Protocol Manual

Modbus Slave V2

Contact

HIMA Paul Hildebrandt GmbH

P.O. Box 1261

68777 Brühl

Phone: +49 6202 709-0

Fax: +49 6202 709-107

E-mail: info@hima.com

Document designation	Description
HI 801 474 D, Rev. 11.00 (1922)	German original document
HI 801 475 E, Rev. 11.00.00 (1926)	English translation of the German original document

Table of Contents

1 Introduction

The Modbus slave V2 manual describes the properties of the Modbus slave V2 protocol and its configuration in SILworX for the safety-related HIMax and HIMatrix controller systems.

Knowledge of regulations and the proper technical implementation of the instructions detailed in this manual performed by qualified personnel are prerequisites for planning, engineering, programming, installing, starting up, operating and maintaining the HIMA controllers.

HIMA shall not be held liable for severe personal injuries, damage to property or the environment caused by any of the following: unqualified personnel working on or with the devices, de-activation or bypassing of safety functions, or failure to comply with the instructions detailed in this manual (resulting in faults or impaired safety functionality).

HIMA automation devices have been developed, manufactured and tested in compliance with the pertinent safety standards and regulations. They may only be used for the intended applications under the specified environmental requirements.

1.1 Structure and Use of This Manual

The manual contains the following chapters:

- Introduction
- Safety
- Product description
- Configuring the Modbus slave V2 protocol in SILworX

Additionally, the following documents must be taken into account:

Name	Content	Document no.
HIMax system manual	Hardware description HIMax system	HI 801 001 E
HIMax safety manual	Safety function HIMax systems	HI 801 003 E
HIMatrix safety manual	Safety function HIMatrix systems	HI 800 023 E
HIMatrix compact system manual	Hardware description HIMatrix compact system	HI 800 141 E
HIMatrix modular system manual	Hardware description HIMatrix modular F 60 system	HI 800 191 E
HIQuad X system manual	Hardware description HIQuad X system	HI 803 211 E
HIQuad X safety manual	Safety function HIQuad X system	HI 803 209 E
Automation security manual	Description of automation security aspects related to the HIMA systems	HI 801 373 E
SILworX first steps manual	Introduction to SILworX.	HI 801 103 E

Table 1: Additional Applicable Manuals

All the current manuals can be obtained upon request by sending an e-mail to: documentation@hima.com. The documentation is available for registered HIMA customers in the download area https://www.hima.com/en/downloads/.

1.2 Target Audience

This document is aimed at the planners, design engineers, programmers and the persons authorized to start up, operate and maintain the automation systems. Specialized knowledge of safety-related automation systems is required.

1.3 Writing Conventions

To ensure improved readability and comprehensibility, the following writing conventions are used in this document:

Bold	To highlight important parts. Names of buttons, menu functions and tabs that can be clicked and used in the programming tool.
Italics	Parameters and system variables, references.
`Courier`	Literal user inputs.
RUN	Operating states are designated by capitals.
Chapter 1.2.3	Cross-references are hyperlinks even if they are not specially marked. In the electronic document (PDF): When the mouse pointer hovers over a hyperlink, it changes its shape. Click the hyperlink to jump to the corresponding position.

Safety notices and operating tips are specially marked.

1.3.1 Safety Notices

Safety notices must be strictly observed to ensure the lowest possible risk.

The safety notices are represented as described below.

- Signal word: warning, caution, notice.
- Type and source of risk.
- Consequences arising from non-observance.
- Risk prevention.

The signal words have the following meanings:

- Warning indicates hazardous situations which, if not avoided, could result in death or serious injury.
- Caution indicates hazardous situation which, if not avoided, could result in minor or moderate injury.
- Notice indicates a hazardous situation which, if not avoided, could result in property damage.

⚠ SIGNAL WORD

Type and source of risk!

Consequences arising from non-observance.

Risk prevention.

NOTICE

Type and source of damage!

Damage prevention.

1.3.2 Operating Tips

Additional information is structured as presented in the following example:

i The text giving additional information is located here.

Useful tips and tricks appear as follows:

TIP The tip text is located here.

1.4 Safety Lifecycle Services

HIMA provides support throughout all the phases of the plant's safety lifecycle, from planning and engineering through commissioning to maintenance of safety and security.

HIMA's technical support experts are available for providing information and answering questions about our products, functional safety and automation security.

To achieve the qualification required by the safety standards, HIMA offers product or customer-specific seminars at HIMA's training center or on site at the customer's premises. The current seminar program for functional safety, automation security and HIMA products can be found on HIMA's website.

Safety Lifecycle Services:	
Onsite+ / On-Site Engineering	In close cooperation with the customer, HIMA performs changes or extensions on site.
Startup+ / Preventive Maintenance	HIMA is responsible for planning and executing preventive maintenance measures. Maintenance actions are carried out in accordance with the manufacturer's specifications and are documented for the customer.
Lifecycle+ / Lifecycle Management	As part of its lifecycle management processes, HIMA analyzes the current status of all installed systems and develops specific recommendations for maintenance, upgrading and migration.
Hotline+ / 24 h Hotline	HIMA's safety engineers are available by telephone around the clock to help solve problems.
Standby+ / 24 h Call-Out Service	Faults that cannot be resolved over the phone are processed by HIMA's specialists within the time frame specified in the contract.
Logistics+/ 24 h Spare Parts Service	HIMA maintains an inventory of necessary spare parts and guarantees quick, long-term availability.

Contact details:	
Safety Lifecycle Services	https://www.hima.com/en/about-hima/contacts-worldwide/
Technical Support	https://www.hima.com/en/products-services/support/
Seminar Program	https://www.hima.com/en/products-services/seminars//

2 Safety

All safety information, notes and instructions specified in this document must be strictly observed. The product may only be used if all guidelines and safety instructions are adhered to.

The product is operated with SELV or PELV. No imminent risk results from the product itself. Use in the Ex zone is only permitted if additional measures are taken.

2.1 Intended Use

To use the HIMA controllers, all pertinent requirements must be met, see relevant manuals in Table 1.

2.2 Residual Risk

No imminent risk results from a HIMA system itself.

Residual risk may result from:

- Faults related to engineering.
- Faults in the user program.
- Faults related to the wiring.

2.3 Safety Precautions

Observe all local safety requirements and use the protective equipment required on site.

2.4 Emergency Information

A HIMA system is a part of the safety equipment of an overall system. If the controller fails, the system enters the safe state.

In case of emergency, no action that may prevent the HIMA system from operating safely is permitted.

2.5 Automation Security for HIMA Systems

Industrial controllers must be protected against IT-specific problem sources. Those problem sources are:

- Attackers inside and outside of the customer's plant.
- Operating failures.
- Software failures.

All requirements for protection against manipulation specified in the safety and application standards must be met. The operator is responsible for authorizing personnel and implementing the required protective actions.

⚠ WARNING

Physical injury possible due to unauthorized manipulation of the controller!

Protect the controller against unauthorized access!

For example:

- **Changing the default settings for login and password!**
- **Controlling the physical access to the controller and PADT!**

Careful planning should identify the measures to implement. The required measures are to be implemented after the risk analysis is completed. Such measures can include:

- Meaningful allocation of user groups.
- Maintained network maps help to ensure that secure networks are permanently separated from public networks and, if required, only a well-defined connection exists (e.g., via a firewall or a DMZ).
- Use of appropriate passwords.

A periodical review of the security measures is recommended, e.g., every year.

The user is responsible for implementing the necessary measures in a way suitable for the plant!

For further details, refer to the HIMA automation security manual (HI 801 373 E).

3 Product Description

The Modbus slave V2 protocol is not based on a new Modbus specification, but is an enhanced HIMA variant concerning the internal processing of the protocol data on HIMA controllers. The standard Modbus function codes remain the same, the HIMA specific Modbus function codes no longer exist.

The Modbus slave V2 protocol enables HIMA controllers to exchange process data with third-party systems via the RS485 or Ethernet interfaces of the controllers. However, Modbus slave V2 must not be used for safety-related communication.

3.1 Equipment and System Requirements

The Ethernet interfaces on the processor module may not be used for Modbus TCP.

Element	Description
Controller and operating system	HIMax CPU operating system ≥ V11 COM operating system ≥ V11
	HIQuad X CPU operating system ≥ V11 COM operating system ≥ V11
	HIMatrix F CPU operating system ≥ V13 COM operating system ≥ V18
Programming tool	SILworX ≥ V11
Activation	Software activation code required, see Chapter 6.1

Table 2: System Requirements for Operating Modbus Slave V2

3.2 System Quantities for Modbus Slave V2

The HIMA system supports the following properties for Modbus slave V2.

Properties	Description
Max. number of Modbus slave V2 protocols	HIMax: A maximum of 20 Modbus slave V2 protocols, 1 for each X-COM 01 module. HIQuad X: A maximum of 20 Modbus slave V2 protocols, 1 for each F-COM 01 module. HIMatrix: A maximum of 1 Modbus slave V2 protocol.
Number of Modbus V2 master accesses	RTU: Due to the RS485 transmission technology, only one Modbus master can have access to an installed RS485 interface. TCP: A maximum of 20 Modbus masters can have access to the slave.
Maximum number of TCP Connections	A maximum of 20 TCP connections can be configured. The *Number of TCP Connections* parameter must be configured for each Modbus master. In total, a maximum of 20 TCP connections to the configured Modbus masters are permitted.
Maximum number of data views	A maximum of 20 data views per COM module.
Maximum input process data volume of a data view	8192 BOOLs or 4096 registers. With respect to the calculation of the maximum amount of process data available, a BOOL variable must always be calculated with 1 byte since it is processed in unpacked form. This applies irrespective of whether a BOOL variable is transferred via Modbus as a packed bit or as an unpacked byte.

