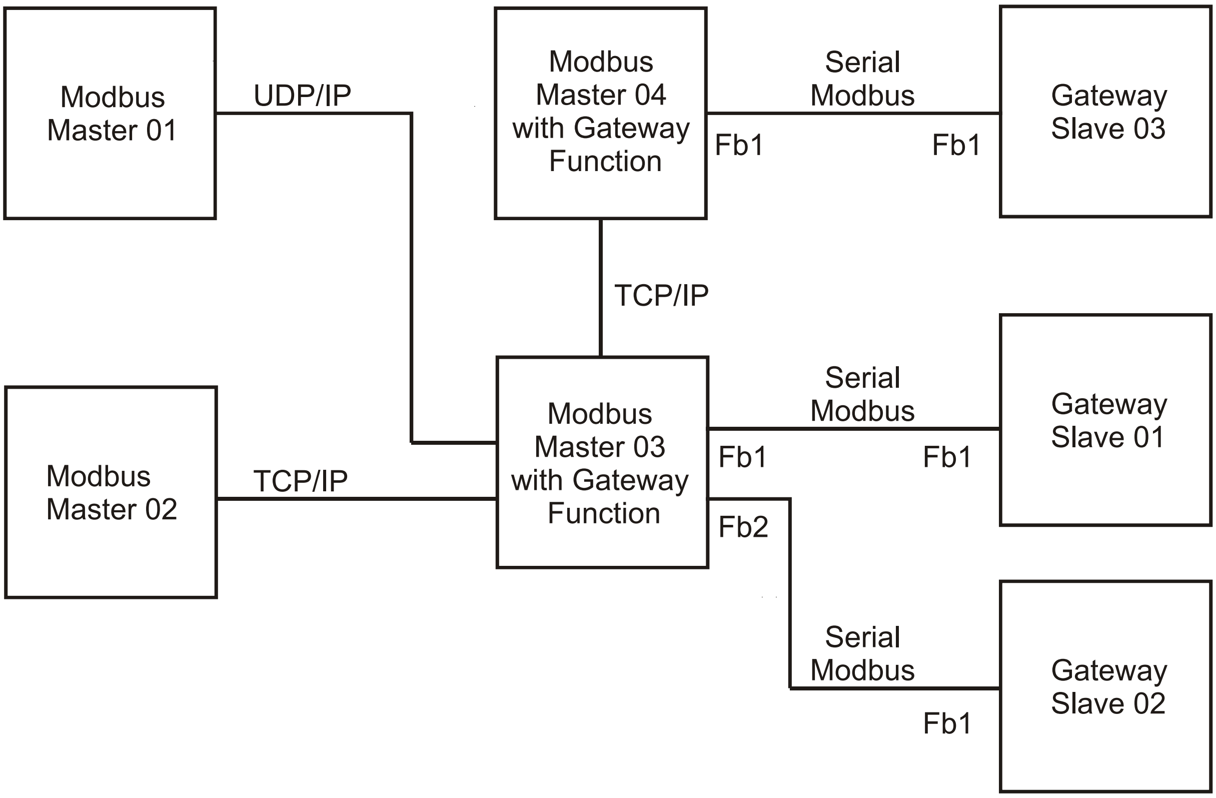


Figure 8: Modbus Communication via Modbus Master with Gateway Function

Modbus master 01, 02 and 03 communicate with its serial slave via a gateway. For this reason, the serial Modbus slaves 01 to 03 must be configured as “Gateway Slaves” in the corresponding Modbus masters.

Note: The Modbus slave’s Receive Timeout [ms] must be set higher in the Modbus master than in the gateway’s Modbus master.

- Modbus master 01 communicates with Modbus slave 01 via Modbus master 03 (with gateway function and serial interfaces fb1 and fb2).
- Modbus master 02 communicates with Modbus slave 02 via Modbus master 03 (with gateway function and serial interfaces fb1 and fb2).
- Simultaneously, Modbus master 03 communicates with Modbus slave 03 via Modbus master 04 (with gateway function and serial interface fb1).

Note: If the Modbus gateway and the Modbus master are located in different subnets, the routing table must contain the corresponding user-defined routes.

Regard the ***ELOP II Factory*** online help to define the "Routing table".

Up to 122 serial Modbus slaves can be addressed per serial interface. The Slave address range is 1 to 247.

The Modbus master can contemporary set up a maximum of 32 TCP/IP and 32 UDP/IP connections.

The Modbus gateway's context menu contains the following functions.

Modbus Gateway
Validate
New
Copy
Add
Delete
Print
Properties

1.10.1 Validate

Prior to generating the code, the Modbus gateway's parameterisation may be tested. The user selects the Modbus gateway from the structure view and *Validate* from the context menu. The fault status field displays then potential faults and warnings.

The validation function is always performed automatically prior to code generating. If a fault is detected during validation, the code generation is aborted.

1.10.2 New

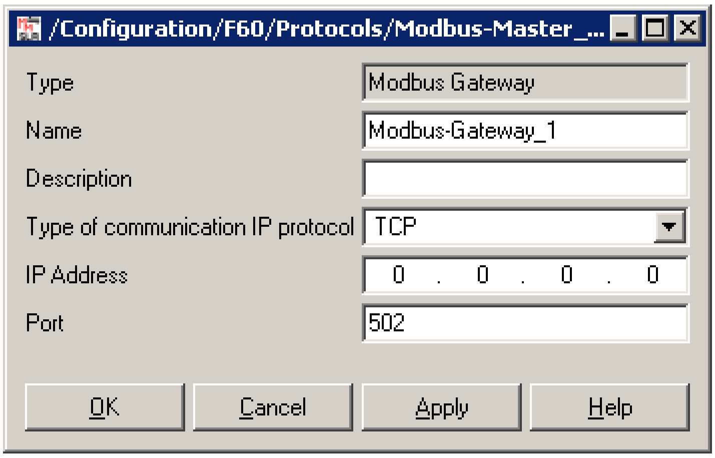
Using *New*, the user can add a new gateway slave to the Modbus Master.

1.10.3 Properties

Using *Properties*, the user can set up the parameters and the IP address which enable the Modbus gateway to communicate with the Modbus slave.

Element	Description
Type	Modbus gateway
Name	Any unique name for the gateway
Description	Any unique description for the TCP/UDP slave
Communications IP Protocol	TCP or UDP Default value: TCP
IP Address	Gateway's IP address which the Modbus Master should use to communicate with its Modbus slave. Default value: (0.0.0.0)
Port	Standard: 502

Table 18: Modbus Gateway's Properties



/Configuration/F60/Protocols/Modbus-Master_...

Type	Modbus Gateway
Name	Modbus-Gateway_1
Description	
Type of communication IP protocol	TCP
IP Address	0 . 0 . 0 . 0
Port	502

Figure 9: Dialog Box "Properties" Associated with the Modbus Gateway

1.10.4 Gateway Slave

The gateway slave’s context menu contains the following functions:

Gateway Slave
Connect signals
Validate
New
Import
Export
Copy
Add
Delete
Print
Properties

1.10.4.1 Connect signals

Two status signals are available in the dialog box ”Signal Connections“:

- Slave status and
- Slave activation control

These signals enable the user to assess the TCP/UDP slave’s status in the user program and to control the gateway slave.

Inputs	Description
Slave Status	Gateway slave’s connection status: up to COM OS version 8.0: 0: Connected 1: Disabled 2: Not connected from COM OS version 8.0: 0: Not connected 1: Disabled 2: Connected

Table 19: Dialog Box ”Signal Connections“ Associated with the Gateway Slave

Outputs	Description
Slave Activation Control	Using this function, the user program can enable or disable the gateway slave. 0: Enable 1: Disable

Table 20: Dialog Box ”Signal Connections“ Associated with the Gateway Slave

1.10.4.2 Validate

Prior to generating the code, the gateway slave's parameterisation may be tested. The user select " gateway slave" from the structure view and *Validate* from the context menu. The fault status field then displays faults and warnings, if any.

The validation function is always performed automatically prior to code generating. If a fault is detected during validation, the code generation is aborted.

1.10.4.3 New

Using "New" the user can add a new Modbus request telegram to the gateway slave.

1.10.4.4 Import

In the menu function "Import" of the context menu are the submenus:
-Orders of Slaves
-Signals of Slaves
By this submenus the configured *.CSV files for the configuration of the gateway slave can be imported.

1.10.4.5 Export

In the menu function "Export" of the context menu are the submenus:
-Orders of Slaves
-Signals of Slaves
By this submenus the configuration of the gateway slave can be exported in *.CSV files.

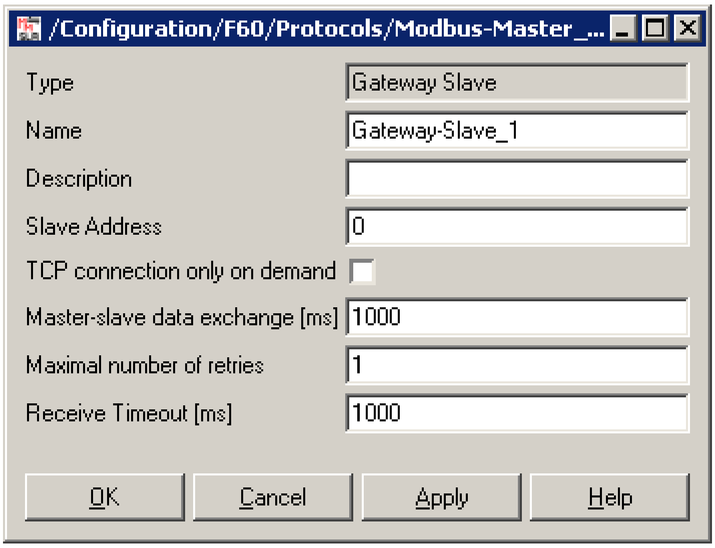
1.10.4.6 Properties

Using "Properties", the user can set up the gateway slave's parameters.

Element	**Description**
Type	Gateway Slave
Name	Any unique name for the gateway slave
Description	Any unique description for the gateway slave
Slave Address	1 to 247
TCP connection only on demand	If the transport protocol's type is TCP, one can set here whether the connection to this slave should be automatically removed after each data exchange: TRUE: The connection should be removed, FALSE The connection is not removed. Default value: FALSE
Master-slave data exchange	Updating rate for this slave.

[ms]	If the number of retries set did not help to reach the slave, the updating rate is set four times higher Default value: 1000 See also Refresh Rate [ms] between CPU/COM 1.8.8.2.
Maximal number of retries	Number of send attempts when the slave does not respond. When TCP/IP is active then “0”. The number of retries can be freely set. A number from 0 to 8 is recommended. Default value: 1
Receive Timeout [ms]	Receive Timeout [ms] for the slave. After this time a new send attempt is undertaken. Default value: 1000

Table 21: Gateway Slave’s Properties



/Configuration/F60/Protocols/Modbus-Master_...

