Manual

HIMax®

Safety Manual

Railway Applications

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 801 326 D, Rev. 11.00 (1938)	German original document
HI 801 327 E, Rev. 11.00.00 (1942)	English translation of the German original document

Table of Contents

1 Introduction

This manual contains information on how to operate the safety-related programmable electronic system HIMax in the intended manner.

The following conditions must be met to safely install and start up the system and to ensure safety during their operation and maintenance:

- Knowledge of regulations.
- Proper technical implementation of the safety instructions detailed in this manual performed by qualified personnel.

HIMA will 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 systems.
- De-activation or bypassing of safety functions.
- Failure to comply with the instructions detailed in this manual.

HIMA develops, manufactures and tests the HIMax system in compliance with the pertinent safety standards and regulations. The use of the systems is only allowed if the following requirements are met:

- They are only used for the intended applications.
- They are operated under the specified environmental conditions.
- They are only connected to the approved external devices.

To provide a clearer exposition, this manual does not specify all details of all system versions.

This safety manual represents the "Original instructions" as of Machinery Directive (Directive 2006/42/EC).

The "Original documentation" for the HIMA system is written in German language. The statements made in the German documentation shall apply.

1.1 Validity and Current Version

This safety manual was created for the following versions:

- HIMax Operating systems in accordance with revision list.
- As of SILworX V11.

For details on how to use previous HIMax and SILworX versions, refer to the corresponding previous versions of this manual.

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 a 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 Use of the HIMax System

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

2.1 Intended Use

This chapter describes the intended use of the safety-related automation system HIMax.

The automation system is designed for the industrial process market to control and regulate processes, protective systems, burner control applications, machine controllers and process plants, as well as for factory automation plants. SILworX, HIMA's programming tool, is used for programming, configuring, monitoring, operating and documenting the HIMax system.

The safety-related HIMax system can be used up to safety integrity level SIL 4 in accordance with EN 50126, EN 50128 and EN 50129.

Redundancy operation of HIMax modules does not preclude simultaneous operation of other non-redundant modules.

2.1.1 Application in Accordance with the De-Energize to Trip Principle

The HIMax system is designed in accordance with the de-energize to trip principle.

A system operating in accordance with the de-energize to trip principle switches off, for instance, an actuator to perform its safety function.

2.1.2 Application in Accordance with the Energize to Trip Principle

The HIMax system can also be used in applications that operate in accordance with the energize to trip principle.

A system operating in accordance with the energize to trip principle switches on, for instance, an actuator to perform its safety function.

When designing the automation system, the requirements specified in the application standards must be taken into account. For instance, line monitoring (SC/OC) for inputs and outputs or message reporting a triggered safety function may be required.

2.2 Non-Intended Use

During the transfer of (safety-relevant) data, IT security rules must be observed. When data is transferred through public networks like the Internet, additional measures such as VPN tunnel or firewall must be implemented to increase security.

2.3 Tasks of Operators and Machine and System Manufacturers

Operators as well as machine and system manufacturers are responsible for ensuring that HIMax systems are safely operated in automated systems and plants.

Machine and system manufacturers must sufficiently validate that the HIMax systems were properly programmed.

2.3.1 Connecting to Communication Partners

Only devices with electrically protective separation may be connected to the communication interfaces.

2.3.2 Implementing Safety-Related Communications

When implementing safety-related communications between various devices, ensure that the overall response time does not exceed the process safety time.

The calculation basis provided in Chapter 10.1.13 and in the communication manual (HI 801 101 E) must be applied.

2.4 ESD Protective Measures

Only personnel with knowledge of ESD protective measures may work on the HIMax system.

NOTICE

Damage to the HIMax system due to electrostatic discharge!

- **When performing the work, make sure that the workspace is free of static, and wear a grounding strap.**
- **If not used, ensure that the modules are protected from electrostatic discharge, e.g., by storing them in their packaging.**

2.5 Additional System Documentation

In addition to this manual, the following documents for configuring the HIMax systems are also available:

Document	Content	Document no.
HIMax safety manual	Safety functions of the HIMax system	HI 801 003 E
HIMax system manual	Hardware description of the modular system	HI 801 001 E
Certificates	Test results	
Revision list	Operating system versions certified by the TÜV	
Component-specific manuals	Description of the individual components	
Maintenance manual	Description of significant operational and maintenance actions.	HI 801 171 E
Communication manual	Description of safe**ethernet** communication and list of the available protocols.	HI 801 101 E
Automation security manual	Description of automation security aspects related to the HIMA systems.	HI 801 373 E
SILworX first steps manual	Introduction to the use of SILworX for engineering, start-up, testing and operation.	HI 801 103 E
SILworX online help	Instructions on how to use SILworX	

Table 1: Overview of the System Documentation

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/.

3 Safety Concept for Using the PES

This chapter contains important general information on the functional safety of HIMax systems.

- Safety and availability.
- Time parameters important for safety.
- Safety requirements.
- Automation security.
- Additional test requirements for railway applications.

3.1 Safety and Availability

The HIMax systems are certified for use in process controllers, protective systems, burner controllers, and machine controllers.

The safety-related HIMax system can be used up to safety integrity level SIL 4 in accordance with EN 50126, EN 50128 and EN 50129.

No imminent risk results from the HIMax automation systems.

⚠ WARNING

Possible physical injury caused by safety-related automation systems improperly connected or programmed.

Check all connections and test the entire system for compliance with the specified safety requirements before start-up!

Depending on the required availability, the HIMax system can be equipped with redundant processor modules (X-CPU 01, X-CPU 31), redundant system bus modules (X-SB 01), redundant communication modules (X-COM 01) and redundant I/O modules.

Redundant modules increase availability. If a module fault occurs, the faulty module automatically enters the safe state and the redundant module maintains operation with no interruption.

HIMA strongly recommends replacing failed modules as soon as possible.

A replacement module that is used instead of a failed one starts operation with no operator action. It adopts the function of the failed module, provided that is of the same type or is an approved replacement model.

3.1.1 Calculating the HR Values

The HR values for the HIMax systems have been calculated in accordance with IEC 61508.

The HR values are provided by HIMA upon request.

The safety functions, consisting of a safety-related loop (input, processing unit, output and safety communication among HIMA systems), meet the requirements described above in all combinations.

3.1.2 Self-Test and Fault Diagnostics

The operating system of the modules executes comprehensive self-tests at start-up and during operation. In particular, the following components are tested:

- Processors.
- Memory areas (RAM, NVRAM).
- The watchdog.
- Connections between modules.
- Individual I/O channels of the I/O modules.

If faults are detected during the tests, the operating system switches off the defective module or, for remote I/Os, the faulty I/O channel. If a module fault is detected during start-up, the modules will not start operation at all.

In non-redundant systems, this means that sub-functions or even the entire PES may be shut down. If a fault is detected in a redundant system, the redundant module or redundant channel assumes the function to be performed.

All HIMax modules are equipped with LEDs to indicate that faults have been detected. This allows the user to quickly diagnose faults detected in a module or the external wiring.

Additionally, the user program can evaluate various system variables displaying the module status.

Extensive diagnostics of the system performance and detected faults are stored in the diagnostic memory of the processor module or other modules. The diagnostics can also be read out after a system fault using the PADT.

For further details on how to evaluate diagnostic messages, refer to the system manual (HI 801 001 E).

For a very small number of component failures that do not affect safety, the HIMax system does not provide any diagnostic information.

3.1.3 PADT

The PADT is used to configure the controller and create the user program. The safety concept of the PADT supports the user in the proper implementation of the control task. The PADT implements numerous actions to verify the information entered.

The PADT is a personal computer installed with the SILworX programming tool.

3.1.4 Redundancy

To improve availability, all parts of the system including active components can be set up redundantly and, if necessary, replaced while the system is operating.

The component redundancy does not impair the system safety. SIL 4 is still guaranteed even if system components are used redundantly.

3.1.5 Structuring Safety Systems in Accordance with the Energize to Trip Principle

Safety systems operating in accordance with the energize to trip principle have the following function:

1. The safe state of a module is the de-energized state. This state is adopted, for instance, if a fault has occurred in the module.
2. The controller can trigger the safety function on demand by switching on an actuator.

3.1.5.1 Detection of Failed System Components

Thanks to the automatic diagnostic function, the safety system is able to detect that modules have failed.

3.1.5.2 Safety Function in Accordance with the Energize to Trip Principle

The safety function is performed when the safety system energizes one or several actuators.

The users must plan the following actions:

- Include and configure a redundant module for every I/O module.
- Every module must be provided with short-circuit and open-circuit monitoring. Short-circuit and open-circuit monitoring must be configured for each channel.
- The operation of the actuators can be monitored through a position feedback.

3.1.5.3 Redundancy of Components

It may be necessary to redundantly structure the following components:

- Power supply of the controller.
- HIMax modules.
- Sensors and actuators.

If redundancy is lost, the controller must be repaired as soon as possible.

For details on component redundancy, refer to the system manual (HI 801 001 E).

It is not required to design the safety system modules redundantly if, in the event of a safety system failure, the required safety level can otherwise be achieved, e.g., by implementing organizational measures.

3.2 Safety-Relevant Time Parameters

The following time parameters must be taken into account for the controller's safety considerations:

- Process safety time.
- Safety time (of the resource).
- Watchdog time (of the resource).
- Response time.

i Resource refers to the image of the controller (PES) in the SILworX programming tool.

3.2.1 Process Safety Time

According to IEC 61508-4, the process safety time is the time interval between a failure of the EUC or the EUC control system with the potential to cause a hazardous event and the point in time when the EUC response must be completed to prevent the hazardous event from occurring.

During the process safety time, the process may allow faulty signals to exist without a hazardous state occurring.

A safety-related response of the controller including all delays due to sensors, actuators, I/O modules and process (response of the plant to a tripping) must occur within the process safety time.

3.2.2 Safety Time [ms] Parameter of the Resource

The *Safety Time [ms] parameter in the resource properties* $\mathbf{t}_{SR}$ affects the response time of the resource $\mathbf{t}_{RR}$ as follows:

$\mathbf{t}_{RR}$ **≤** $\mathbf{t}_{SR}$

$\mathbf{t}_{SR}$ The *Safety Time [ms]* parameter

The following factors prolong the response time of the resource and must be taken into account during set-up:

- Physical delays, e.g., due to the switching times of external relays.
- Delays configured in the user program, e.g., the timer function blocks TON and TOF.
- Delays of output signals due to output noise blanking, see Chapter 8.4.2.

The *Safety Time [ms]* parameter $\mathbf{t}_{SR}$ in the resource properties can be set in SILworX within 20…22 500 ms.

To ensure that the fault response is triggered within the configured resource safety time, the following requirements must be met:

- The user program must respond within a RUN cycle.
- No input signal delay due to delay elements configured in the input modules (T on, T off).
- No delays configured through the user program.

3.2.3 Watchdog Time (of the Resource)

The watchdog time t_{WD} is the maximum permissible duration of a RUN cycle (cycle time). The controller is shut down if the cycle time exceeds the watchdog time.

The user can set the watchdog time in accordance with the safety-related requirements of the application.

Condition for safety:

$$t_{WD} \leq ½ \times t_{SR}$$

t_{WD} Watchdog time (of the resource)

t_{SR} *Safety Time [ms]* parameter (of the resource)

The watchdog time (of the resource) must be configured. The *Watchdog Time [ms]* parameter can be set within 6…7500 ms and is configured in the resource properties. The default setting is 200 ms.

The PADT checks the parameters *Safety Time [ms]* and *Watchdog Time [ms]* and rejects the configuration while generating it if the configured watchdog time is greater than ½ the value of the resource safety time.

The watchdog time can only be estimated. For the estimation, the following time requirements must be taken into account.

- Cycle duration of the user programs (RUN cycle of the resource).
  - Time for reading in the data.
  - Data processing.
  - Process data communication.
  - Time for issuing the data.
- Processor module synchronization.
- Special time requirements for reload.

NOTICE

The user must consider and oberserve the mentioned restrictions when performing online changes to the controller!

Carefully check the settings before any online change!

3.2.4 Estimating the Watchdog Time

HIMA strongly recommends the following setting to ensure sufficient availability:

$$2 \times t_{WD} + t_{Sync} + 2 \times t_{I/O\ cycle} \leq t_{SR}$$ **(***Safety Time [ms]* **parameter)**

t_{Sync} Maximum synchronization time of the processor modules, see Chapter 3.2.4.

$t_{I/O\ cycle}$ I/O cycle time = 2 ms

If no reliable assessment of the max. CPU cycle time can be made, set the watchdog time as follows:

$$3 \times t_{WD} + 2 \times t_{I/O\ cycle} \leq t_{SR}$$

3.2.5 Determining the Watchdog Time through Testing

For time-critical applications or systems including more than one controller (PES), it is necessary to determine the watchdog time $\mathbf{t}_{WD}$ during start-up. This must be done during RUN operation and under full load. To this end, all engineered modules must be inserted and all the configured communication connections (e.g., safe**ethernet** and other standard protocols) must be operating.