Properties	Description
Maximum process data volume for each direction and controller via all the data views	HIMax: Send 128 kB, receive 128 kB. HIQuad X: Send 64 kB, receive 64 kB HIMatrix: Send 64 kB, receive 64 kB
Maximum number of request telegrams for each COM module	For each COM module, the Modbus slave can simultaneously process and respond to up to 64 request telegrams from Modbus masters (receive buffer). Additional request telegrams are rejected.

Table 3: System Properties for Modbus Slave V2

3.3 Differences Between Modbus Slave Set and Modbus Slave V2

Property	Modbus slave set	Modbus slave V2
Write request	Global variables are sent to the processor module in each system cycle.	After a write request, global variables are written once and sent to the processor module, see Chapter 3.5.
Write response	The write response is sent to the Modbus master directly after the process values have been saved on the process module.	The write response is only sent to the Modbus master after the process values have been saved, processed and acknowledged by the process module.
Read response	Modbus slave response telegram to a read request. In a redundant configuration, each of the two protocol variants sends the oldest process data values of the redundant partners.	
Reload	If the protocol has changed due to a reload, a cold reload is performed on the COM module, i.e., the module is temporarily stopped.	Reloadable, see Table 6.
Function code	Standard and HIMA specific	Standard
Master identification	None	The system determines the identity of a master by using its external transport path (fieldbus interface or IP address).

Table 4: Differences Between Modbus Slave Set and Modbus Slave Variante2

3.4 General User Requirements

During start-up, the user must ensure that the assignment of the slave variables transported via Modbus slave V2 to the master variables is implemented as requested. It is not possible to simultaneously operate Modbus slave set and Modbus slave V2 on one COM module. This is already rejected by the PADT during code generation.

i If the Ethernet interfaces are used as transmission channel, the HIMA controller and the communication partner must be located in the same subnet, or, if a router is used, they must have the corresponding routing settings.

3.5 Concept of Data Access to Global Variables

In Modbus slave V2, write access to a global variable is initiated by a write request and is performed once or cyclically by a user program.

The global variables are updated as soon as a corresponding write request is received via the Modbus slave V2 protocol. In this process, the last writing action (performed by a Modbus source or user program) determines the value of the global variables. This behavior can also be used, e.g., to structure redundancy, see Chapter 4.

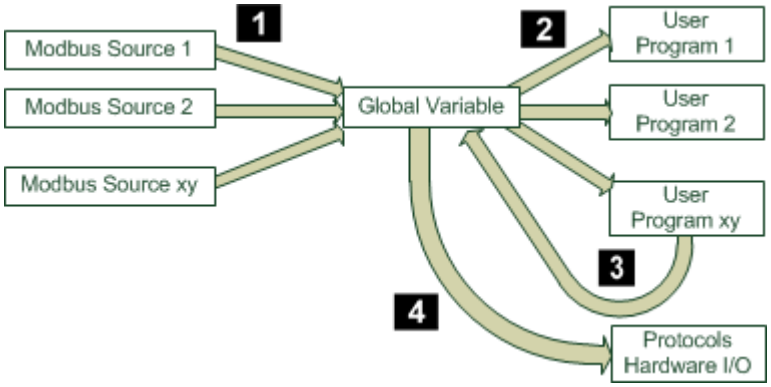


Figure 1: Concept of Data Access to Global Variables

Figure 1 shows the temporal sequence of the various possible access types to individual global variables within a system cycle when the Modbus slave V2 protocol is used. In the version comparison, these global variables are also referred to as global multisource variables, see Chapter 10.2.

1 One or multiple sources can write to the global variables, depending on whether a request for the corresponding Modbus source is pending or not. The value of the global variables is determined by the last Modbus source that has written to it.

2 One or multiple user programs can read the global variables to further process them in the corresponding program logic. Reading does not necessarily occur in every system cycle, if the user program is executed over multiple system cycles.

3 Just 1 or no user program can write to the global variables after processing of the program logic. Writing does not necessarily occur in every system cycle, if the user program is executed over multiple system cycles.

4 Different consumers, such as protocols or I/O modules, can read the global variables to further process them, i.e., usually to output them externally. Reading occurs in every system cycle.

i **Global multisource variables must not be used for the safety functions of the user program.**

3.5.1 Input Variables Written to by the User Program

If an input variable is not only written to by Modbus sources, but also by a user program, it is reset to its initial value once as soon as the user program enters one of the following states:

- STOP
- ERROR STOP
- Cold start of the user program

3.5.2 Write Request Notes

- If write requests from multiple Modbus sources are sent within one controller's cycle, the processing order of the write requests is unpredictable. Nor can this processing be derived from the sequence of the write requests. The value of the last write request becomes effective in the user program.
- In principle, there is no guarantee that a specific Modbus master will reliably read back the value that it had previously written. Unless that Modbus master is the only one writing to that address range.
- If a user program is not processed within one cycle (e.g., due to a corresponding priority configuration of the user program or test mode), there is no guarantee that the evaluation and feedback of a global variable will be effective in the user program within the same cycle. As a result, the value of a global variable, which was written to based on a write request, is retained (and thus is not overwritten by the user program) and becomes effective via the output data.
- The value of a global variable assigned based on a write request and also written to by the user program, may have external effects if the user program is stopped.
- During a reload, the response time for a write request can temporarily increase.
- The minimum response time to a write request (demand) is 2 CPU cycles. To obtain the lowest possible response times, *Schedule Time Slice Async* must be set sufficiently high to ensure that the process data can be processed within 1 CPU cycle.
- Processing of write requests may cause occasional extension of the CPU cycle times, which must be taken into account when determining the *Schedule Time Slice Async* and the CPU watchdog time. The CPU cycle is increased by a maximum of 1 ms per 1 kByte of process data. The maximum increase of the CPU cycle is 20 ms, i.e., no additional increase of the CPU cycle time is to be expected with more than 20 kByte process data.

3.5.3 Data View

Global variables are assigned to a data view to allow access to Modbus sources. The data views define address areas (Modbus offsets) on the system, see Chapter 7.2.1. Write access to the input variables of a data view is initiated through a corresponding write request from the Modbus source.

3.5.4 Data View Link

In Modbus slave V2, data view links are created as child elements of the master object. A data view can be assigned to one or multiple master objects via data view links.

3.5.5 Behavior Upon Loss of Data View Connection

The behavior when the connection between a master and a data view is lost, is configured in the corresponding data view of the Modbus slave. The behavior can be set to *Adopt Initial Values* or *Retain Last Value*. The selected behavior applies to all the input variables of the corresponding data view.

i If global variables are written to by multiple sources, only the option *Retain Last Value* can be used in the data view, otherwise the code generation is rejected.

3.5.6 Retain Behavior of Global Variables

The following requirements must be met to store global variables that were written to through a write request, in the non-volatile memory:

- The global variables must be both read and written to within the user program.
- The Retain property must be set for the global variables.

A demand variable that is declared as retain and written to from a user program, will adopt the retain value during a warm start of the user program.

4 Redundancy

The use of data views simplifies the redundant configuration of the Modbus slave V2 protocol. This multiple writing from different Modbus sources may be used to create redundancy. A slave's global variable can be written to through data views from all the Modbus sources. Writing can only be initiated via a write request from a Modbus source. The most recent write request determines the value of the process data, as described in Chapter 3.5.

4.1 Logical Redundancy

The system supports the user while configuring a logical redundancy by allowing several Modbus sources to write to a global variable, see Chapter 3.5.

The access to global variables occurs via data views that define an address area ((Modbus offset) on the corresponding communication module, where the different global variables can be addressed. Such a data view can also be used by multiple masters. These can use separated transport paths, but they must communicate via the same communication module.

i The global variable can be written to by all the masters (1…xy) such that, depending on the application, all the masters can be considered as redundant.

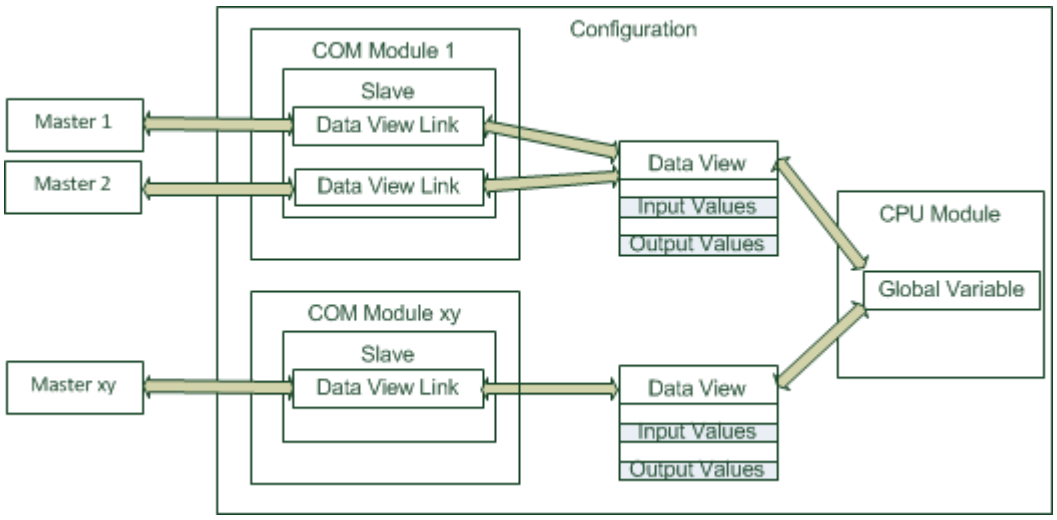


Figure 2: Redundant Master and Mono Slave (HIMax and HIMatrix F*03 Controller)

4.2 COM Redundancy of two Communication Modules

The example in Figure 3 shows a redundant connection to two Modbus slave V2 on two communication modules via two continuously separate transport paths.