Type	Gateway Slave
Name	Gateway-Slave_1
Description	
Slave Address	0
TCP connection only on demand	☐
Master-slave data exchange [ms]	1000
Maximal number of retries	1
Receive Timeout [ms]	1000

Figure 10: Dialog Box ”Properties“ Associated with the Gateway Slave

1.11 Serial Modbus

The Modbus master communicates with its serial slave via the serial Modbus.

Note	The pin assignment of the 9-pin sub-d connectors (fb1, fb2, fb3) is described in the data sheets of the respective *HIMatrix* controller.

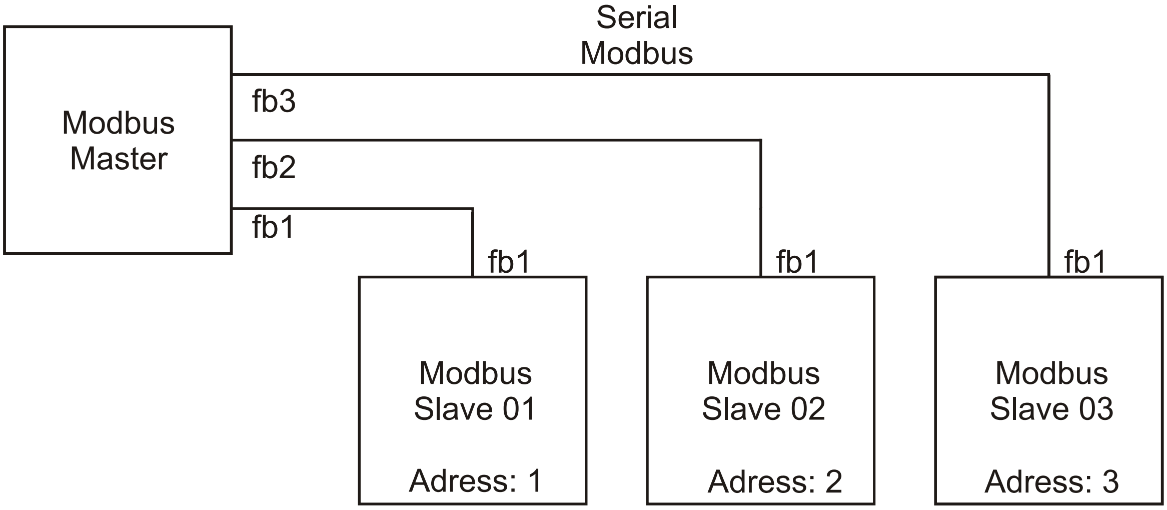
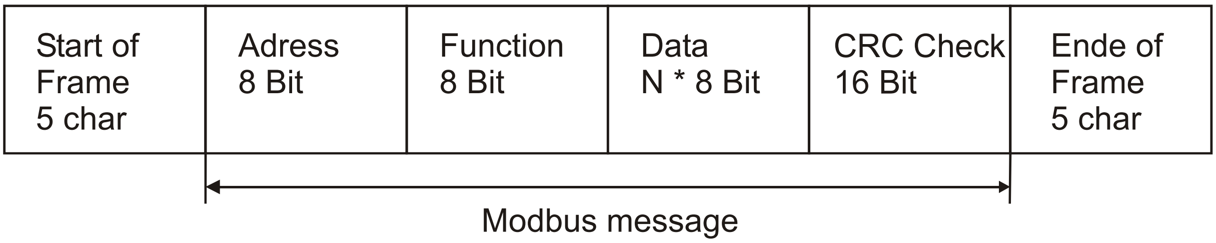


Figure 11: Modbus Communication via Serial Bus

Up to 247 serial slaves may be configured for each Modbus master.
Each slave address (1 to 247) may only be unequivocally to a single serial slave.

In according to the standard, three repeaters are permissible. Maximum 122 bus stations are possible per serial interface of a master.

The HIMA Modbus Master supports data transfer in RTU format (Remote Terminal Unit).
The HIMA RTU message frame starts and ends with the idle characters set by the user (Default value: 5 idle characters).



Start of Frame 5 char	Adress 8 Bit	Function 8 Bit	Data N * 8 Bit	CRC Check 16 Bit	Ende of Frame 5 char

Figure 12: HIMA RTU message frame

The serial Modbus context menu contains the following functions.

Serial Modbus
Validate
New
Copy
Add
Delete
Print
Priorities

1.11.1 Validate

Prior to generating the code, the serial Modbus' parameterisation may be tested. The user selects the serial Modbus from the structure view and *Validate* from the context menu. The fault status field displays then faults and warnings, if any.

The validation function is always performed automatically prior to code generating. If a fault is detected during validation, the code generation is aborted.

1.11.2 New

Using *New*, the user can add a new Modbus slave to the serial Modbus.

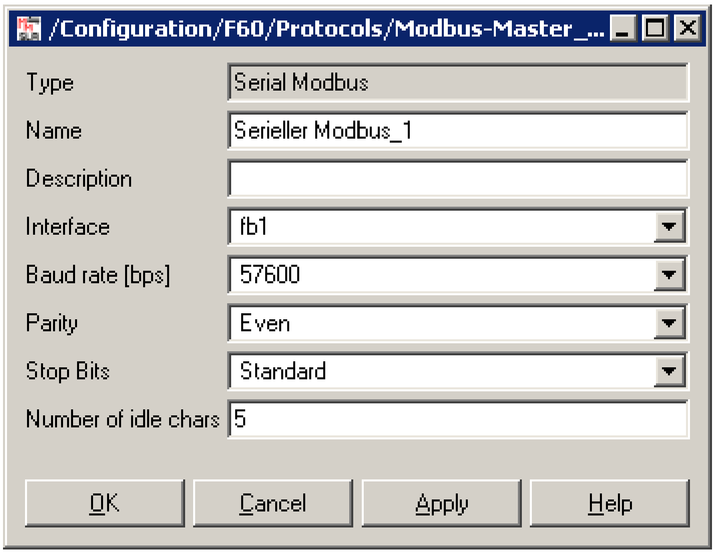
1.11.3 Properties

Using *Properties*, the user can set up the serial Modbus' parameters.

Element	Description
Type	Serial Modbus
Name	The user may select the serial Modbus name
Description	Any unique description for the serial Modbus
Interface	Field bus interface which should be used for the Modbus Master (fb1, fb2, fb3).
Baud Rate [bps]	Transfer rate for RS-485 Possible values: 115200 bps 76800 bps 62500 bps 57600 bps 38400 bps 19200 bps

Element	Description
	9600 bps 4800 bps 2400 bps 1200 bps 600 bps 300 bps Default value: 57600
Parity	None Odd Even Default value: even
Stop Bits	Standard One Stop bit Two Stop bits Default value: Standard
Number of idle chars	The number of idle characters at the start and the end of an RTU telegram frame. Value range: 0 up to 65535 Default value: 5 characters

Table 22: Serial Modbus Properties



/Configuration/F60/Protocols/Modbus-Master_...

Type	Serial Modbus
Name	Serieller Modbus_1
Description	
Interface	fb1
Baud rate [bps]	57600
Parity	Even
Stop Bits	Standard
Number of idle chars	5

Figure 13: Dialog Box ”Properties“ Associated with the Serial Modbus

1.11.4 Modbus Slave

The context menu of the Modbus slave contains the following functions.

Modbus Slave
Connect signals
Validate
New
Import
Export
Copy
Add
Delete
Print
Properties

1.11.4.1 Connect signals

Using the option ”Connect signals”, the dialog box ”Signal Connections“ opens.

Inputs	Description
Slave Status	Connection status of the Modbus slave: up to COM OS version 8.0 0: Connected 1: Disabled 2: Not connected from COM OS version 8.0 0: Not connected 1: Disabled 2: Connected

Table 23: Dialog Box ”Signal Connections“ Associated with the Modbus Slave

Outputs	Description
Slave Activation Control	Using this function, the user can enable or disable the Modbus slave. 0: Enable 1: Disable

Table 24: Dialog Box ” Signal Connections “ Associated with the Modbus Slave

1.11.4.2 Validate

Prior to generating the code, the Modbus slave's parameterisation may be tested. The user selects the Modbus slave from the structure view and *Validate* from the context menu. The fault status field then displays faults and warnings, if any.

The validation function is always performed automatically prior to code generating. If a fault is detected during validation, the code generation is aborted.

1.11.4.3 New

Using *New*, the user can add a new Modbus request telegram to the Modbus slave.

1.11.4.4 Import

In the menu function "Import" of the context menu are the submenus:
-Orders of Slaves
-Signals of Slaves
By this submenus the configured *.CSV files for the configuration of the Modbus Slave can be imported.

1.11.4.5 Export

In the menu function "Export" of the context menu are the submenus:
-Orders of Slaves
-Signals of Slaves
By this submenus the configuration of the Modbus Slave can be exported in *.CSV files.

1.11.4.6 Properties

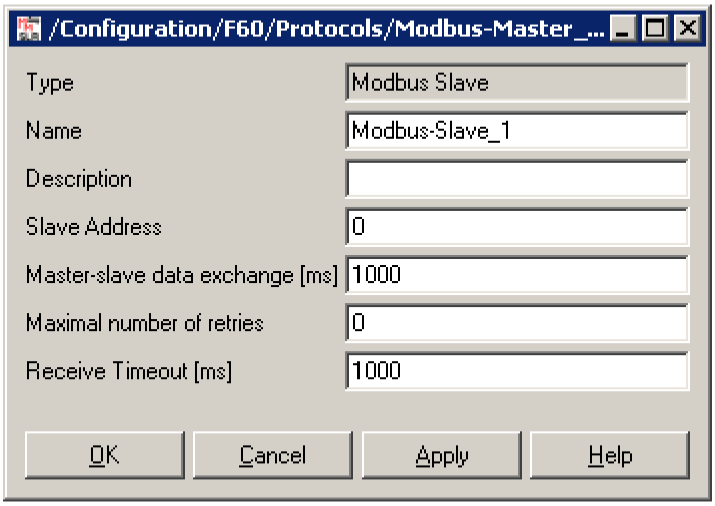
Using the option *Properties*, the user can set up the parameters of the Modbus slave.

Element	Description
Type	Modbus Slave
Name	The Modbus Slave name may be selected by the user
Description	Any unique description for the Modbus Slave
Slave Address	1 to 247
Master-slave data exchange [ms]	Updating rate for this slave. If the number of retries set did not help to reach the slave, the updating rate is set for times higher Default value: 1000 ms
Maximal number of retries	Number of send attempts when the slave does not answer. The number of retries can be freely set (0 to

	65535). A number from 0 to 8 is recommended.
Receive Timeout [ms]	Receive Timeout [ms] for the slave. After this time a new send attempt is undertaken. Default value: 1000 ms

Table 25: Properties of the Modbus Slave

Note: In the serial Modbus slave, the *Receive Timeout* depends on the transfer rate which has been set.
If the baud rate is 19200 [bps] or higher, the default value for the *Receive Timeout* may be used. If the baud rate is lower than 19200 [bps], the *Receive Timeout* must be set higher.



