HIMA

SMART SAFETY.

Manual

HIMatrix®F

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 800 436 D, Rev. 4.00 (1938)	German original document
HI 800 437 E, Rev. 4.00.00 (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 HIMatrix 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 HIMatrix 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:

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

For details on how to use previous HIMatrix 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 HIMatrix 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 HIMatrix.

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 HIMatrix system.

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

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

The HIMatrix 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.

Thus, if faults occur, the de-energized state is adopted as the safe state for inputs and outputs.

2.1.2 Application in Accordance with the Energize to Trip Principle

The HIMatrix 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

The transfer of safety-relevant data through public networks like the Internet is permitted if additional security measures such as VPN tunnel or firewall have been implemented to increase security.

No safety-related communication can be ensured with fieldbus interfaces.

2.3 Tasks of Operators and Machine and System Manufacturers

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

Machine and system manufacturers must sufficiently validate that the HIMatrix 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 9 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 HIMatrix system.

NOTICE

Damage to the HIMatrix 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 HIMatrix systems are also available:

Name	Content	Document no.
HIMatrix safety manual	Safety functions of the HIMatrix system.	HI 800 023 E
HIMatrix system manual	Hardware description of the system	HI 800 141 E
HIMatrix F60 system manual	Hardware description of the modular F60 system	HI 800 191 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 800 673 E
Communication manual	Description of safe**ethernet** communication and 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 (OLH)	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 HIMatrix systems.

- Safety and availability.
- Safety-relevant time parameters.
- Safety requirements.
- Automation security.
- Additional test requirements for railway applications

3.1 Safety and Availability

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

The safety-related HIMatrix 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 HIMatrix automation systems.

⚠ WARNING

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!

3.1.1 Calculating the HR Values

The HR values for the HIMatrix system 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. The controllers, remote I/Os and F60 modules meet these requirements.

3.1.2 Self-Test and Fault Diagnostics

The operating system of the controllers executes comprehensive self-tests at start-up and during operation.

The scope of the testing includes:

- Processors.
- Memory areas (RAM, NVRAM).
- The watchdog.
- Individual I/O channels.
- The power supply.
- If faults are detected during the tests, the operating system switches off the faulty controller, module remote I/O or I/O channel.
- In non-redundant systems, this means that sub-functions or even the entire PES may be shut down.

All HIMatrix controllers, remote I/Os and modules are equipped with LEDs to indicate that faults have been detected. This allows the user to quickly diagnose internal faults or faults detected in the external wiring.

Additionally, the user program can evaluate various system variables displaying the system status, e.g., the temperature state.

Extensive diagnostics of the system performance and detected faults are stored in the diagnostic memory of the controllers. The diagnostics can also be read out after a system fault or supply voltage failure using the PADT.

For further details on how to evaluate diagnostic messages, refer to the HIMatrix system manual (HI 801 141 E).

For a very small number of component failures that do not affect safety, the HIMatrix 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 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 functions:

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.

3.1.4.1 Detection of Failed System Components

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

3.1.4.2 Safety Function in Accordance with the Energize to Trip Principle

The safety function is performed when the safety system energizes one or several actuators, thus ensuring that the safe state is adopted.

The users must plan the following actions:

- Line monitoring (short-circuits and open-circuits) with input and output modules. These functions must be configured accordingly.
- The operation of the actuators can be monitored through a position feedback.

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 The 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:

$\mathbf{t}_{RR} \leq \mathbf{t}_{SR}$

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

When using an F60 AO 8 01, also observe the following:

To determine the worst case response time of the analog outputs, add the double watchdog time of the AO CPU (2 x $\mathbf{t}_{WD\ AO\ \mu P}$) to the double watchdog time (2 x $\mathbf{t}_{WD\ CPU}$).

$\mathbf{t}_{RR} \leq \mathbf{t}_{SR} + \mathbf{12\ ms}$

$\mathbf{t}_{SR}$ 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.

The *Safety Time [ms]* parameter $\mathbf{t}_{SR}$ in the resource properties can be set in SILworX within 20…22500 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 delays configured through the user program.

3.2.3 Worst Case Response Time

The worst case response time applies to an undisturbed system. It is the maximum time that a HIMatrix system may require to respond to a change of an input signal with an output signal. In HIMatrix controllers running in cycles, the worst case response time is twice the maximum cycle time. The requirements are:

- The user program logic is designed so that delays, e.g., due to unfavorable execution order, cannot occur.
- A complete user program cycle is finished within a processor system cycle.
- Response-essential data are not exchanged between various user programs.

The cycle time of the controller includes processing of the following tasks:

- Process data communication: Receive processing.
- Reading the inputs.
- Processing the user program/s.
- Writing to the outputs.
- Process data communication: Send processing
- Executing the test routines.

For details on the calculation of the response time during communication, refer to Chapter 9 or the communication manual (HI 801 101 E).

3.2.4 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:

$$t_{WD} \leq \frac{1}{2} \times t_{SR}$$

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 4…5000 ms and is configured in the resource properties. The default setting is 200 ms for all the controllers and 100 ms for the remote I/Os.The PADT checks the parameters *Safety Time [ms]* and *Watchdog Time [ms]* and rejects the configuration while generating it if the 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!

i Determine the safety time and the watchdog time for the system to be controlled.

3.2.5 Estimating the Watchdog Time

HIMA recommends meeting the following conditions to ensure sufficient availability of the controller:

3 x t_{WD} ≤ tSR (*Safety Time [ms]* parameter*)*

3.2.6 Determining the Watchdog Time through Testing

The watchdog time t_{WD} can be determined through testing during commissioning or start-up. To this end, the system must be in RUN and operated under full load. All engineered modules must be inserted and all the configured communication connections (e.g., safe**ethernet** and other standard protocols) must be operating.

Test requirements:

- The HIMatrix hardware is completely mounted, e.g., the F60 rack includes all designated modules.
- Communication partners, including remote I/Os, are available and connected.
- The user program logic is completely available.
- *Target Cycle Time [ms]* is set to 0.
- *Program's Maximum Number of CPU Cycles* is set to 1 (program properties).
- *Max. Duration for Each Cycle [µs]* is set to 0 (program properties).
- *Max.Com. Time Slice [ms]* is set to a suitable value.
- *Max. Duration of Configuration Connections [ms]* is set to a suitable value.

To determine the minimum value for the watchdog time

1. Operate the system under full load. Communication should also run under full load.
2. Specify input data to preferably pass through the longest program paths. To this end, input value sequences may be necessary.
3. Reset the cycle time statistics in the Control Panel.
4. Perform the reload multiple times, if required by the application.
5. In the Control Panel, observe the maximum cycle time values.
   ☑ t_{Cycle} is identified.
6. Determine the maximum deviation between the user program's total execution time and the average total execution time.
   ☑ Δt_{Peak} is identified.
7. Calculate the minimum watchdog time t_{WD} using:

$t_{WD} = t_{Cycle} + t_{Res} + t_{Com} + t_{Config} + \Delta t_{Peak}$, where

t_{Cycle}	Observed maximum cycle time (basic load, already includes portions of t_{Com} and t_{Config})
$t_{Reserve}$	Safety margin 6 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_{Peak}	Maximum load peak of the cycle time (t_{Peak}) less observed basic load, see step 6.

