HIMax®

Safety Manual

SAFETY
NONSTOP

HIMA

SAFETY

All of the instructions and technical specifications in this manual have been written with great care and effective quality assurance measures have been implemented to ensure their validity. For questions, please contact HIMA directly. HIMA appreciates any suggestion on which information should be included in the manual.

Equipment subject to change without notice. HIMA also reserves the right to modify the written material without prior notice.

For further information, refer to the CD-ROM and our website http://www.hima.de and http://www.hima.com.

Contact

HIMA Address

HIMA Paul Hildebrandt GmbH + Co KG

P.O. Box 1261

68777 Brühl, Germany

Tel: +49 6202 709-0

Fax: +49 6202 709-107

E-mail: info@hima.com

HI 801 003 E Rev.3.00 (0940)

Table of Contents

1 Safety Manual

Knowledge of regulations and the proper technical implementation of the safety instructions detailed in this manual performed by qualified personnel are prerequisites for safely planning, engineering, programming, installing and starting up the HIMax automation devices, as well as for ensuring safety during their operation and maintenance.

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

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

1.1 Validity and Current Version

Version Rev.3.0	This revision must be used with version 3 and higher of the HIMax system.

The most current version of the Safety Manual, indicated by the highest version number, is applicable and valid. The most current version is available on HIMA website at *www.hima.de*.

1.2 Objectives of the Manual

This manual contains information on how to operate the HIMax safety-related automation device in the intended manner. It provides an introduction to the safety concept of the HIMax system and should increase the reader's safety awareness.

The Safety Manual is based upon the contents of the certificate and of the test report for the certificate.

1.3 Target Audience

This manual addresses system planners, configuration engineers, programmers of automation devices and personnel authorized to start up, operate and maintain the devices and systems. Specialized knowledge of safety-related automation systems is required.

1.4 HIMA Address

© HIMA Paul Hildebrandt GmbH + Co KG
P.O. Box 1261
D-68782 Brühl near Mannheim

Telephone	(+49) 06202 709-0
Fax	(+49) 06202 709-107
E-Mail	info@hima.com
Internet	http://www.hima.com

1.5 Formatting Conventions

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

Bold:	To highlight important parts Names of buttons, menu functions and tabs that can be clicked and used in SILworX.
Italics:	For parameters and system variables
`Courier`	Literal user inputs
RUN	Operating state are designated by capitals
Chapter 1.2.3	Cross references are hyperlinks even though they are not particularly marked. When the cursor hovers over a hyperlink, it changes its shape. Click the hyperlink to jump to the corresponding position.

Safety notes and operating tips are particularly marked.

1.5.1 Safety Notes

The safety notes are represented as described below.
These notes must absolutely be observed to reduce the risk to a minimum. The content is structured as follows:

- Signal word: danger, warning, caution, notice
- Type and source of danger
- Consequences arising from the danger
- Danger prevention

⚠ SIGNAL WORD

Type and source of danger!

Consequences arising from the danger

Danger prevention

The signal words have the following meanings:

- Danger indicates hazardous situation which, if not avoided, will result in death or serious injury.
- Warning indicates hazardous situation which, if not avoided, could result in death or serious injury.
- Warning indicates hazardous situation which, if not avoided, could result in minor or modest injury.
- Notice indicates a hazardous situation which, if not avoided, could result in property damage.

NOTICE

Type and source of damage!

Damage prevention

1.5.2 Operating Tips

Additional information is structured as presented in the following example:

i The text corresponding to the additional information is located here.

Useful tips and tricks appear as follows:

TIP The tip text is located here.

2 Intended Use

This chapter describes the conditions for using HIMax systems.

2.1 Scope

The HIMax safety-related controllers can be used in applications up to:

- SIL 3 in accordance with IEC 61508.
- Category 4 in accordance with EN 954-1.
- Performance level e in accordance with ISO 13849-1.

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

All HIMax input and output modules (I/O modules) can be operated with an individual processor module or with several redundant processor modules.

When implementing safety-related communications between various devices, ensure that the overall response time does not exceed the fault tolerance time. All calculations must be performed in accordance with the rules given in the Safety Manual.

Only connect devices with safe electrical isolation to the communications interfaces.

2.1.1 Application in accordance with the 'De-Energize to Trip Principle'

The automation devices have been designed in accordance with the 'de-energize to trip' principle.

A system that operates in accordance with the 'de-energize to trip principle' does not require any power to perform its safety function.

Thus, if a fault occurs, the input and output signals adopt a de-energized, safe state.

2.1.2 Application in accordance with the Energize to Trip Principle

The HIMax controllers can 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 requires power (such as electrical or pneumatic power) to perform its safety function.

When designing the controller system, the requirements specified in the application standards must be taken into account. For instance, line diagnosis for the inputs and outputs may be required

2.1.3 Use in Fire Alarm Systems

All HIMax systems with analog inputs are tested and certified for used in fire alarm systems in accordance with DIN EN 54-2 and NFPA 72. To contain the hazard, these systems must be able to adopt an active state on demand.

The operating requirements must be observed!

2.2 Non-Intended Use

The transfer of safety-relevant data through public networks like the Internet is not permitted unless additional security measures such as VPN tunnel or firewall have been implemented to increase security.

With fieldbus interfaces, no safety-related communication can be ensured.

The use under environmental conditions other than those specified in the following section is not permitted.

2.3 Operating Requirements

The devices have been developed to meet the following standards for EMC, climatic and environmental requirements:

Standard	Content
EC/EN 61131-2	Programmable controllers, Part 2 Equipment requirements and tests
IEC/EN 61000-6-2	EMC Generic standards, Parts 6-2 Immunity for industrial environments
IEC/EN 61000-6-4	Electromagnetic Compatibility (EMC) Generic emission standard, industrial environments

Table 1: Standards for EMC, Climatic and Environmental Requirements

When using the safety-related HIMax control systems, the following general requirements must be met:

Requirement type	Requirement content
Protection class	Protection class II in accordance with IEC/EN 61131-2
Pollution	Pollution degree II in accordance with IEC/EN 61131-2
Altitude	< 2000 m
Enclosure	Standard: IP 20 If required by the relevant application standards (e.g., EN 60204, EN 954-1), the device must be installed in an enclosure of the specified protection class (e.g., IP 54).

Table 2: General requirements

2.3.1 Climatic Requirements

The following table lists the key tests and thresholds for climatic requirements:

IEC/EN 61131-2	Climatic tests
	Operating temperature: 0...+60 °C (test limits: -10...+70 °C)
	Storage temperature: -40...+85 °C
	Dry heat and cold resistance tests: +70 °C / -25 °C, 96 h, power supply not connected
	Temperature change, resistance and immunity test: -25 °C / +70 °C und 0 °C / +55 °C, power supply not connected
	Cyclic damp-heat withstand tests: +25 °C / +55 °C, 95 % relative humidity, power supply not connected

Table 3: Climatic Requirements

2.3.2 Mechanical Requirements

The following table lists the key tests and thresholds for mechanical requirements:

IEC/EN 61131-2	Mechanical tests
	Vibration immunity test: 5...9 Hz / 3.5 mm 9...150 Hz, 1 g, EUT in operation, 10 cycles per axis
	Shock immunity test: 15 g, 11 ms, EUT in operation, 2 cycles per axis

Table 4: Mechanical Tests

2.3.3 EMC Requirements

Higher interference levels are required for safety-related systems. HIMax systems meet these requirements in accordance with IEC 62061 and IEC 61326-3-1 (DIS). See column "Criterion FS“ (Functional Safety).

IEC/EN 61131-2	Interference immunity tests	Criterion FS
IEC/EN 61000-4-2	ESD test: 6 kV contact, 8 kV air discharge	-
IEC/EN 61000-4-3	RFI test (10 V/m): 26 MHz...1 GHz, 80 % AM RFI test (20 V/m): 26 MHz...2.7 GHz, 80 % AM: EN 298	- 20 V/M
IEC/EN 61000-4-4	Burst test: 2 kV power supply-, 1 kV signal lines	4 kV
IEC/EN 61000-4-12	Damped oscillatory wave test 2.5 kV L-,L+ / PE 1 kV L+ / L -	

Table 5: Interference Immunity Tests

IEC/EN 61000-6-2	Interference immunity tests	Criterion FS
IEC/EN 61000-4-6	High frequency, asymmetrical 10 V, 150 kHz...80 MHz, AM 20 V, 150 kHz...80 MHz, AM: EN 298	20 V
IEC/EN 61000-4-3	900 MHz pulses	
IEC/EN 61000-4-5	Surge: 2 kV, 1 kV	2 kV / 1 kV

Table 6: Interference Immunity Tests

IEC/EN 61000-6-4	Noise emission tests
EN 55011 Class A	Emission test: radiated, conducted

Table 7: Noise Emission Tests

2.3.4 Power Supply

The following table lists the key tests and thresholds for the device's power supply:

IEC/EN 61131-2	Review of the DC supply characteristics
	Alternatively, the power supply must comply with the following standards: IEC/EN 61131-2 or SELV (Safety Extra Low Voltage) or PELV (Protective Extra Low Voltage)
	HIMax devices must be fuse protected as specified in this manual
	Voltage range test: 24 VDC, -20 %...+25 % (19.2 V...30.0 V)
	Momentary external current interruption immunity test: DC, PS 2: 10 ms
	Reversal of DC power supply polarity test: Refer to corresponding chapter of the system manual or data sheet of power supply.
	Backup duration withstand test: Test B, 1000 h

Table 8: Review of the DC Supply Characteristics

2.3.5 ESD Protective Measures

Only personnel with knowledge of ESD protective measures may modify or extend the system or replace a module.

NOTE

Electrostatic discharge can damage the electronic components within the controllers!

- **When performing the work, make sure that the workspace is free of static and wear an ESD wrist strap.**
- **If not used, ensure that the module is protected from electrostatic discharge, e.g., by storing it in its packaging.**

Only personnel with knowledge of ESD protective measures may modify or extend the system wiring.

2.4 Requirements to be met by the Operator and the Machine and System Manufacturers

The operator and the machine and system manufacturers are responsible for ensuring that HIMax systems are safely operated in automated systems and plants.

The machine and system manufacturers must validate that the HIMax systems are correctly programmed.

2.5 Additional System Documentation

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

Name	Content	Document no.
		D = German
		E = English

HIMax System Manual	Hardware description of the modular system	HI 801 000 D HI 801 001 E
Certified test report [1)]	Test principles, safety requirements, results	
Manuals for the Components	Description of the individual components	
Communication Manual	safe**ethernet** and standard protocols	HI 801 100 D HI 801 101 E

[1)] Only supplied with the HIMax system

Table 9 Overview System Documentation

The documents are available as PDF files on HIMA website at *www.hima.com*.

3 Safety Concept for Using the PES

This chapter contains important general items on the fuctional safety of HIMax systems.