The maximum system load results from synchronization, when the modules are removed and reinserted. The watchdog time must be set so that synchronization at full load is always possible.

To perform the test

1. In the resource properties, set the *Safety Time [ms]* to the maximum value (22 500 ms).
2. In the resource properties, set the *Watchdog Time [ms]* to the maximum value (7 500 ms).
3. The values for $\mathbf{t}_{Com}$, $\mathbf{t}_{Config,}$ $\mathbf{t}_{Latency}$ must be calculated and set as described in the safety manual.
4. Compile the configuration and load it into the controller by performing a download.
5. Start the resource (cold start).
6. Open the Control Panel for the resource and reset the cycle time statistics.

For the following steps, the system must be operated under full load.

7. Read out the maximum execution time of all user programs (UP) in the Control Panel, wait several minutes and note down the variations and load peaks.
   Then calculate $\mathbf{t}_{peak}$:
   $\mathbf{t}_{peak}$ = execution time (max.) - execution time (min.), calculate it for each UP and add the resulting values.
8. In succession, remove and reinsert every processor module in the base plate. Prior to removing a processor module, wait until the processor module just inserted is synchronized.

i When a processor module is inserted in the base plate, it automatically synchronizes with the configuration of the existing processor modules. The time required for synchronization extends the controller cycle to the maximum cycle time.

The synchronization time increases with the number of processor modules that have already been synchronized.

For further details on how to insert and remove a processor module, refer to the X-CPU 01 manual (HI 801 009 E) or the X-CPU 31 manual (HI 801 355 E).

9. In the diagnostic history of the non-synchronized modules, read the synchronization time from n to n+1 processor modules in every synchronization process. The largest synchronization time is used to determine the watchdog time.

10. Use the noted times in the following equation:

$$t_{WD} = t_{Sync} + t_{Reserve} + t_{Com} + t_{Config} + t_{Latency} + t_{Peak}$$

t_{Sync} Determined processor module's synchronization time

$t_{Reserve}$ Safety margin 12 ms.

t_{Com} System parameter *Max. Com.Time Slice ASYNC [ms]*, which is configured in the resource properties

t_{Config} System parameter *Max. Duration of Configuration Connections [ms]*, which is configured in the resource properties.

$t_{Latency}$ Configured system parameter *Maximum System Bus Latency [µs]* x 4

t_{Peak} Sum of all UP peaks calculated in step 7.

3.3 Safety Requirements

For using the safety-related HIMax automation system, the following safety requirements must be met:

3.3.1 Product-Independent Hardware Requirements

Personnel configuring the HIMax hardware must observe the following product-independent safety requirements.

- To ensure safety-related operation, approved fail-safe hardware and software components must be used. Approved HIMA components are listed in the HIMax version list.
  The latest versions can be found in the version list, which is maintained together with the test authority.
- The conditions of use specified in this safety manual about EMC, mechanical, chemical and climatic influences must be observed.
- Non-fail-safe, interference-free hardware components and software components can be used for processing non-safety-relevant signals, but not for handling safety-related tasks. Non-fail-safe components must not be used for processing safety-related tasks.
- The de-energize to trip principle must be applied to all safety circuits externally connected to the system.

3.3.2 Product-Dependent Hardware Requirements

Personnel configuring the HIMax hardware must observe the following product-dependent safety requirements.

- Only devices with electrically protective separation from the power supply may be connected to the system
- Only safety-related modules may be used to process safety-related tasks.
- The operating requirements detailed in the system manual, particularly those concerning supply voltage and ventilation, must be observed.
- To comply with the protective provisions for electrical safety and grounding, the manufacturer of the specific application must ensure that proper measures are implemented for separating the indoor and outdoor equipment in accordance with EN 50122. This must protect the HIMax systems against influences from the outdoor equipment in the overhead contact line zone or the pantograph zone, as well as against traction return currents. Power supply devices allowed for railway applications must be used.

3.3.3 Product-Independent Programming Requirements

Personnel developing user programs must observe the following product-independent safety requirements:

- In safety-relevant applications, ensure that the safety-relevant system parameters are properly configured.
- In particular, this applies to the system configuration, maximum cycle time and safety time.

3.3.4 Requirements for Using the Programming Tool

The SILworX programming tool must be used for programming the HIMax system. The following requirements for using SILworX must be met.

- Compiling the program twice in SILworX and comparing the two CRCs ensures ensures that the program was properly compiled.
- The application described in the specification must be validated, verified and its proper implementation must be documented. A complete test of the logic must be performed by trial.
- The system response to faults in fail-safe input and output modules must be defined in the user program in accordance with the system-specific safety-related conditions.
- The SILworX programming tool is provided with a feature that, after the user program or system configuration has changed, only displays the performed changes. The analysis of the changes (change impact analysis IA) must define the required test scope. This impact analysis must take the expected changes based on the performed modifications, the result of the SILworX comparison feature and the required regression tests into account.

3.3.5 Communication

The following requirements for communication of data and to systems must be met.

- When implementing safety-related communications between various HIMA systems, ensure that the overall response time of a system does not exceed the worst case response time. All calculations must be performed in accordance with the rules given in Chapter 11.2.
- Data transmission in Category 1 and Category 2 transmission systems in accordance with EN 50159 is possible with no additional measures.
- Transmission systems (Category 3) in accordance with EN 50159 may be used, if additional measures are taken to guarantee that the transmission channel is secure (e.g., firewalls or encryption).
- Never use the standard protocols to transfer safety-related data.
- Only devices with electrically protective separation may be connected to the communication interfaces.

3.3.6 Requirements for Railway Applications

The following requirements must be observed when using the HIMax system in railway applications:

- The HIMax system can be used and operated in environments with pollution degree 2 and overvoltage category 2 in accordance with EN 50124-1.
- The relevant standards must be used for railway applications.
- The digital outputs are equipped with short-circuit monitoring. Responses to detected short-circuits must be programmed in the user program.
- The temperature state (operating temperature) of the HIMax systems must be evaluated in the user program. Also safety-related measures must be triggered with the user program. For further details, refer to the corresponding chapter in the HIMax system manual (HI 801 001 E).
- Error messages must be evaluated in the user program. Errors are signaled by state bits and are thus available to the user program. Additionally, errors are stored in the diagnostic memory of the controller and can be read out using the programming tool. For further details, refer to the corresponding chapter in the HIMax system manual (HI 801 001).
- Detection of ground faults must be configured externally.

3.4 Automation Security

HIMA distinguishes between the terms *safety*, which refers to functional safety, and *security*, which refers to the system protection against manipulation.

Industrial controllers (PES) must be protected against IT-specific problem sources, for instance:

- Inadequate protection of IT equipment (e.g., open WLAN, obsolete operating systems).
- Lack of awareness of proper use of the equipment (e.g., USB sticks).
- Direct access to protected areas.
- Attackers inside the company premises.
- Attackers via communication networks inside and outside the company premises.

HIMA safety systems are composed of the following parts to be protected:

- Safety-related automation system.
- PADT.
- Optional X-OPC servers (X-OPC DA, X-OPC AE).
- Optional communication connections to external systems.

3.4.1 Product Properties

The HIMax controller with basic settings already fulfils the requirements for automation security.

Protective mechanisms for preventing unintentional or unapproved modifications to the safety system are integrated into the controllers and the programming tool:

- Each change to the user program or controller configuration results in a new configuration CRC.
- Online changes of the safety parameters can be deactivated in the controller. Therefore, changes to the safety parameters are only possible by performing a download or reload.
- The user can set up a user management scheme to increase security. This scheme is used to specify the user groups, user accounts, access permissions for PADT and controllers (PES) for each project. In the user management scheme, the user can define if an authorization is required to open the project and log in to a controller.
- The data of a controller can only be accessed if the user projects loaded in the PADT and controller are the same. The CRCs must be identical (archive maintenance!).
- A physical connection between PADT and controller (PES) is not required during operation and must be interrupted for security reasons. The PADT can be reconnected to the controller for diagnostic and maintenance purposes.

The requirements of the safety and security standards must be complied with. The operator is responsible for authorizing personnel and implementing the required protective actions.

⚠ WARNING

Physical injury possible due to unauthorized manipulation of the controllers!

Protect the controllers against unauthorized access!!

- **Change the default settings for login and password.**
- **Supervise access to controllers and PADTs!**
- **For further protection measures, refer to the automation security manual (HI 801 373 E).**

3.4.2 Risk Analysis and Planning

Security is a process, not a product. Maintained network maps, for instance, help to ensure that secure networks are permanently separated from public networks. It is recommended to only have one well-defined connection, e.g., via a firewall or a DMZ (demilitarized zone).

Careful planning should identify the necessary measures. The required measures are to be implemented after the risk analysis is completed, and may include:

- Assignment of access permissions for user groups and user accounts according to the intended tasks.
- Use of passwords in accordance with the security requirements.

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

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

Refer to the HIMA automation security manual (HI 801 373 E) for more details.

3.5 Test Requirements

Refer to the HIMax safety manual (HI 801 003 E) for the standards used to test and certify the HIMax system for industrial use.

3.6 Additional Test Requirements for Railway Applications

The following table shows the HIMax components that are approved for railway applications:

Designation	Description
X-CPU 01	Processor module
X-CPU 31	Processor module
X-SB 01	System bus module
X-COM 01	Communication module
X-AI 32 01	Analog input module (32 channels)
X-DI 32 01	Digital input module (32 channels)
X-DI 32 02	Digital input module (32 channels), for proximity switches (NAMUR)
X-DI 32 03	Digital input module (32 channels), 48 VDC
X-DI 64 01	Digital input module (64 channels)
X-DO 12 01	Relay module (12 channels)
X-DO 24 02	Digital output module, (24 channels), 48 VDC
X-DO 32 01	Digital output module (32 channels)
X-BASE PLATE	HIMax base plate

Table 2: Approved HIMax Components

3.6.1 Height Range

The following classes in the specified height range apply to the HIMax components:

- For use in signaling applications in accordance with EN 50125-3: AX up to 2000 m.
- For use on rolling stock in accordance with EN 50125-1: AX up to 2000 m.

3.6.2 Climatic Requirements

The HIMax components were tested in accordance with EN 50125-3 and EN 50155.

3.6.2.1 Use in Signaling Applications

In accordance with EN 50125-3, the following climatic classes can be derived for the temperature range 0…+60 °C:

- In a container with temperature monitoring: T1, T2 and TX.
- In buildings with no climatic control: T1.
- In buildings with climatic control: T1, T2 and TX.

3.6.2.2 Use on Rolling Stock

For use on rolling stock, no temperature class in accordance with EN 50155 can be assigned to the HIMax modules.

i The user must implement suitable measures in the application to ensure that the HIMax temperature range of 0…60 °C is maintained.

As for the extended operating temperature when powering on, class ST0 applies to the HIMax system, as defined in EN 50155, Chapter 4.3.3.

With respect to fast temperature change, temperature class H1 applies to the HIMax system in accordance with EN 50155, Chapter 4.3.4.

Since the PCB in the modules of the HIMax system are provided with a protective coating, they achieve the protective coating class PC2, as defined in EN 50155, Chapter 10.7.

3.6.3 Mechanical Requirements

The HIMax components were tested in accordance with EN 50125-3 and EN 50155.

3.6.3.1 Use in Signaling Applications

The following table lists the most important tests and limits for mechanical requirements:

Test standard	Mechanical tests
EN 50125-3	Vibration immunity test: 2.3 m/s^2 between 5…2000 Hz, EUT in operation
	Shock immunity test: 20 m/s^2, 11 ms, EUT in operation

Table 3: Mechanical Requirements for Use in Signaling Applications

3.6.3.2 Use on Rolling Stock

The HIMax components listed in Table 2 were mechanically tested in accordance with EN 50155. Testing was performed in accordance with EN 61373, Category 1, Class B.

The HIMax system has no sockets for integrated circuits and/or edge connectors. This is why class K2 is complied with, as defined in the EN 50155, Chapter 10.1.5.

3.6.4 EMC Requirements

The HIMax components specified in Table 2 were successfully tested and met the EMC requirements in accordance with EN 50121-4 and EN 50121-3-2.