Type	Modbus Slave
Name	Modbus-Slave_1
Description	
Slave Address	0
Master-slave data exchange [ms]	1000
Maximal number of retries	0
Receive Timeout [ms]	1000

Figure 14: Dialog Box "Properties" Associated with the Modbus Slave

2 HIMA Modbus Slave

The HIMA Modbus slave can communicate with the Modbus master at the same time via serial interface (RS 485) and TCP/UDP (Ethernet).

2.1 Equipment and System Requirements

HIMatrix Controllers	F30, F35 und F60 from Hardware Revision:00 on
Modbus Slave serial	- HIMA ***ELOP II Factory*** from Version 3.2.x on - COM OS from Version 3.14 on - CPU OS from Version 3.14 on
Modbus Slave TCP/IP	- HIMA ***ELOP II Factory*** from Version 5.2.0 on - COM OS from Version 6.22 on - CPU OS from Version 4.50 on
HIMA Modbus Module	See data sheet of the respective *HIMatrix* controller in chapter „Connections for Field Bus Communication".
License key	Each of the following Modbus Master functions must be activated by a License key. - Modbus Slave serial - Modbus Slave TCP/IP

2.2 Modbus Slave Properties

Modbus slave	Only one Modbus slave per resource may be configured.
Max. size of transmit data	8192 Bytes (up to COM OS V8.32) 16384 Bytes (From COM OS V8.32 on)
Max. size of receive data	8192 Bytes (up to COM OS V8.32) 16384 Bytes (from COM OS V8.32 on)
Display format of the Modbus data	The *HIMatrix* controller is using the Big Endian Format for the data. Example 32 Bit data (e.g. DWORD, DINT):

32 Bit data (hex)	0x12345678			
Memory offset	0	1	2	3
Big Endian (*HIMatrix*)	12	34	56	78
Middle Endian (H51q)	56	78	12	34
Little Endian	78	56	34	12

Note	In addition to the Modbus protocol, other protocols (e.g. Profibus-DP, TCP S/R,…) can operate at the same time on a *HIMatrix* control device. In sum, 16284 bytes of data can be transmitted and 16284 bytes of data can be received per *HIMatrix* control device. The transmit and receive data can be arbitrarily separated between the protocols. The maximum size of the transmit and receive data per protocol is depended by the COM OS (see 2.2).

2.3 Modbus Functions

The following Modbus functions are supported by the HIMA Modbus Slave:

Element	Code	Type	Description
READ COIL	01	BOOL	Reads several signals (BOOL) from the slave's import or export[1)] area.
READ DISCRETE INPUT	02	BOOL	Reads several signals (BOOL) from the slave's export area.
READ HOLDING REGISTER	03	WORD	Reads several signals of any type from the slave's import or export[1)] area.
READ INPUT REGISTER	04	WORD	Reads several signals of any type from the slave's export area.
READ WRITE HOLDING REGISTER	23	WORD	Writes and reads several signals of any type in and from the slave's import area or read from export area.
WRITE MULTIPLE COIL	15	BOOL	Writes several signals (BOOL) in the slave's import area.
WRITE MULTIPLE REGISTER	16	WORD	Writes several signals of any type in the slave's import area.
WRITE SINGLE COIL	05	BOOL	Writes one single signal (BOOL) in the slave's import area.
WRITE SINGLE REGISTER	06	WORD	Writes one single signal (WORD) in the slave's import area.
Diagnostics	08	x	Only Subcode 0: Loopback-Function of the Slave
Read Device Identification	43	x	Transmits the identification data of the slave to the master.

Table 26: Description of Modbus Functions

[1] Export range can be selected only in HIMA Slaves

Note for the Modbus Function: Read Device Identification (43)
The HIMA Modbus Slave supplies identification data to the master and support the following Object-Ids:

Basic:		
0x00	VendorName	"HIMA Paul Hildebrandt GmbH"
0x01	ProductCode	"<ProductCode>"
0x02	MajorMinorRevision	"<CPU Vx.y CRC / COM Vx.y CRC>"
Regular:		
0x03	VendorUrl	"http://www.hima.de"
0x04	ProductName	"HIMatrix" or “GuardPLC”
0x05	ModelName	"<RessourceTyp>" z.B. "F30", "F31", "F35", "F60"
0x06	UserApplicationName	"<UserApplicationName>[S.R.S]" from ELOP-Project
Extended:		
0x80	CPU OS Version/CRC	"<Vx.y / 0x234adcef>"
0x81	CPU OSL Version/CRC	"<Vx.y / 0x234adcef>"
0x82	CPU BL Version/CRC	"<Vx.y / 0x234adcef>"
0x83	COM OS Version/CRC	"<Vx.y / 0x234adcef>"
0x84	COM OSL Version/CRC	"<Vx.y / 0x234adcef>"
0x85	COM BL Version/CRC	"<Vx.y / 0x234adcef>"
0x86	Configuration-CRC	„< 0x234adcef>“

For further information about Modbus, HIMA refer to the specifications:

- Modbus Application Protocol Specification
- Modbus over serial line specification and implementation guide V1.0

www.modbus.org

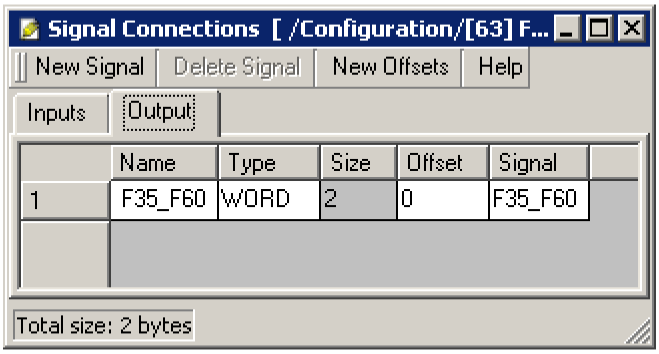
The context menu of the HIMA Modbus slave contains the following functions.

Context menu
Connect signals
Validate
New
Import
Export
Copy
Add
Delete
Print
Properties

2.3.1 Connect signals

Connect signals opens the dialog box ”Signal Connections“.
In the dialog “Signal Connections”

- all the signals transferred by the Modbus slave to the Modbus master, are entered into the tab ”Output“.

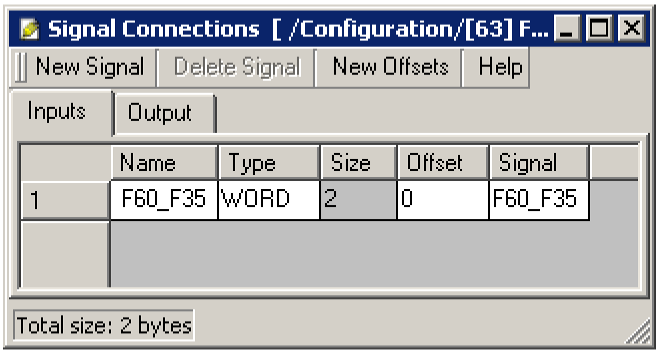


	Name	Type	Size	Offset	Signal
1	F35_F60	WORD	2	0	F35_F60

Total size: 2 bytes

Figure 15: Tab “Output” in dialog ”Signal Connections“ of the Modbus slave

- all the signals received by the Modbus slave from the Modbus master are entered in the tab ”Input“.



	Name	Type	Size	Offset	Signal
1	F60_F35	WORD	2	0	F60_F35

Total size: 2 bytes

Figure 16: Tab “Input” in dialog ”Signal Connections“ of the of the Modbus slave

2.4 Example: Write to the Import Area of a *HIMatrix* Modbus Slave

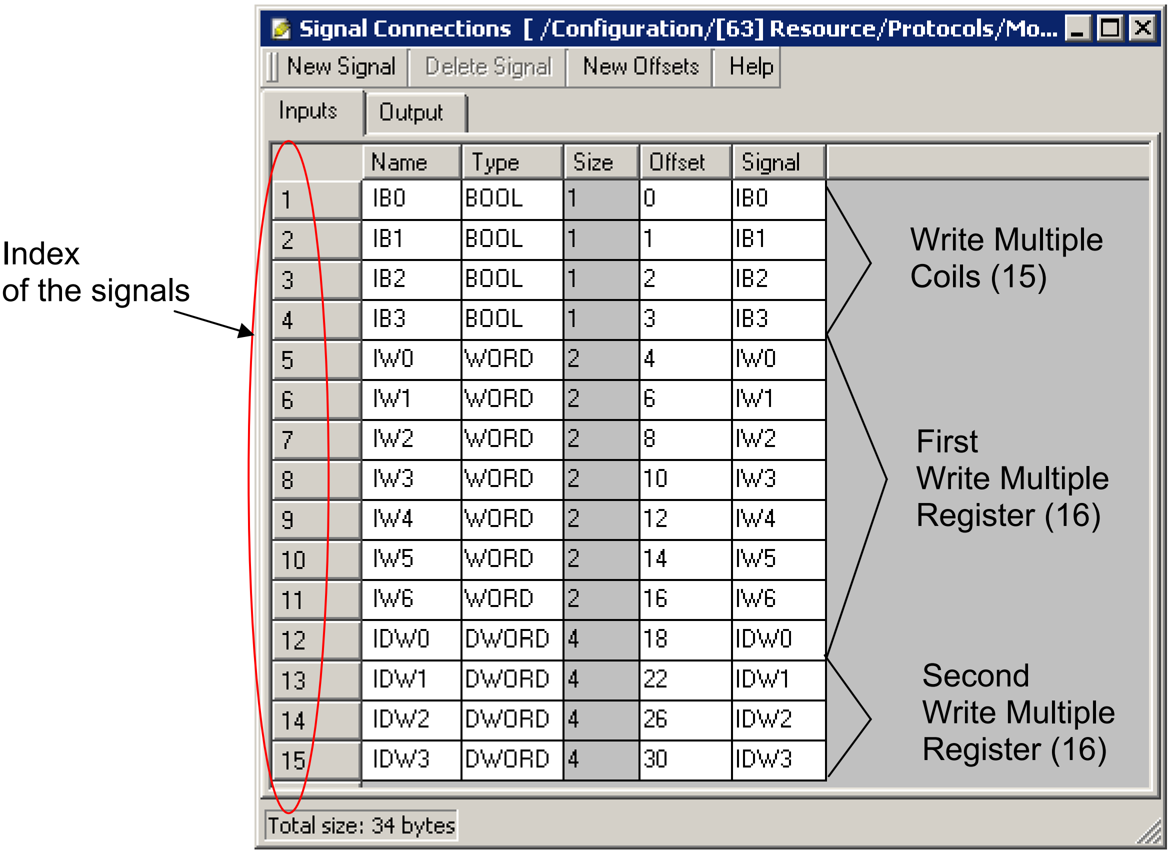
The following example describes the access of a Modbus Master to the import area of the *HIMatrix* Modbus Slave.
To write the import area of the *HIMatrix* Modbus Slave in this example, the following request telegrams are used:

- One request telegram „Write Multiple Coils (15)"
- Two request telegrams „Write Multiple Registers (16)"

2.4.1 The Import Range of the *HIMatrix* Modbus Slave

Signals of different types can be created in the import area of the *HIMatrix* Modbus Slave see Figure 17.
The signals must be stored by the column „Offset".
The address of a signal in the address range of the *HIMatrix* Modbus Slave must be determined from the Index of the signal (address of a signal = index – 1).