In Modbus slave V2, the COM redundancy of two communication modules is automatically supported. Once two Modbus slaves have been created in Modbus slave set V2, the two communication modules cooperate in processing all the write and read accesses.

For pure write accesses to global variables, this has few differences (in terms of impact and thus redundancy of the masters) with the logical redundancy described in Chapter 4.2.

This kind of redundancy, however, has advantages for read access to global variables:

- Read accesses are coordinated by both communication modules such that the temporal sequence of the data read via the two communication modules is consistent. The data from both communication modules can thus be related to one another and, e.g., if one of the communication modules fails, the data of the remaining communication module can seamlessly be used, without fearing inconsistent temporal progression of the read process data.
  This refers to the process of reading back the data from the slave's import area.
- This behavior during Read accessenables redundancy operation of two communication modules and ensures seamless communication, even if one of the communication modules fails.

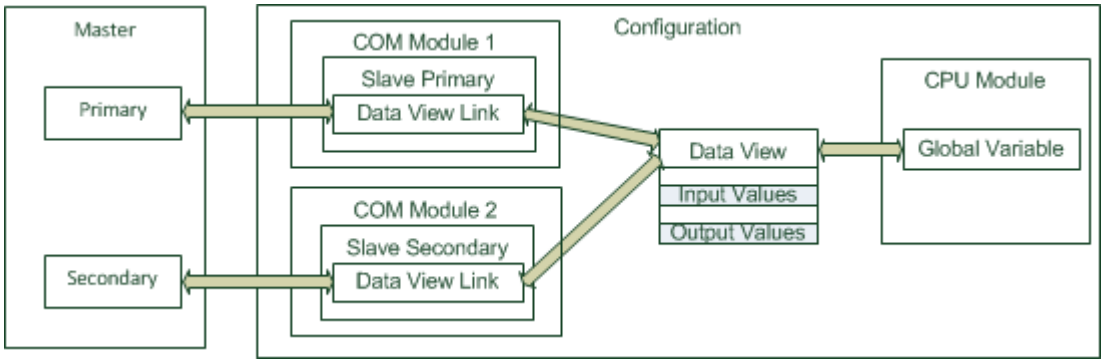


Figure 3: Redundant Master and Redundant Slave (HIMax Controller with two COM Modules)

4.2.1 Support for Configuring Redundant Communication Modules

As soon as two Modbus slaves have been created in Modbus slave V2, the two communication modules cooperate and automatically synchronize the common data views.

To provide support for error-free configuration of redundancy, it is possible to define in SILworX which Modbus masters should operate redundantly in the two communication modules, and thus to enable their consistency check.

Two 2 levels of consistency check exist:

1. All relevant properties are identical, except for the master's transport path
   In **Modbus Slave Set V2**, select **Modbus Slave, Master** and activate *Explicit Check of Redundant Master* in the properties.
2. All relevant properties are identical, including the master's transport path
   In **Modbus Slave Set V2**, select **Modbus Slave, Master** and activate *Check Redundant Transport Path for Identical Setting* in the properties.

4.2.1.1 Configuration Restriction (Identical Masters)

- The system uses the external transport path (fieldbus interface or IP address) of a master to determine its identity.
- If two masters on the two redundantly operated communication modules are to be considered identical to respect to the configuration, the master's properties must also be identical.

4.2.1.2 Combining Redundancies

The redundant accesses to global variables described in this chapter can be freely combined. This means that global variables that are used by two redundant communication modules can also be accessed by other communication modules and 2 redundant communication modules can also be configured differently with respect to data views and masters.

4.3 Notes for Configuring the Redundancy of two Communication Modules in a HIMax System

4.3.1 Rules for Redundant HIMax Modbus Slave V2

To redundantly operate the HIMax Modbus slave communication modules, HIMA recommends a redundant configuration of the HIMax system, refer to the system manual (HI 801 001 E) for more details. The consistent behavior of the Modbus slave communication module pairs towards their external partner (Modbus master) cannot otherwise be ensured as soon as a first fault occurs in the HIMax system.

4.3.2 Slots Allowed for the Redundant HIMax Modbus Slave V2 COM Modules

The system bus segments (1...3) on the base plate must be taken into account to minimize the risk of collisions on the HIMax system bus. For this reason, the redundant Modbus slave communication modules should only be inserted in the same segment of a base plate in the following slots:

Segment	Slot
1	3…6 (as long as no processor module has been planned)
2	7…14
3	15…18

Table 5: Slots Allowed for the Redundant Modbus Slave COM Modules

4.3.3 Redundant Modbus Slave COM Modules in Different Base Plates

No more than two redundant Modbus slave communication module pairs may be operated if their redundant Modbus slave communication modules are located in different base plates (0...15).
In such cases, the redundant Modbus slave communication modules may only be located in adjacent base plates.

Furthermore, 8 additional Modbus slave communication module pairs may be operated on the same HIMax system in accordance with the rules specified in Chapter 4.3.2.

NOTICE

System malfunction possible!

Only use slots for redundant Modbus slave communication modules in compliance with the stated rules!

Between an X-COM module and the X-CPU modules, a maximum additional delay of 50 µs is allowed (due to cable length, switches, etc).

Recommendation: Operate the X-COM modules as close as possible to the X-CPU modules (e.g., rack 0, rack 1).

5 Reload

The use of reload for changing the resource configuration must be agreed upon with the competent test authority! For further details on reload, refer to the safety manual of the corresponding system family.

5.1 Changes to the Modbus Slave V2 Configuration

Adding a Modbus slave V2 via reload requires that either no computing time budget was configured, or that sufficient computing time budget reserve exists throughout all protocols. The computing time budget in other standard protocols can only be changed through cold reload.

The following table provides an overview of the changes to the Modbus slave V2 configuration and their effects on the Modbus slave V2 reload.

Changes to	HIMax X-COM 01	HIQuad X F-COM 01
Adding or deleting global variables from data views	•	•
Adding or deleting data views	•	•
Data type of a global variable	-	-
Adding or deleting assigned Modbus masters	•[1)]	•[1)]
Master configuration of assigned Modbus master	•[1)]	•[1)]
Adding or deleting a Modbus slave V2 protocol	•	•
Budgeting of the computing time for Modbus slave V2	•	•
TCP settings		
Changing the IP address	•	•
Changing the port	•[2)]	•[2)]
Adding connections	•	•
Deleting connections	•[3)]	•[3)]

- • Modbus slave V2 reload is possible
- \- Modbus slave V2 reload is not possible

n.a.: not applicable

1) If the reload changes the master's identity (IP address or fieldbus interface), all system variables assigned to the master are reset to the initial values.

2) If the ports are changed, the reload is performed, but all connections to the Modbus master are aborted. This is indicated via a relaod warning.

3) If the number of TCP connections is reduced and the number of connections actually used is greater than that number in the reload configuration, all connections to the Modbus master are closed. This is indicated via a relaod warning. Irrespective of this, SILworX issues a warning during code generation if the number of TCP connections in the configuration is reduced.
If another connection is established during reload, the firmware may not issue a reload warning, but the connections may nevertheless be closed.

Table 6: Modbus Slave V2 Reload after Changes

6 Creating the Modbus Slave V2 in SILworX

6.1 Registering and Activating the Protocol

The software activation code with the required liceses is generated on the HIMA website using the system ID of the controller (e.g., 60000). Follow the instructions provided on the HIMA website
www.hima.com-> Products & Services -> Product Registration-> Options SILworX .

i The license is intrinsically bound to the system ID. A license can only be used once for a specific system ID. For this reason, only activate the code when the system ID has been uniquely defined.

A software activation code may include a maximum of 32 licenses. It is also possible to specify multiple activation codes in the license management. A maximum of 64 licenses may be loaded into one controller.

To enter the software activation code in SILworX

1. In the structure tree, select **Configuration, Resource, License Management**.
2. Right-click **License Management** and select **New, License Ke**y from the context menu.
   ☑ A new license key is created.
3. Right-click the **License Key** and select **Properties** from the context menu.
4. Enter the new software activation code in the **Activation Code** field.

6.2 Creating the Modbus Slave Set V2 Protocol

The Modbus slave set V2 is created within the Protocols directory of a resource and serves for configuring the Modbus slave V2 protocol.

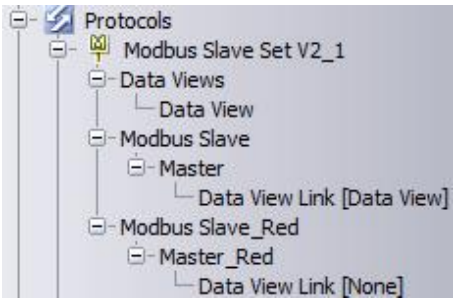


Figure 4: Modbus Slave Set V2 in the SILworX Structure Tree

To created a Modbus slave set V2

1. In the structure tree, select **Configuration, Resource, Protocols**.
2. Select New, Modbus Slave Set V2 from the context menu of Protocols to add a new Modbus slave set V2.
3. Select Edit from the context menu of **Modbus Slave V2** to open the Modbus Slave Editor. The editor includes the tabs **Set Objects** and **Properties** see Chapter 7.1.
4. In the Overview Editor *for the Modbus Slave Set V2, select the* **Properties** *tab.*
   - **Enter the Modbus slave name.**

   The remaining properties may initially retain the default settings.