3.6.4.1 Use in Signaling Applications

The following table lists the most important tests and limits for EMC requirements:

Test standard	Type of test	Interference immunity tests	
EN 61000-4-2	ESD test	6 kV contact discharge, 8 kV air discharge	
EN 61000-4-3	EM field	80 MHz…1 GHz:	10 V/m
		800 MHz…1 GHz:	20 V/m
		1.4…2 GHz:	10 V/m
		2…2.7 GHz:	5 V/m
		5.1…6 GHz:	3 V/m
EN 61000-4-4	Burst test	Supply voltage:	2 kV
		I/O lines:	2 kV
		Ground:	1 kV
EN 61000-4-5	Surge	Supply voltage:	2 kV CM
			1 kV DM
		I/O lines:	2 kV CM
			1 kV DM
EN 61000-4-6	Injected RF currents	Supply voltage:	10 V
		I/O lines:	10 V
		Ground:	10 V
EN 61000-4-8	Power frequency magnetic field	16 2/3 Hz, 50 Hz, 60 Hz:	100 A/m
		DC:	300 A/m

Table 4: EMC Requirements for Use in Signaling Applications According to EN 50121-4

i If the modules X-DI 32 03, X-DI 64 01, X-DO 24 02 and X-DO 32 01 are used, the following measures must be implemented to meet the radio disturbance voltage requirements:

- Split core ferrite WE 742 715 5 with 7 windings on the system's 24 V supply voltage.
- Split core ferrite WE 742 715 5 with 4 windings on the 48 V supply voltage of the X-DO 24 02.
- Split core ferrite WE 742 715 5 with 4 windings on the 48 V supply voltage of the X-DI 32 03.

i If the X-DO 24 02 module is fed with 24 V power supply from a DC network, the H 7013 filter must be installed directly near the supply voltage connection to the module. For 48 VDC, the corresponding H 7021 filter must be used.

3.6.4.2 Use on Rolling Stock

The following table lists the most important tests and limits for EMC requirements:

Test standard	Type of test	Interference immunity tests	
EN 61000-4-2	ESD test	6 kV contact discharge, 8 kV air discharge	
EN 61000-4-3	EM field	80 MHz…1 GHz: 1.4…2 GHz: 2…2.7 GHz: 5.1…6 GHz:	20 V/m 10 V/m 5 V/m 3 V/m
EN 61000-4-4	Burst test	Supply voltage: I/O lines:	2 kV 2 kV
EN 61000-4-5	Surge	Supply voltage:	2 kV CM 1 kV DM
EN 61000-4-6	Injected RF currents	Supply voltage: I/O lines:	10 V 10 V

Table 5: EMC Requirements for Use on Rolling Stock According to EN 50121-3-2

3.6.5 Severe Conditions

The HIMax system must be installed in enclosures with suitable degree of protection (e.g., IP54) to ensure protection against the environmental influences as of classes 4C3, 4B1 and 4S2.

3.6.6 Supply Voltage

The following table lists the most important tests and limits for the supply voltage of the HIMax components:

IEC/EN 61131-2	Verification of the DC supply characteristics
	Alternatively, the power supply must comply with the following standards: IEC/EN 61131-2 or SELV (Safety Extra Low Voltage) or PELV (Protective Extra Low Voltage)
	HIMax devices must be fuse protected as specified in the manuals for the X-BASE PLATE (HI 801 025 E and HI 801 371 E).
	Voltage range test: 24 VDC, -20…+25 % (19.2…30.0 V)
	Momentary external current interruption immunity test: DC, PS 2: 2 ms
	Reversal of DC power supply polarity test: Refer to the corresponding chapter of the system manual or data sheet of the power supply.
	Backup duration, withstand test: Test B, 1000 h

Table 6: Verification of the DC Supply Characteristics

3.6.6.1 Supply Voltage Requirements for Use on Rolling Stock

The HIMax systems are supplied from an accumulator battery with 24 V nominal voltage.

The following values apply to the HIMax supply voltage: 24 VDC, -15…+20 %, 5 % ripple.

This results in the following tolerance values:

Minimum continuous voltage: 19.2 V (0.8 UN)

Maximum continuous voltage: 30 V (1.25 UN)

- The HIMax components specified in Table 2 were tested in accordance with EN 50155, Chapter 5.1.
- Taking external measures, users must ensure that the minimum continuous voltage of 0.8 UN is maintained, since otherwise individual modules or the entire system will reboot.

Taking external measures, the user must be able to intercept voltage fluctuations above 1.25 UN in accordance with EN 50155, Chapter 5.1.1.3.

HIMax systems are designed for interruptions of up to 2 ms. As such, the HIMax meets the requirements of Class S1 in accordance with EN 50155, Chapter 5.1.1.4.

The HIMax system meets the requirements for DC voltage ripple factor in accordance with EN 50155, Chapter 5.1.1.6.

The requirements in accordance with EN 50155, Chapter 5.1.3, for switching 2 supply voltages are not met. External measures must be implemented by the user.

4 Processor Module

The safety-related processor module is composed of 2 microprocessors, each with its own RAM, that simultaneously process the operating system and the user program. A hardware comparator continuously aligns the data from the two microprocessors and those from the memories. The processor module reports detected differences and automatically enters the ERROR STOP state.

The processor module carries out additional self-tests such as the program sequence monitoring (watchdog).

4.1 Processor Module X-CPU 01

The X-CPU 01 processor module can be operated with up to 4-fold redundancy. It may be inserted in rack 0 or rack 1, slots 3…6.

4.2 Processor Module X-CPU 31

The X-CPU 31 processor module combines the functions of processor and system bus modules. For this reason, it can only be inserted into slot 1 or slot 2 of rack 0. If so, no further processor module can be used in slots 3…6 of racks 0 and 1!

4.3 Self-Tests

The operating system of the processor module executes comprehensive self-tests at start-up and during operation. If the operating system detects single faults that could cause a hazardous operating state to occur, the faulty components are switched off. This is the safe state and is performed within the safety time.

The scope of the testing includes:

- The microprocessors.
- The redundant memories.
- The NVRAMs.
- The watchdog.
- The hardware comparator.

4.4 Responses to Faults in the Processor Module

If the processor module detects an internal module fault, an entry is written to the diagnostic history. Subsequently, a reboot is performed.

After the first reboot due to faults, the processor module restarts and, once all self-tests are complete, attempts to start system operation. If the internal module fault is still present, the processor module performs a second reboot.

If a further internal fault occurs within the first minute after restart, the processor module no longer participate in the system's operation.

If the last processor module fails, the entire system stops system operation, i.e., the protocol connections are closed, I/O outputs are de-energized.

If an automatic restart is not desired, the resource parameter *Autostart* must be deactivated.

4.5 Replacing Processor Modules

Prior to replacing a processor module, ensure that the replacement will not cause a running HIMax system to stop.

In particular, this applies for systems running in accordance with the energize to trip principle. The failure of such systems causes the loss of the safety function.

Redundant processor modules can be replaced during operation, provided that at least one processor module is available that can maintain safety-related operation while the other module is being replaced.

NOTICE

Interruption of safety-related operation possible!

Replacing a processor module with a lit or blinking Ess **LED can result in the interruption of a controller's operation.**

Do not remove processor modules if the Ess **LED is lit or blinking.**

A lit or blinking **Ess** LED indicates that the processor module is essential for the system to function.

Even if the LED is not lit or blinking, the system redundancies, which this processor module is part of, must be checked using SILworX. The communication connections processed by the processor module must also be taken into account.

For further details on how to replace processor modules, refer to the processor module manuals (HI 801 009 E and HI 801 355 E) and to the system manual (HI 801 001 E).

5 System Bus Module

A system bus module manages one of the two safety-related system buses. The two system buses are mutually redundant. Each system bus interconnects the various modules and base plates. The system buses are used to transmit safe data via a safety-related protocol.

A HIMax system that **only** contains **one processor module** can be operated at a reduced availability level using one system bus only.

Processor modules of type X-CPU 31 can also be used in rack 0 instead of system bus modules. The statements made in this chapter also apply for X-CPU 31 modules. The X-CPU 31 modules require a special double-width connector board.

5.1 Rack ID

The rack ID identifies a base plate within a resource and must be unique for each base plate.

The rack ID is the safety parameter for addressing the individual base plates and the modules mounted on them!

The rack ID is stored in the connector board of the system bus module.

For details on how to proceed for configuring the rack ID, refer to the system manual (HI 801 001 E) and the SILworX first steps manual (HI 801 103 E).

5.2 The Responsible Attribute

Only one of the system bus modules contained in each system bus may have the Responsible attribute and thus be configured as responsible for system bus operation.

- For system bus A, the Responsible attribute is reserved for the system bus module or the X-CPU 31 processor module in rack 0, slot 1.
- The following applies to system bus B:
  - If system bus modules are used, the attribute can be configured with SILworX. The Responsible attribute can either be set for the system bus module in rack 0, slot 2, or for the system bus module in rack 1, slot 2.
  - If the processor module X-CPU 31 is used, the attribute is fixed for the module in rack 0, slot 2.

Prior to starting safety-related operation, ensure that the Responsible attribute is properly configured for both system buses.

For details on how to set the Responsible attribute, refer to the SILworX first steps manual (HI 801 103 E).

⚠ WARNING

Physical injury possible!

SILworX must be used to verify the configuration.

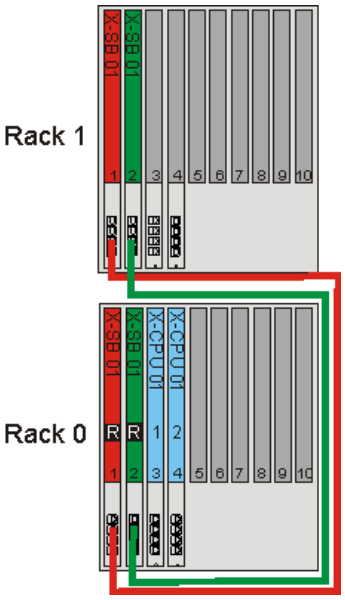
The following procedure must be followed:

In SILworX, log in to the system bus module in rack 0, slot 2.

- **In SILworX, log in to the system bus module in rack 1, slot 2.**
- **In the Control Panel of both system bus modules, ensure that the Responsible attribute is only set for the proper system bus module (see Figure 1 and Figure 2)!**

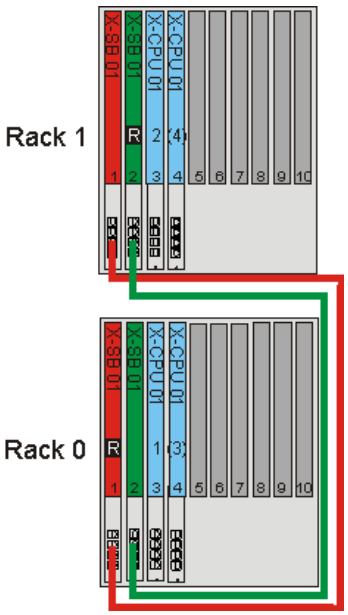
Recommended configurations:

- If processor modules are only contained in rack 0, both system bus modules in rack 0 must be set to Responsible (Figure 1).
- If processor modules are also contained in rack 1 (Figure 2), the Responsible attribute must be set as follows:
  - For the system bus module in rack 0, slot 1 (automatically).
  - For the system bus module in rack 1, slot 2.



R System bus module set to Responsible

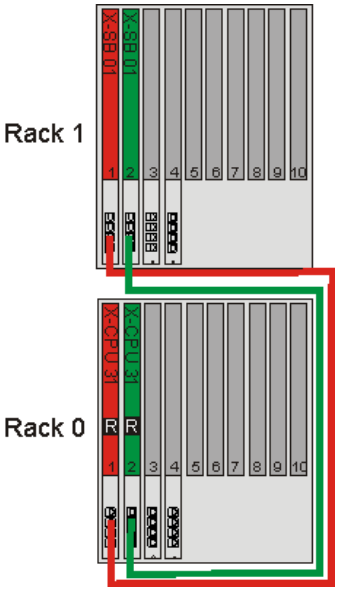
Figure 1: Recommended Configuration: All Processor Modules in Rack 0



R System bus module set to Responsible

Figure 2: Recommended Configuration: X-CPU 01 Processor Modules in Rack 0 and Rack 1

- If X-CPU 31 processor modules are inserted in rack 0, slot 1 and slot 2 (Figure 3), the Responsible attribute must always be set for the processor modules. The Responsible attribute must not be set for the system bus module in rack 1, slot 2.



R Processor module set to Responsible

Figure 3: Configuration with X-CPU 31 Processor Modules in Rack 0, Slot 1 and Slot 2

6 Communication Module

Communication modules are used for both exchanging safety-related data with other HIMA controllers and for exchanging standard data via fieldbuses and Ethernet.

- The processor module controls safety-related data traffic using the safety-related transmission protocol safeethernet. The communication module forwards the data to the connected HIMA controllers. The safety-related safe**ethernet** protocol ensures that corrupted messages are detected (black-channel principle).

  This allows safety-related communication via non safety-related transmission paths, i.e., standard network components.
- The standard protocols are for instance:
  - Modbus
  - PROFIBUS master/slave
  - Send/Receive TCP
  - PROFINET IO
  - SNTP

For further details on communication and communication modules, refer to the following documents:

- This manual, Chapter 11.1
- Communication module manual, HI 801 011 E
- Communication manual, HI 801 101 E
- System manual, HI 801 001 E

7 Input Modules

The following table provides an overview of the input modules of the HIMax system:

Digital input modules [1)]	Channels	Safety-related	Remark
X-DI 32 01	32	SIL 4	
X-DI 32 02	32	SIL 4	Proximity switch (NAMUR)
X-DI 32 03	32	SIL 4	48 VDC
X-DI 64 01	64	SIL 4	
Analog input modules [1)]	Channels	Safety-related	Remark
X-AI 32 01	32	SIL 4	
[1)] Interference-free: When a module performing part of a safety function is not affected by other operating modules. This applies irrespective of whether the modules are safety-related or not.			