Note: Take into consideration, that the signals that are bigger then one byte, are stored in the Big Endian Format in the *HIMatrix* contollers.



	Name	Type	Size	Offset	Signal
1	IB0	BOOL	1	0	IB0
2	IB1	BOOL	1	1	IB1
3	IB2	BOOL	1	2	IB2
4	IB3	BOOL	1	3	IB3
5	IW0	WORD	2	4	IW0
6	IW1	WORD	2	6	IW1
7	IW2	WORD	2	8	IW2
8	IW3	WORD	2	10	IW3
9	IW4	WORD	2	12	IW4
10	IW5	WORD	2	14	IW5
11	IW6	WORD	2	16	IW6
12	IDW0	DWORD	4	18	IDW0
13	IDW1	DWORD	4	22	IDW1
14	IDW2	DWORD	4	26	IDW2
15	IDW3	DWORD	4	30	IDW3

Total size: 34 bytes

Figure 17: Import range of the Modbus slave

Note: The master can read from, and write to the import area of the slave.

The master can only read from the export area of the slave.

In the column „Size" of the dialog window „Signal Connections", the size of the signals is shown in bytes.

2.4.2 Configuration of the Request telegrams in the Modbus Master

The following sections describe the configuration of the request telegrams „Write Multiple Coils (15)“ and „Write Multiple Registers (16)“, that access the *HIMatrix* Modbus slave.

2.4.2.1 Request telegram „Write Multiple Coils (15)“

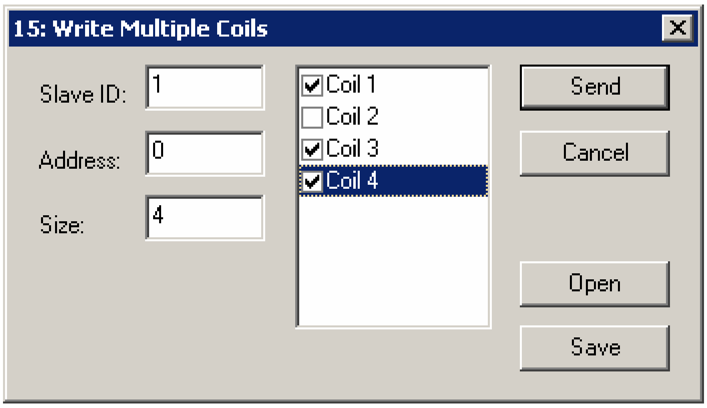
Determine the start address of the Request telegram
The start address of the request telegram must be determined from the Index of the first signal, that is written by this request telegram.

Start address = Index - 1 (*HIMatrix* Modbus Slave, see Figure 17)
Start address = 1 – 1 = 0
Enter **0** into the field „Address“.

Determine the size of the Request telegram
Since the type of the request telegram is „Write Multiple Coils (15)“, the number of signals must be entered in Bytes:

Note: A BOOL signal allocate one Byte in the memory of a *HIMatrix* controller.

Number of BOOL signals = 4
The size of the request telegram = 4 Byte
Enter **4** into the field “Size”.


15: Write Multiple Coils

Slave ID: 1
Address: 0
Size: 4

☑ Coil 1
☐ Coil 2
☑ Coil 3
☑ Coil 4

Figure 18: Request telegram „Write Multiple Coils (15)“ of the Modbus Master

2.4.2.2 First Request telegram „Write Multiple Register (16)“

Determine the start address of the request telegram
The start address of the request telegram must be determined from the Index of the first signal, that is written by this request telegram.

Start address = Index - 1 (*HIMatrix* Modbus Slave, see Figure 17)
Start address = 5 – 1 = 4
Enter **4** into the field „Address“.

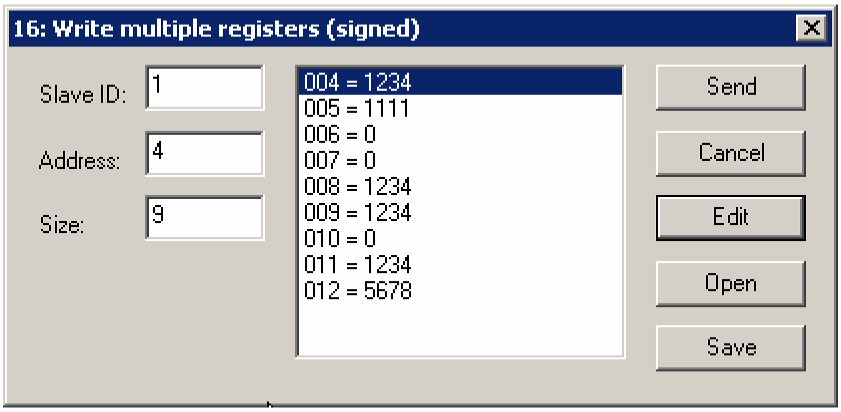
Determine the size of the request telegram

Since the type of the request telegram is a „Write Multiple Register (16)“, the number of signals must be entered in Word:

Number of WORD signals = 7	->	Size = 7 Word
Number of DWORD signals = 1	->	Size = 2 Word

The size of the request telegram = 9 Word

Enter **9** into the field “Size”.


16: Write multiple registers (signed)

Slave ID: 1
Address: 4
Size: 9

```
004 = 1234
005 = 1111
006 = 0
007 = 0
008 = 1234
009 = 1234
010 = 0
011 = 1234
012 = 5678
```

Figure 19: Request telegram „Write Multiple Register(16)“ of the Modbus Master

Note: Take into consideration, that signals bigger than one byte are stored in the Big Endian Format in the *HIMatrix* contollers.
If you use a Modbus Master from another manufacturer, then you must make sure to interpret the reading and writing signal types correctly. (see also 2.2).

2.4.2.3 Second Request telegram „Write Multiple Register (16)“

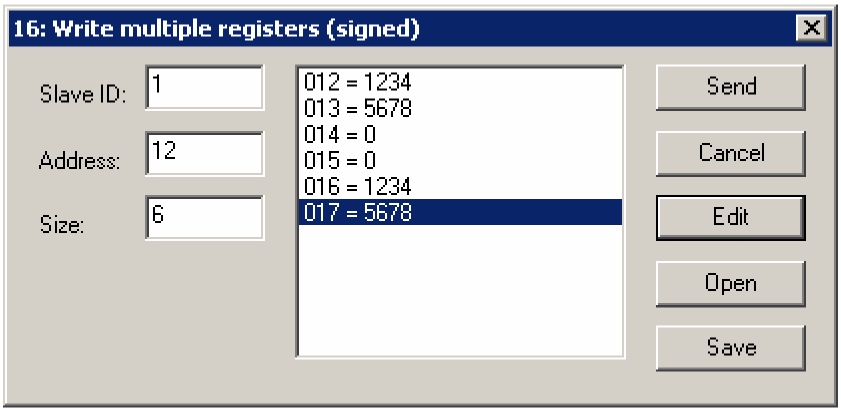
Determine the start address of the request telegram
The start address of the request telegram must be determined from the Index of the first signal, that is written by this request telegram.

Start address = Index - 1 (*HIMatrix* Modbus Slave, see Figure 17)
Start address = 13 – 1 = 12
Enter **12** into the field „Address“.

Determine the size of the request telegram

Since the type of the request telegram is „Write Multiple Register (16)“, the number of signals must entered in Word:

Number of DWORD signals = 3 -> Size = 6 Word
The size of the request telegram = 6 Word
Enter **6** into the field “Size”.



Slave ID: 1
Address: 12
Size: 6

```
012 = 1234
013 = 5678
014 = 0
015 = 0
016 = 1234
017 = 5678
```

Figure 20: Request telegram „Write Multiple Register(16)“ of the Modbus Master

Note:	Take into consideration, that signals bigger than one byte are stored in the Big Endian Format in the *HIMatrix* contollers. If you use a Modbus Master from another manufacturer, then you must make sure to interpret the reading and writing signal types correctly. (see also 2.2).

2.4.3 Validate

Prior to generating the code, the Modbus slave's parameterisation may be tested. The user select Modbus slave from the structure view and *Validate* from the context menu. The fault status field displays then faults and warnings, if any.

The validation function is always performed automatically prior to code generating. If a fault is detected during validation, the code generation is aborted.

2.4.4 Import

In the menu function "Import" of the context menu are the submenus:
-Signals
By this submenus the configured *.CSV files for the configuration of the Modbus slave can be imported.

2.4.5 Export

In the menu function "Export" of the context menu are the submenus:
-Signals
By this submenus the configuration of the Modbus slave can be exported in *.CSV files.

2.4.6 Copy

This function copies the Modbus master, including all configuration settings, to the clipboard.

2.4.7 Delete

Important	”Delete“ deletes the Modbus slave. Make sure that the Modbus slave really has to be deleted. Archive the project before using ”Delete“.

2.4.8 Properties

Properties opens the dialog box ”Properties“. In this box, the user can configure the Modbus slave's properties.

2.4.8.1 Tab ”General“

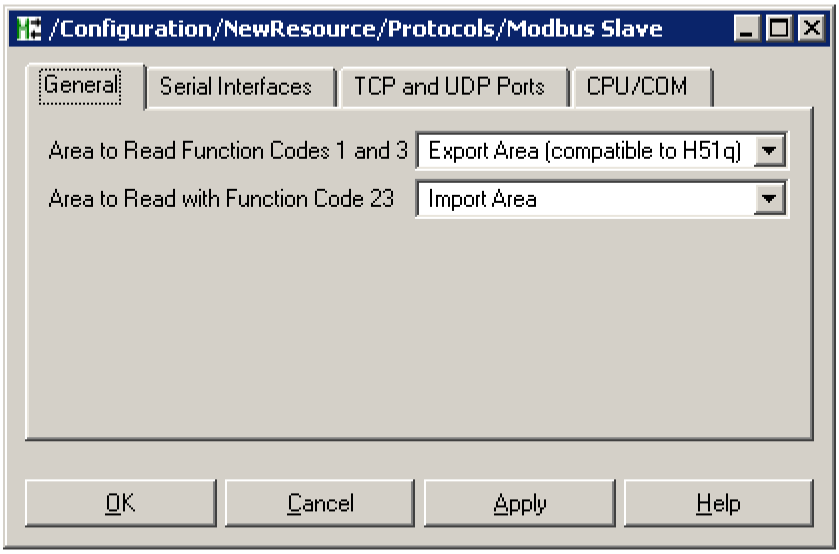
The following Modbus slave’s parameters can be set in the tab ”General“.