6.3 Creating a Data View

1. In the structure tree, select **Configuration**, **Resource**, **Protocols**, **Modbus Slave Set V2**, **Data Views**.
2. Select **New**, **Data View** from the context menu of *Data Views* to add a new data view.
3. Select Edit from the context menu of *Data View* to open the *Data View Connection Editor*. The editor includes the tabs **Register Variables, Bit Variables** and **Properties**.
4. In the Data View Connection Editor, **select the Register Variables** tab.
   - Register Inputs
     To assign the receive variables, select the suitable global variables in the Object Panel and drag them into the Register Inputs field.
   - Register Outputs
     To assign the send variables, select the suitable global variables in the Object Panel and drag them into the Register Outputs field.
5. In the Data View Connection Editor, *select the* **Bit Variables** tab.
   - Bit Inputs
     To assign the receive variables, select the suitable global variables in the Object Panel and drag them into the Bit Inputs field.
   - Bit Outputs
     To assign the send variables, select the suitable global variables in the Object Panel and drag them into the **Bit Outputs** field.
6. In the Data View Connection Editor, *select the* **Properties** tab.

The properties may retain the default settings.

6.4 Configuring the Modbus Slave

1. In the structure tree, select **Configuration, Resource, Protocols, Modbus Slave Set V2.**
2. Select New, Modbus Slave from the context menu of *Modbus Slave Set V2* to add a new Modbus slave.
3. Select Edit from the context menu of **Modbus Slave** to open the Modbus Slave Editor. The editor includes the tabs **Master Objects**, **System Variables** and **Properties**.
4. In the Modbus Slave Editor*, select the* **Properties** *tab.*
   - Select the **COM module** on which the Modbus slave should run.
   - The remaining properties may initially retain the default settings.

6.4.1 Creating the Connection to the Modbus Master in the Modbus Slave

1. Select New*,* Modbus Master from the context menu of *Modbus Slave* to add a new Modbus master.
2. Select Edit from the context menu of **Modbus Master** to open the Modbus Master Editor. The editor includes the tabs **Data View Links, System Variables** and **Properties**.
3. In the Modbus Slave Editor*, select the* **Properties** *tab.*
   - **Enter the master IP address with which this Modbus slave should be connected**.

   The remaining properties may initially retain the default settings.

6.4.1.1 Referencing to the Data View Link with Data View

The data view link object is a child element of the master object.

1. Select Edit *from the context menu of* **Data View Link** *to open the Data View Link Editor.* The editor includes the tabs **System Variables** and **Properties**.
2. In the Data View Link Editor*,* select the *Properties* tab.
   - From the **Reference to Data View Object** drop-down menu, select the data view (e.g., the data view that was created in Chapter 6.3) that should be connected to the Modbus master.

   The remaining properties may initially retain the default settings.

6.5 Configuring the Redundant Modbus Slave

The COM redundancy of two communication modules is automatically supported in the Modbus slave set V2. As soon as two Modbus slaves have been created in the Modbus slave set V2, the two communication modules cooperate in processing all the write and read accesses, if they have access to the same data view (e.g., the data view created in Chapter 6.3).

The COM redundancy of two communication modules is automatically supported in the Modbus slave set V2, see Chapter 4.2.

7 Description of the Modbus Slave V2 Editors and Objects

This section describes the characteristics of the HIMA Modbus slave V2 as well as the SILworX menu functions and dialog boxes required for configuring the HIMA Modbus slave V2.

7.1 Modbus Slave Set V2

The Modbus slave set V2 object is created in the *Protocols* directory and serves for configuring the Modbus slave V2 protocol.
Select Edit from the context menu of Modbus Slave Set V2 to open the Modbus Slave Set V2 Editor. The dialog box contains the following tabs:

- The Set Objects Tab
- The Properties tab

7.1.1 The Set Objects Tab

The Set Objects tab provides the list of all the Modbus slave set V2 objects that are created in the Modbus slave set V2.

Set objects	Description
Data Views	The Data Views object contains all the data views that have been created for the Modbus slave V2.
Modbus Slave	Used to configure the properties of the Modbus slave and the connections to the master(s).

Table 7: Set Objects of a Modbus Slave Set V2

7.1.2 The Properties Tab

The Properties tab contains the parameters to set up the Modbus slave set V2.

Properties	Description
Type	Modbus slave set V2
Name	Changeable Default: Modbus slave set V2
Max. µP Budget in [%]	Maximum µP budget of the module that can be used for processing the protocols. ▪ No limitation ▪ Range of values: 1…100% Default value: 30%
Warning if µP budget exceeded, in [%]	Whenever this warning threshold is exceeded, a warning message is issued. ▪ Do not use ▪ Range of values: 1…100% Default value: 80%

Table 8: Modbus Slave Set V2 Properties

7.2 Data Views

The Data Views object provides the list of all the data view objects that are created in the Modbus slave set V2.

7.2.1 Data View

The process variables for the corresponding communication connection are logically assigned within the Data View Editor.

Select Edit from the context menu of **Data View** to open the Data View Editor. The dialog box contains the following tabs:

- The Register Variables tab.
- The Bit Variables tab.
- The Properties tab.

7.2.1.1 The Register Variables Tab

The global variables in the **Register** area are created in the **Register Variables** tab.

The global variables in the **Register** area can only be accessed using Modbus function codes 3, 4, 6, 16, 23.

7.2.1.2 The Bit Variables Tab

The global variables in the **Bit** area are created in the *Bit Variables* tab.

The global variables in the **Bit** area can only be accessed using Modbus function codes 1, 2, 5, 15.

7.2.1.3 The Properties Tab

Properties	Description
Type	Data View
Name	Name of the data view, changeable. Standard: Data View
Behavior on Connection Loss	If the connection of the processor module to the communication module or master is lost, the input variables are initialized or used unchanged in the process module, depending on this parameter. If global variables are written to by multiple sources, i.e., from multiple data views, only the option *Retain Last Value* can be used in the data view, otherwise the code generation is rejected. Apply Initial Data — Input variables are reset to their initial values. Retain Last Value — The input variables retain the last value.

Table 9: Data View Properties

7.3 Modbus Slave

Select Edit from the context menu of Modbus Slave to open the Modbus Slave Editor. The dialog box contains the following tabs:

- The Master Objects Tab.
- The System Variables tab.
- The Properties tab.

7.3.1 The Master Objects Tab

The Master Objects tab lists all the master objects that are logically assigned to the Modbus slave.

7.3.2 The System Variables Tab

Properties	Data type	Description
Number of System Accesses by Non-Configured Masters	UDINT	Data View
IP Address of an Unconfigured Master	DWORD	The IP address of a not configured master that accessed the controller via a Modbus slave V2 protocol, is provided as system variable. Range of values: Valid unicast IP address Standard: 0
Redundant State	BYTE	The current redundancy state is provided for each COM module. Range of values: 0: The COM module is passive (P). 1: The COM module operates in exclusive mode (E). 2: The COM module operates in redundant mode (R). Default value: 0
Statistics Reset	BYTE	Reset the statistics.

Table 10: System Variables

7.3.3 The Properties Tab

Properties	Description
Type	Data View
Name	Changeable Default: Modbus Slave
Module	The COM module on which the Modbus slave is running.
TCP Port	TCP port for the TCP connections to the Modbus masters. Range of values: 1…65535 Standard: 502 0, if the Modbus slave does not communicate via TCP.
UDP Port	UDP port for the UDP connections to the Modbus masters. Range of values: 1…65535 Standard: 502 0, if the Modbus slave does not communicate via UDP.

Table 11: Modbus Slave Properties

7.4 Modbus Master

Select Edit from the context menu of Modbus Master to open the Modbus Master Editor. The dialog box contains the following tabs:

- The Data View Links tab.
- The System Variables tab.
- The Properties tab.

7.4.1 The Data View Links Tab

The Data View Links tab lists all the data view link objects that were created for the connection to the Modbus master object. A data view link references to a data view within which the global variables for input and output variables are created and can be adjusted, see Chapter 7.5.

7.4.2 The System Variables Tab

Properties	Data type	Description
Current Connection State of a Configured Master	BOOL	Current connection state TRUE Connected FALSE Not connected

Table 12: System Variables

7.4.3 The Properties Tab

Element (editor)	Description
Type	Master
Name	The name can be changed. Default value: Master
Byte Sequence	Possible settings: ▪ BE (Big Endian) ▪ BEB (Big Endian with Byte Swapping) ▪ LE (Little Endian) ▪ LEB (Little Endian with Byte Swapping) For further information, refer to Table 17. Default value: BE (Big Endian)
Read Area for Coil, Holding Register and Read/Write Multiple Register Access Count	Possible settings: ▪ Export Area ▪ Import Area Default value: Export Area
Maximum Response Delay [ms]	Possible settings: ▪ No Limitation ▪ Maximum Response Delay [ms] Default value: 5000
Transport path	Possible settings: ▪ TCP ▪ UDP ▪ RS485 Default value: TCP
Master IP Address	IP address of the TCP/UDP master.