► The value set for the watchdog time should be: determined minimum value t_{WD} + margin for future changes or extensions.

The maximum cycle time values during the reload depend on the configured watchdog time. If the PES should be optimized to the lowest possible watchdog time, the value of the **configured** watchdog time must be gradually reduced in a series of measurements.

In the following cases, contact HIMA technical support:

- If the requisites for the above strategy for determining the watchdog time cannot be complied with.
- If the result is not satisfying.

The HIMatrix system allows settings that ensure an even better performance. In-depth knowledge in several areas is required to identify these settings.

3.3 Safety Requirements

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

3.3.1 Product-Independent Hardware Requirements

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

- Only devices that are safely separated from the power supply may be connected to the system.
- The safe, electrically protective separation of the power supply must be guaranteed within the 24 V system supply. Only power supply units ensuring that the controllers and remote I/O modules are supplied with 24 V low voltage may be used.
- 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 shall protect the HIMatrix 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 HIMatrix 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 9.2.
- 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).
- At this stage, the serial interfaces may only be used for non-safety-related purposes.
- 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 HIMatrix system in railway applications:

- The standard variants of the HIMatrix system family as specified in Table 4, can be used in containers or in buildings with controlled temperature and air humidity.
- The standard variants of the HIMatrix system family are not approved for use on rolling stock.
- The HIMatrix variants for railway applications (see Table 3) 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 line short-circuit monitoring. Responses to detected short-circuits must be programmed in the user program.
- The temperature state (operating temperature) of the HIMatrix 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 HIMatrix system manual (HI 800 141 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 evaluated using the programming tool. For further details, refer to the HIMatrix system manual (HI 800 141 E).
- 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 HIMatrix 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 HIMatrix safety manual (HI 800 023 E) for the standards used to test and certify the HIMatrix system for industrial use.

3.6 Additional test requirements for railway applications

The following tables show the HIMatrix components that are approved for railway applications:

Compact controllers
F30 03
F35 03
Remote I/Os
F1 DI 16 01
F2 DO 4 01
F2 DO 8 01
F2 DO 16 01
F2 DO 16 02
F3 AIO 8/4 01
F3 DIO 8/8 01
F3 DIO 16/8 01
F3 DIO 20/8 02
Modular F60 System
AI 8 01
CIO 2/4 01
CPU 03
DI 32 01
DIO 24/16 01
DO 8 01
GEH 01
MI 24 01
PS 01

Table 2: HIMatrix Standard Variants

All the HIMatrix standard variants listed in Table 2 are only approved for use as equipment for signaling and telecommunications in accordance with EN 50125-3.

Compact controllers
F30 034
F35 034
Remote I/Os
F1 DI 16 014
F2 DO 8 014
F2 DO 16 014
F3 AIO 8/4 014
F3 DIO 8/8 014
F3 DIO 16/8 014
F3 DIO 20/8 024
Modular F60 system
AI 8 014
CIO 2/4 014
CPU 034
DI 32 014
DIO 24/16 014
GEH 014
MI 24 014
PS 014

Table 3: HIMatrix Variants for Railway Applications

All the HIMatrix components listed in Table 3 are approved for use on rolling stock in accordance with EN 61373, Category 1, Class B, and, as equipment for signaling and telecommunications in accordance with EN 50125-3, for the position outside the track (1…3 m from the rail). These variants of the standard components are identified by the suffix 4 in the type designation.

The HIMatrix components have been additionally developed to meet the following EMC, climatic and environmental requirements:

3.6.1 Height Range

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

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

The following classes in the specified height range apply to the HIMatrix components that are specified in Table 3:

- 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

All HIMatrix standard variants are designed and tested for a temperature range of 0…60 °C and a relative air humidity of 10…95 % (non-condensing). The following temperature classes result for railway applications in accordance with EN 50125-3:

HIMatrix	In external ambient	In control cabinet	In container		In building	
			N.T.C	T.C	N.C.C.	C.C
Standard	-	-	-	T1, T2, TX	-	T1, T2, TX

Table 4: HIMatrix Temperature Classes of the Standard Variants According to EN 50125-3

The standard variants of the HIMatrix system family as specified in Table 4, can be used in containers or in buildings with controlled temperature and air humidity.

NOTICE

The standard variants of the HIMatrix system family are not approved for use on rolling stock in accordance with EN 50155.

3.6.2.1 Use in Signaling Applications

The HIMatrix variants for railway applications are designed for a temperature range of -25…+70 °C. All the HIMatrix variants for railway applications were tested in accordance with EN 50125-3 and can be used in the following temperature classes:

HIMatrix	In external ambient	In control cabinet	In container		In building	
			N.T.C	T.C	N.C.C.	C.C
F30 034	T1	T1	T1, T2	T1, T2, TX	T1, T2, TX	T1, T2, TX
F35 034	T1	T1	T1, T2	T1, T2, TX	T1, T2, TX	T1, T2, TX
F1 DI 16 014	T1	T1	T1, T2	T1, T2, TX	T1, T2, TX	T1, T2, TX
F2 DO 8 014	T1	T1	T1, T2	T1, T2, TX	T1, T2, TX	T1, T2, TX
F2 DO 16 014	T1	T1	T1, T2	T1, T2, TX	T1, T2, TX	T1, T2, TX
F3 AIO 8/4 014	T1	T1	T1, T2	T1, T2, TX	T1, T2, TX	T1, T2, TX
F3 DIO 8/8 014	T1	T1	T1, T2	T1, T2, TX	T1, T2, TX	T1, T2, TX
F3 DIO 16/8 014	T1	T1	T1, T2	T1, T2, TX	T1, T2, TX	T1, T2, TX
F3 DIO 20/8 024	T1	T1	T1, T2	T1, T2, TX	T1, T2, TX	T1, T2, TX
PS 014	T1	T1	T1, T2	T1, T2, TX	T1, T2, TX	T1, T2, TX
CPU 034	T1	T1	T1, T2	T1, T2, TX	T1, T2, TX	T1, T2, TX
AI 8 014	T1	T1	T1, T2	T1, T2, TX	T1, T2, TX	T1, T2, TX
CIO 2/4 014	T1	T1	T1, T2	T1, T2, TX	T1, T2, TX	T1, T2, TX
DI 32 014	T1	T1	T1, T2	T1, T2, TX	T1, T2, TX	T1, T2, TX
DIO 24/16 014	T1	T1	T1, T2	T1, T2, TX	T1, T2, TX	T1, T2, TX
MI 24 014	T1	T1	T1, T2	T1, T2, TX	T1, T2, TX	T1, T2, TX

Table 5: Temperature Classes According to EN 50125-3

3.6.2.2 Use on Rolling Stock

All the HIMatrix variants for railway applications were tested in accordance with EN 50155 and can be used in the following temperature classes:

HIMatrix	Temperature classes
F30 034	OT3
F35 034	OT3
F1 DI 16 014	OT3
F2 DO 8 014	OT3
F2 DO 16 014	OT3
F3 AIO 8/4 014	OT3
F3 DIO 8/8 014	OT3
F3 DIO 16/8 014	OT3
F3 DIO 20/8 024	OT3
PS 014	OT3
CPU 034	OT3
AI 8 014	OT3
CIO 2/4 014	OT3
DI 32 014	OT3
DIO 24/16 014	OT3
MI 24 014	OT3

Table 6: Temperature Classes According to EN 50155

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

With respect to fast temperature change, temperature class H1 applies, as defined in EN 50155, Chapter 4.3.4.