Table 7: Overview of the Input Modules

7.1 General

Safety-related inputs may be used for safety-related as well as for non-safety-related signals. Non-safety-related signals, however, may not be used for safety functions!

The controllers generate and save error and status messages for the diagnostic LEDs of the modules. The PADT can read out the saved diagnostic messages.

Safety-related input modules automatically perform a high-quality, cyclic self-test during operation.

Detection of faults during the self-tests automatically triggers a safety-related response. The initial value is provided to the user program as a global variable and corresponding error messages are created. The detailed error messages can be evaluated in the user program by reading out the error codes.

For further details on the input modules, refer to the module manuals.

7.2 Response in the Event of a Fault

If a fault is detected at the signal inputs, the user program processes the input's initial value. A module fault in the input module causes the user program to process the initial value for all the inputs. The initial value of the global value must be configured in SILworX accordingly (default value = 0). The module activates the *Error* LED.

The status and error messages as well as the system variables can be used to program application-specific fault responses. For further details, refer to the manual of the corresponding module.

7.3 Safety of Sensors, Encoders and Transmitters

In safety-related applications, the controller (PES) and connected sensors, encoders and transmitters must all meet the safety requirements and achieve the specified SIL. For details on how to achieve the required SIL for sensors, refer to the IEC 61511-1 standard, Section 11.4.

7.4 Safety-Related Digital Input Modules

The input modules read the digital signals at the inputs and provide failsafe values to the user program in every processor module cycle. The modules cyclically test the inputs' safe operation.

7.4.1 Test Routines

The test routines check whether the input channels are able to forward both signal levels (low and high), irrespective of the signals actually present on the input. This functional test is performed while the input signals are being read. Whenever a fault occurs in the input module, the low level (safe state) is processed in the user program.

7.4.2 Redundancy of Digital Inputs

The digital inputs may be connected redundantly. The redundant connection is used to increase availability of the inputs.

7.4.3 Surges on Digital Inputs

Due to the short cycle time of the HIMax systems, a surge pulse as described in EN 61000-4-5 can be read in to the digital inputs as a short-term high level.

If shielded cables are used for digital inputs, no additional precautionary measures are required to protect against surges.

The following measures ensure proper operation in environments where surges may occur:

- Install shielded input wires.
- Program noise blanking in the user program. A signal must be present for at least two cycles before it is evaluated. The fault response is triggered with a corresponding delay.

i The measures specified above are not necessary if the plant design precludes surges within the system.

In particular, the design must include protective measures with respect to overvoltage, lightning, ground grounding and plant wiring in accordance with the relevant standards and the instructions specified in the system manual (HI 801 001 EE).

7.5 Safety-Related Analog Input Modules

Analog input channels convert the measured input currents to a value of type DINT (double integer), i.e., the raw value, and to a value of type REAL, i.e., the process value. The Raw Value parameter contains the measured input signal whereas the process value is a scaled value.

Proximity switch inputs create a digital value by comparing the raw value with the configured thresholds.

7.5.1 Test Routines

The module captures the analog values in two ways and compares the results with one another. Additionally, it cyclically tests the input path function.

7.5.2 Redundancy of Analog Inputs

The analog inputs may be connected redundantly. The redundant connection is usually used to increase availability.

7.5.3 State of LL, L, N, H, HH in X-AI 32 01

For safety-related applications of the X-AI 32 01 module, the following applies:

If scalar events were defined for a channel's limit values, the state variables *-> State LL, -> State L, -> State N, -> State H, -> State HH* must be connected to the *Channel OK* variable!

If faults occur, the state variables return FALSE.

7.6 Checklists for Inputs

HIMA recommends using the available checklist for engineering, programming and starting up safety-related inputs. The checklist can be used as a planning document and also serves as proof of careful planning.

When engineering and starting up the system, a checklist must be filled out for each of the safety-related input channels used in the system to verify the requirements to be met. This is the only way to ensure that all requirements were considered and clearly recorded. The checklist also provides documentation about the relationship between the external wiring and the user program.

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

8 Output Modules

Digital output modules	Channels	Safety-related	Remark
X-DO 24 02	24	SIL 4	48 VDC, ≤ 0.5 A
X-DO 32 01	32	SIL 4	24 VDC, ≤ 0.5 A
Relay module [1)]			
X-DO 12 01	12	SIL 4	230 VAC/VDC
[1)] With electrically protective separation			

Table 8: Overview of the Output Modules

8.1 General

Values are written to the safety-related output modules once per cycle, the generated output signals are read back and compared with the specified output data.

The safe state of the outputs is 0 or an open relay contact.

The corresponding error code provides additional options for programming fault responses in the user program.

For further details on the output modules, refer to the module manuals.

8.2 Response in the Event of a Fault

If the test routines detect a faulty output, the controller switches off the affected output, i.e., it enters the safe state. The module activates the *Error* LED.

Failure of the overall output module causes all outputs to enter the safe state.

The error code and other system variables can be used to program application-specific fault responses. For further details, refer to the manual of the corresponding module.

8.3 Safety of Actuators

In safety-related applications, the controller (PES) and connected actuators must all meet the safety requirements and achieve the specified SIL. For details on how to achieve the required SIL for actuators, refer to the IEC 61511-1 standard, Section 11.4.

8.4 Safety-Related Digital Output Modules

The safety-related output channels are equipped with three testable switches connected in series, in addition to individual channel switch-off. This ensures compliance with the SIL 4 requirement for a second safe independent shutdown option. If a fault occurs, this integrated safety shutdown safely de-energizes individual channels of the defective submodule (de-energized state).

Additionally, the watchdog signal of the module is the second shutdown option: If the watchdog signal is lost, the module immediately enters the safe state.

8.4.1 Test Routines

The modules are tested automatically during operation. The main test functions are:

- Reading back the output signals.
- Checking the integrated redundant safety shutdown.
- Shutdown test of the outputs.
- Operating voltage monitoring.

8.4.2 Output Noise Blanking

If the Output Noise Blanking option is activated, the output module delays the channel's switch-off response.

If the Output Noise Blanking option is activated and transient interference has been suppressed, a delay in the response to *safety time - watchdog time* may occur.

In all cases, the module also indicates the fault through the *Error* LED on the front plate.

8.4.3 Behavior in the Event of External Short-Circuit or Overload

If the output is short-circuited to L- or overloaded, the module is still safe.

In this state, the outputs are checked every few seconds to determine whether the overload is still present. In a normal state, the outputs are switched on again.

8.4.4 Redundancy of Digital Outputs

The digital outputs may be connected redundantly. The redundant connection is used to increase availability.

8.5 Safety-Related Relay Modules

Relay modules are connected to the actuator under any of the following circumstances:

- Electric and galvanic separation is required.
- Higher amperages are to be connected.
- Alternating currents are to be connected.

The module outputs are equipped with two safety relays with forcibly guided contacts. The outputs can thus be used for safety shutdowns in accordance with SIL 4.

Additionally, the watchdog signal of the module is the second shutdown option: If the watchdog signal is lost, the module immediately enters the safe state.

8.5.1 Test Routines

The modules are tested automatically during operation. The main test functions are:

- Reading the output signals back from the switching amplifiers located before the relays.
- Testing the switching of the relay with forcibly guided contacts.
- Checking the integrated redundant safety shutdown.
- Operating voltage monitoring.

8.5.2 Redundancy of Relay Outputs

The digital relay outputs may be connected redundantly. The redundant connection is used to increase the availability of relay outputs.

8.6 Checklists for Outputs

HIMA recommends using the available checklist for engineering, programming and starting up safety-related outputs. The checklist can be used as a planning document and also serves as proof of careful planning.

When engineering and starting up the system, a checklist must be filled out for each of the safety-related output channels used in the system to verify the requirements to be met. This is the only way to ensure that all requirements were considered and clearly recorded. The checklist also provides documentation about the relationship between the external wiring and the user program.

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

9 Software

The software for the safety-related HIMax automation system includes the following parts:

- SILworX programming tool in accordance with IEC 61131-3.
- Operating system.
- User program.

The user program, which contains the application-specific functions to be performed by the automation system, is used to create the user program. The programming tool is used to configure and operate the operating system functions of the hardware components.

The code generator integrated in the programming tool translates the user program into a machine code. The programming tool uses the Ethernet interface to transfer this machine code to the flash EPROM of the automation system.

9.1 Safety-Related Aspects of Operating Systems

Each approved operating system is clearly identified by the revision number and the CRC signature. The valid versions of the operating system and corresponding signatures (CRCs) - approved by the TÜV for use in safety-related automation devices - are subject to a revision control and are documented in a version list.

The Revision List of HIMax Systems of HIMA Paul Hildebrandt GmbH is created and maintained by HIMA Paul Hildebrandt GmbH in co-operation with the TÜV Rheinland GmbH.

The current version of the operating system can only be read using the SILworX programming tool. Users must ensure that the operating system versions loaded in the modules are valid.

9.2 Operation and Functions of Operating Systems

The operating system executes the user program cyclically. In a simplified form, it performs the following functions:

- Reading of the input data.
- Processing of the logic functions, programmed in accordance with IEC 61131-3.
- Writing of the output data.

The following basic functions are also executed:

- Comprehensive self-tests.
- Test of inputs and outputs during operation.
- Data transmission.
- Diagnostics.

9.3 Safety-Related Aspects of Programming

When creating or changing a user program, the requirements detailed in this chapter must be observed.

9.3.1 Safety Concept of SILworX

The safety concept for the SILworX programming tool includes the following points:

- When SILworX is installed, a CRC checksum ensures the programming tool's integrity on the way from the manufacturer to the user.
- SILworX performs validity checks to reduce the likelihood of faults while entering data.
- SILworx compiles the program twice and compares the resulting configuration CRCs (checksums) to one another. This ensures that data corruption in the application due to temporary faults in the PC in use is detected.
- SILworX and the measures defined in this safety manual make it sufficiently improbable that a code generated properly from a semantic and syntactic view point can still contain undetected systematic faults resulting from the code generation process.

When starting up a safety-related controller for the first time, a comprehensive functional test must be performed by the user to verify the safety of the entire system.

- Verify whether the control tasks were properly implemented based on the data and signal flows.
- Verify the logic of all functions by trial.

If a user program is changed, at least the program components affected by the change must be tested. The safety-related SILworX version comparison can be used to determine and prove changes compared to a previous version.

Whenever the safety-related controller is started up, the verification and validation requirements specified in the application standards must be observed!

9.3.2 Verifying the Configuration and the User Programs

To check the user programs for compliance with the safety functions, the user must create suitable test cases that validate the specified safety functions.

An independent test of each individual loop (consisting of input, processing including user connections, output) is usually sufficient.

Suitable test cases must be created for the numerical evaluation of formulas. The evaluation can be performed, for instance, using equivalence class tests. The test cases must be selected such that the calculations can be proven to be correct. The required number of test cases depends on the formula used and must include critical value pairs.

HIMA recommend performing an active simulation with data sources. This will prove that the sensors and actuators in the system are properly wired. The same also applies to sensors and actuators that are connected to the system via remote I/Os.

SILworX can be used as test equipment for:

- Checking inputs.
- Forcing outputs.

This procedure must be followed both when initially creating the user program and when modifying it.

9.3.3 Archiving a Project

HIMA recommends archiving the project after each download or reload.

SILworX stores all a project's data to a single file. For reasons of data security, HIMA recommends additionally storing the project on an external medium.

9.3.4 Identifying Configuration and Programs

Changes to a program cause the CRC to change and therefore affect the configuration CRC.

To determine the changes to the current configuration, the project is compared to a saved or loaded configuration. The individual changes can be proved using the safe SILworX version comparison.

9.4 Resource Parameters

Some parameters are defined in SILworX for actions permitted during the resource's safety-related operation and are referred to as safety parameters.

⚠ WARNING

Physical injury possible due to invalid configuration!

Neither the programming tool nor the controller can verify some of configured project-specific parameters. For this reason, enter the safety parameters correctly in the programming tool and verify the whole entry upon completion of the PES load from within the controller.

These parameters are:

- **For the rack ID, refer to Chapter 5.1 and the system manual (HI 801 001 E).**
- **Responsible attribute of system bus or processor modules, see Chapter 5.1 for details.**
- **The parameters marked in Table 9**

Settings that may be defined for safety-related operation are not firmly bound to any specific requirement classes. Instead, each of these must be agreed upon together with the competent test authority for each separate implementation of the controller.

9.4.1 Resource System Parameters

The system parameters of the resource determine how the controller will behave during operation. The system parameters can be set in SILworX, in the *Properties* dialog box of the resource.