- Safety and availability
- Time parameters important for safety
- Proof test
- Safety requirements
- Certification

3.1 Safety and Availability

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

They can be used in applications up to safety integrity level SIL 3 in accordance with IEC 61508 or up to safety category Cat. 4 and up to performance level PL e in accordance with EN ISO 13849

All input and output modules (I/O modules) can be used with an individual processor module or with several redundant processor modules.

The HIMax systems have been tested and certified for use in fire alarm and fire-fighting systems in accordance with EN54 and NFPA72. To contain the hazard, these systems must be able to adopt an active state on demand.

No imminent danger results from the HIMax systems.

⚠ DANGER

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

Check all connections and test the entire system before starting up!

NOTE

System damage!

System damage caused by safety-related automation systems improperly connected or programmed.

Check all connections and test the entire system before starting up!

HIMA strongly recommends replacing failed modules as soon as possible.

3.1.1 Calculating the PFD and the PFH Values

The PFD and the PFH values have been calculated for the HIMax systems in accordance with IEC 61508.

IEC 61508-1 defines SIL 3 to have a PFD of 10^{-4}...10^{-3} and a PFH value of 10^{-8}...10^{-7} per hour.

HIMA will gladly provide the PFD, PFH and SFF values upon request. The "SILence“ tool is available for more detailed calculations.

A proof test interval of 10 years has been defined for the HIMax systems (offline proof test, see IEC 61508-4, paragraph 3.8.5).

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

3.1.2 Self-Test and Fault Diagnosis

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

- Processors
- Memory areas (RAM, non-volatile memory)
- Watchdog
- Connections between modules
- Individual channels of the I/O modules

If faults are detected during these tests, the defective module or the defective channel of the I/O module is switched off. If the tests detect a module fault while starting up the module, the module will not begin to operate.

In non-redundant systems, this means that sub-functions or even the entire PES will shut down. In case of a detected failure within a redundant system, the redundant module or redundant channel takes over the function to be performed

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

Further, the user program can also be used to evaluate various system variables that report the module status.

An extensive diagnostic record of the system's performance and detected faults are logged and stored in the diagnostic memory of the processor module or that of other modules. After a system fault, the recorded data can be read using the PADT.

For more information on evaluating diagnostic messages, see "Diagnostics“ in the System Manual HI 801 001.

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

3.1.3 PADT

Using the PADT, the user creates the program and configures the controller. The safety concept of the PADT supports the user in the correct implementation of the control task. The PADT takes numerous measures to check the entered information.

3.1.4 Redundancy

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

- The system bus has been designed to allow the use of redundant components. Two redundant system bus modules can be installed into each base plate (rack). Two redundant cables are used to connect the base plates.
- All processor modules can be used up to fourfold redundancy. Two variants are possible:
  - All processor modules can operate in the same base plate.
  - Processor modules can also be operated in two different base plates. In doing so, processor modules and I/O modules located in different places, can also be operated redundantly.
- With input and output modules, two or three modules can be defined as redundant to each other. A redundant channel will thus exist on each module for each corresponding channel on the other modules.

- The power supply can be designed for (two-fold) redundancy. The base plates contain a wiring for redundant supply of all modules.
- Using the safe**ethernet** certified by TÜV, it is possible to implement redundant communication connections between HIMax systems.

Defective modules can be replaced during operation. To do this, the defective module is removed from the base plate, and a new module is inserted. The new module automatically starts operation. A new processor module assumes the user program from the redundant processor module and is thus quickly ready for operation.

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

System operating in accordance with the 'energize to trip' principle, e.g., fire alarm and fire-fighting systems , have the following "safe states":

1. Safe state in the event of system shutdown.
2. State entered on demand, i.e., when performing the safety function. In such a case, the actuator is activated.

Observe the following points when structuring safety systems in accordance with the energize to trip principle:

- Ensuring the safety function in hazardous situations.
- Detection of failed system components and reaction:
  - Failure notification.
  - Automatic switching to redundant components, if necessary and possible.

Ensuring the Safety Function

The planner must make sure that the safety system is able to perform its safety function in hazardous situations. The safety function is performed when the safety system energizes one or several actuators and, as a consequence, a safe state is adopted, e.g., a fire compartment door is closed.

A redundant structure of the safety system components can be required to ensure the safety function, see Chapter 3.1.4:

- Power supply of the controller.
- Components of the controller: HIMax modules.
- When relay outputs are used, HIMA recommends to configure the relay outputs and the actuators' power supply redundantly.
  Reason:
  - A relay output has no line monitoring.
  - This step can be necessary to achieve the required SIL.

If the components are no longer operating redundantly due to a failure, repair of the failed component must be ensured at the earliest opportunity.

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

Detection of Failed System Components

The safety systems recognizes that components are not functioning and activates the redundant components. This is done with

- Self-tests of the HIMax modules.
- Line monitoring (short-circuits and open-circuits) with input and output modules. The modules must be configured accordingly.
- Additional inputs for monitoring the actuators, if required by the project.

3.2 Time Parameters Important for Safety

These are:

- Fault tolerance time
- Watchdog time
- Safety time
- Response time

3.2.1 Fault Tolerance Time (FTT)

The fault tolerance time (FTT) is a property of the process and describes the span of time during which the process allows faulty signals to exist before the system state becomes dangerous. A dangerous state can result if the fault exists for longer than the FTT.

3.2.2 Resource Watchdog Time

The watchdog time is set in the dialog for configuring the resource properties. This time is the maximum permissible duration of a RUN cycle (cycle time). If the cycle time exceeds the preset watchdog time, the processor module adopts the error stop state.

When determining the watchdog time, the following factors must be taken into account:

- Time required by the application, e.g., the duration of a cycle in the user program.
- Time required to manage the redundant processor modules.
- Time internally required to perform a reload.

The setting range for the watchdog time of the resource ranges

from 6 ms to maximum 7 500 ms.

The default setting is 200 ms.

When setting the watchdog time, the following must apply: **watchdog time ≤ ½ * safety time**

The watchdog time for a project is determined by a test on a complete system. During the test, all the processor modules are inserted in the base plate. The system operates in RUN mode with full load.

All communication links are operating (safe**ethernet** and standard protocols).

To determine the watchdog time

1. Set the watchdog time high for testing.
2. Use the system under the maximum load: In the process, all communication connections must be operating both via safeethernet and standard protocols. Frequently read the cycle time in the Control Panel and note the variations of the cycle time.
3. In succession, remove and reinsert every processor module in the base plate. Prior to removing one processor module, wait that the processor module that has just been inserted is synchronized.

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

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

For more information on how to insert and remove a processor module, refer to the X-CPU 01 manual HI 801 009.

4. In the diagnostic history, read the synchronization time from n to n+1 processor modules in every synchronization process and not it down. The longest synchronization time is used to determine the watchdog time.

5. Calculate the minimum watchdog time from the longest synchronization time + 12 ms spare + spare for the noted variations of the cycle time.

This allows one to calculate a suitable value for the watchdog time.

i In particular cases, the watchdog time calculated as described above might be too short for performing a reload.

TIP The determined watchdog time can be used as maximum cycle time in the safe**ethernet** configuration, see Communication Manual HI 801 101.

3.2.3 Watchdog Time of the User Program

Each user program has its own watchdog and watchdog time.

The watchdog time for the user program cannot be set directly. To calculate the watchdog time for a user program, HIMax uses its parameters *Max Duration for Each Cycle [µs]* and M*aximum Number of Cycles*. Refer to Chapters 10.2.3 and 10.2.10 for more details.

3.2.4 Safety Time (of PES)

The safety time is the maximum permissible time within which the PES must react to a safety requirement event. Safety requirement events include:

- Changes in input signals from process.
- Faults occurring in the controller.

In HIMax controllers, the safety time can be set anywhere between 20 ms and 22 500 ms.

Within the safety time of the controller, the self-test facilities detect whether there are any potentially dangerous faults. They trigger predefined fault reactions that set the faulty components to a safe state.

When determining the safety time, the effects of the following factors must be taken into account:

- With input modules, consider the following:
  Time-on/time-off delay settings for input channels:
  enter maximum delay time setting in µs + 4 ms
- Noise blanking also needs time reserves.

Choose a safety time that is long enough to account for the most significant factor mentioned above, but still lower than the FTT of the process. It is important not to neglect the sensor and actuator time parameters for the safety function.

The safety time for the controller is:

Safety time = 2 * watchdog time + reserve X

In the actual application, the user should measure reserve X by replacing a redundant processor module. Enter the average cycle time determined for the entire system as the reserve X into the above formula. This ensures maximum availability for the system.

3.2.5 Response Time

Assuming that no delay results from the configuration or the user program logic, the response time of HIMax controllers running in cycles is twice the system cycle time.

The cycle time of the controller consists of the following main components:

- Input processing.

  - Processing input data on input module.
  - Reading process data from communication interfaces.
  - Reading process data from input modules.
- Processing user program logic.
- Output processing.
  - Writing process data to output modules.
  - Writing process data to communication interfaces.
  - Processing output data on output modules.
- Additional processing of final actions for reloading, additional processor modules, etc.

3.3 Proof Test

A proof test is a periodic test performed to detect any hidden faults in a safety-related system so that, if necessary, the system can be restored to a state where it can perform its intended functionality.

HIMA safety systems must be subjected to a proof test **in intervals of 10 years**. This interval can often be extended by calculating and analyzing the implemented safety loops.

3.3.1 Proof Test Execution

The execution of the proof test depends on how the system (EUC = equipment under control) is configured, its intrinsic risk potential and the standards applicable to the equipment operation and required for approval by the responsible test authority.

According to IEC 61508 1-7, IEC 61511 1-3, IEC 62061 and VDI/VDE 2180 sheets 1 to 4, the operator of the safety-related systems is responsible for performing the proof tests.

3.3.2 Frequency of Proof Tests

The HIMA PES can be proof tested by testing the entire safety loop.

In practice, shorter proof test intervals are required for the input and output field devices (e.g., every 6 or 12 months) than for the HIMax controller. Testing the entire safety loop together with a field device automatically includes the test of the HIMax controller. There is therefore no need to perform additional proof tests of the HIMax controller.

If the proof test of the field devices does not include the HIMax controller, the HIMax controller must be tested for SIL 3 at least once every 10 years. This can be achieved by restarting the HIMax controller.

3.4 Safety Requirements

The following safety requirements must be met when using the safety-related PES of the HIMax system:

3.4.1 Hardware Configuration

Personnel configuring the HIMax hardware must observe the following safety requirements.

Product-Independent Requirements

- To ensure safety-related operation, only approved fail-safe hardware modules and software components may be used. The approved hardware modules and software components are specified in the *Versionsliste der Module und der Firmware der HIMax-Systeme der Firma HIMA Paul Hildebrandt GmbH + Co KG (version list of modules and firmware for HIMax systems from HIMA Paul Hildebrandt GmbH + Co KG*). The latest versions can be found in the version list maintained together with the test authority.
- The operating requirements specified in this safety manual (see Chapter 'Operating Requirements') about EMC, mechanical, chemical, climatic influences must be observed.