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

3.6.2.3 Derating of Digital Outputs

With an operating temperature higher than 60 °C the load of the digital outputs must be derated. In this case, each output can be loaded with a maximum of 0.5 A, see the manuals of the components.

3.6.3 Mechanical Requirements

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

3.6.3.1 Use in Signaling Applications

All HIMatrix components were mechanically tested in accordance with EN 50125-3. The following table lists the most important tests and limits for mechanical requirements:

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

Table 7: Mechanical Requirements for Use in Signaling Applications

3.6.3.2 Use on Rolling Stock

The components listed in Table 3 were mechanically tested in accordance with EN 50155. Testing was performed in accordance with EN 61373, Category 1, Class B.

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

3.6.4 EMC Requirements

The following chapters contain the tests and limit values of the EMC requirements for use in signaling technology and on railway vehicles.

3.6.4.1 Use in Signaling Applications

All HIMatrix components were successfully tested and meet the EMC requirements in accordance with EN 50121-4. The following table lists the most important tests and limits:

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…1000 MHz:	10 V/m
		800…1000 MHz:	20 V/m
		1400…2000 MHz:	10 V/m
		2000…2700 MHz:	5 V/m
		5100…6000 MHz:	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
		Shielded wires:	2 kV CM
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 8: EMC Requirements for Use in Signaling Applications According to EN 50121-4

Remarks to Surge with 2 kV (CM) / 1 kV (DM):

The following notes apply to the standard variants and the variants for railway applications, even if these are not explicitly mentioned.

The external H 7013 filter from HIMA is absolutely required if HIMatrix compact systems are used to act against the DC supply voltage surge. The supply voltage of the HIMatrix F35 03 must not be provided from outside, but must be generated within the same control cabinet.

i In the following cases, external surge filters are also required for all unshielded input and output lines:

- Connection to equipment within the 3-meter range.
- Connection to equipment within the 10-meter range with connection within the 3-meter range.
- Connection to equipment within the 10-meter range with lines that are longer than 30 m.

For compact systems F30 03, F1 DI 16 01, F3 DIO 20/8 02 and for the F60 modules DIO 24/16 01 and DI 32 01, the DCO RK ME24 surge absorber from the DEHN (currently DCO SD2 ME24) must be used to protect the digital inputs against surge pulses.

For F30 03 controller and for the DIO 24/16 01 F60 modules, the DCO RK MD24 surge absorber from the DEHN (currently DCO SD2 MD24) must be used to protect the digital outputs against surge pulses.

Surge absorbers from other manufacturers may be used, if the specifications provided in the data sheets are equivalent or better.

3.6.4.2 Use on Rolling Stock

The HIMax components specified in Table 3 were successfully tested and met the EMC requirements in accordance with EN 50121-4 and EN 50121-3-2. The following table lists the most important tests and limits:

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…1000 MHz:	20 V/m
		1400…2000 MHz:	10 V/m
		2000…2700 MHz:	5 V/m
		5100…6000 MHz:	3 V/m
EN 61000-4-4	Burst test	Supply voltage:	2 kV
		I/O lines:	2 kV
EN 61000-4-5	Surge	Supply voltage:	2 kV CM
			1 kV DM
EN 61000-4-6	Injected RF currents	Supply voltage:	10 V
		I/O lines:	10 V

Table 9: EMC Requirements for Use on Rolling Stock According to EN 50121-3-2

Remarks to Surge with 2 kV (CM) / 1 kV (DM):

The external H 7013 filter from HIMA is absolutely required if HIMatrix compact systems are used to act against the DC supply voltage surge. The supply voltage of the HIMatrix F35 034 must not be provided from outside, but must be generated within the same control cabinet.

Surge absorbers from other manufacturers may be used, if the specifications provided in the data sheets are equivalent or better.

3.6.5 Severe Conditions

The HIMatrix 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 HIMatrix systems:

IEC/EN 61131-2	Verification of the DC supply characteristics
	Voltage range test: 24 VDC, -15…+20 %, $r_P \leq 5$ %
	Momentary external current interruption immunity test: DC, PS 2: 10 ms
	Reversal of DC power supply polarity test: Tested for 10 s

Table 10: Supply Voltage Failures Immunity Test

3.6.6.1 Supply Voltage Requirements for Use on Rolling Stock

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

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

This results in the following tolerance values:

- Minimum continuous voltage: 19.2 V (0.8 U_N).
- Maximum continuous voltage: 30 V (1.25 U_N).

The HIMatrix variants 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 U_N is maintained, since otherwise individual devices or the entire system will reboot.

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

HIMatrix systems are designed for voltage dropouts of up to 20 ms. As such, the HIMatrix meets the requirements of Class S3 in accordance with EN 50155, Chapter 5.1.1.4.

The HIMatrix 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 two supply voltages are not met. External measures must be implemented by the user.

4 Central Functions

The controllers and remote I/Os of type F1…, F2…, F3… are compact systems that cannot be modified.

The controllers of type F60 are modular systems. In addition to a processor module and a power supply module, one controller of this type may include up to 6 I/O modules.

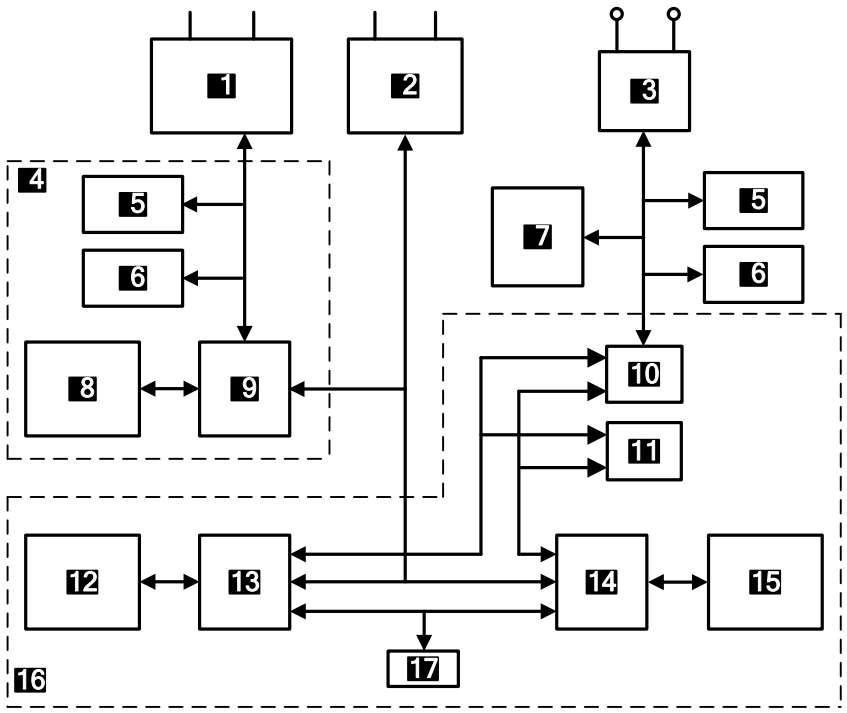
4.1 Power Supply Units

The HIMatrix systems must be supplied by power supply units ensuring a 24 V low voltage to the controllers and remote I/Os.