System parameter	S [1)]	Description	Setting for safe operation
Name	N	Name of the resource.	Any
System ID [SRS]	Y	System ID of the resource. Range of values: 1…65535 Default value: 60 000 The value assigned to the system ID must differ from the default value, otherwise the project is not able to run!	Unique value within the controller network. This network includes all controllers that can potentially be interconnected.
Safety Time [ms]	Y	For details on the safety time of the resource (in milliseconds), refer to the safety manual (HI 801 003 E). Range of values: 20…22500 ms Default value: 600 ms (can be changed online)	Application-specific
Watchdog Time [ms]	Y	For details on the watchdog time of the resource (in milliseconds), refer to the safety manual (HI 801 003 E). Range of values: 6...7500 ms Default value: 200 ms (can be changed online)	Application-specific
Target Cycle Time [ms]	N	Target or maximum cycle time, see *Target Cycle Time Mode*. Range of values: 0...7500 ms Default value: 0 ms (can be changed online) The maximum target cycle time value may not exceed the configured *Watchdog Time [ms]* minus the minimum value that can be set for *Watchdog Time [ms]* (6 ms, see above); otherwise the entry is rejected. If the default value is set to 0 ms, the target cycle time is not taken into account. For further details, refer to the following chapters.	Application-specific
Target Cycle Time Mode	N	For details on the use of the *Target Cycle Time [ms],* see the following chapters. The default setting is *Fixed-tolerant* (can only be changed online).	Application-specific
Multitasking Mode	N	Mode 1 The duration of a CPU cycle is based on the required execution time for all user programs. Mode 2 The processor provides the execution time portion not needed by lower priority user programs to higher priority user programs. Operation mode for high availability. Mode 3 The processor waits until the execution time not needed by the user programs has expired, thus increasing the cycle. Default value: Mode 1	Application-specific
Max. Com.Time Slice [ms]	N	Highest value in ms for the time slice used for communication during a resource cycle, see the communication manual (HI 801 101 E). Range of values: 2…5000 ms	---

System parameter	S [1)]	Description		Setting for safe operation
		Default value: 60 ms		
Optimized Use of Com. Time Slice	N	The system parameter reduces the response times for communications via processor module(s). i This can affect the temporal utilization of *Max.Com. Time Slice ASYNC [ms]* and the system parameter *Max. Duration of Configuration Connections [ms]* such that these two times can be subject to more demands (e.g., during reload).		---
Max. Duration of Configuration Connections [ms]	N	This defines how much time within a CPU cycle is available for configuration connections. Range of values: 2...3500 ms Default value: 20 ms For further details, refer to the following chapters.		Application-specific
Maximum System Bus Latency [µs]	N	Maximum delay of a message between an I/O module and a processor module. 100...50 000 µs, Default value: *System Defaults* i A license is required for setting the maximum system bus latency to a value ≠ *System Defaults*.		Application-specific
Allow Online Settings	Y	TRUE:	**All** the switches/parameters listed under FALSE can be changed online using the PADT. This is only valid if the system variable *Read-only in RUN* has the value FALSE. Default value: TRUE.	HIMA recommends using the FALSE setting.
		FALSE:	The following parameters **cannot** be changed online: ▪ *System ID* ▪ *Autostart* ▪ *Global Forcing Allowed* ▪ *Global MultiForcing Allowed* ▪ *Global Force Timeout Reaction* ▪ *Load Allowed* ▪ *Reload Allowed* ▪ *Start Allowed* The following parameters can be changed online if *Reload Allowed* is TRUE. ▪ *Watchdog Time (for the resource)* ▪ *Safety Time* ▪ *Target Cycle Time* ▪ *Target Cycle Time Mode*	
		Allow Online Settings can only be TRUE when the controller is stopped or by performing a reload.		

System parameter	S [1)]	Description		Setting for safe operation
Autostart	Y	TRUE:	If the processor module is connected to the supply voltage, the user programs start automatically. Default value: TRUE.	Application-specific
		FALSE:	The user program does not start automatically after connecting the supply voltage.	
		Observe the settings in the resource program properties!		
Start Allowed	Y	TRUE:	Cold start or warm start permitted with the PADT in RUN or STOP. Default value: TRUE.	Application-specific
		FALSE:	Start not allowed.	
Load Allowed	Y	TRUE:	Configuration download is allowed. Default value: TRUE.	Application-specific
		FALSE:	Start not allowed.	
Reload Allowed	Y	TRUE:	Configuration reload is allowed. Default value: TRUE.	Application-specific
		FALSE:	Configuration reload is not allowed. A running reload process is not aborted when switching to FALSE.	
Global Forcing Allowed	Y	TRUE:	Global forcing is permitted for this resource. Default value: TRUE.	Application-specific
		FALSE:	Global forcing is not permitted for this resource.	
Global Force Timeout Reaction	N	Specifies how the resource should behave when the global force timeout has expired: ▪ Stop Forcing Only. ▪ Stop Forcing and Stop Resource. Default value: Stop Forcing Only.		Application-specific
Global MultiForcing Allowed	Y	TRUE:	Users with MultiForcing access can write force data (force values and individual force switches) for global variables in a resource if the required higher-order conditions have been met and the force permissions have been granted.	Application-specific
		FALSE:	Users with MultiForcing access cannot force global variables. Default value: FALSE (can be changed online)	

System parameter	S [1)]	Description		Setting for safe operation
Minimum Configuration Version	N	With this setting, it is possible to generate code that is compatible with previous or newer HIMax operating system versions in accordance with the project requirements. Default value: SILworX V11 for new projects.		Application-specific
		SILworX V2	The code is generated like in SILworX V2 for HIMax prior to V3.	
		SILworX V3	The code is generated like in SILworX V3 for HIMax V3.	
		SILworX V4	The code is generated like in SILworX V4 for HIMax V4.	
		SILworX V5	The code is generated like in SILworX V5 for HIMax V5.	
		SILworX V6	The code is generated like in SILworX V6.48 for HIMax V6.	
		SILworX V6b	The code is generated like in SILworX V6.114 for HIMax V6.	
		SILworX V7	The code is generated like in SILworX V7 for HIMax V7.	
		SILworX V8	The code is generated like in SILworX V8 for HIMax V8.	
		SILworX V9	The code is generated like in SILworX V9 for HIMax V9.	
		SILworX V10	The code is generated like in SILworX V10 for HIMax V10.	
		SILworX V11	The code is generated like in SILworX V11 for HIMax V11.	
Fast Start-Up	N	Not applicable to HIMax.		---
[1)] The operating system handles the system parameter in a safety-related manner, yes (Y) or no (N).				

Table 9: Resource System Parameters

9.4.1.1 Use of the Parameters *Target Cycle Time* and *Target Cycle Time Mode*

Using the settings for the *Target Cycle Time Mode* system parameter, the cycle time can be maintained as constant as possible at the value of *Target Cycle Time [ms]*. To do this, the system parameter must be set to a value > 0.

In doing so, HIMax limits reload and synchronization on the redundant modules to ensure that the target cycle time is maintained.

The following table describes the settings for the *Target Cycle Time Mode* system parameter.

Setting	Description
Fixed	If a CPU cycle is shorter than the defined Target Cycle Time, the CPU cycle is extended to the target cycle time. If the CPU cycle takes longer than the target cycle time, the CPU resumes the cycle without delay. i A reload or synchronization process is rejected if the reserve time is not sufficient (target cycle time minus actual cycle time).
Fixed-tolerant	Similar to *Fixed*, but with the following differences: 1. To ensure that the synchronization process can be performed successfully, the target cycle time may be violated for a CPU cycle. 2. To ensure that the reload can be performed successfully, the target cycle time may be violated for 1 to n CPU cycles (where n is the number of changed user programs). The default setting is *Fixed-tolerant*! i After the first reload activation cycle, the values of watchdog time, target cycle time and target cycle time mode apply in accordance with the new configuration. A maximum of every fifth cycle can be extended during the reload. One single cycle may be extended during synchronization.
Dynamic	The CPU processes each CPU cycle as fast as possible. This corresponds to a target cycle time of 0 ms. i A reload or synchronization process is rejected if the reserve time is not sufficient (target cycle time minus actual cycle time). A maximum of every fifth cycle can be extended during the reload. One single cycle may be extended during synchronization.
Dynamic-tolerant	Similar to *Dynamic*, but with the following differences: 1. If necessary, the target cycle time is automatically increased for one CPU cycle to ensure that the synchronization process can be performed successfully. 2. To ensure that the reload can be performed successfully, the target cycle time may be automatically increased for 1 to n CPU cycles (where n is the number of changed user programs). i After the first reload activation cycle, the values of watchdog time, target cycle time and target cycle time mode apply in accordance with the new configuration. A reload or synchronization process is rejected if the reserve time is not sufficient (target cycle time minus actual cycle time).

Table 10: Settings for Target Cycle Time Mode

9.4.1.2 Maximum Communication Time Slice

The maximum communication time slice is the time period in milliseconds (ms) per CPU cycle assigned to the processor module for processing the communication tasks. Even if the protocol processing could not be completed within one communication time slice, the CPU still executes the safety-relevant monitoring for all protocols within one CPU cycle.

i If not all upcoming communication tasks can be processed within one CPU cycle, the whole communication data is transferred over multiple CPU cycles. The number of communication time slices is then greater than 1.

For calculating the maximum response time, the number of communication time slices must be equal to 1.

9.4.1.3 Determining the Maximum Duration of the Communication Time Slice

For a first estimate of the maximum duration of the communication time slice, the sum of the following times must be entered in the *Max. Com. Time Slice [ms]* system parameter located in the properties of the resource.

- For each X-COM module: 3 ms.
- For each redundant safe**ethernet** connection: 1 ms.
- For non-redundant safe**ethernet** connection: 0.5 ms.
- For each kilobyte user data of non-safety-related protocols, e.g., Modbus: 1 ms.

HIMA recommends comparing the value estimated for *Max. Com. Time Slice [ms]* with the value displayed in the Control Panel and, if necessary, correcting it in the properties of the resource. This can be done during an FAT (factory acceptance test) or SAT (site acceptance test).

To determine the actual duration of the maximum communication time slice

1. Operate the HIMax system under full load (FAT, SAT):
   All communication protocols are in operation (safe**ethernet** and standard protocols).
2. Open the **Control Panel** and select the **Com. Time Slice** structure tree folder.
3. Read the vale displayed for *Maximum Com. Time Slice Duration per Cycle [ms].*
4. Read the value displayed for *Maximum Number of Required Com. Time Slice Cycles*.

The duration of the communication time slice must be set so that, when using the communication time slice, the CPU cycle cannot exceed the watchdog time specified by the process.

9.4.1.4 Calculating the *Maximum Duration of Configuration Connections [ms]* T_{Config}

The *Max. Duration of Configuration Connections [ms]* system parameter corresponds to the time budget ($\mathbf{t}_{Config}$) required for the system-internal communication connections (tasks):

- PADT online connections (e.g., download/reload, OS update, online test, diagnostics).
- Remote I/O status connections (start, stop and diagnostics).
- Configuration of modules (e.g., loading of replaced modules).

If these tasks cannot be completed within one CPU cycle, the remaining tasks are processed in the next CPU cycle. This can cause unexpected delays for these tasks.

i HIMA recommends dimensioning $\mathbf{t}_{Config}$ in such a way that all tasks can be processed in a single CPU cycle.

$\mathbf{t}_{Config}$ for HIMax CPU operating systems ≤ V3 is fixed and set by SILworX to 6 ms. The time required to process the mentioned tasks may, however, exceed the default value in a CPU cycle.

$\mathbf{t}_{Config}$ for HIMax CPU operating systems ≥ V4 is calculated as follows:

X-CPU 01: $\mathbf{t}_{Config} = (\mathbf{n}_{Com} + \mathbf{n}_{PADT} + \mathbf{n}_{RIO}) * 0.25\ ms + 4\ ms + 4 * (\mathbf{t}_{Latency} * 2 + 0.31\ ms)$

X-CPU 31: $\mathbf{t}_{Config} = (\mathbf{n}_{Com} + \mathbf{n}_{PADT}) * 1\ ms + \mathbf{n}_{RIO} * 0.25\ ms + 4\ ms + 4 * (\mathbf{t}_{Latency} * 2 + 0.8\ ms)$

$t_{Config:}$	System parameter *Max. Duration of Configuration Connections [ms].*
$n_{COM:}$	Number of modules with Ethernet interfaces (X-SB, X-CPU, X-COM).
$n_{PADT:}$	5, maximum number of PADT connections.
$n_{RIO:}$	Number of configured remote I/Os.
$t_{Latency:}$	Use the active maximum system bus latency, see the following descriptions. If the value of the maximum system bus latency is expressed in µs, it must be divided by 1000 before the calculation to obtain the value in ms.