Product-Dependent Requirements

- Only connect devices to the system that are safely electrically isolated from the power supply.
- The operating requirements detailed in the system manual, particularly those concerning supply voltage and ventilation, must be observed.

3.4.2 Programming

Personnel developing user programs must observe the following safety requirements.

Product-Independent Requirements

- In safety-related 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.

3.4.3 Requirements for Using the Programming Tool

- SILworX must be used for programming.
- Compiling the program twice in SILworX and comparing both of the created files ensures that the program was properly compiled.
- The correct implementation of the application specifications must be validated, verified and documented. A complete test of the logic must be performed by trial.
- In case of a change of the user program, at minimum test all the parts of the logic concerned by the changes.
- The system response to faults in the safe input and output modules must be defined in the user program in accordance with the system-specific safety-related conditions.

3.4.4 Communication

- When implementing safety-related communications between the various devices, ensure that the system's overall response time does not exceed the fault tolerance time. All calculations must be performed in accordance with the rules given in 11.2.
- The transfer of safety-relevant data through public networks like the Internet is not permitted unless additional security measures have been implemented such as: VPN tunnel.
- If data are transferred through company-internal networks, administrative or technical measures must be implemented to ensure sufficient protection against manipulation (e. g. using a firewall to separate the safety-relevant components of the network from other networks).
- Never use the standard protocols to transfer safety-related data.
- All devices to be connected to the communication interfaces must be equipped with safe electrical isolation.

3.4.5 Maintenance Work

- Maintenance work must be performed in accordance with the current version of the document "Maintenance Override“ document published by TÜV Rheinland and TÜV Product Service.
- Whenever necessary, the operator must consult with the test authority responsible for the final inspection of the system and define administrative measures appropriate for regulating access to the systems.

3.5 Certification

HIMA safety-related automation devices (Programmable Electronic Systems, PES) of the HIMax system have been tested and certified by TÜV for functional safety in accordance with CE and the standards listed below:

TÜV Rheinland Industrie Service GmbH

Automation, Software und Informationstechnologie
Am Grauen Stein
51105 Köln

Certificate and test report
safety-related automation devices HIMax

Intended use: Safety Related Programmable Electronic System for process control, Burner Management (BMS), emergency shut down and machinery, where the demand safe state is the de-energized state.
Applications, where the demand state is the de-energized or energized state".

International standards:

EN / IEC 61508, Parts 1-7: 2000		SIL 3
EN / IEC 61511: 2004		SIL 3
EN 954-1: 1996		Category 4
EN / ISO 13849-1: 2008		Performance level e
EN / IEC 62061: 2005	Incl. Cor 1 und Cor 2: 2009	
EN 50156-1: 2006		
EN 12067-2: 2004		
EN 298: 2004	+Cor 1: 2006	
EN 230: 2005		
NFPA 85: 2007		
NFPA 86: 2007		
EN / IEC 61131-2: 2007		
EN / IEC 61000-6-2: 2005		
EN 61000-6-4: 20071		
EN 54-2: 1997	/A1: 2007	
NFPA 72: 2002		

Chapter 'Operating Requirements' contains a detailed list of all environmental and EMC tests performed.

All devices have received the CE mark of conformity.

To program the HIMax devices, a PADT is required, that it a PC running

SILworX

programming software. This software helps the user operate the automation devices and create safety-related programs using Function Block Diagrams (FBD) and Sequential Function Charts (SFC) in accordance with IEC 61131-3. For more details, refer to the SILworX documentation.

4 Processor Modules

The processor module's safety function is maintained by processing the user program with two processors that constantly compare their data. If a fault occurs, the watchdog sets the module to the safe state and reports the CPU state.

Refer to the manual for further details about the processor modules.

4.1 Self-Tests

The following section specifies the most important self-test routines of controllers' safety-related processor modules and the coupling to the I/O level.

- Processor test
- Memory test
- Comparator test
- CRC test with non-volatile memories
- Watchdog test

4.1.1 Reactions to Faults in the Processor Module

A hardware comparator within the processor module permanently compares the data of the microprocessor system 1 to those of the microprocessor system 2. If they are different, or if the test routines detect failures in the processor module, the controller automatically assumes the error stop state and the watchdog signal is switched off. The processor module does no longer process the user program and sets the outputs into the de-energized, switched-off state.

4.2 Slots Permitted for Processor Modules

The following rules must be observed when assigning the slots to the processor modules, also in the Hardware Editor:

1. A maximum of four processor modules may be used.
2. Processor modules may only be inserted in the following slots:
   - Slots 3 to 6 on rack 0.
   - Slots 3 to 4 on rack 1.
3. Slot 5 on rack 0 and slot 4 on Rack 1 may not simultaneously contain processor modules.
4. Slot 6 on Rack 0 and Slot 3 on Rack 1 may not simultaneously contain processor modules.

NOTE

System malfunction possible!

Only slots complying with these rules may be used for processor modules.

The table specified the recommended variants complying with the rules:

Variant	Base plate 0 Processor module(s) in slot:	Rack 1 Processor module(s) in slot:	Required system busses
1	3 for mono operation[1)]	-	A
2	3	-	A + B
3	3, 4	-	A + B
4	3, 4, 5	-	A + B
5	3, 4, 5, 6	-	A + B
6	3	3	A + B
7	3, 4	3	A + B
8	3, 4	3, 4	A + B
9	3, 4, 5	3	A + B

[1)] Mono operation:The project is configured in SILworX for mono operation and has only one processor module in Slot 3, at least one system bus module in Slot 1, I/O modules and possibly communication modules. The switch for mono start-up must be set within SILworX. It is always possible (and recommended!) to configure the system bus modules redundantly!

Table 10: Slot Positions Recommended for Processor Modules

HIMA recommends to use variant 3 even if variant 1 would be possible. In doing so, the processor module can be replaced without interrupting operation.

Since the operating system is designed to ensure maximum availability, other combinations are possible, but not recommended. This allows HIMax to offer more flexibility, e.g., when replacing modules or modifying the system. However, after such measures have been completed, the system should be structured such that it corresponds to one of the recommended variants noted in Table 10.

4.3 Availability of Processor Modules

One to four redundant processor modules can be used on a HIMax system. The system is still functional even if only one processor modules is available.

When a failed processor module has been replaced, the new processor module copies the user program from the still functioning processor module (self education) and initiates redundant operation.

The availability can be also increased during operation by adding additional processor modules – up to a total of four processor modules.

In this scenario, the user program must be configured for redundant operation.

4.4 Replacing Processor Modules

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

NOTE

⚠ **Disruption of the safety-related operation possible!**

The operation of the controller can be interupted by exchanging a processor module, on which the Ess **LED is lit or blinking.**

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

If the **Ess** LED is lit or blinking, the processor module is required for the system to function.

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

Refer to the manual of the processor module HI 801 009 and to the system manual HI 801 001 for more details about replacing processor modules.

5 System Bus Module

A system bus module administrates one of the two safety-related system busses. The two system busses are redundant to one another. Each system bus interconnects the various modules and base plates. The system busses transfer data using a safety-related protocol.

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

5.1 Slots Permitted for System Bus Modules

Only Slot 1 and Slot 2 of each base plate are permitted for the system bus modules. No other modules may (nor can) be plugged in to these slots.

If just one system bus is being used, the system bus modules must be inserted into Slot 1 of each base plate. In such a case, a blank module must be inserted in Slot 2.

NOTE

System malfunction possible!

Slot configurations, other than those specified here, are not permitted for system bus modules!

5.2 Rack ID

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

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

The rack ID is stored in the connector bioard of the system bus module and must be modified using the system bus module. Whenever the rack ID must be changed, e.g., when installing a new HIMax system, follow the instructions given in the system manual.

5.3 Responsibility

Only one of the system bus module contained in each system bus may receive the "responsible" attribute and thus be configured as "responsible" for the system bus operation.

- For system bus A, the "responsible" attribute is reserved for the system bus module in Base Plate 0, Slot 1.
- For system bus B, the user can use SILworX to set the attribute.

The responsible system module must be either located in Base Plate 0 or Base Plate 1.

Make sure that this requirements are met prior to starting safety-related operation.

⚠ WARNING

Physical injury possible!

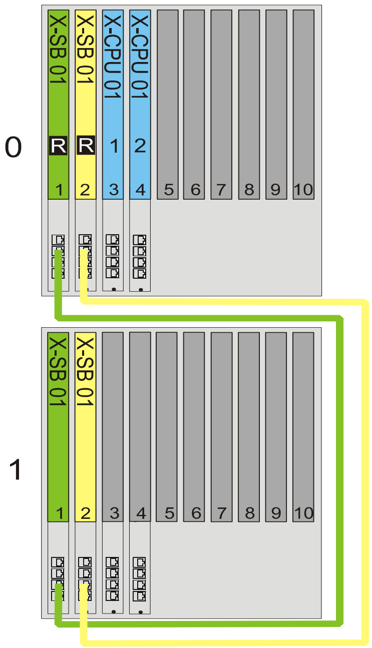
SILworX must be used to verify the configuration.

Proceed as follows:

- **In SILworX, log in to the system module on Base Plate 0, Slot 2.**
- **In SILworX, log in to the system module on Base Plate 1, Slot 2.**
- **Check the Control Panels of both system bus modules to ensure that the "responsible" attribute has only been set for the correct system bus module (see Figure 1 and Figure 2)!**

Recommended configurations:

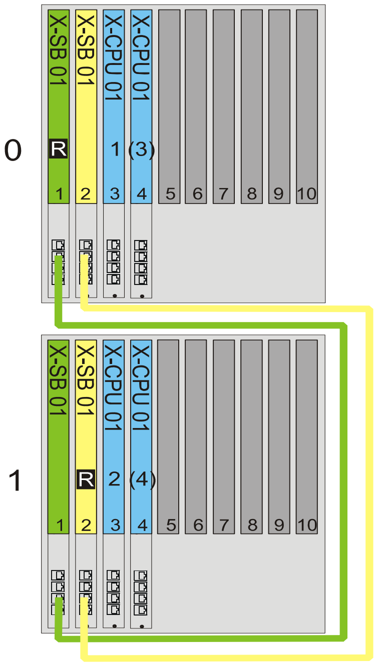
- If processor modules are only contained on base plate 0, both system bus modules on base plate 0 must be set to 'responsible' (Figure 1).
- If processor modules are also contained on base plate 1 (Figure 2), the following system bus modules must be set to 'responsible'.
  - On base plate 0, the system bus module in slot 1.
  - On base plate 1, the system bus module in slot 2.