Element	Description
Area for Read Function Codes 1 and 3	The user can specify the Modbus slave area from which the function codes (1 and 3) should read (Refer to “Description of the Slave). Import Area Export Area (compatible with 51q)
Area to Read with Function Code 23 (From COM OS V.6 on)	The user can specify the Modbus slave area from which the function code 23 should read. Import Area The master can read from, and write to the slave’s import area. Export Area The master can read from the slave’s export area and write to the slave’s import area. Default value: Import Area

Table 27: Tab ”General“ Located in the Dialog Box ”Properties“ Associated with the Slave

Note: For the *HIMatrix* control devices (COM OS’ versions V.3 to V.6), the following read areas are available for function codes 1 and 3:

COM OS V.3:	Import area
COM OS V.4:	Export area (compatible to H51q)
COM OS V.6:	configurable



Area to Read Function Codes 1 and 3: Export Area (compatible to H51q)

Area to Read with Function Code 23: Import Area

Figure 21: Tab ”General“ in the dialog box ”Properties“ from Modbus Slave

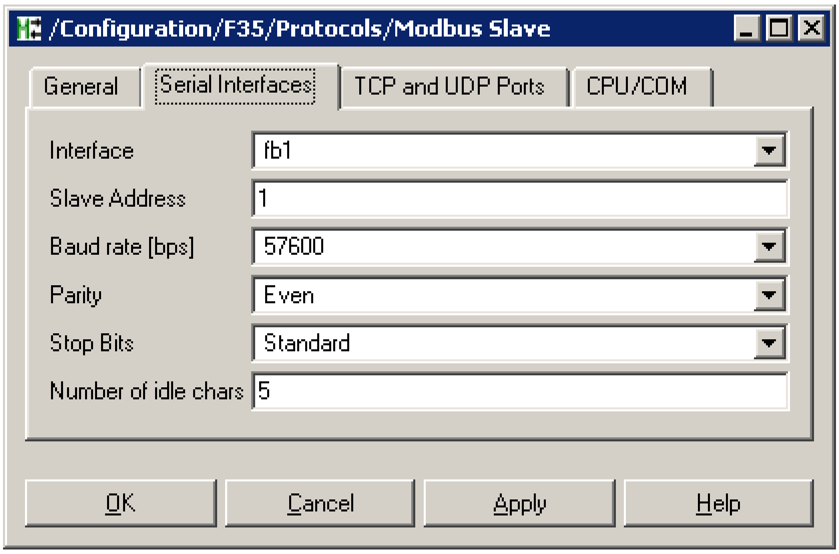
2.4.8.2 Tab ”Serial Interface“

The parameters of the Modbus slave’s serial interface can be set in the tab ”Serial Interface“.

Note	The pin assignment of the 9-pin sub-d connectors (fb1, fb2, fb3) is described in the data sheets of the respective *HIMatrix* controller.

Element	Description
Interface	Field bus interface, which should be used for the Modbus Slave (fb1, fb2, fb3).
Slave Address	Slave’s bus address Value range: 1 to 247
Baud rate [bps]	Transfer rate for RS-485 Possible values: 115200 bps 76800 bps 62500 bps 57600 bps 38400 bps 19200 bps 9600 bps 4800 bps 2400 bps 1200 bps 600 bps 300 bps Default value: 57600
Parity	None Odd Even Default value: even
Stop Bits	Standard (adapt the number of stop bits to the parity: with parity = 1 stop bit, no parity = 2 stop bit) One Stop bit Two Stop bits Default value: Standard
Number of idle chars	The number of idle characters at the start and the end of a RTU telegram frame. Value range: 0 up to 65535 Default value: 5 characters

Table 28: Tab “Serial Interface” in the dialog box „Properties“ from Modbus Slave



Interface: fb1

Slave Address: 1

Baud rate [bps]: 57600

Parity: Even

Stop Bits: Standard

Number of idle chars: 5

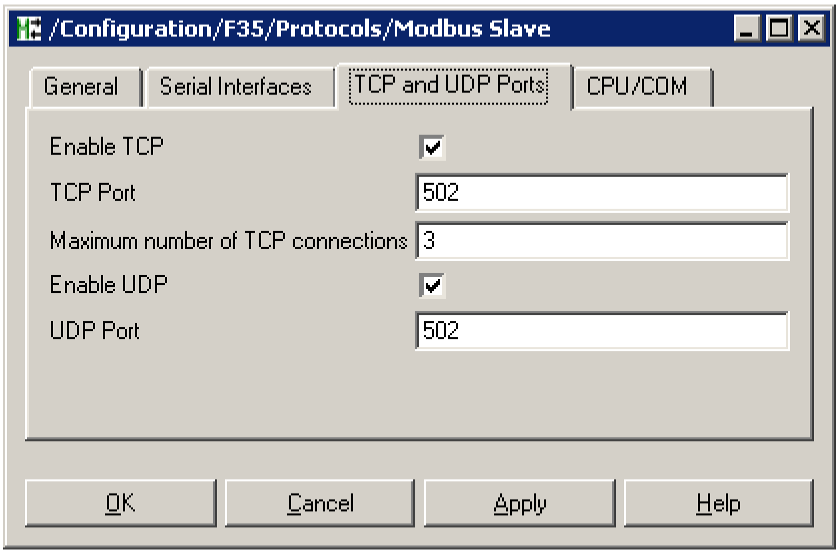
Figure 22: Tab “Serial Interface“ in the dialog box „Properties“

2.4.8.3 Tab „TCP and UDP Ports“

The following parameters of the Modbus slave’s TCP and UDP ports can be set in the tab „ TCP and UDP Ports “.

Element	Description
Enable TCP	Enables the TCP/IP connections
TCP Port	Standard: 502
Maximum Number of TCP Connections	Maximum number of TCP connections opened simultaneously and functioning as server. Value range: 1 to 20 Default value: 3
UDP Enable	Enables UDP/IP connections
UDP Port	Standard: 502

Table 29: Tab “TCP and UDP Ports” in the Dialog Box „Properties“ from Modbus Slave



Enable TCP ☑

TCP Port 502

Maximum number of TCP connections 3

Enable UDP ☑

UDP Port 502

Figure 23: Tab „TCP and UDP Ports“ in the dialog box „Properties“ from Modbus Slave

2.4.8.4 Tab „CPU/COM“

The default values for the parameters provide the fastest possible data exchange of the Modbus data between the COM processor (COM) and the CPU processor (CPU) in the *HIMatrix* controller.

These parameters should only be changed if a reduction of the COM and/or CPU load for an application is necessary and if the process does permit it.

⚠

The modification of the parameters are only recommended for the experienced programmer.

An increase of the refresh rate of the COM and the CPU means also that the real refresh rate of the Modbus data will be increased. The time requirements of the plant must be verified.

Please regard also the parameter "Master Slave Data Exchange [ms]“ in the Modbus master (see 1.9.4.6 and 1.10.4.6) , which determines the refresh rate of the Modbus data from/to the Modbus slave.

This one can be increased according to the COM/CPU refresh rate.

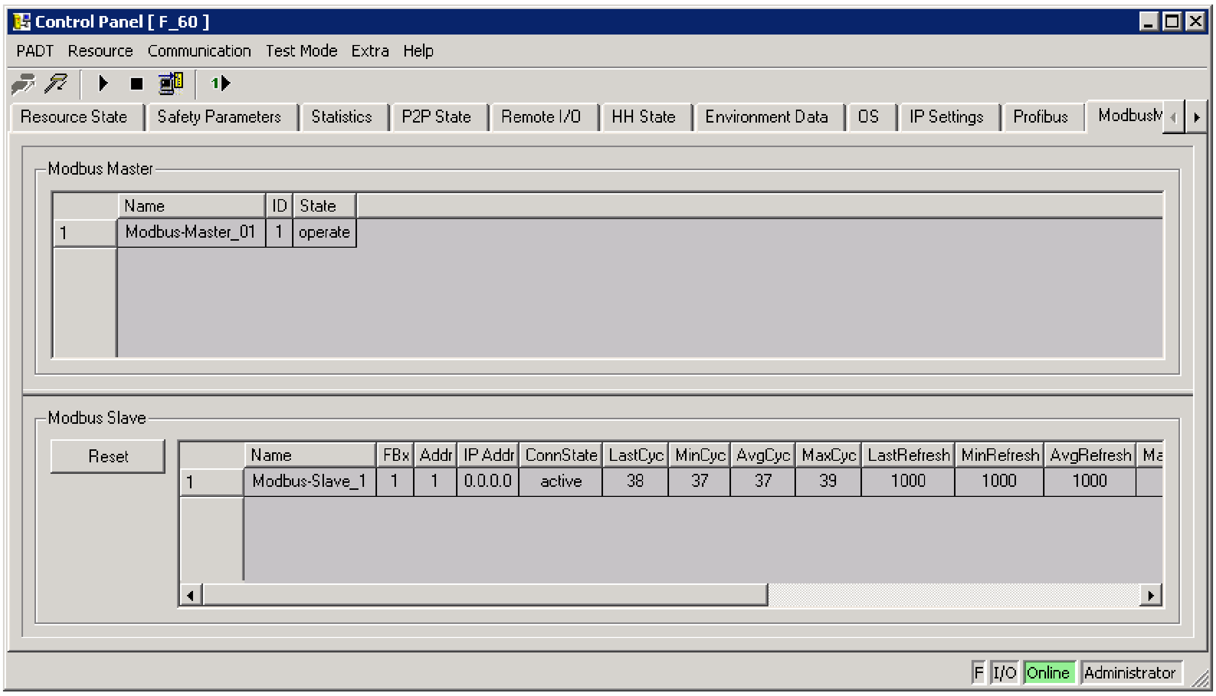
Element	Description
Refresh Rate [ms]	Refresh rate in milliseconds with which the data of the protocol are exchanged between COM and CPU. If the "Refresh Rate“ zero or smaller than the cycle time of the controller then the data exchange is as fast as possible. Range of value: 0 to (2^{31}-1) Default value: 0
Within one cycle	Activated Transfer of the whole data of the protocol from the CPU to the COM within a cycle time of the CPU. Deactivated Transfer of the whole data of the protocol from the CPU to the COM distributed over several CPU cycles of each 900 byte per data direction. Thereby the cycle time of the controller can be reduced. Default value: activated

Table 30: Tab “CPU/COM” in the Dialog Box „Properties“ from Modbus Slave

3 Diagnosis and error codes

3.1 Control Panel (Tab ”ModbusMs.“)

On the control panel located in the hardware management window, the user may select the tab ”ModbusMs.“.
The tab “ModbusMs.” is divided into two sections, ”Modbus Master“ and ”Modbus Slave“.



	Name	ID	State
1	Modbus-Master_01	1	operate

	Name	FBx	Addr	IP Addr	ConnState	LastCyc	MinCyc	AvgCyc	MaxCyc	LastRefresh	MinRefresh	AvgRefresh	Ma
1	Modbus-Slave_1	1	1	0.0.0.0	active	38	37	37	39	1000	1000	1000	

Figure 24: Tab ”ModbusMs.“ in the dialog box ”Control Panel“

3.1.1 Section ”Modbus Master“

The section ”Modbus Master“ contains the fields displaying the current master’s status.

Element	Description
Name	Master name
ID	Master ID
Status	Current operation status OFFLINE, OPERATE, UNDEFINED

Table 31: Values Displayed in the Field ”Master Data“

3.1.2 Section ”Modbus Slave“

The section ”Modbus Slave“ contains the fields displaying the selected slave’s current status information (e.g. min./max. cycle time etc.).