Depending on which system bus structure was selected for the HIMax system, the following value must be used for the system bus latency:

Network structure:	If 100…50 µs was manually entered for *Maximum System Bus Latency [µs]*, then this value must be used in the equation as $\mathbf{t}$*Latency*.
Line structure:	If *Maximum System Bus Latency [µs]* is set to System Defaults, the standard value of the maximum system bus latency specified for $\mathbf{t}$*Latency* in the following table should be used in the equation. As an alternative to the value indicated in the table, the maximum value can first be used: 550.4 µs for X-CPU 01 and 1166.4 µs for X-CPU 31.

When generating the code or converting the project, a warning message is displayed in the PADT logbook if the value defined for $\mathbf{t}_{Config}$ is less than the value resulting from the previous equation.

i Setting the value for $\mathbf{t}_{Config}$ too low can significantly impair the performance of PADT online connections (tasks) and cause the connection to remote I/Os to be aborted.

HIMA recommends comparing the value calculated for $\mathbf{t}_{Config}$ with the value displayed in the Control Panel and, if necessary, correcting it in the properties of the resource. This can be done during a SAT (site acceptance test).

For test purposes, $\mathbf{t}_{Config}$ can also be set online in the Control Panel.

The value set for $\mathbf{t}_{Config}$ must be taken into account for dimensioning the required watchdog time. For details, refer to the section on safety-relevant time parameters.

9.4.1.5

Maximum rack distance	Maximum system bus latency in µs				Example: the system consists of the mentioned racks
	X-CPU 01		X-CPU 31		
	Min	Max [1)]	Min	Max [1)]	
0	49.1	-	665.2	-	Only rack 0
1	105.5	155.5	721.6	771.6	Racks 0 and 1
2	161.9	211.9	778.0	828.0	Racks 0, 1, 3
3	218.4	268.4	834.4	884.4	Racks 0, 1, 3, 5
4	274.8	324.8	890.8	940.8	Racks 0, 1, 3, 5, 7
5	331.2	381.2	947.2	997.2	Racks 0, 1, 3, 5, 7, 9
6	387.6	437.6	1003.6	1053.6	Racks 0, 1, 3, 5, 7, 9, 11
7	444.0	494.0	1060.9	1110.9	Racks 0, 1, 3, 5 , 7, 9, 11, 13,
8	500.4	550.4	1116.4	1166.4	Racks 1, 0, 2, 4, 6, 8, 10, 12, 14
1) Maximum system bus latency including the maximum additional latency caused by the network infrastructure					

Table 11: Default Values for Maximum System Bus Latency

9.4.1.6 The *Minimum Configuration Version* Parameter

- The highest *Minimum Configuration Version* is always selected for new projects. Verify that this setting is in accordance with the operating system version in use.
- In a previous project converted to the current SILworX version, the value for *Minimum Configuration Version* remains the value set in the previous version. This ensures that the configuration CRC resulting from the code generation is the same as in the previous version and the configuration is still compatible with the operating systems of the modules.

  The value of *Minimum Code Generation* only needs to be increased for converted projects if additional functions of a controller should be used.
- If features requiring a higher configuration version are used in the project, SILworX automatically generates a configuration version higher than the preset *Minimum Configuration Version*. This is indicated by SILworX in the code generation logbook. The modules reject loading configurations if their version and operating system do not match.

  The safety-related SILworX version comparison can be used to determine and prove changes performed to the current project version compared to a previous one.

9.4.1.7 System Variables of Racks

These system variables are used to change the behavior of the controller while it is operating in specific states. These variables can be set in the *System* tab located in the rack detail view of the SILworX Hardware Editor.

System variables	S [1)]	Function	Setting for safe operation
Force Deactivation	Y	Prevents the forcing process from starting and terminates a running forcing process. Default setting: FALSE.	Application-specific
Spare 0…Spare 16	Y	No function!	---
MultiForcing Denied	Y	MultiForcing can be enabled and disabled using the *MultiForcing Denied* system variable so that the associated functions can be controlled by the user program. For MultiForcing, the system variable must be set to FALSE. Default setting: FALSE.	Application-specific
Emergency Stop 1...Emergency Stop 4	Y	Shuts down the controller if faults are detected by the user program. Default setting: FALSE.	Application-specific
Read-only in RUN	Y	After the controller is started, the access permissions are downgraded to *Read-Only*. Exceptions are forcing and reload. Default setting: FALSE.	Application-specific
Reload Deactivation	Y	Locks the execution of reload. Default setting: FALSE.	Application-specific
[1)] The operating system handles the system parameter in a safety-related manner, yes (Y) or no (N).			

Table 12: Hardware System Variables

Global variables can be connected to these system variables; the value of the global variables is modified using a physical input or the user program logic.

9.4.2 Locking and Unlocking the Controller

Locking the controller locks all functions and prevents users from accessing them during operation. This also protects against unauthorized manipulations to the user program.

Unlocking the controller deactivates any locks previously set, e.g., to perform work on the controller.

The system variables *Read-Only in RUN*, *Reload Deactivation*, *Force Deactivation* and *MultiForcing Denied* are used to lock the controller.

If all of the above system variables are TRUE, no access to the controller is possible. In this case, the controller can only enter the STOP state by restarting all processor modules. Only then can a new user program be loaded. The example describes a simple case, in which a key-operated switch is used to lock or unlock all interventions to the resource.

Example: To make a controller lockable

1. Define global variables of type BOOL and set initial values to FALSE.
2. Assign the global variable as output variables to the above system variables.
3. Assign the global variable to the channel value of a digital input.
4. Connect a key switch to the digital input.
5. Compile the program, load it into the controller, and start it.

► The owner of a corresponding key-operated switch is able to lock and unlock the controller. If the corresponding digital input module fails, the controller is automatically unlocked.

This simple example can be modified using multiple global variables, digital inputs and key switches. The permissions for forcing, reload, MultiForcing and other operating functions can be distributed on different keys and persons.

9.5 Forcing

Forcing is the procedure of manually writing to variables with values that do not result from the process, but are defined by the user, while the controller is processing the user program.

There are different types of globally forcible data sources in a system:

- All input and status information from modules (e.g., I/O modules) and communication protocols.
- All global variables that have not been written, but have been read (VAR_EXTERNAL).
- All global variables that have been written to by a user program (VAR_EXTERNAL).

In addition to the globally forcible data sources in a system, there are also different types of locally (in the user program) forcible data sources:

- All user program variables that have not been written, but have been read (VAR).
- All variables from a user program that have been written (VAR).

i When a variable is forced, forcing always applies to its data source! A forced variable does not depend on the process since its value is defined by the users.

9.5.1 Use of Forcing

Forcing supports users during the following tasks:

- Testing of the user program for cases that do not, or only infrequently occur during normal operation and are therefore only testable up to a certain extent.
- Simulation of sensor values, e.g., of unconnected sensors.
- Service and repair work.
- General troubleshooting.

⚠ WARNING

Physical injury due to forced values is possible!

- **Only force values after consent of the person responsible for the plant and the test authority during commissioning.**
- **Only remove existing forcing restrictions with the consent of the person responsible for the plant and the test authority during commissioning.**

When forcing values, the person in charge must take further technical and organizational measures to ensure that the process is sufficiently monitored in terms of safety. HIMA recommends setting a time limit for the forcing procedure, refer to Chapter 9.5.3 for details.

⚠ WARNING

Failure of safety-related operation possible due to forced values!

- **Forced value may lead to unexpected output values.**
- **Forcing prolongs the cycle time. This can cause the watchdog time to be exceeded.**

Forcing can operate at two levels:

- Global forcing: Global variables are forced for all applications.
- Local forcing: Local variables are forced within a user program.

9.5.2 Assigning a Data Source Changed through Reload

Assigning variables to a new data source by performing a reload may have unexpected results in conjunction with the following inputs:

- Hardware.
- Communication protocols.
- System variables.

The following changes resulting from a reload lead to changed force states:

1. A global variable A is assigned to a forced data source and is thus forced itself.
2. The assignment of global variable A is removed by performing a reload. The data source maintains the property *Forced*. Global variable A is no longer forced.
3. The forced data source is assigned another global variable (global variable B).
4. During the next reload, global variable B will be forced, even if unintentionally.

Consequence

To prevent this effect, stop forcing a variable before changing the data source. To this end, deactivate the individual force switch.

The *Inputs* tab in the Force Editor displays which channels are being forced.

i Global variables having the user program as data source retain the *forced* setting even when the assignment is changed.

9.5.3 Time Limits

Different time limits can be set for global or local forcing. Once the defined time has expired, the controller stops forcing values.

The behavior of the HIMax system upon expiration of the time limit can be configured:

- For global forcing, the following settings can be selected:
  - *Stop Resource.*
  - *Stop Forcing Only,* i.e., the resource continues to operate.
- For local forcing, the following settings can be selected:
  - *Stop Program.*
  - *Stop Forcing Only,* i.e., the user program continues to run.

Forcing can also be used without time limit. In this case, the forcing procedure must be stopped manually.

The person responsible for forcing must clarify what effects stopping forcing have on the entire system!

9.5.4 Restricting the Use of Forcing

The user can limit the use of forcing; disturbed operation which may be caused by forcing, is to be avoided. The following measures can be implemented in the configuration:

- Configuration of different user profiles with or without forcing permissions.
- Explicit enabling of forcing for a resource (PES).
- Set-up of MultiForcing user accounts in the PES User Management.
- Explicit enabling of local forcing for a user program.
- Immediate stop of forcing via the *Force Deactivation* system variable using the key switch.
- Disabling of MultiForcing through the *MultiForcing Denied* system variable.

9.5.5 MultiForcing

Users with MultiForcing access can write force data (force values and individual force switches) for global variables in a resource if the required higher-order conditions have been met and the force permissions have been granted. To all other functions of a resource, users have Read-Only access. Starting, stopping or resetting a force process is not possible.

The use of MultiForcing is limited to a maximum of 5 users at a time. The users can be working from separate locations and also independently of each other in terms of time. The separation of the tasks performed by the individual users must be ensured by the operator through organizational measures.

⚠ WARNING

Behavior that cannot be controlled by the user, is possible!

The operator must ensure that different Force Users do not force the same variables simultaneously and that there can be no overlaps in timing. If several Force Users write to the same variables, those force values and force switches will prevail which were written last by the firmware. Because force data are transferred in several blocks , it would otherwise be possible for the settings of different Force Users to take effect on one single controller. This behavior cannot be controlled by the user.

⚠ WARNING

Existing force data is not deactivated, if *MultiForcing Denied* = TRUE!

If *MultiForcing Denied* is TRUE, users with MultiForcing access cannot modify force values or the force switches. Existing force data is not deactivated, if *MultiForcing Denied* = TRUE! Global Forcing, if allowed, is then only possible for a single user with at least Operator permissions.

For further details on forcing, refer to the system manual (HI 801 001 E) and the SILworX online help.

9.5.5.1 Objectives of MultiForcing

For commissioning, normative and functional loop tests are prescribed as part of the site acceptance test, whereby a loop represents the path from the sensor to the actuator. MultiForcing makes it possible to distribute the resulting tasks to up to 5 PADTs thus processing them efficiently.

Based on loop tests, the nominal operating range is checked as well as the responses in the event of open-circuits and short-circuits. Because numerous loops must be tested frequently, the duration of site acceptance testing is a significant cost factor. MultiForcing can help to optimize these tasks.

- The behavior of actuators and linked information (e.g., end position feedback) is tested through forcing. The output signals are forced directly. This tests the wiring and the external circuit.
- In a system which is only partially functional, sensors are tested through forcing in such a way that the tests have no effect on the actuators. This approach can also be used for troubleshooting in connection with sensors.

9.5.5.2 Global MultiForcing

Global MultiForcing is the simultaneous writing of force data (force values and force switches) for global variables by more than one user (Force Users).

A Force User is a person who is logged into a controller with either MultiForcing, Operator, Write or Administrator permissions. Every Force User is able to read and also at least write force data. A maximum of 5 Force Users can be logged into each controller. The number of current Force Users is displayed in the SILworX status bar.

Force values and force switches set by a Force User with MultiForcing access may only take effect if the user is logged into the controller with at least Operator permissions. Only this user can start or stop forcing.

i To perform Global MultiForcing, Global Forcing must be allowed as well! The settings are displayed online.

9.6 Safe Version Comparison

During the code generation, SILworX creates various files. This data set is referred to as the resource configuration. The complete resource configuration is loaded to the resource whenever a download or reload is performed.

During a safe version comparison, different resource configurations are compared to one another and the differences between the individual files are detected.

Essentially, there are three types of resource configurations:

1. The created resource configuration which is the result of the last code generation.
2. The loaded resource configuration which is the configuration that was loaded into the controller by performing a reload or download.
3. An unknown resource configuration which was exported and saved. This represents any state of the resource configuration.

To verify the program changes, the safe version comparison must be started **before** the program is loaded to the controller.

The version comparison exactly determines the changed parts of the resource configuration. This facilitates testing and identifying the changes. The result has SIL 4 quality and may be submitted to the inspection authority as a piece of evidence.