Element (editor)	Description
Number of TCP Connections	A maximum of 20 TCP connections can be configured. The *Number of TCP Connections* parameter must be configured for each Modbus master. In total, a maximum of 20 TCP connections to the configured Modbus masters are permitted. The *Transport Path* parameter must be set to TCP, otherwise the input field is locked. Range of values: 1…20 Default value: 1
The following elements can only be edited if RS485 was selected as transport path.	
Interface	Fieldbus interface that should be used for the Modbus slave (fb1, fb2, fb3, fb4). Range of values: Depending on the COM module type.
Slave Address	Slave address for the fieldbus interface. Range of values: 1…247 Default value: 1
Baud Rate [bps]	Possible value for transfer rate for RS485: 300 bit/s 600 bit/s 1200 bit/s 2400 bit/s 4800 bit/s 9600 bit/s 19200 bit/s 38400 bit/s 57600 bit/s (maximum baud rate for HIMax) 62500 bit/s (HIMatrix) 76800 bit/s (HIMatrix) 115000 bit/s (HIMatrix)
Parity	Possible settings: ▪ None ▪ Odd ▪ Straight Default value: Even
Stop Bits	Standard (adapts the number of stop bits to the parity: with parity = 1 stop bit, no parity = 2 stop bits). Possible settings: ▪ One stop bit ▪ Two stop bits Default value: Default
Number of Idle Chars	Number of idle characters at the start and the end of a RTU telegram frame. Range of values: HIMax: 0…65535 HIMatrix F*03: 0…93 Default value: 5 characters For HIMax with RS485 operation, no fewer than 4 idle characters should be configured since the hardware idles the RTS signal for 3.1 character lengths after the last low data bit is sent. In the RS485 module, the RTS signal prevents the slave from receiving its own data transmissions, i.e., the RS485 receiver is switched off for 3.1 character lengths after sending the last data bit.

Element (editor)	Description
Explicit Check of Redundant Master	The function to explicitly check redundant masters for identical configuration can be used to facilitate the consistency of redundant masters. TRUE The explicit check of redundant masters is active. FALSE No explicit check.
The names of the redundant masters differ from own name	If the *Explicit Check of Redundant Master* parameter is active, the Master object provides the property *The names of the redundant masters differ from own name*. TRUE The user must enter the name of the redundant master in the *Redundant Master* field. FALSE The name of the redundant master is the same as the own name.
Redundant Master	If the checkbox *The names of the redundant masters differ from own name* is set to TRUE, the *Redundant Master* field is active. The names entered for the redundant masters can be separated by commas. The button next to the field serves for starting a dialog box, in which the redundant masters of the slave can be selected. If a master is no longer a redundant master in the properties, the dialog box reports accordingly through a message and the selection is ignored.
Check Redundant Transport Path for Identical Setting	If the *Explicit Check of Redundant Master* parameter is active, the Master object provides the property *Check Redundant Transport Path for Identical Setting*. This checks whether the redundant master has the same transport path settings (baud rate, IP address, etc.).

Table 13: Properties of the Modbus Master

7.5 Data View Link

Select Edit from the context menu of **Data View Link** to open the Data View Link Editor. The dialog box contains the following tabs:

- The System Variables tab.
- The Properties tab.

i If multiple data view links were created for a master, the Modbus addresses must not overlap!

7.5.1 The System Variables Tab

Properties	Data type	Description
Connection Status Byte	Byte	The system provides a connection status byte for each connection between a master and a data view. The connection status byte has the following range of values: Number of configured data views as bit field 0-0x07 [BYTE]

Bit 0	Current connection state Master<-> Data View:
0	The Master <-> Data View are currently not connected.
1	The Master <-> Data View are currently connected or the connection is not monitored.
Bit 1	Historic connection state Master<-> Data View.
0	The monitoring time has never expired since the last reset or system reboot, or the master accesses to the current data view are not monitored.
1	The monitoring time for the data view has at least expired once since the last reset or system start.
Bit 2	Invalid master requests.
0	There was no invalid master request since the reboot or last reset.
1	There was at least one invalid master request.

Table 14: Data View Links System Variables

7.5.2 The Properties Tab

Properties	Description
Type	Data view link
Name	Data view link
Reference to Data View Object	The reference is the name of the data view that is connected to the current master via the data view link (selection among the available data views).
Master Monitoring Time [ms]	The master monitoring time is when the master had the last access to the data view through this link per read or write request, see Table 14. Possible settings: ▪ No Limitation ▪ Master monitoring time [ms] = 1…2147483647 Default value: No Limitation

Table 15: Data View Links Properties

8 Modbus Function Codes

8.1 Standard Modbus Function Codes

The HIMA Modbus slave V2 supports the following Modbus function codes:

Element	Code	Description
READ COILS	01	Reading of multiple coils from the configured read area[1)] of the Modbus slave. Maximum length of the process data for each Modbus telegram: 250 bytes (2000 coils).
READ DISCRETE INPUT	02	Reading of multiple discrete inputs from the export area of the Modbus slave. Maximum length of the process data for each Modbus telegram: 250 bytes (2000 coils).
READ HOLDING REGISTER	03	Reading of multiple registers from the configured read area[1)] of the Modbus slave. Maximum length of the process data for each Modbus telegram: 250 bytes (125 registers).
READ INPUT REGISTER	04	Reading of multiple registers from the export area of the Modbus slave. Maximum length of the process data for each Modbus telegram: 250 bytes.
WRITE SINGLE COIL	05	Writing of a single coil in the import area of the Modbus slave. Maximum length of the process data for each Modbus telegram: 1 byte.
WRITE SINGLE Register	06	Writing of a single register in the import area of the Modbus slave. Maximum length of the process data for each Modbus telegram: 2 bytes.
DIAGNOSIS	08	Only sub-code 0 (loop-back function) of FC08.
WRITE MULTIPLE COILS	15	Writing of multiple coils in the import area of the Modbus slave. Maximum length of the process data for each Modbus telegram: 246 bytes (1968 coils).
WRITE MULTIPLE REGISTER	16	FC 16 for writing multiple registers in the import area of the Modbus slave. Maximum length of the process data for each Modbus telegram: 246 bytes (123 registers).
READ WRITE MULTIPLE REGISTER	23	Reading of multiple registers from the configured read area[1)] of the Modbus slave. Writing of multiple registers in the import area of the Modbus slave. Maximum length of the process data for each Modbus telegram: 242 bytes (121 registers), from the Modbus master. 250 bytes (125 registers), to the Modbus master.
Read Device Identification	43	Transmits the slave identification data to the Modbus master.
[1)] Export or import area		

Table 16: Modbus Function Codes of the HIMA Modbus Slave

Function codes 01, 02, 05 and 15 support the BOOL data type. Function codes 03, 04, 16 and 23 support any data type of 2-bytes in length.

The start address of the first variable to be transferred and the number of registers/bits of the variables to be transferred must be entered for each request.

Error codes:

- If the master sends a telegram with unknown function codes, the controller responds with error code 1 (invalid code).
- If the telegram of the master does not match the Modbus slave configuration (the data layout and offsets of the linked data views are relevant), the slave returns error code 2 (invalid data).
- If the master sends a telegram with incorrect values (e.g., length fields), the slave responds with error code 3 (invalid value).

Communication only takes place if the COM module is in RUN. If the COM module is in any of the other operating states, no requests from the master is answered.

Note for Modbus Function: Read Device Identification (43)

The HIMA Modbus slave provides the identification data to the master and supports the following object IDs:

Basic:
0x00 VendorName “HIMA Paul Hildebrandt GmbH”
0x01 ProductCode “<Module serial number>”
0x02 MajorMinorRevision “<COM Vx.y CRC / COM Vx.y CRC>”

Regular:
0x03 VendorUrl “http://www.hima.com”
0x04 ProductName “HIMax /HIMatrix”
0x05 ModelName “ModulType”
0x06 UserApplicationName “--------[SRS]”

Extended:
0x80 blank “--------”
0x81 blank “--------”
0x82 blank “--------”
0x83 blank “--------”
0x84 blank “--------”
0x85 blank “---------”
0x86 CRC of the file modbus.config
(Configuration file of the Modbus slave protocol within the CPU file system. To compare with the information provided in SILworX, under Online/Version Comparison).

The following ReadDevice ID codes are supported:

(1) Read basic device identification (stream access)
(2) Read regular device identification (stream access)
(3) Read extended device identification (stream access)
(4) Read one specific identification object (individual access)

For further details on Modbus, refer to the *Modbus Application Protocol Specification* www.modbus.org)

8.2 Byte Order

The byte order describes the endian format of the incoming and outgoing process data.

The following table shows how the byte order influences the data stream of the Modbus V2 protocol if the data type does not begin or end at a register limit.

Tab	Data byte	Data type	Little endian [LE]	Big endian [BE]	Little endian with byte swapping [LEB]	Big endian with byte swapping [BEB]
0	0	Byte A	Byte A	Byte A	Byte A	Byte A
	1	DWord	DWord[0]LSB	DWord[3]MSB	DWord[1]	DWord[2]
1	2		DWord[1]	DWord[2]	DWord[0]LSB	DWord[3]MSB
	3		DWord[2]	DWord[1]	DWord[3]MSB	DWord[0]LSB
2	4		DWord[3]MSB	DWord[0]LSB	DWord[2]	DWord[1]
	5	Word	Word[0]LSB	Word[1]MSB	Word[1]MSB	Word[0]LSB
3	6		Word[1]MSB	Word[0]LSB	Word[0]LSB	Word[1]MSB
	7	Byte B	Byte B	Byte B	Byte B	Byte B

Table 17: Byte Order

9 Views and Diagnostics in the Control Panel

The Control Panel appears after a successful system login.

The *Modbus Slave* node collects all the details on the *Modbus V2 via Ethernet* protocol operation.