Observing the permitted voltage limits guarantees the controller's proper operation.

4.2 Functional Description of the Processor System

The processor system is the central component of the controller. The following figure shows the block diagram of the processor system based on the example of the CPU 03 in the F60 modular system:



1 Fieldbus interfaces
2 Ethernet interfaces
3 I/O bus submodule
4 Communication system
5 NVSRAM
6 Flash
7 V_{CC} and temperature monitoring
8 SDRAM of the communication system
9 Processor of the communication system
10 Comparator
11 Watchdog
12 SDRAM 1 of the processor system
13 Processor 1 of the processor system
14 Processor 2 of the processor system
15 SDRAM 2 of the processor system
16 Safety-related processor system
17 Real time clock

Figure 1: CPU 03 Block Diagram

Characteristics of the Processor System

- Two synchronous microprocessors (processor 1 and processor 2).
- Each microprocessor has its own SDRAM memory.
- Testable hardware comparators for all external accesses of both microprocessors.
- In the event of an error, the watchdog is set to the safe state.
- Flash as the program memory for operating systems and user programs, suitable for at least 100 000 memory cycles.
- Data memory in NVSRAM.
- Gold capacitor for buffering date/time.
- Communication processor for fieldbus and Ethernet connections.
- Interface for exchanging data between devices, F60 controllers and the PADT, based on Ethernet.
- Optional interface(s) for data exchange via fieldbus.
- LEDs for indicating the system states.
- I/O bus logic for connection to I/O modules.
- Safe watchdog (WD).
- Monitoring of power supply units, testable (1.8 VDC / 3.3 VDC).
- Temperature monitoring.

4.3 Self-Tests

The operating system of the processor system 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 diagnostic measures mandatory for complying with the safety standards are implemented in the safety-related processor system.

The following section specifies the most important self-test routines of safety-related processor systems.

4.3.1 Microprocessor Test

The following is tested:

- All commands and addressing modes used.
- The writability of the flags and the commands affected by the flags.
- The writability and crosstalk of the registers.

4.3.2 Memory Areas Test

The operating system, user program, constants and parameters as well as the variable data are stored in memory areas of both processors and are tested by a hardware comparator.

4.3.3 Protected Memory Areas

The operating system, user program and parameter range are each stored in one memory. They are secured by write protection and a CRC test.

4.3.4 RAM Test

A write and read test is performed to check the modifiable RAM areas, in particular for stuck-at issues and crosstalk.

4.3.5 Watchdog Test

The watchdog signal is switched off if it is not triggered by both CPUs within a defined time window and if the hardware comparator test fails. An additional test determines the switch-off ability of the watchdog signal.

4.3.6 Testing the I/O Bus Within the Controller

The connection between the CPU and the associated inputs and outputs (I/O modules) is tested.

4.4 Responses to Faults in the Processor System

A hardware comparator within the processor system constantly checks if the data from microprocessor 1 is identical to the data from microprocessor 2. If this is not the case or the test routines detect a fault, the watchdog signal is switched off. This means that the input signals are no longer processed by the controller, and the outputs switch to the de-energized, switched-off state.

If such a fault occurs for the first time, the controller is restarted (reboot). If a further fault occurs within the first minute after start-up, the controller enters the STOP/INVALID CONFIGURATION state and will remain in this state.

4.5 Fault Diagnostics

Each F60 module has an own LED for reporting module malfunctions or faults in the external wiring. This allows the user to quickly diagnose faults detected in a module.

In the F1…, F2…, F3… compact systems, these fault indications are grouped into one common error message.

Additionally, the user program can evaluate various system variables associated with the inputs, outputs or the controller.

Faults are only signaled if they do not hinder communication with the processor system, i.e., the processor system must still be able to evaluate the faults.

The user program logic can evaluate the error codes of the system variables and of all the input and output signals.

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

For further details on how to evaluate diagnostic messages, refer to the system manual (HI 801 141 E).

5 Inputs

The notes in this chapter apply to the standard variants and the variants for railway applications, even if these are not explicitly mentioned.

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

Component	Type	Number	Safety-related	Interference-free	Galvanically separated
Compact systems					
F30 03	Digital	20	•	•	– [1]
F35 03	Digital	24	•	•	– [1]
	24-bit counter	2	•	•	– [1]
	Analog	8	•	•	– [1]
F1 DI 16 01	Digital	16	•	•	– [1]
F3 DIO 8/8 01	Digital	8	•	•	– [1]
F3 DIO 16/8 01	Digital	16	•	•	– [1]
F3 AIO 8/4 01	Analog	8	•	•	– [1]
F3 DIO 20/8 02	Digital	20	•	•	– [1]
Modular F60 System					
DIO 24/16 01	Digital	24	•	•	•
DI 32 01 (configurable for line control)	Digital	32	•	•	•
CIO 2/4 01	24-bit counter	2	•	•	•
AI 8 01	Analog	8	•	•	•
MI 24 01	Analog or digital	24	•	•	•
[1] Reference potential: L-					

Table 11: Overview of the HIMatrix System Inputs

5.1 General Information

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 provide status and fault information as follows:

- Through diagnostic LEDs.
- Using system variables that the user program can evaluate.
- Storing messages in the diagnostic memory that the PADT can read.

Safety-related input modules are automatically tested during operation through high-quality, cyclic self-tests. These test routines are TÜV-tested and monitor the safe functioning of the corresponding module.

For a small number of component failures that do not affect safety, no diagnostic information is generated.

5.2 Response in the Event of a Fault

If the test routines detect an error, they trigger the following responses:

- The user program processes the initial value of the global variables assigned to the input.
- 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 component.

If a fault occurs, a compact system activates the ERROR LED, an F60 module the *ERR* LED.

5.3 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.

5.4 Safety-Related Digital Inputs

The described properties apply to both the digital input channels of F60 modules and the digital input channels of all compact systems (unless stated otherwise).

5.4.1 General Information

The digital inputs are read once per cycle and saved internally; cyclic tests are performed to ensure their safe functioning.

Under certain circumstances, input signals that are present for shorter than the time between two samplings, are not detected.

5.4.2 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 bevor the input signals are read.

5.4.3 Surges on Digital Inputs

Due to the short cycle time of the HIMatrix 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.

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. This measure increases the maximum response time!

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 HIMatrix system manual (HI 800 141 E).

5.4.4 Configurable Digital Inputs

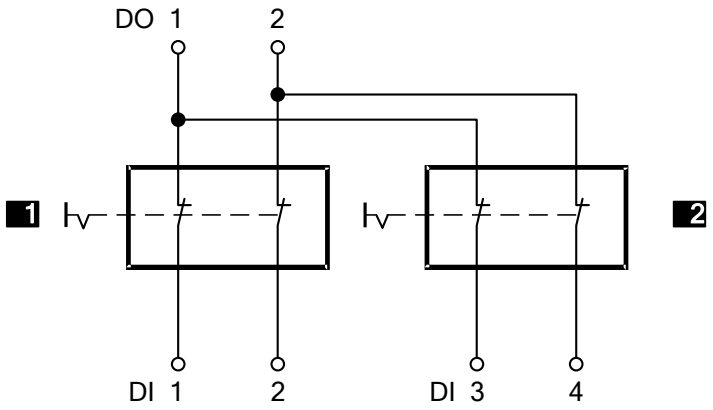
The digital inputs of the F35 03 controller and the MI 24 01 module operate as analog inputs, but return digital values due to the configuration of switching thresholds.