Structured programming, and the use of significant names from the first resource configuration on, facilitate understanding of the comparison result.

For further details, refer to the version comparison manual (HI 801 286 E).

10 Safety-Related Aspects of User Programs

This chapter describes the safety-related aspects that are important for the user programs.

Programming goals for a user program:

- Understandable.
- Traceable.
- Testable.
- Easy to modify.

10.1 Safety-Related Usage

The user programs must be created with the programming tool SILworX.

SILworX can only be installed on a PC with Microsoft Windows operating system. The minimum requirements for the computer used to run SILworX are specified on the corresponding installation DVD.

The SILworX programming tool includes the following functions:

- Global Variable Editor (for creating global variables with symbolic names and data types).
- Hardware Editor (for assigning the controllers of the HIMax system).
- FBD Editor (for creating the user program).
- Code generator (for translating the user program into a machine code).
- Configuration of communication.
- Monitoring and documentation.

The safety requirements specified in this manual must be observed, see Chapter 3.4.

10.1.1 Programming Basics

The tasks to be performed by the controller must be defined in a specification or a requirements specification. This documentation serves as the basis for checking its proper implementation in the user program.

The documentation depends on the control task and can be represented in two ways.

Combinational logic:

- Cause/effect diagram.
- Logic of the connection with functions and function blocks.
- Function blocks with specified characteristics.

Sequential controllers (sequence control system):

- Written description of the steps and their enabling conditions and of the actuators to be controlled.
- Flow charts.
- Matrix or table form of the step enabling conditions and the actuators to be controlled.
- Definition of constraints, e.g., operating modes, emergency stop.

10.1.1.1 I/O Concept

The I/O concept of the system must include the analysis of the field circuits, i.e., the type of sensors and actuators:

Digital and analog sensors:

- Signals during normal operation (de-energize to trip principle with digital sensors, 'life-zero' with analog sensors).
- Signals if a fault occurs.
- Definition of safety-related redundancies required for safety (1oo2, 2oo3).
- Discrepancy monitoring and response.

Actuators:

- Positioning and activation during normal operation.
- Safe response/positioning at shutdown or after power loss.

10.1.2 Programming Steps

To program HIMax systems for safety-related applications, perform the following steps:

1. Specify the control functions.
2. Write the user programs.
3. Compile the user programs using the C code generator.
   - The user programs are free from errors and able to run.
4. Verify and validate the user programs.
5. Tests the user programs.

After these steps, the user programs are ready to start safety-related operation!

10.1.3 User Program Functions

The user program functions can be freely programmed.

- Only elements complying with IEC 61131-3 together with their functional requirements are used within the logic.
- The physical inputs and outputs usually operate in accordance with the de-energize to trip principle, i.e., their safe state is 0.
- The user programs are built of logic and/or arithmetic functions irrespective of the de-energize to trip principle of the physical inputs and outputs.
- The program logic should be clear and easy to understand and well documented to assist in debugging. This includes the use of functional diagrams.
- To simplify the logic, the inputs and outputs of all function blocks and variables can be inverted in any given order.
- The programmer must evaluate the fault signals from the inputs/outputs or from logic blocks.

HIMA recommends encapsulating functions to user-specific function blocks and functions based on standard functions. This ensures that user programs can be clearly structured in modules (functions, function blocks). Each module can be viewed and tested on an individual basis. By grouping modules into larger ones and combining them into a single user program, users are effectively creating a comprehensive, complex function.

10.1.4 User Program System Parameters

The following user parameters can be set in the *Properties* dialog box of the user programs:

System parameters	S [1)]	Description		Setting for safe operation
Name	N	Name of the user program. The name must be unique within the resource.		Any
Program ID	Y	ID for identifying the program when displayed in SILworX. Range of values: 0…4 294 967 295 Default value: 0 If *Code Generation Compatibility* is set to *SILworX V2*, only the value 1 is permitted.		Application-specific
Priority	Y	Priority of the user program. Range of values: 0…31 Default value: 0 (highest priority) This setting is only required if several user programs are used!		Application-specific
Program's Maximum Number of CPU Cycles	Y	Maximum number of CPU cycles that a user program cycle may take. Range of values: 1…4 294 967 295 Default value: 1 This setting is only required if several user programs are used!		Application-specific
Max. Duration for Each Cycle [µs]	N	Maximum time in each processor module cycle for executing the user program. Range of values: 0…4 294 967 295 Standard value: 0 (no limitation) The safety-related response is ensured through the watchdog. This setting is only required if several user programs are used!		Application-specific
Watchdog Time [ms] (calculated)	---	Monitoring time of the user program, calculated from the product of the watchdog time of the resource and the configured maximum number of CPU cycles. Not changeable!		
Classification	N	Classification of the user program in *Safety-related* or *Standard*; the setting is for documentation only and has no effects on the program's performance. Default value: Safety-related.		Application-specific
Allow Online Settings	Y	If *Allow Online Settings* is deactivated, the settings of the reamaining program switches cannot be changed online (from within the Control Panel). Only applies if the *Allow Online Settings* switch for the resource is set to TRUE! Default value: TRUE.		
Autostart	Y	Enabled type of Autostart: Cold Start, Warm Start, Off. Default value: Warm start.		Application-specific
Start Allowed	Y	TRUE:	The PADT may be used to start the user program. Default value: TRUE.	Application-specific
		FALSE:	The PADT may not be used to start the user program.	

System parameters	S [1]	Description		Setting for safe operation
Test Mode Allowed	Y	TRUE:	The test mode is permitted for the user program.	Application-specific [2]
		FALSE:	The test mode is not permitted for the user program. Default value: FALSE.	
Reload Allowed	Y	TRUE:	The user program reload is permitted. Default value: TRUE.	Application-specific
		FALSE:	The user program reload is not permitted.	
		Observe the settings in the resource properties!		
Local Forcing Allowed	Y	TRUE:	Forcing is permitted at program level.	FALSE is recommended
		FALSE:	Forcing is not permitted at program level. Default value: FALSE.	
Local Force Timeout Reaction	Y	Behavior of the user program after the forcing time has expired: ▪ Stop Forcing Only. ▪ Stop Program. The default setting is *Stop Forcing Only*.		
Code Generation Compatibility	-	Code generation is compatible with previous versions of SILworX.		Application-specific
		SILworX V2	Code generation is compatible with SILworX V2.	
		SILworX V3	Code generation is compatible with SILworX V3.	
		SILworX V4 – V6b	Code generation is compatible with SILworX V4 up to SILworX V6b.	
		SILworX V7 and higher	Code generation is compatible with SILworX V7.	
		The default setting for all new projects is *SILworX V7 and higher*.		

[1] The operating system handles the system parameter in a safety-related manner, yes (Y) or no (N)

[2] Once the test mode has stopped, a cold start must be performed prior to starting a safety-related operation!

Table 13: System Parameters of the User Program

10.1.5 Notes on the *Code Generation Compatibility* Parameter

Observe the following points in conjunction with the *Code Generation Compatibility* parameter:

- In a new project, SILworX selects the current setting for the *Code Generation Compatibility* parameter. This ensures that the current, enhanced features are activated and the current module and operating system versions are supported. Verify that this setting is in accordance with the hardware in use.
- In a previous project converted to the current SILworX version, the value for *Code Generation Compatibility* remains the value set in the previous version. This ensures that the configuration CRC resulting from the code generation is the same as in the previous version and the configuration is still compatible with the operating systems of the modules.
  The value of *Code Generation Compatibility must only be changed for converted projects if additional functions of a controller should be used.*
- If a *Minimum Configuration Version* of *SILworX V4* and higher is set in the resource properties, the *Code Generation Compatibility* parameter must be set to *SILworX V7 and Higher* in every user program.

10.1.6 Code Generation

The code is generated after entering the complete user program and the I/O assignments of the controller. During these steps, the configuration CRC, i.e., the checksum for the configuration files, is created.

This is a signature for the entire configuration and is issued as a 32-bit, hexadecimal code. It includes all of the configurable or modifiable elements such as the logic, variables or switch parameter settings.

i Before loading a user program for safety-related operation, the user program must first be compiled twice. The two generated versions must have the same checksum.

By default, SILworX automatically compiles the resource configuration twice and compares the checksums.

The result of the CRC comparison is displayed in the logbook.

By compiling the user program twice and comparing the checksums of the generated code, the user can detect potential corruptions of the user program resulting from random faults in the hardware or operating system of the PC in use.

10.1.7 Loading and Starting the User Program

A resource configuration can only be loaded into a controller through download if the controller is in the STOP state.

The user program can be started after successful resource configuration download.

i The PADT is only able to operate the controller, e.g., by performing a reload and forcing, if the project matching the resource configuration is opened in SILworX.

HIMA recommends archiving the project after each download or reload.

SILworX stores all a project's data to a single file. For reasons of data security, HIMA recommends additionally storing the project on an external medium.

The backup ensures that the project data matching the resource configuration remains available even if the PADT fails.

10.1.8 Reload

If changes were performed to a project, they can be transferred to the controller by performing a reload. After being tested by the operating system, the modified project is activated and assumes the control task.

The reload can only be performed if the *Reload Allowed* system parameter is set to TRUE and the *Reload Deactivation* system variable is set to FALSE.

i A reload is only permitted after receiving consent from the test authority responsible for the acceptance test. During the entire reload process, the person in charge must take further technical and organizational measures to ensure that the process is sufficiently monitored in terms of safety.

Observe the following points when reloading sequence chains:

The reload information for sequence chains does not take the current sequence status into account. A reload can therefore cause the sequence to change setting it to an undefined state. The user is responsible for properly performing the reload.

Examples:

- Deletion of the active step causes all the steps within the step sequence to lose the *active* state!
- Renaming an initial step while another step is active leads to a step sequence with two active steps!

Observe the following points when reloading actions:

During the reload, actions are loaded with their complete data. All potential consequences must be carefully analyzed prior to performing a reload.

Examples:

- If a timer action qualifier is deleted due to the reload, the timer expires immediately. Depending on the remaining settings, the *Q* output can therefore be set to TRUE.
- If the status action qualifier (e.g., the *S* action qualifier) is deleted for a set element, the element remains set.
- Removing a *P0* action qualifier set to TRUE actuates the trigger function.

Prior to performing a reload, the operating system checks if the required additional tasks would increase the cycle time of the current user programs to such an extent that the defined watchdog time is exceeded. In this case, the reload process is aborted with an error message and the controller continues operation with the previous resource configuration.

The controller can abort a reload.

Reload can be performed successfully by planning a sufficient reserve for the reload when determining the watchdog time or temporarily increasing the controller watchdog time by a reserve.

Any temporary increases in the watchdog time must be agreed upon with the competent test authority.

Exceeding the target cycle time can also lead to an abort of the reload.

The user is responsible for ensuring that the watchdog time includes a sufficient reserve time. This should allow the user to manage the following situations:

- Variations in the user program's cycle time.
- Sudden, strong cycle loads, e.g., due to communication.
- Expiration of time limits during communication.

10.1.9 Online Test

Online test fields (OLT fields) can be used in the user program logic to display variables while the controller is operating.

For further details on how to use OLT fields, use OLT field as keyword in the SILworX online help and refer to the SILworX first steps manual (HI 801 103 E).

10.1.10 Test Mode

To diagnose faults, the user program operating in online mode can be run in single steps, i.e., cycle by cycle. Each cycle is triggered by a command from the PADT. In the period between two cycles, the global variables written to by the user program remain **frozen**. The assigned physical outputs and communication data then no longer respond to process changes!

The test mode can be configured individually for each user program by activating or deactivating the *Test Mode Allowed* parameter.

Test Mode Allowed	Description
Deactivated	Test mode deactivated (default setting).
Activated	Test mode activated.

Table 14: User Program Parameter *Test Mode Allowed*

NOTICE

Failure of safety-related operation possible!

If the user program is frozen in test mode, it cannot provide a safety-related response to inputs and thus control the outputs! The values of the outputs cannot change in test mode.

Test mode is therefore not permitted in safety-related operation!

For safety-related operation, the *Test Mode Allowed* parameter must be deactivated!

10.1.10.1 Changing the System Parameters during Operation

The system parameters specified in Table 15 may be changed during operation (online).

A typical application case is the temporary increase of the watchdog time to be able to perform a reload.

Prior to using an online command to set parameters, make sure that this change will not result in a dangerous state of the plant. If required, organizational and/or technical measures must be implemented to preclude any damage. The application standards must be observed!

The safety time and watchdog time values must be checked and compared to the safety time required by the application and to the actual cycle time. These values cannot be verified by the controller!

The controller ensures that the watchdog time is not set to a value less than the watchdog time value of the configuration loaded in the controller.