9.1 Modbus Online Pane

The online pane displays the Modbus connections of the *Modbus V2* master:

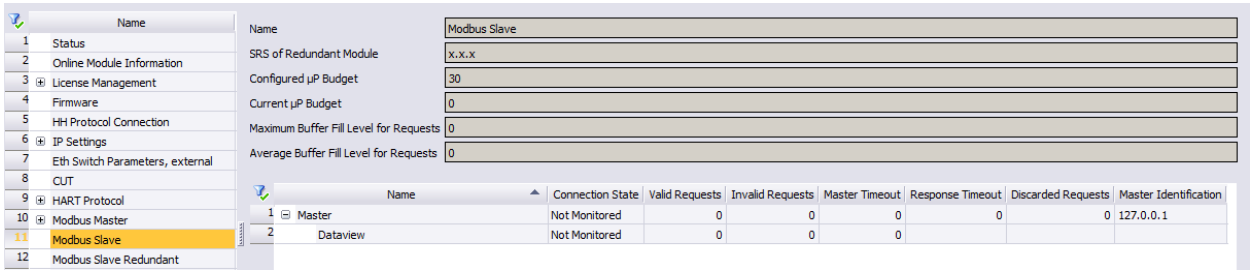


Figure 5: The Modbus Online Pane

The following table specifies all the parameters of the Modbus online view that are available for online diagnostics.

Modbus Slave (Online)	Description
Name	Modbus slave
SRS of Redundant Module	System.Rack.Slot
Configured µP Budget	See Table 8.
Current µP Budget	See Table 8.
Maximum Buffer Fill Level for Requests	Maximum request buffer fill level for a Modbus slave V2 protocol.
Average Buffer Fill Level for Requests	Average request buffer fill level for a Modbus slave V2 protocol.
Modbus Master (Online)	**Description**
Name	The name of the master.
Connection State	See Table 12.
Valid Requests	Number of valid requests that can be successfully processed and answered by the Modbus slave.
Invalid Requests	Number of invalid requests upon which the Modbus slave sends an error code (see Modbus exception codes).
Master Timeout	If the master request timeout is not 0, each connection Master <-> Data View must be checked to detect if a valid request from the configured master occurred within the timeout. Any Modbus function code is considered as a valid request. The fact that a master has multiple connections to multiple data views and that more than one connection to the data views are lost, is still reported as a single and not as multiple communication errors.
Response Timeout	Number of timed-out responses from the Modbus slave. This means that the response did not occur within the configured maximum response time and is therefore rejected. The counter is revolving.
Discarded Requests	Number of discarded requests from a configured master. The counter is revolving.

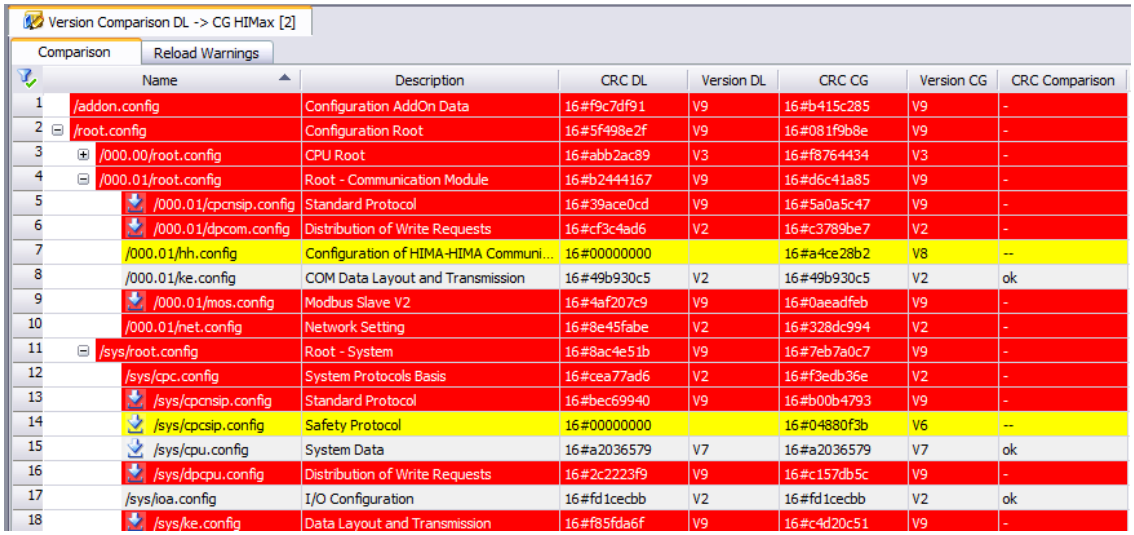
Master Identification	Provides the identification for each master configured on a communication module. ▪ The identification of Modbus UDP/TCP masters occurs through the IP address. ▪ The identification of the Modbus RTU master occurs through the RS485 interface identifier used to connect the corresponding communication module.
TCP Connections Currently Open	The number of currently open TCP connections is displayed for each configured master. Range of values: 0…20
Data View (Online)	Description
Name	Name of the data view.
Connection State	See Table 12.
Valid Requests	Parameters such as Modbus master (online), but related to the offset area of the data view.
Invalid Requests	
Master Timeout	

Table 18: Elements of the Modbus Online Pane

10 Version Comparison

The version comparison is based on the project checksums (CRCs) created by the code generator. For further information, refer to the version comparison manual (HI 801 286 E).

During a version comparison, different resource configurations are compared to one another and the differences between the individual configuration files are detected. The result of the version comparison has SIL 3 quality and is based on the configuration files describing the executable code.



	Name	Description	CRC DL	Version DL	CRC CG	Version CG	CRC Comparison
1	/addon.config	Configuration AddOn Data	16#f9c7df91	V9	16#b415c285	V9	-
2	/root.config	Configuration Root	16#5f498e2f	V9	16#081f9b8e	V9	-
3	/000.00/root.config	CPU Root	16#abb2ac89	V3	16#f8764434	V3	-
4	/000.01/root.config	Root - Communication Module	16#b2444167	V9	16#d6c41a85	V9	-
5	/000.01/cpcnsip.config	Standard Protocol	16#39ace0cd	V9	16#5a0a5c47	V9	-
6	/000.01/dpcom.config	Distribution of Write Requests	16#cf3c4ad6	V2	16#c3789be7	V2	-
7	/000.01/hh.config	Configuration of HIMA-HIMA Communi...	16#00000000		16#a4ce28b2	V8	--
8	/000.01/ke.config	COM Data Layout and Transmission	16#49b930c5	V2	16#49b930c5	V2	ok
9	/000.01/mos.config	Modbus Slave V2	16#4af207c9	V9	16#0aeadfeb	V9	-
10	/000.01/net.config	Network Setting	16#8e45fabe	V2	16#328dc994	V2	-
11	/sys/root.config	Root - System	16#8ac4e51b	V9	16#7eb7a0c7	V9	-
12	/sys/cpc.config	System Protocols Basis	16#cea77ad6	V2	16#f3edb36e	V2	-
13	/sys/cpcnsip.config	Standard Protocol	16#bec69940	V9	16#b00b4793	V9	-
14	/sys/cpcsip.config	Safety Protocol	16#00000000		16#04880f3b	V6	--
15	/sys/cpu.config	System Data	16#a2036579	V7	16#a2036579	V7	ok
16	/sys/dpcpu.config	Distribution of Write Requests	16#2c2223f9	V9	16#c157db5c	V9	-
17	/sys/ioa.config	I/O Configuration	16#fd1cecbb	V2	16#fd1cecbb	V2	ok
18	/sys/ke.config	Data Layout and Transmission	16#f85fda6f	V9	16#c4d20c51	V9	-

Figure 6: Modbus Slave V2 in the Version Comparison

10.1 Configuration Files Relevant for Modbus Slave V2

The relevant configuration files for the Modbus slave V2 protocol, which are specified in Table 19, are located within the *sys/root.config* configuration file.

<table>
<tr><th>Line</th><th>Configuration file</th><th>Description</th></tr>
<tr><td>4</td><td>/000.01/root.config</td><td>Communication module main file. This configuration file is referenced to subordinated configuration files and always changes if one of the subordinated file is modified.</td></tr>
<tr><td>6</td><td>/000.01/dpcom.config</td><td>Configuration file of the Modbus protocol. This specifies the changed settings for e.g., Number of Data Views, Length of Data View Input Area/Export Area.</td></tr>
<tr><td>9</td><td>/000.01/mos.config</td><td>Modbus slave V2 parameters, properties of the connections. The PADT generates a configuration file mos.config in the code generation, depending on the value of the Number of TCP Connections parameter, with the following versions:
<table>
<tr><th>Value Number of TCP Connections</th><th>mos.config version</th></tr>
<tr><td>1</td><td>1.0</td></tr>
<tr><td>2…20</td><td>2.0</td></tr>
</table></td></tr>
<tr><td>13</td><td>/sys/cpcnsip.config</td><td>The settings for the transfer of non-safe protocol data between CPU and COM modules are specified in the configuration file.</td></tr>
<tr><td>18</td><td>/sys/ke.config</td><td>Configuration file for reading and writing global variables in protocols (ke indicates the communication endpoint, i.e., global variable). This file contains a table that can be used to check global variables for plausibility with respect to the intended use as global multisource variables, see Chapter 10.1.</td></tr>
</table>

Table 19: Modbus Slave V2 Configuration Files

10.2 Global Multisource Variable

A global variable is considered to be a global multisource variable as soon as it is used in a data view. This also applies for derived data types of global variables (e.g., arrays), if at least one child element is used in a data view.