Base plate 0: base plate 0 Base plate 1: base plate 1

R System Bus Module set to Responsible

Figure 1: Recommended Configuration: All Processor Modules on Base Plate 0



Base plate 0: base plate 0 Base plate 1: base plate 1

R System Bus Module set to Responsible

Figure 2: Recommended Configuration: Processor Modules on Base Plate 0 and on Base Plate 1

6 Communication Module

Communication modules control both non-safety-related data transfer through field busses and Ethernet, and safety-related data transfer to other HIMA controllers.

- The processor module controls safety-related data transfer using the safeethernet transfer protocol. The communication module forwards the data packets to the other systems. The safety-related protocol ensures that message any corrupted data in messages are detected.
- The standard protocols are for instance:
  - Modbus
  - PROFIBUS master/slave

A HIMax system can be equipped with a maximum of 20 communication modules.

For more information, see Chapter 11.1, the module manual HI 801 011 and the Communication Manual HI 801 101.

6.1 Slots Permitted for Communication Modules

The communication modules can be plugged in to any slot in any base plate with the exception of the following:

- Slot 1 and Slot 2 of each base plate, which are reserved for the system bus modules.
- Slots used for processor modules. Depending on the system configuration, these can be slots 3, 4, 5 and 6 of base plate 0 and slots 3 and 4 of base plate 1.

NOTE

System malfunction possible!

The slots reserved for processor and system bus modules must not be used for communication modules!!

7 Input Modules

Module	Number of channels	Safety-related	Non-reactive channels	Remark
Digital inputs X-DI 16 01	16	SIL 3	•	120 VAC
X-DI 32 01	32	SIL 3	•	
X-DI 32 02	32	SIL 3	•	Proximity switches (NAMUR)
X-DI 32 03	32	SIL 3	•	48 VDC
X-DI 32 04	32	SIL 3	•	With sequence of events recording
X-DI 32 05	32	SIL 3	•	Proximity switches (NAMUR), with sequence of events recording
X-DI 64 01	64	SIL 3	•	
Analog inputs X-AI 32 01	32	SIL 3	•	0/4...20 mA
X-AI 32 02	32	SIL 3	•	With sequence of events recording
Counter inputs X-CI 24 01	24	SIL 3	•	

Table 11: Overview of the Input Modules

7.1 General

Safety-related inputs can be used for both safety-related signals and non-safety-related signals. Non-safety-related signals, however, may not be used for safety functions!

In addition to the diagnostic LEDs for the module, the controllers also generate status messages for the user program that can be evaluated. I/O faults stored in the diagnostic memory can be read using the PADT.

Safety-related input modules automatically perform stringent, cyclic self-tests during operation.

If a fault occurs, a 0 signal is sent to the user program and, if possible, fault information is issued. The user program can read out the error code and evaluate this fault information.

For more information on the input modules, refer to the individual module manuals.

7.2 Safety of Sensors, Encoders and Transmitters

In safety-related applications, the PES and its connected sensors, encoders and transmitters must all meet the safety requirements and achieve the specified SIL. Also refer to Annex "Increasing the SIL Value of Sensors and Actuators".

7.3 Redundancy of Inputs and Input Modules

To increase availability, the input channels can be defined redundant to one another from within the SILworX Hardware Editor. The following conditions must be ensured:

- Input channels are located on different input modules of the same type.
- Input channels have the same channel number.

Both input channels are assigned to the same variable. If a sensor or an input module fails, the redundant channel supplies the values to the variable.

It is permitted to assign all of the input module channels or only a few of them to the corresponding channels of another input module.

7.4 Slots Permitted for Input Modules

The input modules can be plugged in to any slot in any base plate with the exception of the following:

- Slot 1 and slot 2 of each base plate, which are reserved for the system bus modules.
- Slots used for processor modules. Depending on the system configuration, these can be slots 3, 4, 5 and 6 of base plate 0 and slots 3 and 4 of base plate 1.

NOTE

System malfunction possible!

The slots reserved for processor and system bus modules must not be used for input modules!!

7.5 Safety-Related Digital Inputs

The digital input module reads its digital inputs one time per module cycle and stores the value internally. The module cyclically checks that the inputs are safely functioning.

Input signals that exist for a time shorter than the time between two samplings, i.e. shorter than a cycle time of the input module, are not be detected.

7.5.1 Test Routines

The online test routines check whether the input channels are able to forward both signal levels (L and H levels), regardless of the signals actually present on the input. This function test is performed each time the input signals are read.

7.5.2 Reaction in the Event of a Fault

If the test routines detect a fault of a digital input, the module sets the channel value in a way that the global variable assigned to the channel assumes the following values:

- For the most faults, the global variable assumes its initial value configured. The module sets the status Channel OK to FALSE
- For certain types of faults the module is able to put the channel into the safe state, but the module cannot issue a diagnosis entry.
  For these faults, the global variable adopts the value 0.

If the test routines detect a module or submodule fault, the module sets the *Module OK* or *Submodule OK* status to FALSE. Additionally, the module or submodule sets *Channel OK* to FALSE for all its channels.

In all these cases, the module activates the *Error* LED on the front plate.

7.5.3 Surges on Digital Inputs

NOTE

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

If shielded cables are not used, surge pulses (as defined in EN 61000-4-5) on digital inputs may be read as a temporary high level due to the short cycle time of HIMax systems.

Use noise blanking to avoid these types of faults: a signal must be present for at least two cycles before it is evaluated. At the same time, the safety time must not be exceeded.

7.6 Safety-Related Analog Inputs

Analog input channels convert the measured input currents into a value of type DINT (double integer), i.e., the "raw value“, and into a "process value“ of type REAL.

The safety-related precision is the guaranteed accuracy of the analog input without module fault reaction. This value must be taken into account when configuring the safety functions.

7.6.1 Test Routines

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

7.6.2 Reaction in the Event of a Fault

If the test routines detect a fault of an analog input, the module sets the channel value in a way that the global variable assigned to the channel assumes the following values:

- For the most faults, the global variable assumes its initial value configured.
- For certain types of faults the module is able to put the channel into the safe state, but the module cannot issue a diagnosis entry.
  For these faults, the global variable adopts the value 0.

The module sets the status *Channel OK* to FALSE

If the test routines detect a module or submodule fault, the module sets the *Module OK* or *Submodule OK* status to FALSE. Additionally, the module or submodule sets *Channel OK* to FALSE for all its channels.

In all these cases, the module activates the *Error* LED on the front plate.

7.7 Safety-Related Counter Inputs

Depending on its configuration, a safety-related counter input can provide the following process values:

- A counter reading as an integer value or as a scaled floating-point value.
- A rotation speed or frequency as an integer value or as a scaled floating-point value.
- Additional auxiliary values such as overflow.

For further details, refer to the module manual HI 801 113.

7.7.1 Test Routines

The module captures the counter values in parallel along three paths and compares the results with one another. Additionally, it tests the input path function cyclically.

7.7.2 Reaction in the Event of a Fault

If the test routines detect a fault of a counter input, the module sets the channel value in a way that the global variables assigned to the channel assume the following values:

- The global variables assigned to the -> *Rotation Speed [mHz] [INT]* and -> *Rot. Speed (scal.) [REAL]* parameter adopt the value 0.
- The global variable assigned to the -> *Counter Reading* parameter adopts the last valid value.

The module sets the status *Channel OK* to FALSE

If the test routines detect a module or submodule fault, the module sets the *Module OK* or *Submodule OK* status to FALSE. Additionally, the module or submodule sets *Channel OK* to FALSE for all its channels.

In all these cases, the module activates the *Error* LED on the front plate.

7.7.3 Observe the following points when using the X-CI 24 01 counter module.

When the X-CI 24 01 is used, the following particularities must be observed, also refer to the module manual HI 801 113:

- During a reload, the pulses on the input can be lost for up to three user program cycles if the following parameters are changed:
  - Edge evaluation
  - Channel pairs in use
  - Adding unused channels does not influence the already used channels.
- The module filters and suppresses input signals whose frequency is outside the specified range. No fault reaction is triggered. Furthermore, the counter reading is not incremented and a frequency value of 0 Hz is output.
- During the edge evaluation "2 phases, 4 edges", the frequency or rotation speed is limited to 10 kHz. Incorrect process values can result if frequencies of > 10 kHz are used.
- If the channel sensor fails during the edge evaluation "2 phases, 4 edges", and no short-circuit or open-circuit was detected, the module only registers half of the actual frequency value.
- The channel parameters -> *Level* and -> *Count rev.* must not be used for safety-related applications!
- If the pulses to be counted are not present, the module cannot perform any safety tests.
- Pulse to be counted can be lost due to noise blanking with automatic restart.
- The automatic or manual module restart must be considered as application-specific.
- Application recommendations:
  - The use of redundant sensors is recommended with multiphase evaluation and recognition of rotation direction, otherwise a sensor failure cannot detected.
  - The configuration of noise blanking does not pose a safety risk during frequency measurements.

7.8 Checklists for Inputs

HIMA recommends using the available checklists for engineering, programming and starting up safety-related digital inputs. The checklists can be used for helping with planning as well as to demonstrate later on that the planning phase was carefully completed.

When engineering or starting up the system, you can fill out a checklist 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 you have considered and clearly recorded all requirements. The checklist also documents the relationship between the external wiring and the user program.

The checklists are available in Microsoft® Word® format on the HIMA website.

8 Output Modules

Module	Number of channels	Safety-related	Safely electrically isolated	Remark
digital outputs				
X-DO 12 02	12	SIL 3	-	
X-DO 24 01	24	SIL 3	-	
X-DO 24 02	24	SIL 3	-	48 VDC
X-DO 32 01	32	SIL 3	-	
digital relay outputs				
X-DO 12 01	12	SIL 3	•	230 VAC
analog outputs				
X-AO 2401	24	SIL 3	pairwise	

Table 12: Overview of the Input Modules

8.1 General

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

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

Using the corresponding error codes, the user has additional options for programming fault reactions in the user program.

For more information on the output modules, refer to the individual module manuals.

8.2 Safety of Actuators

In safety-related applications, the PES and its connected actuators must all meet the safety requirements and achieve the specified SIL. Also refer to Annex "Increasing the SIL Value of Sensors and Actuators".

8.3 Redundancy of Outputs and Output Modules

To increase availability, the output channels can be defined redundant to one another from within the SILworX Hardware Editor. The following conditions must be ensured:

- Output channels are located on different output modules of the same type.
- Output channels have the same channel number.

Both output channels are assigned to the same variable. If an actuator or one of the output modules fails, the redundant channel takes up the function.

It is permitted to assign all of the output module channels or only a few of them to the corresponding channels of another output module.

8.4 Slots Permitted for Output Modules

The input modules can be plugged in to any slot in any base plate with the exception of the following slots:

- Slot 1 and slot 2 on each base plate which are reserved for the system bus modules.
- Slots used by the processor modules. Depending on the system configuration, these can be slots 3, 4, 5 and 6 of base plate 0 and slots 3 and 4 of base plate 1.

NOTE

System malfunction possible!

The slots reserved for processor and system bus modules must not be used for input modules!!