Reset:

This option resets the statistical data (e.g. min./max. cycle time etc.) to 0.

Element	Description
Name	Slave name
FBx	Number of the master’s field bus interfaces
Addr	Slave’s address on the serial bus
IPAddr	Slave’s IP address
ConnState	Connection Status Connected Disabled Not connected
LastCyc	Last cycle time
MinCyc	Min. cycle time
AvgCyc	Average cycle duration
MaxCyc	Maximum cycle time occurred
LastRefresh	Last duration of a refresh interval
MinRefresh	Min. duration of a refresh interval
AvgRefresh	Average duration of a refresh interval
MaxRefresh	Max. refresh time occurred
NumReq	Number of transferred requests
NumPosAnswer	Number of positive answers
NumTimeouts	Number of timeouts
LastRtt	Last Round Trip Time (RTT)
MinRtt	Min. RTT
AvgRtt	Average RTT
MaxRtt	Max. RTT occurred

Table 32: Valued Displayed in the Field ”Slave Data“

3.1.3 Diagnosis Field bus (FBx) LEDs of the serial interface

The status of the serial Modbus communication is indicated by the FBx LED of the configured serial interface (fb1, fb2, fb3).

3.1.3.1 Function of the FBx LED of the Modbus Master

Function of the FBx LED before COM OS 10.38

FBx LED	Description
OFF	No Modbus slave is configured to this serial interface in the configuration of the Modbus master.
Flashing in one second cycle time	No data exchange! At least one Modbus slave is configured to this serial interface in the configuration of the Modbus master.
ON	The Modbus master is in the status OPERATE. Data exchange with all Modbus slaves that are configured to this serial interface.
Flashing in ¼ second cycle time	The Modbus master is in the status OPERATE. At least one of the Modbus slaves on this serial interface is not reachable.

Function of the FBx LED from COM OS 10.38 on

FBx LED	Description
OFF	No data exchange!
Flashing	The FBx LED is flashing, if the Modbus master sends or receives data packets.

3.1.3.2 Function of the FBx LED of the Modbus Slave

Function of the FBx LED before COM OS 10.38

LED FBx	Description
OFF	1. The serial interface is not configured in the Modbus slave. 2. The *HIMatrix* controller is in status STOP.
Flashing in one second cycle time	No data exchange! But the Modbus slave is configured on this serial interface.
Flashing in irregular intervals	Data exchange is active. Each reception of a Modbus telegram switches the FBx LED on and off.

Function of the FBx LED from COM OS 10.38 on

FBx LED	Description
OFF	No data exchange!
Flashing	The FBx LED is flashing, if the Modbus master sends or receives data packets.

3.1.4 Error Codes of a TCP/IP connection

The error codes of the TCP/IP connection are displayed in the hardware management in the dialog window "Diagnosis".

Open the dialog window „Diagnosis“:

- ❑ Change into the Hardware Management.
- ❑ Open over „main menu“ *Online->Diagnosis* the dialog box „Diagnosis“.
- ❑ Activate „Com Short Term diagnostic“, to show the error messages.

Error code	Description
35	Operation would block
48	Address already in use
50	Network is down
53	Software caused connection abort
54	Connection reset by peer
55	No buffer space available
60	Operation timed out
61	Connection refused
65	No route to peer host

Table 33: Error codes of TCP/IP connections

4 Application with Modbus Gateway

Modbus Master 01 communicates with Modbus Slave 01 via Modbus Master 02 (with gateway function).

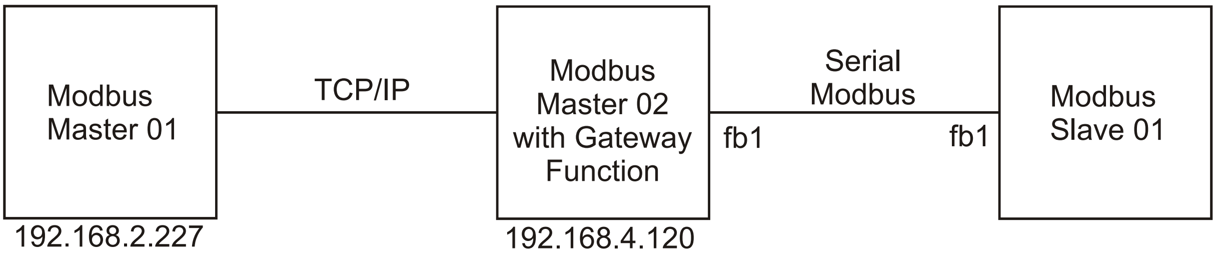


Figure 25: Modbus Communication via a Modbus Master with Gateway Function

Note:	If the Modbus gateway and the Modbus master are located in different subnets, the routing table must contain the corresponding user-defined routes. Regard the ***ELOP II Factory*** online help to define the "Routing table". Consider the equipment and the system requirements specified in Chapter 1.1

4.1 Entering the Modbus Master in the Routing Table

Note:	If Modbus Master 01 and Modbus Master 02 are located in the same subnet, no entries are required in the routing table. In this case, continue with Chapter 4.2.

Routing allows data exchange between control devices located in different subnets. Up to 8 user-defined routes may be specified in the routing table.

Entries in the routing tables

- IP address specific to the communication partners
- Subnet mask 255.255.255.255
- Ethernet interface’s name (for *HIMatrix* control devices 0.0.0.1)

Step1: Entering parameters in the routing tables associated with Modbus master 01:

- ❑ Open the control panel of the resource associated with Modbus master 01.
- ❑ Open the Tab „IP Settings“.
- ❑ Select *Add* to open the pop-up window „Add Routing Settings“.
- ❑ Enter the following parameters in the routing table.

Modbus Master 01’s Routing Table		
Route IP	Route Subnet Mask	Gateway/IF
192.168.4.120	255.255.255.255	0.0.0.1

Table 34: Entries for the Routing Table Associated with Modbus Master 01

Step2: Entering parameters in the routing table associated with the Modbus master 02 with gateway:

- ❑ Open the control panel of the resource associated with Modbus master 02.
- ❑ Open the Tab „IP Settings“.
- ❑ Select *Add* to open the pop-up window „Add Routing Settings “.
- ❑ Enter the following parameters in the routing table.

Modbus Master 02 Gateway’s Routing-Table		
Route IP	Route Subnet Mask	Gateway/IF
192.168.2.227	255.255.255.255	0.0.0.1

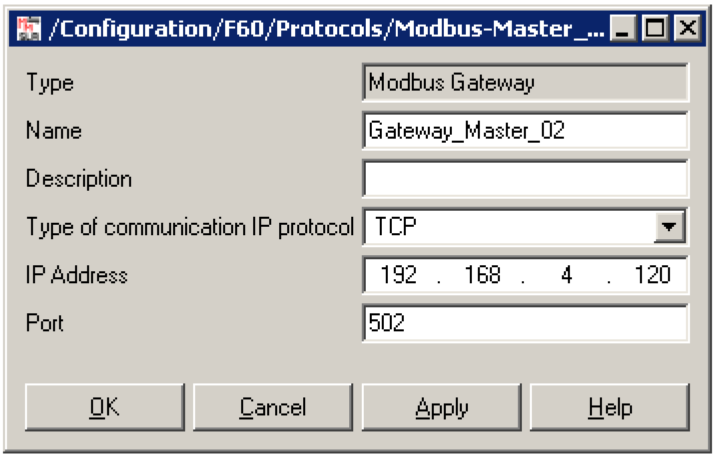
Table 35: Entries for the Routing Table Associated with Modbus Master 02 (with Gateway Function)

4.2 Configuring the Modbus Master 01

The Modbus gateway is created in the Modbus master 01. The parameters and the IP address for communication via Modbus Master 02 (with gateway function) are entered in the Modbus gateway.
All gateway slaves are entered in the directory „Modbus Gateway".

Step1: Configuring the communication with the Modbus gateway:

- Open the structure tree of the Modbus Master 01's resource
- Select *ModbusMaster->New->Modbus Gateway*
- Open the dialog box „Properties, by selecting *Modbus Gateway-> Properties*.
- Enter the parameters displayed in Figure 26.



/Configuration/F60/Protocols/Modbus-Master_...

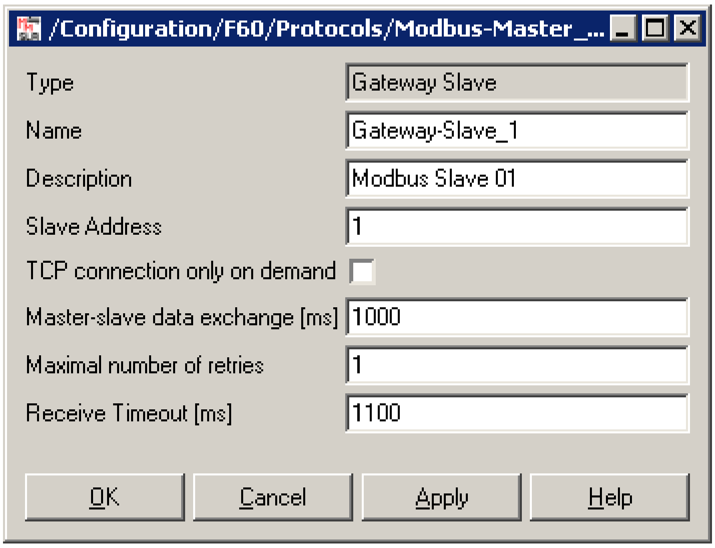
Type	Modbus Gateway
Name	Gateway_Master_02
Description	
Type of communication IP protocol	TCP
IP Address	192 . 168 . 4 . 120
Port	502

Figure 26: Entering the IP Address of the Modbus Master 02 (Gateway)

Step2: Configuring the gateway slave in the Master 01:

- ❑ Select *Modbus Gateway->New->Gateway Slave.*
- ❑ Open the dialog box „Properties“, by selecting *Gateway Slave->Properties*“.
- ❑ Enter the parameters displayed in Figure 27.

Note: The Modbus slave’s Receive Timeout [ms] must be set higher in the Modbus master than in the gateway’s Modbus master.