In addition to their Modbus source, global multisource variables can also be written to by a user program. With respect to the derived data type, it is irrelevant if the user program and the Modbus source write to the same or to different sub-elements.

If a global multisource variable is used in the user program, it must have write and read access. Before the global multisource variable is written to by the user program, it must be read in the user program. If a user program only writes to a global multisource variable, the following message appears in the logbook when the code generation is completed:

Global variable 'Global_Variable_5' has more than one source, but it will only be written to. In this case, it must also be read in the program.

Workaround: Read *Global_Variable_5* in the user program before writing to it!

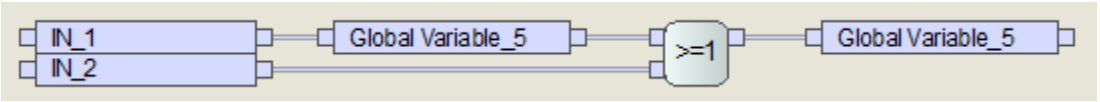


Figure 7: Reading and Writing to a Global Multisource Variable in the User Program

10.2.1 Identifying the Global Multisource Variables in the Version Comparison

With the version comparison, the resource's global multisource variables can be identified based on their use.

Global multisource variables must not be used for the safety functions of the user program.

To open the version comparison for the resource to be checked

1. In the structure tree, select the resource for which the version comparison should be started.
2. Select Extras, Version Comparison from the menu bar.

► The *Version Overview* dialog box appears.
During a version comparison, different resource configurations can be compared to one another. For further information, refer to the version comparison manual (HI 801 286 E).

To open ke.config

1. Double-click the row of the *ke.config* configuration file.
   ☑ The detail view appears.
2. Select the **Global Multisource Variables (CG)** tab.
   ☑ This tab lists all the global multisource variables (CG).
3. Double-click the row of a global variables.
   ☑ The Global Variable Editor with the cross-reference view appears.

► The information is displayed in plain text format. Figure 8 shows the global multisource variable *Global_Variable_5* and the positions in which it is used in the user program and in the Modbus slave set V2.

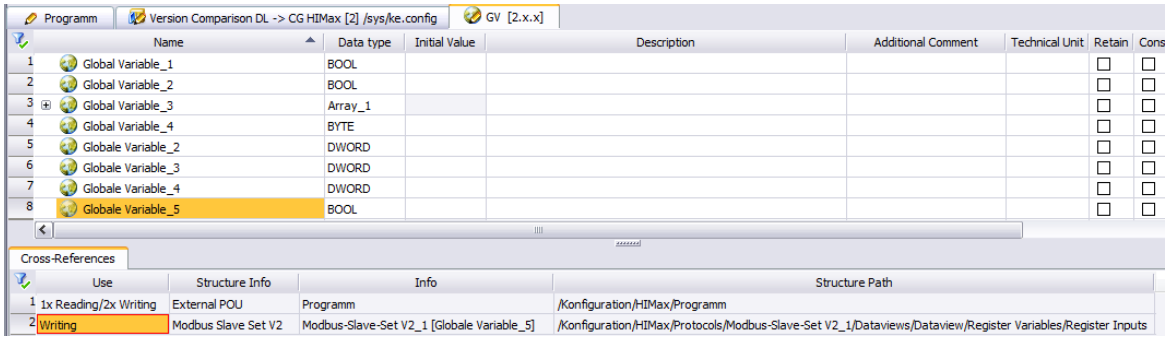


Figure 8: Uses of *Global_Variable_5*

11 Modbus Topologies with DCS

This chapter shows typical combinations for integrating a HIMA controller in a distributed control system (DCS) via the Modbus slave V2 protocol. The DCS must adopt the master role and initiates all the data transmission processes with the HIMA Modbus slave V2.

The Modbus Slave V2 integration options are not limited to those described in this chapter. For questions, please contact HIMA technical support.

The Modbus coupling of the HIMA controller to almost any DCS and visualization systems can be achieved either directly, using the RS485 interfaces, or indirectly, using the Ethernet interfaces of the HIMA controller.

i HIMax/HIMatrix controllers and DCS must be located in the same subnet, if the Ethernet interfaces are used as transmission channel, or they must have the corresponding routing settings if a router is used.

11.1 Modbus Slave Restrictions due to External Transport Path

The system uses the external transport path (fieldbus interface or IP address) of a master to determine its identity.

- If two masters on the two redundantly operated communication modules are to be considered identical to respect to the transport path, the master's properties specified in Chapter 4.3 must also be identical.
- Since the transport path is used as identification of the master, a master specific to each transport path must be created in the Modbus slave V2, even if the same master is logically hidden behind it.

11.2 Modbus via One Transport Path

11.2.1 Single Line Without Redundancy

HIMA does not recommend using this variant if availability requirements exist. A redundant variant should be chosen for high availability.

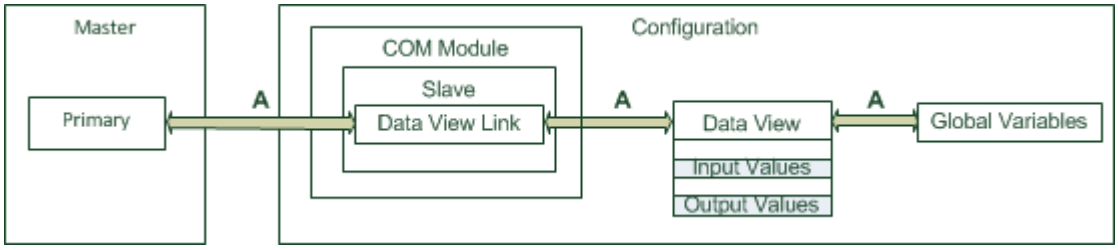


Figure 9: One Line Without Redundancy

11.3 Modbus via Two Transport Paths

This chapter presents some typical combinations for a redundant Modbus slave V2 connection.

These combinations can only be implemented with one or two COM modules and offer redundancy via two lines. No further engineering expense is necessary in the user program thanks to the demand mode of the Modbus slave V2 protocol, see Chapter 4.

i In accordance with the generally accepted regulations for developing Ethernet networks, no network loop may occur. Data packets may only reach a controller over a single path.

11.3.1 Both Lines Active (Both Lines Reading and Writing)

The global variable is written to with the value of the last write request (A or B) of a master. The data view links can be configured in two redundant COM modules or in one common COM module.

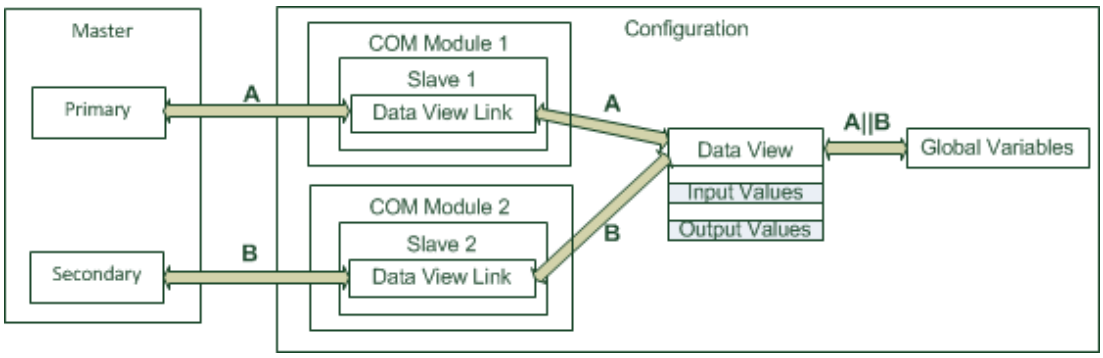


Figure 10: Two COM Modules, Both Lines Active (Both Lines Reading and Writing)

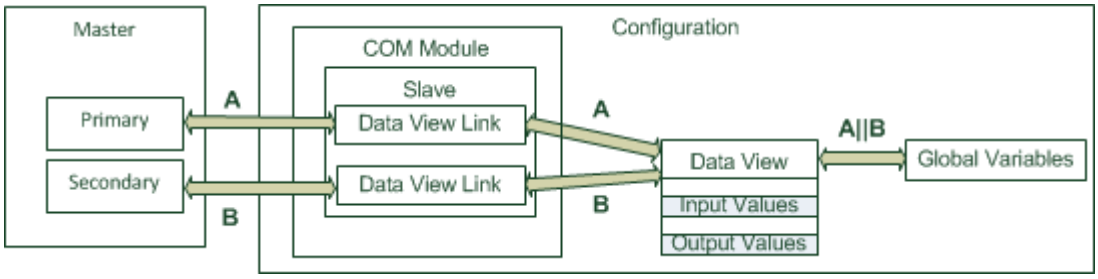


Figure 11: One COM Module, Both Lines Active (Both Lines Reading and Writing)

11.3.2 Both Lines Active with Switching Lines Option

As in the structure depicted in Figure 10, the global variable is written to with the value of the last write request (A or B) of a master. Optional switching between the masters is implemented in the slave by configuring an additional data view link in each of the two redundant COM modules.