8.5 Safety-Related Digital Outputs

The safety-related output channels are equipped with three testable switches connected in series. By this, the requirement for a safe, independent, second way of de-energizing is fulfilled. If a fault occurs, this integrated safety shutdown function safely de-energizes all channels of the defective output module (de-energized state).

Furthermore, the watchdog signal of the module is a second option for safety shutdown: If the watchdog signal is lost, the module immediately enters the safe state.

8.5.1 Test Routines for Digital Outputs

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

- Reading the output signals back from the switching amplifier. The switching threshold of a 0 signal that has been read back is below the valid voltage value for the type of output. The diodes used prevent a feed back of signals.
- Checking the integrated redundant safety shutdown.
- A shutdown test of the outputs is performed cyclically for max. 200 µs. The minimum interval between two tests can be set from within the SILworX Hardware Editor.

If faults occur, the outputs are set to the safe value.

8.5.2 Reaction in the Event of a Fault

If the test routines detect a fault in one or several channels, the module switches those channels off and by this brings them into the safe state. The parameter *Channel OK* is set to FALSE for these channels.

If the test routines detect a module or submodule fault, the module sets the *Module OK* or *Submodule OK* status to FALSE. Additionally, the module or submodule sets *Channel OK* to FALSE for all its channels.

In all cases, the module also indicates the fault by the *Error* LED on the faceplate.

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

If the output is short-circuited to L- or overloaded, the module is still testable. It is not necessary to transfer the module to the safe state.

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

NOTE

System malfunction possible!

The voltage induced during switching off inductive loads could cause faults in the controller or in other electronical systems close to the actor's input leads.

Therefore, it is a good practice to connect inductive loads with a suitable free-wheeling circuit at the actuator to counteract these disturbances.

8.6 Safety-Related Relay Outputs

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

- Electric isolation is required.
- Higher amperages are used.
- Alternating currents are to be connected.

The module outputs are equipped with two safety relays with forcibly guided contacts. The outputs can thus be used for safety shutdowns in accordace with SIL 3.

Furthermore, the watchdog signal of the module provides a second means of safety shutdown: If the watchdog signal is lost, the module immediately adopts the safe state.

8.6.1 Test Routines for Relay Outputs

The module is tested automatically during operation. The main test functions are:

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

8.6.2 Reaction in the Event of a Fault

If a faulty signal is detected, the affected module output is set to the safe, de-energized state using the safety switches. If a module fault occurs, all module outputs are switched off. Both types of faults are also indicated by the Error LED.

NOTE

System malfunction possible!

The voltage induced during switching off inductive loads could cause faults in the controller or in other electronical systems close to the actor's input leads.

Therefore, it is a good practice to connect inductive loads with a suitable free-wheeling circuit at the actuator to counteract these disturbances.

8.7 Safety-Related Analog Outputs

They forward the values determined in the user program to the actuators.

The safety-related analog outputs read back their output values and compare them to the values to be output. If the values differ, a fault reaction is triggered.

8.7.1 Test Routines for Analog Outputs

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

- Read back of the output signal.
- Checking the integrated redundant safety shutdown.

If faults occur, the outputs are set to the safe value 0 mA.

8.7.2 Reaction in the Event of a Fault

If the test routines detect a fault in one or several channels, the module switches the channel groups off and brings them into the safe state. The parameter *Channel OK* is set to FALSE for these channels.

If the test routines detect a module or submodule fault, the module sets the *Module OK* or *Submodule OK* status to FALSE. Additionally, the module or submodule sets *Channel OK* to FALSE for all its channels.

In all cases, the module also indicates the fault by the *Error* LED on the faceplate.

8.7.3 Behavior in the Event of External Open-Circuit

If an open-circuit occurs, the module switches the current off for approx. 8 ms and then checks if the open-circuit is still present. If this is the case, it switches off for approx. 10 s. This process can repeat indefinitely.

8.7.4 Observe the following points when using the analog X-AO 16 01 output module!

When the analog output module is used, the following particularities must absolutely be observed, also refer to the module manual HI 801 111:

- Only the wiring options specified in the module manual HI 801 111 are allowed!
- If more than two modules are redundantly connected in series, the SELV voltage can be exceeded!
- With serial redundancy, only one channel of each group of two channels may be used!
- If HART communication occurs between the connected actuator and a HART terminal, the output signal can deviate by up to 2 % of the upper value!
- If a fault occurs, the time to reach the safe state can take up to 16 ms in the worst case. Take this time into account when defining the reaction and safety times!
- The user program must not write to analog outputs in cycles shorter than 6 ms.
- If faults occur, the module outputs the safe value 0 mA, even if the upper limit of the setting range is exceeded.

8.8 Checklists for Outputs

HIMA recommends using the available checklists for engineering, programming and starting up safety-related digital outputs. The checklists can be used for helping with planning as well as to demonstrate later on that the planning phase was carefully completed.

When engineering or starting up the system, you can fill out a checklist 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 you have considered and clearly recorded all requirements. The checklist also documents the relationship between the external wiring and the user program.

The checklists are available in Microsoft® Word® format on the HIMA website.

9 Software

The software for the safety-related automation devices of the HIMax systems consist of the following components:

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

The *operating system* is loaded into the controller's processor module. HIMA recommends using the latest version valid for the safety-related applications.

The user program is created using the SILworX programming system and contains the application-specific functions to be performed by the automation device. SILworX is also used to configure it.

The user program is compiled with the code generator and transferred to the non-volatile memory automation device through an Ethernet interface.

9.1 Safety-Related Aspects of the Operating System

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 list maintained together with TÜV,

The current version of the operating system can be read using SILworX. The user must verify wether a valid version of the operating system has been loaded into the modules (see 10.3 Checklist for Creating a User Program).

9.2 Operation and Functions of the Operating System

The operating system executes the user program cyclically. Basically, this involves performing the following functions:

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

The following basic functions are also executed:

- Comprehensive self-tests
- Test of I/O modules during operation,
- Data transfer
- Diagnosis
- Handling redundancy if processor modules are added or removed.

Fore more information, see HIMax System Manual HI 801 001. For more information, see HIMax System Manual HI 801 001.

9.3 Safety-Related Aspects of Programming

When creating a user program, the requirements detailed in this section must be observed.

9.3.1 Safety Concept of SILworX

The safety concept of SILworX:

- When SILworX is installed, a CRC checksum helps ensure the program package's integrity on the way from the manufacturer to the user.
- SILworX performs validity checks to reduce the likelihood of faults while entering data.

- Compiling the program twice and comparing the two CRC checksums ensures that data corruption in the application is detected that can result from random faults in the PC in use.

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

Function Test of the Controller

1. Verify that the tasks to be performed by the controller were properly implemented using the data and signal flows
2. Perform a comprehensive function test of the logic by trial (see Testing the configuration and the appl.).

The controller and the application are sufficiently tested.

If a user program is modified, only the program components affected by the change must be tested. To do this, the safe revision comparator in SILworX can be used to determine and display all changes relative to the previous version.

9.3.2 Verifying the Configuration and the User Program

To verify that the user program created performs the required safety function, the user must create suitable test cases for the required system specification.

An independent test of each loop (consisting of input, the key interconnections in the application and output) is usually sufficient.

Suitable test cases must also be created for the numerical evaluation of formulas. Equivalence class tests are reasonable . These are tests within defined ranges of values, at the limits of or within invalid ranges of values. 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 reccommends not to do without performing an active simulation with data sources, since this is the only way to prove that the sensors and actuators in the system (also those connected to the system via communication with remote I/Os) are properly wired. This is also the only way to verify the system configuration.

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

9.4 Resource Parameters

⚠ DANGER

Physical injury possible due to defective configuration!

Neither the programming system nor the controller can verify certain project-specific parameters. For this reason, enter these parameters correctly and verify the whole entry.

These parameters are:

- **System ID**
- **Rack ID, see 5.2 and System Manual HI 801 001.**
- **'Responsible' attribute of system bus modules, see 5.3**
- **Safety Time**
- **Watchdog Time**
- **Main Enable**
- **Autostart**
- **Start Allowed**
- **Load Allowed**
- **Reload Allowed**
- **Global Forcing Allowed**

The following parameters are defined in SILworX for the operations permissible in the safety-related operation of the resource and are referred to as safety-related parameters.

Parameters 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 responsible test authority for each separate implementation of the automation device.

9.4.1 System Parameters of the Resource

The system parameters of the resource can be set in SILworX, in the *Properties* dialog box of the resource.

Parameter / Switch	Description	Default value	Setting for safe operation
Name	Resource name		Arbitrary
System ID [SRS]	System ID of the resource 1...65 535 Assign a value different from the default value to the system ID, otherwise the project cannot be executed!	60 000	Unique value within the controller network. This network includes all controllers that can potentially be interconnected
Safety Time [ms]	Safety time in milliseconds 20...22 500 ms	600 ms	Application-specific
Watchdog Time [ms]	Watchdog time in milliseconds 6...7500 ms	200 ms	Application-specific
Main Enable	ON: The following switches/parameters can be changed during operation (= RUN) using the PADT. OFF: The parameters cannot be changed during operation.	ON	OFF is recommended
Autostart	ON: If the processor module is connected to the supply voltage, the user program starts automatically OFF: The user program does not start automatically after connecting the supply voltage.	OFF	Application-specific
Start Allowed	ON: A cold start or warm start permitted with the PADT in RUN or STOP OFF: Start not allowed	ON	Application-specific
Load Allowed	ON: Download of the user program permitted OFF: Download of the user program not permitted	ON	Application-specific
Reload Allowed	ON: Reload of a user program permitted OFF: Reload of a user program not permitted. The reload process currently running is not aborted when switching to OFF	ON	Application-specific
Global Forcing Allowed	ON: Global forcing permitted for this resource OFF: Global forcing not permitted for this resource	ON	Application-specific
Global Force Timeout Reaction	Specifies how the resource should behave when the global force time-out has expired: ▪ Stop Forcing ▪ Stop resource	Stop Forcing	Application-specific
Max.Com. Time Slice ASYNC [ms]	Highest value in ms for the time slice used for communication during a resource cycle (see Communication Manual HI 801 101), 2...5000 ms	10 ms	Application-specific
Target Cycle Time [ms]	Targeted or maximum cycle time, see *Target Cycle Time Mode*, 0...7500 ms	0 ms	Application-specific
safeethernet CRC	SILworX V.2.36.0: The CRC for safe**ethernet** is created as in SILworX version 2.36.0. This setting is required for exchanging data with resources planned with SILworX version 2.36 or previous versions. Current Version: The CRC for safeethernet is created with the current algorithm	Current Version	Application-specific

Parameter / Switch	Description		Default value	Setting for safe operation
Multitasking Mode	Mode 1	The duration of a CPU cycle is based on the required execution time of all user programs.	Mode 1	Application-specific
	Mode 2	The processor provides user programs with a higher priority the execution time not needed by user programs with a lower priority. Operation mode for high availability.		
	Mode 3	The processor waits for the unneeded execution time of user programs to expire and thus increases the cycle.		
Sum of UP Max. Duration for Each Cycle [µs]	Sum of the values indicated for *Max. Duration for each Cycle [µs]* in all the user programs; not changeable.			-
Target Cycle Time Mode	Use of *Target Cycle Time [ms]*		Fixed	Application-specific
	Fixed	HIMax maintains the target cycle time and extends the cycle if necessary. This does not apply if the processing time of the user programs exceeds the target cycle time.		
	Dynamic	HIMax maintains the target cycle time as well as possible, but it also executes the cycle as quickly as possible.		
Minimum configuration version	SILworX-V2	The code is generated as in SILworX version 2, except for the new functions. This setting allows the reload of a project created with version 2.	SILworX-V2	Application-specific
	SILworX-V3	Code generation for HIMax version 3. This setting ensures the compatibility with future versions.		