For configurable digital inputs, the same test routines and safety-related functions defined for analog inputs apply as specified in Chapter 5.5.

5.4.5 Line Control

Line control is used to detect short-circuits or open-circuits e.g., on emergency stop devices and can be configured for the HIMatrix systems with digital inputs (not for the F35 03 controller and MI 24 01 module).

To this end, connect the digital outputs of the system to the digital inputs of the same system as follows (example):

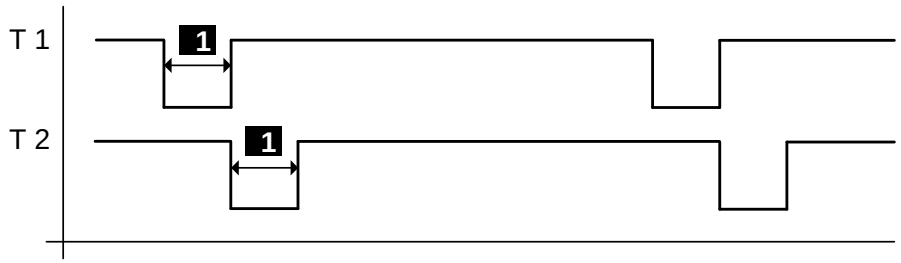


1 Emergency stop 1
2 Emergency stop 2

Emergency stop switches in accordance with EN 60947-5-1 and EN 60947-5-5

Figure 2: Line Control

The controller pulses the digital outputs to detect short-circuits and open-circuits on the wires connected to the digital inputs. To do so, configure the *Value [BOOL]* -> system variable in SILworX. The pulsed outputs can be assigned to any digital inputs.



1 Configurable 5…2000 µs

Figure 3: Pulsed Signals T1, T2

An (evaluable) error code is created, if the following errors occur:

- Cross-circuit between two parallel wires.
- Invalid connections of two lines (e.g., DO 2 to DI 3).
- Ground fault on one of the wires (with grounded reference pole only).
- Open-circuit or open contacts.

For a description of line control and further details, refer to the HIMatrix system manual (HI 800 141 E).

5.5 Safety-Related Analog Inputs (F35 03, F3 AIO 8/4 01 and F60)

The analog input channels convert the measured input currents into an INTEGER value. The values are available to the user program as variables that are assigned to the system variable -> *Value [INT].*

The range of values for the inputs depends on the component:

F35 03 Controller

Input channels	Measurement procedure	Current, voltage	Range of values in the application	
			FS1000 [1]	FS2000 [1]
8	Unipolar	0…+10 V	0…1000	0…2000
8	Unipolar	0…20 mA	0…500 [2] 0…1000 [3]	0…1000 [2] 0…2000 [3]

[1] Configurable by selecting the type in the PADT.

[2] With external 250 Ω shunt adapter.

[3] With external 500 Ω shunt adapter.

Table 12: Analog Inputs of the F35 03 Controller

F3 AIO 8/4 01 Remote I/O

Input channels	Measurement procedure	Current, voltage	Range of values in the application
8	Unipolar	0…+10 V	0…2000
8	Unipolar	0/4…20 mA	0…1000 [1] 0…2000 [2]

[1] With external 250 Ω shunt adapter.

[2] With external 500 Ω shunt adapter.

Table 13: Analog Inputs of the F3 AIO 8/4 01 Remote I/O

F60 Modules

Input channels	Measurement procedure	Current, voltage	Range of values in the application	
			FS1000 [1]	FS2000 [1]
AI 8 01				
8	Unipolar	-10…+10 V	-1000…1000	-2000…2000
8	Unipolar	0…20 mA	0…1000 [3]	0…2000 [3]
8	Unipolar	0…20 mA	0…500 [2]	0…1000 [2]
4	Bipolar	-10…+10 V	-1000…1000	-2000…2000
MI 24 01				
24	Unipolar	0…20 mA	0…2000 [4]	

[1] Configurable by selecting the type in the PADT (F60).

[2] With external 250 Ω shunt.

[3] With external 500 Ω shunt (accuracy 0.05 % 1 W). No longer available at HIMA.

[4] Internal shunts.

Table 14: Analog Inputs of the F60 Controller

The F60 module AI 8 01 can be configured in the user program for 8 unipolar or 4 bipolar functions. However, it is not allowed to combine functions on a module.

The analog inputs of the F35 03 controller, the F3 AIO 8/4 01 remote I/O and the AI 8 01 module operate with voltage measurement. With the analog inputs of the F35 03 and F3 AIO 8/4 01, digital outputs of the own system (F35 03) or other HIMatrix controllers can be monitored to detect open-circuits. For further details, refer to the manuals of the corresponding HIMatrix controllers.

If an open-circuit occurs and line monitoring is not active in the system, random input values are processed at the high-resistance inputs. The value resulting from this floating input voltage is not reliable; for voltage inputs, the channels must be terminated with a 10 kΩ resistor. The internal resistance of the source must be taken into account.

To measure currents, the shunt is connected in parallel to an input; in doing so the 10 kΩ resistor is not required.

The inputs of the MI 24 01 module are only current inputs, because of the internal shunts, and cannot be used as voltage inputs.

If input channels are not used, the measuring input must be connected to the reference potential. If an open-circuit occurs, negative influences (floating input voltages) on other channels can thus be avoided. It is sufficient not to assign unused inputs global variables.

5.5.1 Test Routines

The analog values are processed in parallel via 2 multiplexers and 2 analog/digital converters with 12-bit resolution, and the results are compared. Additionally, test values are applied via the D/A converters, converted back to digital values, and then compared with the default value.

5.6 Safety-Related Counters (F35 03 and F60)

Unless otherwise noted, the points previously mentioned apply to the CIO 2/4 01 counter module of the F60 system as well as to the F35 03 counters.

5.6.1 General Information

A counter channel can be configured for operation as a high-speed up or down counter with 24-bit resolution or as a decoder in Gray code.

If used as high-speed up or down counters, the pulse input and count direction input signals are required in the application. A reset is only carried out in the user program.

The counter encoders have the following resolutions:

- The counters of the F60 module CIO 2/4 01 have 4-bit or 8-bit resolution.
- The F35 03 counters have 3-bit or 6-bit resolution.

A reset is possible.

Two independent 4-bit inputs can only be linked to one 8-bit input (example for F60) via the user program. No switching option is planned for this purpose.

The encoder function monitors the change of the bit pattern on the input channels. The bit patterns on the inputs are directly transferred to the user program. They are represented in the PADT as decimal numbers corresponding to the bit pattern (*Counter[0x].Value*).

Depending on the application, this number (which corresponds to the Gray code bit pattern) can be converted into the corresponding decimal value, for example.

5.7 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/.

6 Outputs

The notes in this chapter apply to the standard variants and the variants for railway applications, even if these are not explicitly mentioned.