Parameter	Can be changed in the following controller state
System ID	STOP
Watchdog Time (for the resource)	RUN, STOP/VALID CONFIGURATION
Safety Time	RUN, STOP/VALID CONFIGURATION
Target Cycle Time	RUN, STOP/VALID CONFIGURATION
Target Cycle Time Mode	RUN, STOP/VALID CONFIGURATION
Allow Online Settings	TRUE -> FALSE: All FALSE -> TRUE: STOP
Autostart	All
Start Allowed	All
Load Allowed	All
Reload Allowed	All
Global Forcing Allowed	All
Global Force Timeout Reaction	All
Global MultiForcing Allowed	All

Table 15: Online Changeable Parameters

10.1.11 Project Documentation for Safety-Related Applications

The SILworX programming tool allows the user to automatically print the documentation for a project. The most important document types include:

- Interface declaration.
- Signal list.
- Logic.
- Description of data types.
- Configurations for system, modules and system parameters.
- Network configuration.
- List of signal cross-references.

This documentation is required for the factory acceptance test (FAT) of a system subject to approval by a test authority, e.g., TÜV.

10.1.12 Multitasking

Multitasking refers to the capability of the HIMax system to process up to 32 user programs within the processor module.

The individual user programs can be started and stopped independently from one another.

A user program cycle can take multiple processor module cycles. This can be controlled with the resource and user program parameters. SILworX uses these parameters to calculate the user program watchdog time:

Watchdog Time$_{\text{User program}}$ = **Watchdog Time**$_{\text{Processor module}}$ * **Maximum Number of Cycles**

The individual user programs operate in an interference-free manner and independently from one another. However, reciprocal influence can be caused by:

- Use of the same global variables in several user programs.
- Unpredictably long runtimes can occur in individual user programs if no limit is configured with *Max. Duration for Each Cycle*.
- The distribution of user program cycles over processor module cycles strongly affects the user program response time and the response time of the variables written to by the user program!
- A user program evaluates global variables written to by another user program at the earliest one CPU module cycle later. Depending on the value set for *Program's Maximum Number of CPU Cycles* in the program properties, the evaluation process may be prolonged by many CPU cycles, which also causes a delayed response.

Refer to the system manual (HI 801 001 E) for further details on multitasking.

10.1.13 Factory Acceptance Test and Test Authority

HIMA recommends involving the test authority as soon as possible when designing a system that is subject to approval.

The factory acceptance test (FAT) only applies to the user functionality, but not to the safety-related modules and automation devices of the HIMax system that have already been approved.

10.2 Checklist for Creating a User Program

To comply with all safety-related aspects during the programming phase, HIMA recommends using the checklist prior to and after loading a new or modified program. The checklist can be used as a planning document and also serves as proof of careful planning.

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

11 Configuring Communication

In addition to using the physical input and output variables, variable values can also be exchanged with other systems through a data connection. In this case, the variables are declared with SILworX, in the Protocols area of the corresponding resource.

11.1 Standard Protocols

Many communication protocols only ensure a non-safety-related data transmission. These protocols can be used for the non-safety-related aspects of an automation task.

⚠ WARNING

Physical injury possible due to usage of non-safe import data!

Do not use data imported from non-safe sources for the user program's safety functions.

The following standard protocols are available:

- On the Ethernet interfaces on the communication module:
  - Modbus TCP (master/slave).
  - Modbus, redundant (slave).
  - SNTP.
  - Send/Receive TCP.
  - PROFINET IO (controller, device).
- On the fieldbus interfaces (RS485) of the communication module, depending on the device model:
  - Modbus (master/slave).
  - Modbus, redundant (slave).
  - PROFIBUS DP (master/slave).

11.2 Safety-Related safeethernet Protocol

Use the safe**ethernet** Editor to configure how safety-related communication is monitored.

For further details on safe**ethernet**, refer to the communication manual (HI 801 101 E).

NOTICE

The safe state may be entered inadvertently!

***Receive Timeout* is a safety-related parameter!**

Receive Timeout is the monitoring time of PES 1 within which a correct response must be received from PES 2.

i *Receive Timeout* also applies in the other direction from PES 2 to PES 1!

If a correct response is not received from the communication partner within *Receive Timeout*, HIMax terminates the safety-related communication. The input variables of this safe**ethernet** connection respond in accordance with the preset parameter *Freeze Data on Lost Connection [ms]*. For safety-related functions, which are implemented via safe**ethernet**, the setting **Use Initial Value** must be used.

In the following equations for determining the worst case response time, the target cycle time can be used instead of the watchdog time, if it is guaranteed that the process module maintains the target cycle time, even in case of reload and synchronization.

In this case, the following requirements apply to the *Fixed-tolerant* or *Dynamic-tolerant* settings of *Target Cycle Time Mode*:

1. *Watchdog Time* ≤ 1.5 * *Target Cycle Time*
2. *Receive Timeout ≤ 5 * Target Cycle Time + 4 x Latency*

   Latency refers to the delay on the transport path.
3. For reload, there is either just one user program or several user programs, the cycle of which is limited to a single processor module cycle.

11.3 Worst Case Response Time for safeethernet

In the following examples, the formulas for calculating the worst case response time only apply for a connection with HIMatrix controllers if their programming does not include noise blanking. These formulas always apply to HIMax controllers.

i The allowed worst case response time depends on the process and must be agreed upon together with the competent test authority.

The following table describes the parameters and conditions that must be taken into account in SILworX to calculate the worst case response time:

Terms	Description
Receive Timeout	Monitoring time of controller 1 (PES 1) within which a valid response from controller 2 (PES 2) must be received. Otherwise, safety-related communication is terminated after the time has expired.
Production Rate	Minimum interval between two data transmissions.
Watchdog Time	Maximum duration permitted for a controller's RUN cycle. The duration of the RUN cycle depends on the complexity of the user program and the number of safe**ethernet** connections. The watchdog time (WDT) must be entered in the resource properties.
Worst Case Response Time	The worst case response time is the time between a change in a physical input signal (in) of PES 1 and a change in the physical output signal (out) of PES 2.
Response time of the HIMax controller	For further details on the response time of the HIMax controller (resource) t_{RR}, see Chapter *Safety-Relevant Time Parameters*.
Delay	Delay of a transport path, e.g., when a modem or satellite connection is used. For direct connections, an initial delay of 2 ms can be assumed. The responsible network administrator can measure the actual delay on a transport path.

Table 16: safe**ethernet** Parameter Description and Conditions

The following conditions apply to the calculations of the maximum response times specified below:

- The signals transmitted over safe**ethernet** must be processed in the corresponding controllers within one CPU cycle.
- The response times of the sensors and the actuators must also be added up.

The calculations also apply to signals in the opposite direction.

11.3.1 Calculating the Worst Case Response Time of 2 HIMax Controllers

The worst case response time T_R is the time between a change on the sensor input signal (in) of controller 1 and a response on the corresponding output (out) of controller 2. It is calculated as follows:

1 Input
2 HIMax controller 1
3 Safety-related protocol
4 HIMax controller 2
5 Output

Figure 4: Response Time when 2 HIMax Controllers are Interconnected

$T_R = t_1 + t_2 + t_3$

T_R Worst case response time

t_1 Safety time of HIMax controller 1

t_2 *Receive Timeout*

t_3 Safety time of HIMax controller 2

11.3.2 Calculating the Worst Case Response Time with 1 HIMatrix Controller

The worst case response time T_R is the time between a change on the sensor input signal (in) of the HIMax controller and a response on the corresponding output (out) of the HIMatrix controller. It is calculated as follows:

1 Input
2 HIMax controller
3 Safety-related protocol
4 HIMatrix controller
5 Output

Figure 5: Response Time when 1 HIMax and 1 HIMatrix Controllers are Interconnected

$T_R = t_1 + t_2 + t_3$

T_R Worst case response time

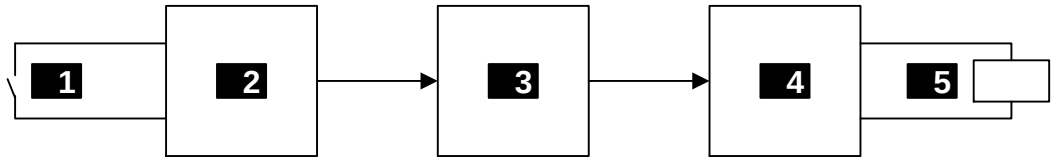
t_1 Safety time of the HIMax controller

t_2 *Receive Timeout*

t_3 2 * Watchdog time of the HIMatrix controller

11.3.3 Calculating the Worst Case Response Time with 2 HIMatrix Controllers or Remote I/Os

The worst case response time T_R is the time between a change on the sensor input signal (in) of the first HIMatrix controller or remote I/O (e.g., F3 DIO 20/8 01) and a response on the corresponding output (out) of the second HIMatrix controller or remote I/O (out). It is calculated as follows:



1 Input
2 Remote I/O 1
3 HIMax controller
4 Remote I/O 2
5 Output

Figure 6: Response Time with 2 Remote I/Os and 1 HIMax Controller

$T_R = t_1 + t_2 + t_3 + t_4 + t_5$

T_R Worst case response time

t_1 2 * watchdog time of the HIMatrix controller 1 or the remote I/O 1

t_2 *Receive Timeout1*

t_3 2 * watchdog time of the HIMax controller.

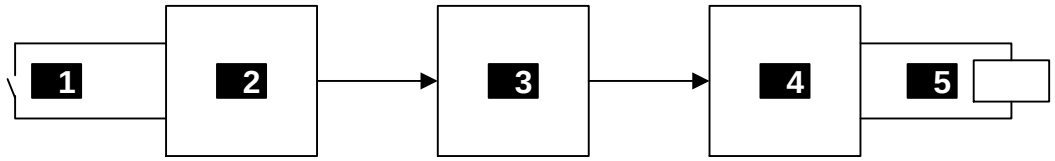
t_4 *Receive Timeout2*

t_5 2 * watchdog time of the HIMatrix controller 2 or the remote I/O 2

i Remote I/O 1 and remote I/O 2 can also be identical. The time values still apply if a HIMatrix controller is used instead of a remote I/O.

11.3.4 Calculating the Worst Case Response Time with 2 HIMax and 1 HIMatrix Controllers

The worst case response time T_R is the time between a change on the sensor input signal (in) of the first HIMax controller and a response on the corresponding output (out) of the second HIMax controller. It is calculated as follows:



1 Input
2 HIMax controller 1
3 HIMatrix controller
4 HIMax controller 2
5 Output

Figure 7: Response Time with 2 HIMax Controllers and 1 HIMatrix Controller

$T_R = t_1 + t_2 + t_3 + t_4 + t_5$

T_R Worst case response time

t_1 Safety time of HIMax controller 1

t_2 *Receive Timeout1*

t_3 2 * watchdog time of the HIMatrix controller

t_4 *Receive Timeout2*

t_5 Safety time of HIMax controller 2

i HIMax controllers 1 and 2 can also be identical.
The HIMatrix controller can also be a HIMax controller.

11.4 Safety-Related PROFIsafe Protocol

The requirements for using the PROFIsafe protocols are specified in the communication manual (HI 801 101 E). These requirements must be met.

The equations for determining the worst case response time are also specified in the communication manual.

Appendix

Glossary

Term	Description
AI	Analog input
AO	Analog output
ARP	Address resolution protocol, network protocol for assigning the network addresses to hardware addresses
COM	Communication module
CRC	Cyclic redundancy check
DI	Digital input
DO	Digital output
EMC	Electromagnetic compatibility
EN	European standard
ESD	Electrostatic discharge
FB	Fieldbus
FBD	Function block diagrams
HW	Hardware
ICMP	Internet control message protocol, network protocol for status or error messages
IEC	International electrotechnical commission
Interference-free	Inputs are designed for interference-free operation and can be used in circuits with safety functions
MAC	Media access control address, hardware address of one network connection
PADT	Programming and debugging tool (in accordance with IEC 61131-3), PC with SILworX
PELV	Protective extra low voltage
PES	Programmable electronic system
R	Read, the variable is read out
R/W	Read/Write, column title for system variable type
Rack ID	Base plate identification (number)
r_P	Peak value of a total AC component
SB	System bus (module)
SC/OC	Short-circuit/open-circuit
SELV	Safety extra low voltage
SFF	Safe failure fraction, portion of faults that can be safely controlled
SIL	Safety integrity level (in accordance with IEC 61508)
SILworX	Programming tool
SNTP	Simple network time protocol (RFC 1769)
SRS	System.Rack.Slot, addressing of a module
SW	Software
TMO	Timeout
W	Write, the variable receives a value, e.g., from the user program
WD	Watchdog, device for monitoring the system's correct operation. Signal for fault-free process
WDT	Watchdog time

Index of Figures

Index of Tables

Index

MANUAL

Safety Manual for Railway Applications

HI 801 327 E

For further information, please contact:

HIMA Rail Segment Team
Phone: +49 6202 709-411

Or contact our Rail Expert Team:
rail@hima.com

Learn more about HIMA solutions for
railway applications online:

https://www.hima.com/en/industries-solutions/rail/

HIMA SMART SAFETY.

www.hima.com