Table 13: System Parameters of the Resource

9.4.2 Hardware System Variables

These variables are used to change the behavior of the controller during operation in specific states. These variables can be set in the detailed view of the hardware, in the SILworX Hardware Editor.

Parameter / Switch	Function	Default setting	Setting for safe operation
Force Deactivation	Used to prevent forcing and to stop it immediately	OFF	Application-specific
Spare 0 ... Spare 16	No function	-	-
Emergency Stop 1 ... Emergency Stop 4	Emergency stop switch to shutdown the controller if faults are detected by the user program	OFF	Application-specific
Read-only in RUN	With the exception of forcing and reload, it is not possible to perform any operations (stop, start, download) with SILworX	OFF	Application-specific
Reload Deactivation	Prevents execution of reload	OFF	Application-specific

Table 14: Hardware System Variables

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

Example: A key switch is connected to a digital input. The digital input is assigned to a global variable associated with the system variable "Read only in Run“. The owner of a key can thus activate or deactivate the operating actions "stop", "start" and "download".

9.5 Forcing

Forcing is the procedure for replacing a variable's current value with a force value. The variable receives its current value from a physical input, communication or a logic operation. If the variable is forced, its value does no longer depend on the process, but is defined by the user.

Forcing is used for the following purposes:

- Testing the user program; especially under special circumstances or conditions that cannot otherwise be tested.
- Simulating unavailable sensors in cases where the initial values are not appropriate.

⚠ WARNING

Use of forced values can disrupt the safety integrity!

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

Forcing is only permitted after receiving consent from the test authority responsible for the final system acceptance test.

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 aspects. HIMA recommends to set a time limit for the forcing procedure, see 9.5.1.

Forcing can operate at two levels:

- Global forcing: Global variables are forced for all applications.
- Local forcing: Values of local variables are forced for an individual user program.

9.5.1 Time Limits

Time limits can be set both for global and local forcing. Once the defined time has expired, the controller stops forcing values.

It is also possible to define how the HIMax system behaves on completion of the forcing procedure:

- With global forcing, the resource is stopped or continues to run.
- With local forcing, the user program is stopped or continues to run.

It is also possible to use forcing without time limit. In this case, the forcing procedure must be stopped manually.

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

9.5.2 Restricting the Use of Forcing

The following measures can be configured to limit the use of forcing and thus avoid potential faults in the safety functionality due to improper use of forcing:

- Configuring different user profiles with or without forcing authorization
- Prohibit global forcing for a resource
- Prohibit local forcing
- Forcing can also be stopped immediately using a key switch.
  To do so, the system variable "Force deactivation“ must be linked to a digital input connected to a key switch.

⚠ **WARNING**

Use of forced values can disrupt the safety integrity!

Only remove existing forcing restrictions with the consent of the test authority responsible for the final system acceptance test.

9.5.3 Force Editor

SILworX Force Editor lists all variables, grouped in global and local variables.

For each variable, the following can be set:

- Force value
- Switch-on or off a force switch to prepare for forcing variables

Forcing can be started and stopped for both local and global variables.

When starting, set a time limit or start forcing for an indefinite time period. If none of the restrictions apply, all variables with an active force switch are set to their force values.

If forcing is stopped manually or because the time limit has expired, the variables will again receive their values from the process or the user program.

For more information about the Force Editor and forcing, refer to the SILworX Online Help.

Basic information on forcing can be found in the TÜV document "Maintenance Override“.

This document is available on the TÜV homepage:

`http://www.tuv-fs.com or`
`http://www.tuvasi.com.`

9.6 Protection against Manipulation

Together with the responsible test authority, the user must define which measures should be implemented to protect the system against manipulation.

Protective mechanisms for preventing unintentional or unapproved modifications to the safety system are integrated into the PES and SILworX:

- Each change to the user program or configuration creates a new CRC. These changes can only be transferred to the PES via download or reload.
- The operating options depend on the user login into the PES.
- SILworX prompts the user to enter a password in order to connect to the PES.
- No connection is required between the PADT and PES in RUN.

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

⚠ **DANGER**

Danger: Physical injury due to unauthorized manipulation of the controller!

Protect the controller against unauthorized access!

For instance, change the default settings for login and password!

PES data can only be accessed if the PADT in use is operating with the current version of SILworX and the user project is available in the currently running version (archive maintenance!).

The user only need to connect the PADT to the PES when loading the user program or performing diagnostics. The PADT is not required during normal operation. Disconnecting the PADT and PES during normal operation helps to prevent against unauthorized access.

9.7 Locking and Unlocking the PES

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

Unlocking the PES: Deactivates any locks previously set (e.g., to perform work on the controller).

Three system variables serve for locking:

Variable	Function
Read only in Run	ON: Starting, stopping, and downloading the controller are locked. OFF: Starting, stopping, and downloading the controller are possible.
Reload Deactivation	ON: Reload is locked. OFF: Reload is possible.
Force Deactivation	ON: Forcing is deactivated. OFF: Forcing is possible.

Table 15: System Variables for Locking and Unlocking the PES

If all three system variables are ON: no access to the controller is possible. In this case the controller can only be put into STOP state by restarting a processor module with the mode switch in the Init position. Then loading a new user program is possible.

Example for using these system variables:

To make a controller lockable

1. Define a global variable of type BOOL and set its initial value to FALSE.
2. Assign the global variable as output variables to the three system variables *Read only in Run*, *Reload Deactivation*, and *Force Deactivation*.
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, lod it on the controller, and start it.

The owner of a corresponding key is able to lock and unlock the controller. In case of a fault of the corresponding digital input module, the controller is unlocked.

10 User Program

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

10.1 General Sequence

General sequence for programming HIMax automation devices for safety-related applications:

1. Specify the controller functionality
2. Write the user program
3. Compile the user program:
   the user program is error-free and can run
4. Verify and validate the user program.

Upon completing these steps, the user program can be tested and the PES can begin the safe operation.

10.2 Scope for Safety-Related Use

(For more on specifications, regulations and explanation of safety requirements, see Chapter 3.4 "Safety Requirements")

The user program must be written using the SILworX programming software. This program runs on Windows XP® Servicepack 2 for PCs.

Essentially, SILworX includes:

- Input (Program Editor), monitoring and documentation
- Global variables with symbolic names and data types (BOOL, UINT, etc.)
- Assignment of HIMax controllers (Hardware Editor)
- Compilation of user program into a format that can be loaded into the PES
- Configuration of communication

10.2.1 Programming Basics

The tasks to be performed by the controller should 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 specification format depends on the tasks to be performed. These include:

Combinational logic

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

Sequential controllers (sequence control system)

This is a written description of the steps and their enabling conditions, and a description of the actuators to be controlled.

- Flow charts
- Matrix or table form of the step enabling conditions and the actuators to be controlled
- Definition der Randbedingungen, z. B. Betriebsarten, NOTAUS usw.

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

Sensors (digital or analog)

- Signals during normal operation ('de-energize-to-trip' principle with digital sensors, 'life-zero' with analog sensors)
- Signals in the event of a fault:

Definition of required safety-related redundancies (1oo2, 2oo3)

- Discrepancy monitoring and reaction

Actuators

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

Programming goals for user program

- Easy to understand.
- Easy to trace and follow.
- Easy to test.
- Easy to modify.

10.2.2 Functions of the User Program

Programming is not subject to hardware restrictions. The user program functions can be freely programmed.

When programming, account for the 'de-energize-to-trip' principle for the physical inputs and outputs. Only elements complying with IEC 61131-3 together with their functional requirements are permitted 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 program includes meaningful 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.

The "packaging" of functions in self-created function blocks and functions consisting of standard functions is reccommended. This ensures that a user program can be clearly structured in modules (functions, function blocks). Each module can be viewed and tested on an individual basis. By grouping smaller modules into larger ones and then all together into a single user program, the user is effectively creating a comprehensive, complex function.

10.2.3 System Parameters of the User Program

The following user program switches and parameters can be set in the *Properties* dialog box of the user program:

Switch / Parameter	Function	Default value	Setting for safe operation
Name	Name of the user program		Arbitrary
Safety Integrity Level	Safety integrity level: SIL0 to SIL3 (for purposes of documentation only).	SIL3	Application-specific
Start	ON: The PADT may be used to start the user program. OFF: The PADT may not be used to start the user program	ON	Application-specific
Program Main Enable	It enables changes of other user program switches: Only the enable switch of the resource is relevant!	ON	-
Autostart	Enabled type of Autostart: Cold Start, Warm Start, Off	Cold start	Application-specific
Test Mode Permitted	ON The test mode is permitted for the user program. OFF The test mode is not permitted for the user program.	OFF	Application-specific
Local Forcing Allowed	ON: Forcing Permitted at Program Level OFF: Forcing not Permitted at Program Level	OFF	OFF is recommended
Reload Allowed	ON: User program reload is permitted OFF: User program reload is not permitted	ON	Application-specific
Max. Duration for Each Cycle [µs]	Maximum time in each processor module cycle for executing the user program: 1...7 500 000 µs, no limitations	0 µs	Application-specific
Local Force Timeout Reaction	Behavior of the user program after the forcing time has expired: ▪ Stop Forcing Only. ▪ Stop Program.	Stop Forcing Only.	-
Program ID	ID for identifying the program as displayed within SILworX, 1..32	1	Application-specific
Watchdog Time [ms] (calculated)	Monitoring time of the user program, calculated from the maximum number of cycles and the watchdog time of the resource Not changeable!		
Code Generation Compatibility	SILworX V3 SILworX V2	SILworX V3	Application-specific

Table 16: System Parameters of the User Program

10.2.4 Code Generation

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

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

NOTE

Faulty operation of controller possible!

Before loading a user program for safety-related operation, the user program must be first compiled twice. Both versions generated must have the same CRC.

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 sporadic faults in the hardware or operating system of the PC in use .

10.2.5 Downloading and Starting the User Program

A PES in the HIMax system cannot be downloaded until it is set to the STOP state.