The following table provides an overview of the output modules of the HIMatrix system:

Component	Type	Number	Safety-related	Galvanically separated
Compact systems				
F30 03 (configurable for line control)	Digital	8	•	–[1)]
F35 03	Digital	8	•	–[1)]
F1 DI 16 01	Pulse	4	-	–[1)]
F2 DO 4 01 [2)]	Digital	4	•	–[1)]
F2 DO 8 01	Relay	8	•	•
F2 DO 16 01	Digital	16	•	–[1)]
F2 DO 16 02 [2)]	Relay	16	•	•
F3 DIO 8/8 01	Digital 1-pole	8	•	–[1)]
	Digital 2-pole	2		
F3 DIO 16/8 01	Digital 1-pole	16	•	–[1)]
	Digital 2-pole	8		
F3 AIO 8/4 01	Analog	4	-	–[1)]
F3 DIO 20/8 02 (configurable for line control)	Digital	8	•	–[1)]
Modular F60 System				
DIO 24/16 01 (configurable for line control)	Digital	16	•	
DO 8 01 (250 V) [2)]	Relay	8	•	•
CIO 2/4 01	Digital	4	•	

[1)] Reference potential L-.

[2)] Only available as standard variant.

Table 15: Overview of the HIMatrix System Outputs

6.1 General Information

The controller writes to the safety-related outputs once per cycle, reads back the output signals and compares them with the specified output data.

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

Three testable switches connected in series are integrated in the safety-related output channels. The required second independent shutdown option is thus integrated in the output module. If a fault occurs, this integrated safety shutdown safely de-energizes all the channels of the defective submodule (de-energized state).

Additionally, the watchdog signal of the CPU is the second safety shutdown option: If the watchdog signal is lost, the CPU immediately enters the safe state of all output channels.

This function is only effective for all the digital outputs and relay outputs of the controller.

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

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

If a fault occurs, a compact system activates the *ERROR* LED, an F60 module the *ERR* LED.

6.3 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.

6.4 Safety-Related Digital Outputs

The points listed below apply to both digital output channels of F60 modules and digital output channels of the compact systems. Unless specified otherwise, the relay modules are an exception in both cases.

6.4.1 Test Routines for Digital Outputs

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

- Reading the output signals back from the switching amplifier. The switching threshold for a read-back low level is 2 V. The diodes used prevent the signals from being fed back.
- Checking the integrated redundant safety shutdown.
- Any shutdown test of the outputs is performed as background test for max. 200 µs. The minimum time between two tests is ≥ 20 s.

The system monitors its operating voltage and de-energizes all outputs at voltages of less than 13 V.

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

If the output is short-circuited to L- or overloaded, the module is still testable. Shutdown via safety shutdown is not required.

The controller monitors the module's total current consumption and sets all output channels to the safe state if the threshold is exceeded.

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.

Line Control

The controller can pulse safety-related digital outputs or special pulsed outputs and use them with the safety-related digital inputs of the same system (not the digital inputs of the F35 03 or F60 MI 24 01) to detect open-circuits and short-circuits (see Chapter 5.4.5).

NOTICE

Malfunctions of the connected actuators are possible!

Pulsed outputs must not be used as safety-related outputs (e.g., for activating safety-related actuators)!

Relay outputs cannot be used as pulsed outputs.

6.5 Safety-Related 2-Pole Digital Outputs

The following points apply to 2-pole digital outputs of the remote I/Os F3 DIO 8/8 01 and F3 DIO 16/8 01.

The remote I/Os are tested automatically during operation. The main test functions are:

- Reading the output signals back from the switching amplifier. The diodes used prevent the signals from being fed back.
- Checking the integrated (redundant) safety shutdown.
- Any shutdown test of the outputs is performed as background test for max. 200 µs. The minimum time between two tests is ≥ 20 s.
- Line diagnosis with 2-pole connection
  F3 DIO 16/8 01:
  - Short-circuit to L+, L-.
  - Short-circuit between the 2-pole connections.
  - Open-circuit in one of the 2-pole connections.

  F3 DIO 8/8 01:
  - Short-circuit to L+, L-.

The system monitors its operating voltage and de-energizes all outputs at voltages of less than 13 V.

With a 2-pole connection, observe the following notes:

i A relay or actuator connected to the output may accidentally be switched on!

A requirement for applications in machine safety is that the outputs DO+, DO- are switched off if an open-circuit is detected.

i If the requirements previously described cannot be met, observe the following case:

If a short-circuit occurs between DO- and L-, a relay may be energized or some other actuator may be set to a different switching state.

Reason: During the monitoring time specified for line diagnosis, a 24 V level (DO+ output) is present on the load (relay, switching actuator) allowing it to receive enough electrical power to potentially switch to another state.

The monitoring time must be configured such that an actuator cannot be activated by the line diagnosis test pulse.

i Detection of open-circuits may be disturbed!

In a 2-pole connection, no DI input may be connected to a DO output. This would inhibit the detection of open-circuits.

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

If the output is short-circuited to L-, L+ or overloaded, the remote I/O is still testable. Shutdown via safety shutdown is not required.

The total current consumption of the remote I/O is monitored. If the threshold is exceeded, the remote I/O sets all channels to the safe state.

In this state, the remote I/O checks the outputs every few seconds to determine whether the overload is still present. In a normal state, the remote I/O switches the outputs on again.

6.6 Relay Outputs

The relay outputs correspond to functional digital outputs, but offer galvanic separation and higher electrical strength.

6.6.1 Test Routines for Relay Outputs

The relay module automatically tests its outputs 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.

The system monitors its operating voltage and de-energizes all outputs at voltages of less than 13 V.

The outputs of the DO 8 01 module and those of the remote I/Os F2 DO 8 01 and F2 DO 16 02 are equipped with three safety relays:

- 2 relays with forcibly guided contacts.
- 1 standard relay.

This enables the outputs to be used for safety switch-off functions.

6.7 Analog Outputs with Safety-Related Shutdown (F3 AIO 8/4 01)

The remote I/O writes to the analog outputs once per cycle and saves the values internally.

The outputs are not safety-related, but they can be safely switched off together.

To achieve SIL 4, the output values must be read back via safety-related analog inputs and evaluated in the user program. Responses to faulty output values must be programmed in the user program as well.

6.7.1 Test Routines

The remote I/O automatically tests the 2 safety switches used to shut down all 4 module outputs during operation.

6.8 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/.

7 Software

The software for the safety-related HIMatrix 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.

7.1 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 HIMatrix 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.

7.2 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.

7.3 Safety-Related Aspects of Programming

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

7.3.1 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!

7.3.2 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.

7.3.3 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.

7.3.4 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.

i During commissioning or after a change to the user program of a safety-related controller, a comprehensive functional test must be performed.

A project archive must be created.

7.4 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 the HIMatrix system manual (HI 800 141 E)**
- **The parameters marked as safety parameters in Table 16.**

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.

7.4.1 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.

Parameters	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…22 500 ms Default value: 600 ms for controllers, 400 ms for remote I/Os (can be changed online)	Application-specific
Watchdog Time [#	Y	For details on the watchdog time *of* the resource (in milliseconds), refer to the safety manual (HI 801 003 E). Range of values: 4…5000 ms Default value: 200 ms for controllers, 100 ms for remote I/Os (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…5000 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] (4 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 Default value: 60 ms	Application-specific

Parameters	S [1)]	Description	Setting for safe operation
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	Not applicable for HIMatrix controllers! Default value*:* System Defaults	---
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.	