Type	Gateway Slave
Name	Gateway-Slave_1
Description	Modbus Slave 01
Slave Address	1
TCP connection only on demand	□
Master-slave data exchange [ms]	1000
Maximal number of retries	1
Receive Timeout [ms]	1100

Figure 27: Gateway Slave

Step3: Creating the request telegrams:

- ❑ Select *Modbus Gateway->Gateway Slave->New*
- ❑ In the context menu, select:
  - ❑ A „Read Holding Register (3)” and
  - ❑ A “Write Single Register (6)”

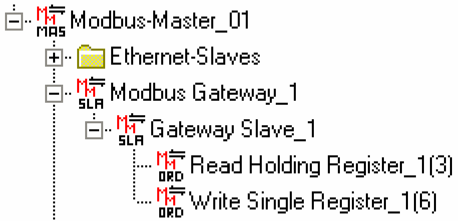
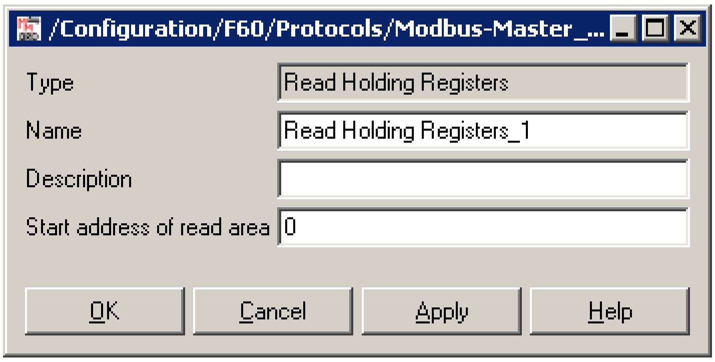

```
Modbus-Master_01
  Ethernet-Slaves
  Modbus Gateway_1
    Gateway Slave_1
      Read Holding Register_1(3)
      Write Single Register_1(6)
```

Figure 28: Creating Two Request Telegrams

Step4: Configuring the request telegrams:

- ❑ Select
  Modbus Gateway->Gateway Slave-> Read Holding Register (3).
- ❑ Open the dialog box „Properties“ by selecting *Properties* from the context menu.
- ❑ Enter the start address “0”.

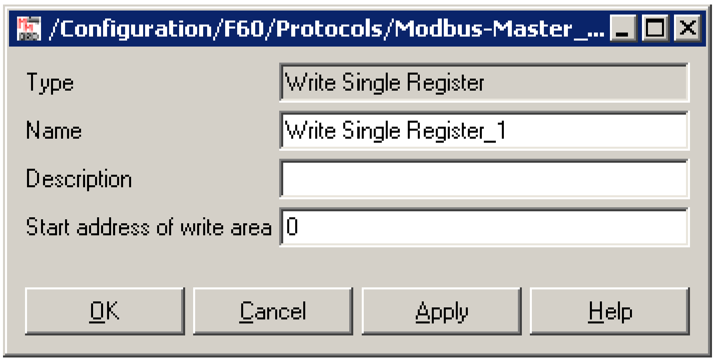

/Configuration/F60/Protocols/Modbus-Master_...

Type: Read Holding Registers
Name: Read Holding Registers_1
Description:
Start address of read area: 0

Figure 29: Request Telegram’s Configuration „Read Holding Register (03)“

Step 5: Configuring the request telegrams:

- ❑ Select
  Modbus Gateway->Gateway Slave-> Write Single Register (6).
- ❑ Open the dialog box „Properties“ by selecting *Properties* from the context menu.
- ❑ Enter the start address “0”.


/Configuration/F60/Protocols/Modbus-Master_...

Type: Write Single Register
Name: Write Single Register_1
Description:
Start address of write area: 0

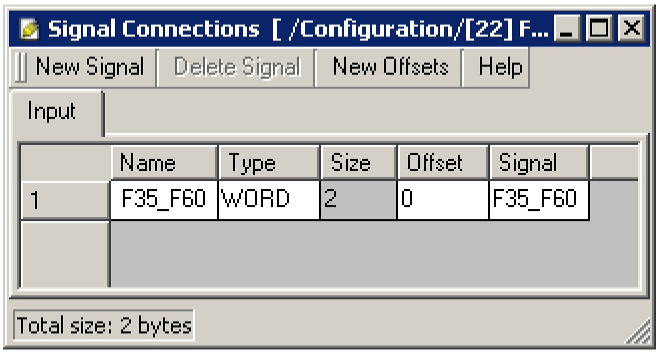
Figure 30: Request Telegram’s Configuration „Write Single Register (06)“

Step 6: Creating signals in the signal editor of the Modbus master:

- Open the signal editor by selecting *Signals -> Editor* from the menu bar.
- Create the signal „F35_F60“ (type WORD).
  This signal is used as an INPUT signal of the Modbus master.
- Create the signal „F60_F35“ (type WORD).
  This signal is used as an output signal of the Modbus master.

Step 7: Configuring the input signals:

- Select
  Modbus Gateway->Gateway Slave-> Read Holding Register (3).
- Open the dialog box „Signal Connections“ by selecting *Connect signals* from the context menu.
- Open the signal editor by selecting *Signals -> Editor* from the menu bar.
- In the signal editor, click the „Name“ corresponding to the signal „F35_F60“: Drag and drop the name to the input signal in the tab "Input" located in the dialog box "Signal Connections"´.
- Click the button *New Offsets* located in the dialog box „Signal Connections“.
- In the pop-up window „Renumber Offsets“, click the button *Renumber*.
- Close the dialog box.



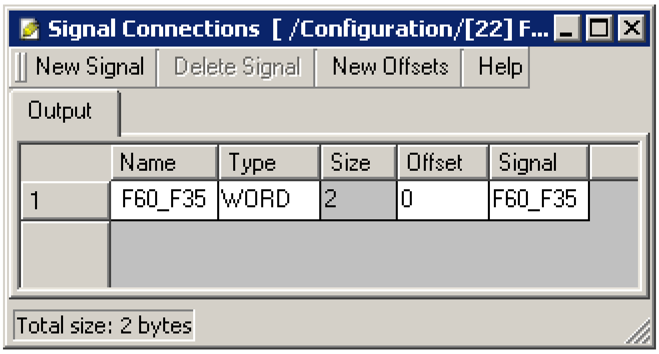
	Name	Type	Size	Offset	Signal
1	F35_F60	WORD	2	0	F35_F60

Total size: 2 bytes

Figure 31: Dialog Box „Signal Connections“, in the „Read Holding Tab (03)“

Step 8: Configuring the output signals:

- Select *Modbus Gateway->Gateway Slave-> Write Single Register (6).*
- Open the dialog box „Signal Connections“ by selecting *Connect signals* from the context menu.
- Open the signal editor by selecting *Signals -> Editor* from the menu bar.
- In the signal editor, click the „Name“ corresponding to the signal „F60_F35“. Drag and drop the name to the output signal in the tab "Output" located in the dialog box "Signal Connections".
- Click the button *New Offsets* located in the dialog box „Signal Connections“.
- In the pop-up window „Renumber Offsets“, click the button *Renumber*.
- Close the dialog box.



	Name	Type	Size	Offset	Signal
1	F60_F35	WORD	2	0	F60_F35

Total size: 2 bytes

Figure 32: Dialog Box „Signal Connections“ Associated with „Write Single Tab (06)“

Step 9: Loading the code into the resource:

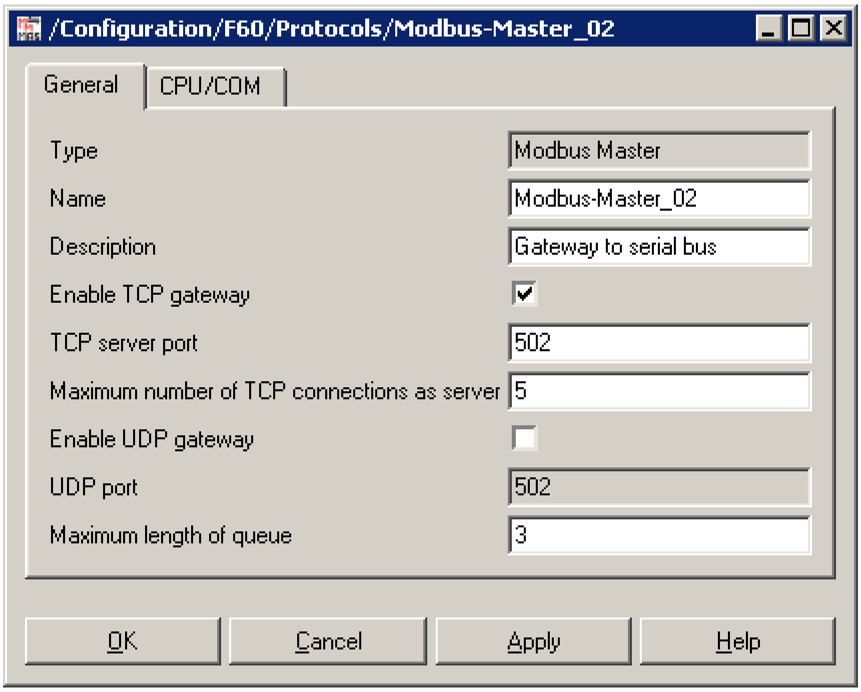
- Start the code generator specific to the resource.
- Make sure that the code has been generated correctly (see the “fault status field”).
- Load the code into the resource.

4.3 Configuring the Modbus Master 02

The gateway function must be enabled in the Modbus master 02's properties. This action connects the gateway slaves configured in the master 01 with the serial slaves.

Step 1: Configuring the TCP gateway in the Modbus master 02:

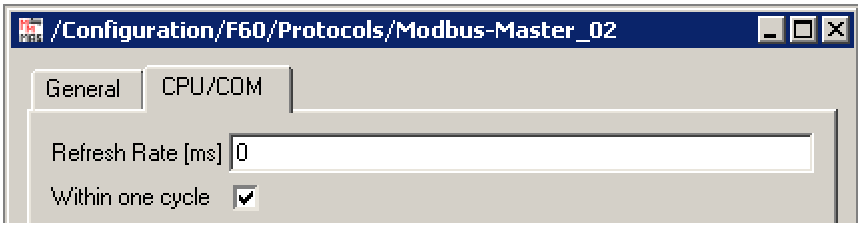
- Open the structure tree of the Modbus master 02's resource
- Select *Modbus Master->Properties.*
- Enable the TCP gateway and enter the parameters displayed in Figure 33 and Figure 34.



/Configuration/F60/Protocols/Modbus-Master_02

Type	Modbus Master
Name	Modbus-Master_02
Description	Gateway to serial bus
Enable TCP gateway	☑
TCP server port	502
Maximum number of TCP connections as server	5
Enable UDP gateway	□
UDP port	502
Maximum length of queue	3

Figure 33: Tab „General“ in the dialog box „Modbus Master 02“



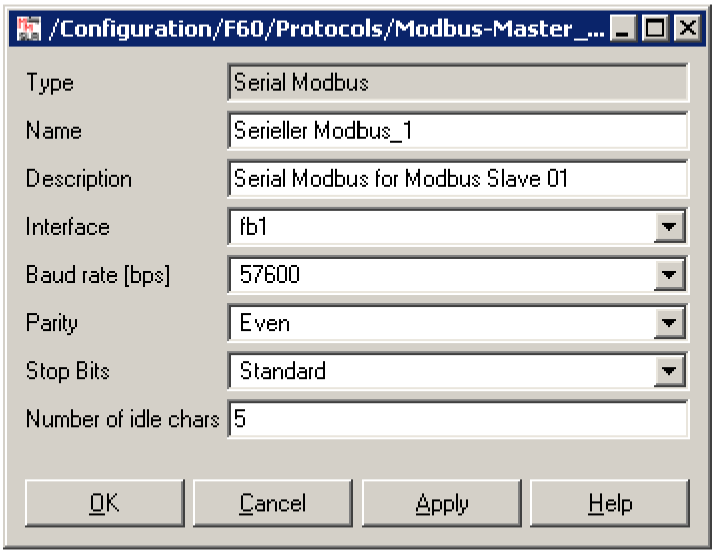
/Configuration/F60/Protocols/Modbus-Master_02

Refresh Rate [ms]	0
Within one cycle	☑

Figure 34: Tab „CPU/COM“ in the dialog box „Modbus Master 02“

Step 2: Configuring the serial Modbus in the Modbus master 02:

- ❑ Open the structure tree of the Modbus master 02's resource.
- ❑ Select *Serial Modbus->Properties.*
- ❑ Enter the parameters displayed in Figure 35.