Currently, only one user program can be loaded into a given PES. The system monitors that the user program is loaded completely. Afterwards, the user program can be started, i.e. the routine begins to be processed in cycles.

i HIMA recommends backing up project data, e.g., on a data storage medium, after loading a user program into the controller, even by performing a reload.

This is done to ensure that the project data corresponding to the configuration loaded into the controller remains available even if the PADT fails.

HIMA recommends a data back up on a regular basis also independently from the program load.

10.2.6 Reload

If user programs were modified, the changes can be transferred to the PES during operation. After being tested by the operating system, the modified user programs are activated and it assumes the control task.

i Note that the reload does not take the current state into account if changes are performed to user programs containing a finite state machine. This can lead to an unpredictable program behavior after the reload process. A program contains a finite state machine if it is composed of sequential function chart elements such as s*teps*, or function blocks with storage capacity, e.g., *RS* Flipflops.

In such a case, the execution of a reload must be planned thoroughly!

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 project configuration.

i **Reload can be aborted**

A successful reload is ensured by planning a sufficient reserve for the reload when determining the watchdog time or temporarily increasing the controller watchdog time by a reserve of at least 30% and by at least 4*X ms.

Any temporary increases in the watchdog time must be coordinated with the responsible test authority.

The reload can only be performed if the "Reload permitted“ system parameter is set to ON and the "Reload deactivation“ system variable is set to OFF.

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

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

10.2.7 Online Test

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

For more information on how to use OLT fields, enter "OLT field" in the SILworX online help.

10.2.8 Single Step Mode

To diagnose faults during the online test, the user program can be run in single steps, i.e., cycle for cycle. Each cycle is triggered by a command from the PADT.

This function can only be used if the **Freeze Allowed** system parameter is set to ON in the corresponding user program.

State	Description
OFF	Single step mode impossible
ON	Single step mode possible (default setting)

Table 17: User Program Switch **Freeze Allowed**

NOTE

Failure of safety-related operation possible!

The single step mode must not be used in safety-related operation!

10.2.9 Program Documentation for Safety-Related Applications

SILworX allows the user to automatically print the documentation for a project. The most important documentation includes:

- 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 acceptance test of a system subjected to approval by a test authority (e.g., TÜV).

10.2.10 Multitasking

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

This allows the project's sub-functions to be separated from one another. The individual user programs can be started, stopped and loaded independently by performing a reload. SILworX displays the states of the individual user programs on the Control Panel and allows the user to operate them.

In a simplified overview, the processor module cycle (CPU cycle) of only one user program is composed of the following phases:

1. Process the input data.
2. Run the user program.
3. Supply the output modules with output data.

The overview does not include special tasks that might be executed within a CPU cycle such as reload or synchronization of processor modules.

Using multitasking, the second phase changes so that a CPU cycle runs as follows:

1. Process the input data.
2. Process all the user programs.
3. Supply the output modules with output data.

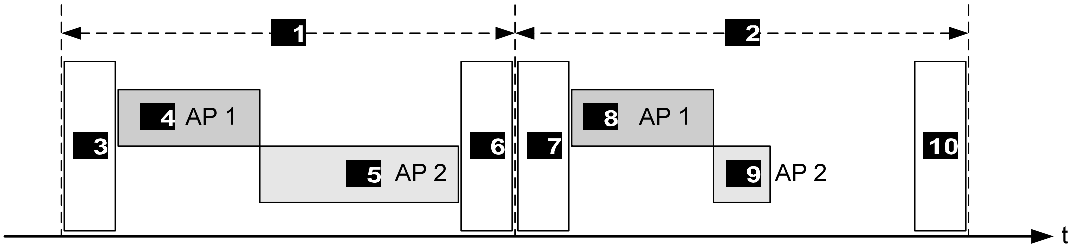
In the second phase, the HIMax can run up to 32 user programs. Two scenarios are possible for each user program:

- An entire user program cycle can be run within a single CPU cycle.
- A user program cycle requires multiple CPU cycles to be completed.

These two scenarios are even possible if only **one** user program exists.

It is not possible to exchange global data between user programs within a single CPU cycle. Data written by a user program is made available immediately before phase 3, but after the user program execution has been completed. This data can thus first be used as input values at the next start of another user program.

The example in Figure 3 shows both scenarios in a project containing two user programs.



1 First CPU Cycle Considered
2 Second CPU Cycle Considered
3 Input Processing in the First CPU Cycle
4 First UP 1 Cycle Considered
5 First Portion of the Considered UP 2 Cycle
6 Output Processing in the First CPU Cycle
7 Input Processing in the Second CPU Cycle
8 Second UP 1 Cycle Considered
9 Second Portion of the Considered UP 2 Cycle
10 Output Processing in the Second CPU Cycle

Figure 3: CPU Cycle Sequence with Multitasking

Each UP 1 cycle is completely processed during each CPU cycle. UP 1 processes an input change registered by the system at the beginning of the CPU cycle 1 and delivers a reaction at the end of the cycle.

One UP2 cycle requires two CPU cycles to be processed. UP 2 needs CPU cycle 1 to process an input change registered by the system at the beginning of CPU cycle 2. For this reason, the reaction to this input change is only available at the end of CPU cycle 2. The reaction time of UP 2 is two times longer than that of UP 1.

The program processing sequence can be controlled by assigning a priority, which indicates how important the corresponding user program is compared to the others (see multitasking mode 2).

To specify the user program processing sequence, use the following parameters in the resources and programs or in the Multitasking Editor:

Parameter	Description	Configurable for
Max. Duration for Each Cycle [µs]	Time permitted for executing the user program within a CPU cycle.	User program, Multitasking Editor
Program ID	ID for identifying the program when displayed in SILworX	User program, Multitasking Editor
Watchdog Time	Resource Watchdog Time	Resource
Target Cycle Time [ms]	Required or maximum cycle time	Resource
Multitasking Mode	Use of the execution duration unneeded by the user program, e. g., the difference between actual execution duration in one CPU cycle and the defined *Max. Duration for Each Cycle [µs]*. Mode 1 The duration of a CPU cycle is based on the required execution time of all user programs. Mode 2 The processor provides user programs with a higher priority the execution time not needed by user programs with a lower priority. Operation mode for high availability. Mode 3 The processor waits for the unneeded execution time of user programs to expire and thus increases the cycle.	Resource, Multitasking Editor
Target Cycle Mode	Use of *Target Cycle Time [ms]*	Resource
Priority	Importance of a user program; highest priority: 0.	Multitasking Editor
Maximum Number of Cycles	Maximum number of CPU cycle required to process one user program cycle.	Multitasking Editor

Table 18: Parameters Configurable for Multitasking

Observe the following rules when setting the parameters:

- If *Max. Duration for Each Cycle [µs]* is set to 0, the execution time of the user program is not limited, e.g., it is always processed completely. Therefore, the number of cycles may be set to 1 in this case.
- The sum of the *Max. Duration for Each Cycle [µs]* parameters in all user programs must not exceed the resource watchdog time. Make sure that sufficient reserve is planned for processing the remaining system tasks.
- The sum of the *Max. Duration for Each Cycle [µs]* parameters in all user programs must be large enough to ensure that sufficient reserve is available to maintain the target cycle time.
- The *Program ID*s of all user programs must be unique.

During verification and code generation, SILworX monitors that these rules are observed. These rules must also be observed when modifying the parameters online.

SILworX uses these parameters to calculate the user program watchdog time:
User program watchdog time = w*atchdog time* * maximum number of cycles

i The sequence control for executing the user programs is run in cycles of 250 µs. For this reason, the values set for *Max. Duration For Each Cycle [µs]* can be exceeded or under-run by up to 250 µs.

Usually, the individual user programs run concurrently in a non-reactive manner. 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 a limit is not configured with *Max Duration for Each Cycle*.

NOTE

An unpredictable behavior of the user program is possible!

The use of the same global variables in several user programs can lead to a variety of consequences caused by the reciprocal influence among 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 in one location, either in a user program or from the hardware!**

i HIMA recommends to set the *Max. Duration for each Cycle [µs]* parameter to an appropriate value ≠ 0. This ensures that a user program with an excessively long runtime is stopped during the current CPU cycle and resumed in the next CPU cycle without affecting the other user programs.

Otherwise, an unusually long runtime for one or several user programs can cause the target cycle time, or even the resource watchdog time, to be exceeded, thus leading to an error stop of the controller.

10.2.11 Acceptance by Test Authority

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

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

10.3 Checklist for Creating a User Program

To comply with all safety-related aspects during the programming phase, HIMA recommends using the following checklist prior to and after loading a new or modified program. The checklist can be used for helping with planning as well as to demonstrate later on that the planning phase was carefully completed.

The checklist is available in Microsoft® Word® format on the HIMA website.

11 Configuring Communication

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

11.1 Standard Protocols

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

⚠ **DANGER**

Physical injury due to usage of unsafe import data

Do not use any data imported from unsafe sources for safety functions in the user program.

The following standard protocols are available:

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

11.2 Safety-Related Protocol (safe**ethernet**)

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

NOTE

Unintentional transition to the safe state possible!

ReceiveTMO is a safety-related parameter!

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

i ReceiveTMO also applies in the other direction from PES 2 to PES 1!

11.2.1 Receive Timeout

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

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

For safety-related functions implemented via safe**ethernet**, only the **Use Initial Data** setting may be used.

Since *ReceiveTMO* is a safety-relevant component of the Worst Case Reaction Time T_R (see Chapter **11.3.1** et seqq.), its value must be determined as described below and entered in the safe**ethernet** Editor.

ReceiveTMO ≥ 4*delay + 5*max. cycle time

Condition: The Communication Time Slice must be sufficiently high to allow all the safeethernet connections to be processed within one CPU cycle.

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

i A wanted fault tolerance of communication can be achieved by increasing *ReceiveTMO,* provided that this is permissible in terms of time for the application process.

11.2.2 Response Time

ResponseTime is the time in milliseconds (ms) that elapses until the sender of the message receives acknowledgement from the recipient.

When configuring using a safe**ethernet** profile, a *Response Time* parameter must be set based on the physical conditions of the transmission path.

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

ResponseTime ≤ ReceiveTMO / n

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

The ratio between *ReceiveTMO* and *ResponseTime* influences the capability to tolerate faults, e.g., when packets are lost (resending lost data packets) or delays occur on the transmission path.

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

min. Response Time ≤ ReceiveTMO / 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** connection.

i If this condition is **not met**, the availability of a safeethernet connection can only be ensured in a collision and fault-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 conditions cannot always be ensured, a corresponding connection system variable for monitoring the response time is available. If the measured response time is not seldom exceeded for over the half P2P ReceiveTMO, the configured response time must be increased.

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

11.3 Worst Case Reaction Time for safeethernet

In the following examples, the formulas for calculating the worst case reaction time only apply for a connection with HIMatrix controllers if the parameter Safety Time = 2 * Watchdog Time is set. These formulas always apply to HIMax controllers.

i The allowed worst case reaction time depends on the process and must be agreed upon together with the test authority responsible for the final inspection.