Parameters	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 Multi-Forcing 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)	

Parameters	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 HIMatrix operating system versions in accordance with the project requirements. Default value: SILworX V11 for new projects.		Application-specific
		SILworX V2	Not applicable for HIMatrix controllers!	
		SILworX V3		
		SILworX V4	The code is generated like in SILworX V4, V5 or V6.48. The generated code is compatible with the CPU operating system V8.	
		SILworX V5		
		SILworX V6		
		SILworX V6b	The code is generated like in SILworX V6.114. The generated code is compatible with the CPU operating system V10.	
		SILworX V7	The code is generated like in SILworX V7. The generated code is compatible with the CPU operating system V11.	
		SILworX V8	The code is generated like in SILworX V8. The generated code is compatible with the CPU operating system V12.	
		SILworX V9	The code is generated like in SILworX V9. The generated code is compatible with the CPU operating system V13.	
		SILworX V10	The code is generated like in SILworX V10. The generated code is compatible with the CPU operating system V13.	
		SILworX V11	The code is generated like in SILworX V11. The generated code is compatible with the CPU operating system V15.	
Fast Start-Up	Y	After connecting the supply voltage, the resource starts up faster, <10 s, see Chapter 7.4.1.4. Default value: FALSE		Application-specific

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

Table 16: Resource System Parameters

7.4.1.1 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, HIMatrix 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 17: Settings for Target Cycle Time Mode

7.4.1.2 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 HIMatrix CPU operating systems is calculated as follows:

HIMatrix CPU $t_{Config} = (\mathbf{n}_{Com} + \mathbf{n}_{PADT} + \mathbf{n}_{RIO}) * 0.25\ ms + 4\ ms$

$t_{Config:}$	System parameter *Max. Duration of Configuration Connections [ms].*
$n_{COM:}$	Number of modules with Ethernet interfaces (CPU, COM)
$n_{PADT:}$	5, maximum number of PADT connections.
$n_{RIO:}$	Number of configured remote I/Os.

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.

7.4.1.3 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.

7.4.1.4 The Fast Start-Up Parameter

The *Fast Start-Up* parameter exists for SILworX V7 and higher, and requires a resource with CPU operating system V11 or higher and a COM operating system V16 or higher. Additionally, the resource must be equipped with a CPU bootloader V11.2 or higher and a COM bootloader V16.8 or higher. The bootloader is not the same as the OS loader (emergency loader) and cannot be replaced by the user.

Fast start-up is only effective when the PES supply voltage is connected. Operation at SIL 3 level is still ensured.

Fast start-up is achieved through the following measures:

- Shortened self-tests.
- No detection of duplicate IP addresses.

  If detection of duplicate IP addresses is deactivated and the network configuration is faulty, duplicate IP addresses might be in use in the network!

  The parameter settings must ensure that no duplicate IP addresses exist in the network!

If an LED test is required during reboot, the *Fast Start-Up* parameter must be set to FALSE!

7.4.1.5 Hardware System Variables

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 SILworX Hardware Editor , in the hardware detail view.

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
Leer 2…Leer 21	N	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
Relay Contact 1… Relay Contact 4	N	Only applicable to F60! OR-linked system variables that control the relay of the FAULT contact on the F60 PS 01. The relay is a change-over contact with common contact 2, break contact 3 and make contact 1. ▪ If the F60 module is in the RUN state and the system variables *Relay Contact 1…4* are FALSE, contact 1-2 is closed (contact 2-3 is open). ▪ If the F60 module is in the RUN state and no global variables are connected to the system variables *Relay Contact 1…4*, contact 1-2 is closed (contact 2-3 is open). ▪ If the F60 module is in the RUN state and at least one of the system variables *Relay Contact 1…4* is TRUE, contact 1-2 is open (contact 2-3 is closed). ▪ If the F60 module is not in the RUN state, contact 1-2 is open (contact 2-3 is closed). ▪ If the F60 module is de-energized, contact 1-2 is open (contact 2-3 is closed).	Application-specific
Reload Deactivation	Y	Locks the execution of reload. Default setting: FALSE.	Application-specific
User LED 1, User LED 2	N	Applicable only for special controllers! Controls the corresponding LED, if existing. Default setting: 0 ms	---

Table 18: 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.

7.4.2 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.

7.5 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.

7.5.1 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 7.5.3 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.

7.5.2 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.

7.5.3 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 HIMatrix 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!

7.5.4 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.

7.5.5 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.

Refer to the system manual (HI 800 141 E) and the SILworX online help for further details on forcing.

7.5.5.1 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.

7.5.5.2 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.

7.6 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).

8 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.

8.1 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 HIMatrix 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.

8.1.1 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.

8.1.1.1 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.

8.1.2 Programming Steps

To program HIMatrix 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!

8.1.3 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.

8.1.4 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 19: System Parameters of the User Program

8.1.5 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.

8.1.6 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.

8.1.7 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.

8.1.8 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.

The use of reload requires a license. For further details on reload, refer to the HIMatrix system manual (HI 800 141 E).

8.1.9 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).

8.1.10 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 20: 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!

8.1.11 Changing the System Parameters during Operation

The system parameters specified in Table 21 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 21: Online Changeable Parameters

8.1.12 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.
- Code generator details.

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

8.1.13 Multitasking

Multitasking refers to the capability of the HIMatrix systems to process up to 32 user programs within the processor system.

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

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

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

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 as many processor system cycles later as the value set in the system parameter *Program's Maximum Number of CPU Cycles*. In the worst case, the following sequence is possible:
  - Program A writes to global variables needed by program B.
  - Program A stops its cycle in the same processor system cycle in which program B starts its cycle.
  - Program B is only able to read the values written to by program A when its next cycle starts.
  - The duration of the cycle just started by program B can be *Program's Maximum Number of CPU Cycles * Cycle Time*. Only at this point, program B adopts the values written to by program A.
  - It may take more than the configured *Program's Maximum Number of CPU Cycles* of the processor system until B reacts to these values!

⚠ CAUTION

Reciprocal influence of user programs is possible!

The use of the same global variables in several user programs can lead to a variety of unintentional consequences caused by the reciprocal influence of the user programs.

- **Carefully plan the use of the same global variables in several user programs.**
- **Use the cross-references in SILworX to check the use of global data. Global data may only be assigned values by one entity, either within a user program, from safety-related inputs or through safety-related communication protocols!**

The user is responsible for ensuring that operation is not disturbed by a reciprocal influence of the user programs!

For further details on multitasking, refer to the HIMatrix system manual (HI 800 141 E).

8.1.14 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 HIMatrix system that have already been approved.

8.2 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/.

9 Configuring Communication

In addition to physical input and output variables, variables can also be exchanged with another system via a data connection. In this case, the variables are declared with SILworX, in the Protocols area of the corresponding resource.

This data exchange can occur in either read-only or read/write mode.

9.1 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.

Depending on the controller variant, the following standard protocols are available:

- SNTP
- Send/Receive TCP
- Modbus (master/slave)
- PROFIBUS DP (master/slave)
- PROFINET and PROFIsafe (as of CPU BS V7)

All standard protocols are interference-free with respect to the safe processor system.