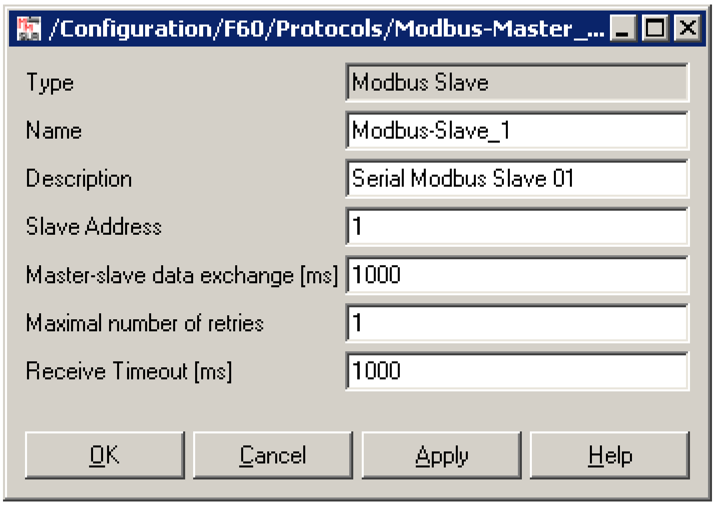
/Configuration/F60/Protocols/Modbus-Master_...

Type	Serial Modbus
Name	Serieller Modbus_1
Description	Serial Modbus for Modbus Slave 01
Interface	fb1
Baud rate [bps]	57600
Parity	Even
Stop Bits	Standard
Number of idle chars	5

Figure 35: Dialog Box „Serial Modbus 01“

Step 3: Configuring the Modbus slaves:

- ❑ Select *Serial Modbus->Modbus Slave->Properties.*
- ❑ Enter the parameters displayed in Figure 36.



/Configuration/F60/Protocols/Modbus-Master_...

Type	Modbus Slave
Name	Modbus-Slave_1
Description	Serial Modbus Slave 01
Slave Address	1
Master-slave data exchange [ms]	1000
Maximal number of retries	1
Receive Timeout [ms]	1000

Figure 36: Dialog box „Modbus Slave“

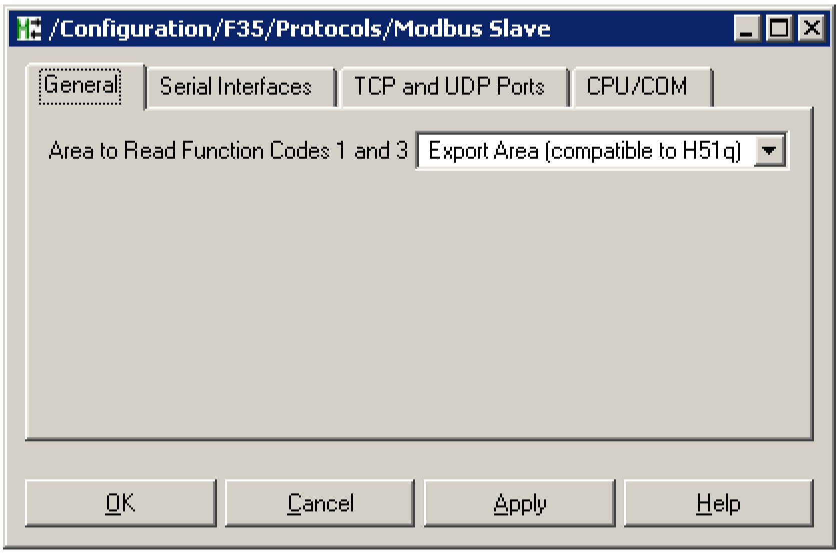
Step 4: Loading the code into the resource:

- Start the code generator specific to the resource.
- Make sure that the code has been generated correctly (see the “fault status field”).
- Load the code into the resource.

4.4 Configuring the Modbus Slave 01

Step 1: Configuring the serial Modbus slave 01:

- Open the structure tree of the Modbus slave 01's resource
- Select *Protocols->Modbus Slave*
- Open the dialog box „Properties“ by selecting *Modbus Slave-> Properties* from the context menu
- Select the tab „General “
- Enter the parameters displayed in Figure 37.



Area to Read Function Codes 1 and 3: Export Area (compatible to H51q)

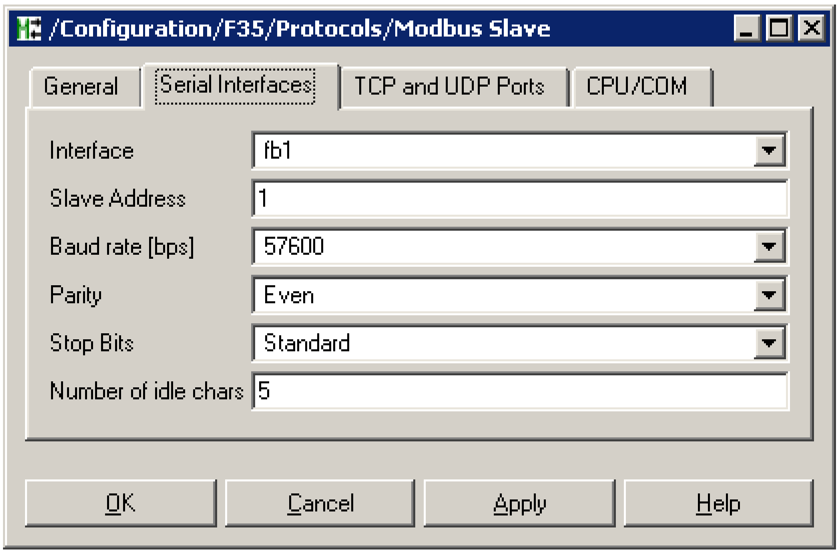
Figure 37: Dialog Box „General“

Note: For the *HIMatrix* control devices (COM OS' versions V.3 to V.6), the following read areas are available (for function codes 1 and 3):

COM OS V.3:	Import area
COM OS V.4:	Export area (compatible with H51q)
COM OS V.6:	configurable

Step 2: Configuring the Modbus slave's serial interface:

- Select the tab „Serial interfaces“
- Enter the parameters displayed in Figure 38.



/Configuration/F35/Protocols/Modbus Slave

Interface: fb1
Slave Address: 1
Baud rate [bps]: 57600
Parity: Even
Stop Bits: Standard
Number of idle chars: 5

Figure 38: Tab „Serial Interfaces“

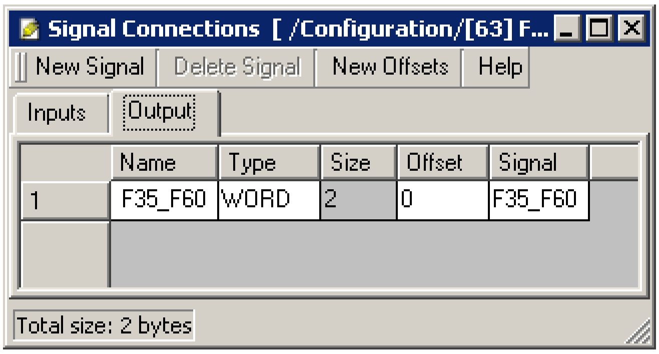
Step 3: Creating signals in the Modbus slave's signal editor:

- Open the signal editor located in the hardware management window by selecting *Signals->Editor*.
- Create the signal „F35_F60“ (type WORD). This signal is used as an output signal of the Modbus slave.
- Create the signal „F60_F35“(type WORD). This signal is used as an input signal of the Modbus slave.

Note: If the Modbus master was created in the same configuration, the signals are already available in the signal editor and may also be used for the Modbus slave.

Step 4: Connecting the Modbus slave’s output signal:

- Open the dialog box „Signals Connections“ by selecting *Connect signals* from the Modbus slave’s context menu.
- Select the tab „Output“ located in the dialog box “Signals Connections “.
- Open the signal editor by selecting *Signals -> Editor* from the menu bar.
- In the signal editor, click the „Name“ corresponding to the signal „F35_F60“. Drag and drop the name to the output signal in the tab "Output" located in the dialog box "Signal Connections".
- Click the button *New Offsets* located in the dialog box „Signal Connections“.
- In the pop-up window „Renumber Offsets“, click the button *Renumber*.
- Close the dialog box.



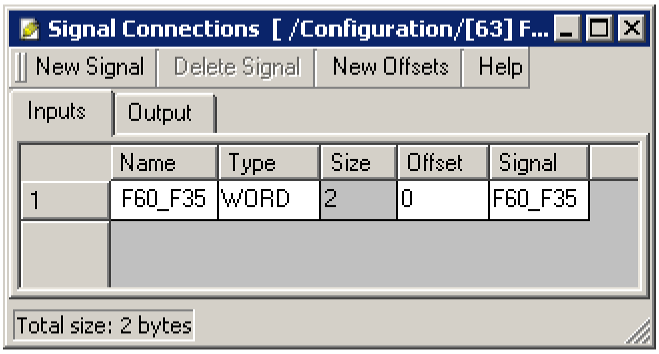
	Name	Type	Size	Offset	Signal
1	F35_F60	WORD	2	0	F35_F60

Total size: 2 bytes

Figure 39: Tab „Outputs“ Located in the Dialog Box „Signal Connections“

Step 5: Connecting the Modbus slave’s input signal:

- Open the dialog Box „Signals Connections“ by selecting *Connect signals* from the Modbus slave’s context menu.
- Select the tab „Input“ located in the dialog box “Signals Connections“.
- Open the signal editor by selecting *Signals -> Editor* from the menu bar.
- In the signal editor, click the „Name“ corresponding to the signal „F60_F35“. Drag and drop the name to the input signal in the tab "Input" located in the dialog box "Signal Connections".
- Click on the button *New Offsets* located in the dialog box „Signal Connections“.
- In the pop-up window „Renumber Offsets“, click the button *Renumber*.
- Close the dialog box.



	Name	Type	Size	Offset	Signal
1	F60_F35	WORD	2	0	F60_F35

Total size: 2 bytes

Figure 40: Tab „Inputs“ Located in the Dialog Box „Signal Connections“

Step 6: Loading the code into the resource:

- Start the code generator specific to the resource.
- Make sure that the code has been generated correctly (see the “fault status field”).
- Load the code into the resource.

HIMA
...the safe decision.

HIMA Paul Hildebrandt GmbH
Industrial Automation
Postfach 1261 • D - 68777 Bruehl
Phone: (+49) 6202 709-0 • Fax: (+49) 6202 709-107
E-mail: info@hima.com • Internet: www.hima.com