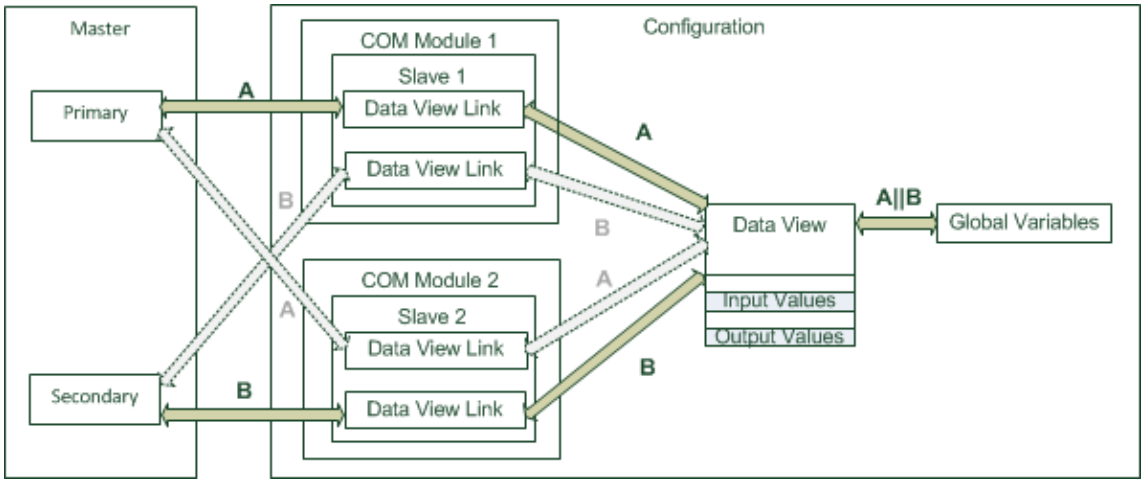


Figure 12: Two COM Modules, Both Lines Active (Both Lines Reading and Writing)

11.3.3 Cold Standby (One Line Active, One Line as Backup)

The global variable is written to with the value of the currently active master. The standby master is activated whenever the active master fails.

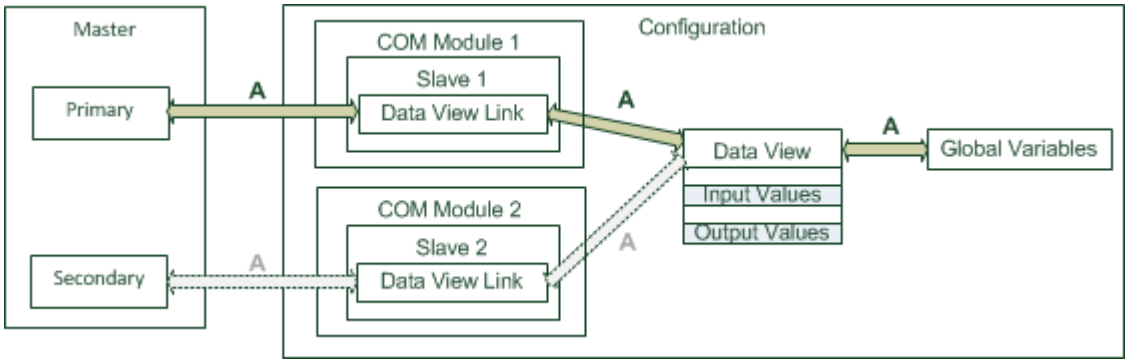


Figure 13: Cold Standby (One Line Active, One Line as Backup)

i The data view links can be configured in two redundant COM modules or in one common COM module if the redundant master allows the use of identical IP addresses for a redundant slave (simplex slaves).

11.3.4 Hot Standby (Both Lines Reading and Only One Line Writing)

Both masters can read, but only one can write. If the write master fails, this variant ensures fast switching between the masters.

The global variable is written to with the value of the currently active write master. The standby master is activated whenever the active master fails.

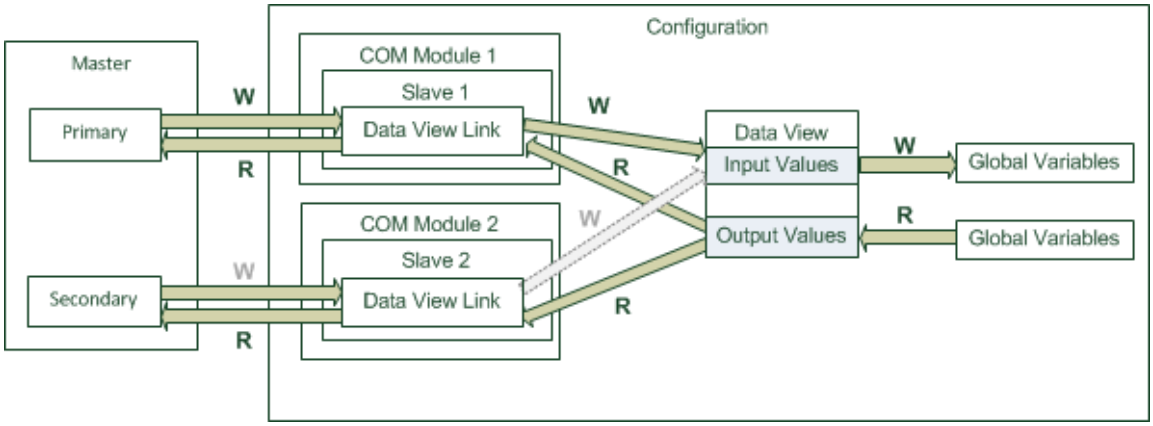


Figure 14: Hot Standby (Both Lines Reading and Only One Line Writing)

i The data view links can be configured in two redundant COM modules or in one common COM module if the redundant master allows the use of identical IP addresses for a redundant slave (simplex slaves).

12 General Information about Data Transfer

For further information on Ethernet and fieldbus communication, refer to the communication manual (HI 801 101 E).

12.1 Permitted Master IP Address Combinations

The general rules for assigning IP address and subnet masks must be adhered to.

i Some DCS allow that the same IP address is assigned to multiple masters, thus providing high availability. The identical IP address of the two masters is then entered in each master object of the Modbus slave.

12.1.1 Network Ports in Use for Ethernet Communication

All the following ports are destination ports.

UDP ports / use

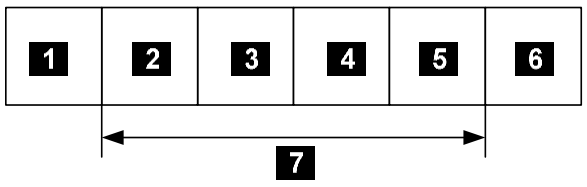
123	SNTP (time synchronization between PES and remote I/O, PES and external devices)
502	Modbus slave (can be changed by the user)
6010	safeethernet and OPC
8001	Configuration of the remote I/Os using the PES
8000	Programming and operation with SILworX
34964	PROFINET endpoint mapper (required for establishing the connection)
49152	PROFINET RPC server
49153	PROFINET RPC client

TCP ports / use

502	Modbus slave (can be changed by the user)
Xxx	TCP SR assigned by the user

12.2 Serial Modbus

The HIMA Modbus slave V2 supports data transmission in RTU format (Remote Terminal Unit). The RTU telegram frame starts and ends with the idle characters set by the user (default value: 5 idle characters), refer to Table 13.



1 Beginning of the frame (idle characters)
2 Address (8-bit)
3 Function (8-bit)
4 Data (N * 8-bit)
5 CRC checksum (16-bit)
6 End of the frame (idle characters)
7 Modbus Telegram

Figure 15: Modbus Telegram in the RTU Format

Appendix

Glossary

Term	Description
ARP	Address resolution protocol, network protocol for assigning the network addresses to hardware addresses
Bit variable	Variable that is addressed bit by bit
CENELEC	Comité Européen de Normalisation Électrotechnique (European Commitee for Electrotechnical Standardization)
COM	Communication module
Connector board	Connector board for the HIMax module
CPU	Processor module
CRC	Cyclic redundancy check
Data view	The global variables for output and output data are assigned to a data view to allow access to Modbus sources
EN	European standard
Export area	The export area is the process data volume that is written to by the system (a user program, hardware input or another protocol) and is read by the Modbus master
FB	Fieldbus
FBD	Function block diagrams
ICMP	Internet control message protocol, network protocol for status or error messages
IEC	International electrotechnical commission
Import area	The import area is the process data volume that is written to by the Modbus master and can be used as input data for the system (in a user program, hardware output or another protocol)
Interference-free	Supposing that two input circuits are connected to the same source (e.g., a transmitter). An input circuit is termed "interference-free" if it does not distort the signals of the other input circuit
KE	Communication end point
MAC address	Media access control address, hardware address of one network connection
NSIP	Not safety-related protocol
PADT	Programming and debugging tool (acc. to IEC 61131-3), PC with SILworX
PE	Protective ground
PELV	Protective extra low voltage
PES	Programmable electronic system
R	Read
R/W	Read/Write
Rack ID	Base rack identification (number)
Register variable	Variable that is addressed word by word
SB	System bus
SFF	Safe failure fraction, i.e., portion of faults that can be safely controlled
SIF	Safety-instrumented function
SIL	Safety integrity level (acc. to IEC 61508)
SILworX	Programming tool for HIMax, HIQuad X and HIMatrix
SIP	Safety instrumented protocol
SNTP	Simple network time protocol (RFC 1769)
SRS	System.Rack.Slot
SW	Software
TMO	Timeout
W	Write
WD	Watchdog
WDT	Watchdog time

Index of Figures

Index of Tables

MANUAL

Modbus V2

HI 801 475 E

For further information, please contact:

HIMA Paul Hildebrandt GmbH
Albert-Bassermann-Str. 28
68782 Brühl, Germany

Phone +49 6202 709-0
Fax +49 6202 709-107
E-mail info@hima.com

Learn more about HIMA solutions online:

www.hima.com/en/

www.hima.com