9.2 Safety-Related safe**ethernet Protocol**

safe**ethernet** must be used for safety-related data exchange between safety-related components.

As a HIMatrix system component, safe**ethernet** is certified up to SIL 4.

Use the safe**ethernet** Editor / P2P Editor to configure how safety-related communication is monitored.

For determining the safe**ethernet** parameters *Receive Timeout* and *Response Time*, the following condition applies:
The communication time slice must be sufficiently high to allow all the safe**ethernet** connections to be processed within one CPU cycle.

For safety-related functions, which are implemented via safe**ethernet**, the setting **Use Initial Value** must be used.

NOTICE

The safe state may be entered inadvertently!

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

If all values must be transferred, the value of a signal must either be present for longer than *Receive Timeout* or it must be monitored using a loop back.

9.2.1 Receive Timeout

Receive Timeout is the monitoring time in milliseconds (ms) within which a valid response from the communication partner must be received.

If a correct response is not received from the communication partner within *Receive Timeout*, safety-related communication is terminated. The input variables of this safe**ethernet** connection respond in accordance with the preset parameter *Freeze Data on Lost Connection [ms]*.

Since *Receive Timeout* is a safety-relevant component of the worst case response time T_R (see Chapter 3.2.3 et seqq.), its value must be determined as described below and entered in the safe**ethernet** Editor.

Receive timeout ≥ 4 * delay + 5 * max. cycle time

Condition: The communication time slice must be sufficiently high to allow all the safe**ethernet** connections to be processed within one CPU cycle.

Delay:	Delay on the transport path, e.g., due to switch or satellite.
Max. cycle time	Maximum cycle time of both controllers.

i A desired fault tolerance of the communication can be achieved by increasing *Receive Timeout*, provided that this is permissible for the application process in terms of time ().

NOTICE

The maximum value permitted for *Receive Timeout* depends on the application process and is configured in the safe**ethernet Editor, along with the expected maximum response time and the profile.**

9.2.2 Response Time

Response Time is the time period expressed in milliseconds (ms) until the sender of the message receives acknowledgement from the recipient.

When configuring the safe**ethernet** protocol, the **Response Time** expected to result from the physical conditions of the transport path must be set and a suitable safe**ethernet** profile must be selected.

The preset *Response Time* affects the configuration of all the safe**ethernet** connection parameters and is calculated as follows:

Response Time ≤ Receive Timeout / n

n = 2, 3, 4, 5, 6, 7, 8.....

The ratio between Receive Timeout and Response Time influences the capability of tolerating faults, e.g., when packets are lost (resending lost data packets) or delays occur on the transport path.

In networks where packets can be lost, the following condition must be given:

Min. response time ≤ receive timeout / 2 ≥ 2 * delay + 2.5 * max. cycle time

If this condition is met, the loss of at least one data packet can be intercepted without interrupting the safe**ethernet** / peer-to-peer connection.

i If this condition is not met, the availability of a safe**ethernet** connection can only be ensured in a collision and noise-free network. However, this is not a safety problem for the processor module!

i Make sure that the communication system complies with the configured response time!

If this condition cannot always be ensured, a corresponding connection system variable for monitoring the response time is available. If more than on occasion the measured response time exceeds the receive timeout by more than a half, the configured response time must be increased.

The receive timeout must be adjusted according to the new value configured for response time.

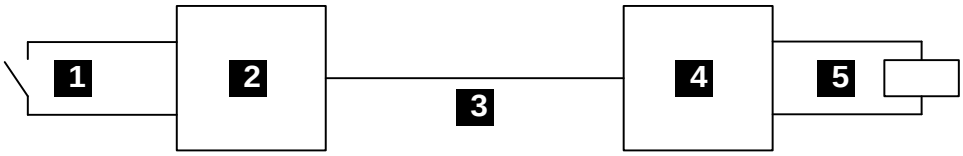
NOTICE

In the following examples, the formulas for calculating the worst case response time only apply for a connection with HIMatrix controllers if the safety time is set as follows.

safety time = 2 * watchdog time

9.2.3 Calculating the Worst Case Response Time

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



1 Input
2 Controller 1
3 Safety-related protocol
4 Controller 2
5 Output

Figure 4: Response Time when 2 HIMatrix Controllers are Interconnected

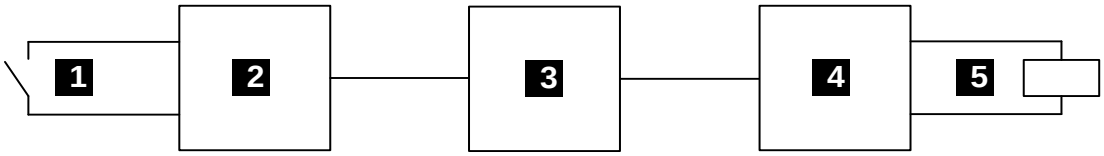
$$T_R = t_1 + t_2 + t_3$$

T_R	Worst case response time
t_1	2 ∗ watchdog time of controller 1.
t_2	Receive Timeout
t_3	2 ∗ watchdog time of controller 2

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

9.2.4 Calculating the Worst Case Response Time with 2 Remote I/Os

The worst case response time T_R is the time between a change on a field component input signal (in) of the first remote I/O module and a response on the corresponding output (out) of the second remote I/O module. It can be calculated as follows:



1 Input
2 Remote I/O 1
3 Controller
4 Remote I/O 2
5 Output

Figure 5: Response Time with Remote I/Os

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

T_R	Worst case response time
t_1	2 * watchdog time of remote I/O 1
t_2	Receive Timeout$_1$
t_3	2 ∗ watchdog time of the controller
t_4	Receive Timeout$_2$
t_5	2 * watchdog time of remote I/O 2

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

9.2.5 Terms

Receive Timeout	Monitoring time of controller 1 within which a valid response from controller 2 must be received. After the time has expired, safety-related communication is terminated.
Receive Timeout$_1$	Remote I/O 1 → controller
Receive Timeout$_2$	Controller → remote I/O 2
Watchdog time	Maximum permissible duration of a PES RUN cycle (cycle time).
Worst case	The worst case response time is the time between a change in a physical input (in) signal of controller 1 and a response in the corresponding output (out) of controller 2.

Data is transmitted using a safety-related protocol.

9.2.6 Assigning safe**ethernet** Addresses

Take the following points into account when assigning network addresses (IP addresses) for safe**ethernet**:

- The addresses must be unique within the network in use.
- When connecting safe**ethernet** to another network (company-internal LAN, etc.), make sure that no disturbances may occur. Potential sources of disturbances include:
  - Data traffic.
  - Coupling with other networks (e.g., Internet).

In these cases, implement suitable measures to counteract against such disturbances using Ethernet switches, firewall and similar.

---

i The operator is responsible for ensuring that the Ethernet used for safe**ethernet** communication or peer-to-peer communication is sufficiently protected against manipulations (e.g., from hackers).

The type and extent of the measures must be agreed upon together with the responsible test authority.

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
PE	Protective earth
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)
r_P	Peak value of a total AC component
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

HIMatrix Safety Manual for Railway Applications

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/

www.hima.com