Terms

ReceiveTMO: Monitoring time of PES 1 within which a correct response from PES 2 must be received. Otherwise, safety-related communication is terminated after the time has expired.

Production Rate: Minimum interval between two data transmissions.

Watchdog Time: Maximum duration permitted for a controller's RUN cycle. The duration of the RUN cycle depends on the complexity of the user program and the number of safe**ethernet** connections. The watchdog time (WDT) must be entered in the resource properties.

Worst Case Reaction Time: The worst case reaction time is the time between a change in a physical input signal (in) of PES 1 and a reaction on the corresponding output (out) of PES 2.

Delay: Delay of a transmission path e.g., with a modem or satellite connection.
For direct connections, an initial delay of 2 ms can be assumed.
The responsible network administrator can measure the actual delay on a transmission path.

To the calculations of the maximum reaction times specified below, the following conditions apply:

- The signals transmitted over safe**ethernet** must be processed in the corresponding controllers within one CPU cycle.
- Further, the reaction time of the sensors and actuators must be added.

The calculations also apply to signals in the opposite direction.

11.3.1 Calculating the Worst Case Reaction Time of Two HIMax Controllers

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

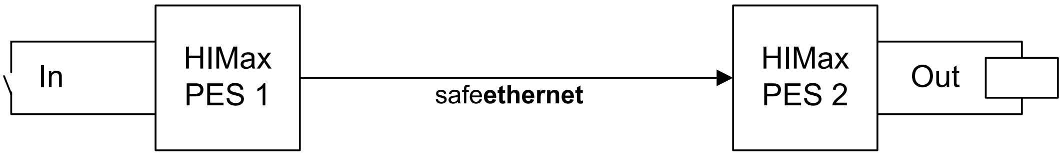


Figure 4: safe**ethernet** Connection of Two HIMax Controllers

Reaction Time when two HIMax controllers are interconnected

$T_R = t_1 + t_2 + t_3$

T_R Worst Case Reaction Time

t_1 Safety Time of PES 1

t_2 *ReceiveTMO*

t_3 Safety Time of PES 2

11.3.2 Calculating the Worst Case Reaction Time in Connection with One HIMatrix PES

Reaction time T_R between a change on the sensor input signal (in) of HIMax PES 1 and a reaction on the corresponding output (out) of HIMatrix PES 2. It is calculated as follows:

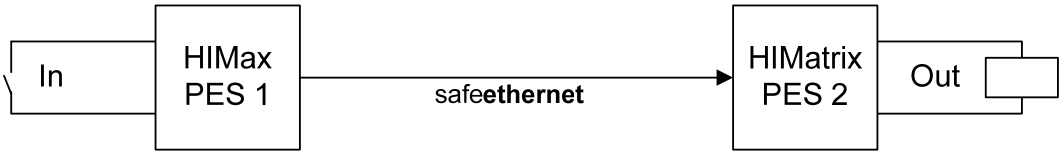


Figure 5: safe**ethernet** Connection between One HIMax and One HIMatrix Controller

Reaction time when one HIMax controller is connected to one HIMatrix controller:

$T_R = t_1 + t_2 + t_3$

T_R Worst Case Reaction Time

t_1 Safety Time of HIMax PES 1

t_2 *ReceiveTMO*

t_3 2 * Watchdog Time of HIMatrix PES 2

11.3.3 Calculating the Worst Case Reaction Time with two HIMatrix Controllers or RIOs

The worst case reaction time T_R is the time between a change on the sensor input signal (in) of the first HIMatrix PES 1 or RIO (e.g., F3 DIO 20/8) and a reaction on the corresponding output (out) of the second HIMatrix PES 2 or RIO. It is calculated as follows

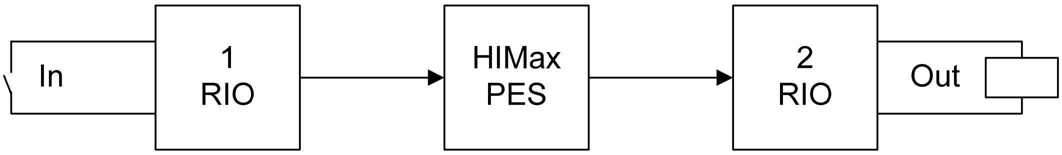


Figure 6: safe**ethernet** Connection in Connection with RIOs

Response Time with RIOs

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

T_R Worst Case Reaction Time

t_1 2 * watchdog time of the 1st RIO

t_2 *ReceiveTMO1*

t_3 2 * Watchdog Time of HIMax PES

t_4 *ReceiveTMO2*

t_5 2 * watchdog time of the 2nd RIO

i The two RIOs can also be identical. The time values still apply if a HIMatrix PES is used instead of a RIO.

11.3.4 Calculating the Worst Case Reaction Time with Two HIMax and One HIMatrix PES

Worst case reaction time T_R between a change on the sensor input signal (in) of the first HIMax PES and a reaction on the corresponding output (out) of the second HIMax PES. It is calculated as follows

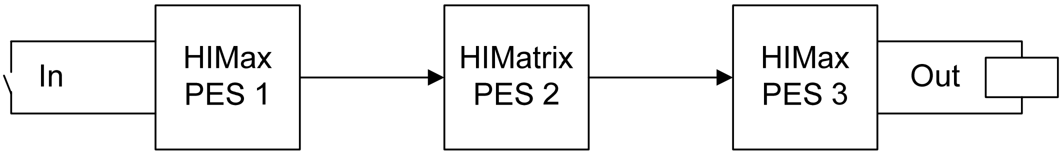


Figure 7: safe**ethernet** Connection between Two HIMax and One HIMatrix PES

Reaction time when two HIMax controllers are connected to one HIMatrix controller:

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

T_R Worst Case Reaction Time

t_1 Safety Time of HIMax PES 1

t_2 *ReceiveTMO1*

t_3 2 * Watchdog Time of HIMatrix PES 2

t_4 *ReceiveTMO2*

t_5 Safety Time of HIMax PES 3

i HIMax PES 1 and HIMax PES 3 can also be identical.
HIMatrix PES 2 can also be a HIMax PES.

11.3.5 safeethernet Profile

safe**ethernet** profiles are combinations of parameters compatible with one another that are automatically set when one of the safe**ethernet** profiles is selected.
When configuring, only the Receive Timeout and the expected Response Time parameters must be individually set.
A safe**ethernet** profile is used to optimize the data throughput within a network taking the physical conditions into account.

To ensure that the optimization is effective the following conditions must be met:

- the Communication Time Slice must be sufficiently high to allow all the safeethernet connections to be processed within one CPU cycle.
- if average CPU cycle time < response time.
- if average CPU cycle time < ProdRate or ProdRate = 0.

NOTE

Disturbance of the safe**ethernet communication up to communication loss!**

Unsuitable combinations of CPU cycle, communication time slice, response time and ProdRate are not rejected during code generation and download/reload, but can cause communication disturbances.

In the Control Panel, verify the *Bad Messages* and *Resends* values for both controllers.

Six safe**ethernet** profiles are available. Select the safe**ethernet** profile the most suitable for the transmission path.

For safety-related process data communication, only the profiles "Fast&Noisy", "Medium&Noisy" and "Slow&Noisy" may be used.

Fast & Cleanroom	Not suitable foor safety-related process data communication!
Fast & Noisy	
Medium & Cleanroom	Not suitable foor safety-related process data communication!
Medium & Noisy	
Slow & Cleanroom	Not suitable foor safety-related process data communication!
Slow & Noisy	

Appendix

Increasing the SIL of Sensors and Actuators

The safety-related HIMax controllers can be used in applications up to SIL 3. This requires that the sensors and actuators (signalers and actuating elements) in use also achieve the required SIL.

In some cases, sensors or actuators may not be available for the requirements defined in the application, such as process value, range of value, SIL, etc. If this is the case, proceed as follows:

- For inputs: Use any of the available sensors that meet all of the requirements with the exception of the SIL value. Use enough of them such that their combination provide an input signal with the required SIL.
- For outputs: Use any of the available actuators that meet all of the requirements with the exception of the SIL. Use enough of them such that their combination affects the process with the required SIL.

With inputs, associate the values of the individual sensors and their status information with a part of the user program such that a global variable with the required SIL results from this combination.

With outputs, distribute the value of a global variable among multiple outputs such that the process adopts the safe state if a fault occurs. Further, the combination of actuators must be able to affect the process in the required manner (for example, the serial or parallel connection of valves).

For both inputs and outputs, design the system to have the required number of sensors and actuators for a given process variable until the greatest possible degree of safety is achieved for the process. Use a calculation tool to determine the SIL.

i The use of multiple sensors and actuators for inputting or outputting a single signal as described here is only intended as a means of increasing the SIL. Do not confuse this with the use of redundant inputs or outputs for improving availability

For information on how to achieve the required SIL for sensors and actuators, see IEC 61511-1, Section 11.4.

Definitions and Abbreviations

Term	Description
ARP	Address Resolution Protocol: Network protocol for assigning the network addresses to hardware addresses
AI	Analog Input
Connector Board	Connector board for the HIMax module
COM	Communication module
CRC	Cyclic Redundancy Check
DI	Digital Input
DO	Digital Output
EMC	Electromagnetic Compatibility
EN	European Norm
ESD	ElectroStatic Discharge
FB	Fieldbus
FBD	Function Block Diagram
FTA	Field Termination Assembly
FTT	Fault Tolerance Time
ICMP	Internet Control Message Protocol: Network protocol for status or error messages
IEC	International Electrotechnical Commission
MAC address	Hardware address of one network connection (Media Access Control)
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
PFD	Probability of Failure on Demand, probability of failure on demand of a safety function
PFH	Probability of Failure per Hour, probability of a dangerous failure per hour
R	Read
Rack ID	Base plate identification (number)
Non-reactive	Supposing that two input circuits are connected to the same source (e.g., a transmitter). An input circuit is termed "non-reactive" if it does not distort the signals of the other input circuit.
R/W	Read/Write
SB	System Bus (Module)
SELV	Safety Extra Low Voltage
SFF	Safe Failure Fraction, portion of safely manageable faults
SIL	Safety Integrity Level (in accordance with IEC 61508)
SILworX	Programming tool for HIMax
SNTP	Simple Network Time Protocol (RFC 1769)
SRS	System.Rack.Slot addressing of a module
SW	Software
TMO	TiMeOut
TMR	Triple Module Redundancy
W	Write
Watchdog (WD)	Time monitoring for modules or programs. If the watchdog time is exceeded, the module or program enters the ERROR STOP state.
WDT	WatchDog Time

Index of Figures

Index of Tables

Index

HI 801 003 E

Albert-Bassermann-Str. 28
68782 Brühl, Germany
Tel. +49 6202 709-0
Fax +49 6202 709-107
HIMax-info@hima.com
www.hima.com

SAFETY
NONSTOP