



HIMax[®]

System Manual

SAFETY
NONSTOP



SYSTEM

All HIMA products mentioned in this manual are protected by the HIMA trade-mark. Unless noted otherwise, this also applies to other manufacturers and their respective products referred to herein.

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>.

© Copyright 2009, HIMA Paul Hildebrandt GmbH + Co KG

All rights reserved

Contact

HIMA Address:

HIMA Paul Hildebrandt GmbH + Co KG

P.O. Box 1261

68777 Brühl

Tel: +49 6202 709-0

Fax: +49 6202 709-107

E-mail: info@hima.com

Table of Contents

1	Introduction	7
1.1	Structure and Use of the Document.....	7
1.2	Target Audience.....	7
1.3	Formatting Conventions	8
1.3.1	Safety Notes	8
1.3.2	Operating Tips	9
2	Safety	10
2.1	Intended Use	10
2.1.1	Scope.....	10
2.1.2	Non-Intended Use.....	10
2.1.3	Operating Requirements.....	11
2.1.4	Requirements to be met by the operator and the machine and system manufacturers	13
2.2	Residual Risk	13
2.3	Safety Precautions.....	14
2.4	Emergency Information.....	14
3	Product Description	15
3.1	Base Plates and Base Plate Types.....	17
3.1.1	Base Plate Structure	18
3.1.2	Ventilation	18
3.1.3	Monitoring the Temperature.....	19
3.1.4	Power Supply	19
3.2	System Bus	20
3.2.1	Extending the System Bus.....	22
3.3	Modules and Connector Boards.....	24
3.3.1	Identifying the Module via SRS.....	25
3.3.2	Permissible Slot Assignments.....	25
3.4	Processor module.....	26
3.4.1	Operating System	27
3.4.2	Behavior in the Event of Faults	29
3.4.3	Noise Blanking	29
3.5	Alarm and Sequence of Events Recording	30
3.5.1	Alarm and Events.....	30
3.5.2	Creating Events	30
3.5.3	Recording Events.....	31
3.5.4	Transfer of Events.....	31
3.6	Communication.....	32
4	Redundancy	33
4.1	Processor Module.....	33
4.1.1	Decreasing Redundancy.....	33
4.1.2	Upgrading Redundancy	33

4.2	I/O Modules.....	33
4.2.1	I/O Module	33
4.2.2	Channel Redundancy	33
4.2.3	Connector Boards for Redundant Module	34
4.3	System Bus	34
4.4	Communication.....	34
4.4.1	safeethernet	34
4.4.2	Standard Protocols	34
4.5	Power Supply	34
5	Programming.....	35
5.1	Connecting the Programming System.....	35
5.2	Using Variables in a Project.....	35
5.2.1	Types of Variables	35
5.2.2	Initial Value	36
5.2.3	System Variables and System Parameters	36
5.2.4	Assignment to I/O Channels	43
5.2.5	Assignment to Communication Connections	46
5.2.6	Configuring the Sequence of Events Recording	46
5.3	Forcing.....	49
5.3.1	Time Limits.....	50
5.3.2	Restricting the Use of Forcing.....	50
5.3.3	Force Editor	50
5.3.4	Forcing and Scalar Events.....	50
5.4	Multitasking	51
5.4.1	Multitasking Mode	54
5.5	Loading User Programs	57
5.5.1	Download	58
5.5.2	Reload.....	58
5.6	Loading Operating Systems	59
5.6.1	Load Process	60
5.6.2	Updating and Downgrading Operating Systems.....	60
6	User Management.....	61
6.1	User Management for SILworX Projects.....	61
6.2	User Management for the Controller.....	61
6.2.1	Default User	62
6.2.2	Parameters for User Accounts	63
6.2.3	Setting Up User Accounts.....	63
7	Diagnosis	64
7.1	Light Emitting Diodes.....	64
7.1.1	Module Status Indicators	65
7.1.2	Redundancy Indicators	65
7.1.3	System Bus Indicators	66
7.1.4	Rack connection indicators	66

7.1.5	Slot Indicators	66
7.1.6	Fault Indicators	67
7.1.7	Maintenance Indicators.....	67
7.1.8	I/O Indicators.....	67
7.1.9	Fieldbus Indicators.....	68
7.1.10	Ethernet Indicators.....	68
7.1.11	Communication Indicators	69
7.2	Diagnostic History	69
7.3	Online Diagnosis.....	70
8	Specifications, Dimensioning	73
9	Life Cycle	74
9.1	Installation	74
9.1.1	Mechanical Structure	74
9.1.2	Connecting the Field Zone to the I/O Module	74
9.1.3	Earthing.....	78
9.1.4	Electrical Connections	84
9.1.5	Mounting a Connector Board	86
9.1.6	Considerations about Heat	87
9.2	Start-Up.....	89
9.2.1	Starting-up the Control Cabinet	89
9.2.2	Starting-up the PES	90
9.2.3	Assigning the Rack ID.....	91
9.3	Maintenance and Repairs.....	92
9.3.2	Disturbances	92
9.3.3	Connecting the redundant Power Supply	93
9.3.4	Repair	93
10	HIMax Documentation and Support.....	94
10.1	HIMax Documentation	94
10.2	HIMA Service, Training and Hotline	96
	Appendix	97
	Application Examples.....	97
	Glossary.....	99
	Index of Figures	100
	Index of Tables.....	101
	Index.....	103

1 Introduction

The System Manual describes the configuration and mode of operation of the safety-related HIMax controller system.

HIMax can be used for various control tasks within the process and factory automation industry.

1.1 Structure and Use of the Document

This System Manual is composed of the following chapters:

Safety	Information on how to safely use the HIMax system
Product Description	Structure of the HIMax systems
Communication	Brief description of the communication between the HIMax and other systems. For more information, refer to the Communication Manual HI 801 101
Redundancy	Options for increasing availability
Programming	Important instructions on how to create a user program
User Management	User management for accessing to the HIMax controllers
Diagnosis	Summary of the diagnostic options
Specifications, Dimensioning	Data related to the entire system. Data concerning the individual components are specified in the corresponding manual
Life Cycle	Phases of one HIMax system life cycle <ul style="list-style-type: none">▪ Installation▪ Start-up▪ Service and maintenance
HIMax Documentation and Support	Overview of the documentation and on how to get access to the support
Appendix	<ul style="list-style-type: none">▪ Configuration examples for the HIMax systems▪ Glossary▪ Index of tables and index of figures▪ Index

1.2 Target Audience

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

All staff members (planning, installation, commissioning) must be informed about the risks and potential consequences resulting from the manipulation of a safety-related automation system.

Planners and configuration engineers must have additional knowledge about the selection and use of electrical and electronic safety systems within automated systems, e.g., to prevent improper connections or faulty programming.

The operator is responsible for qualifying the operating and maintenance personnel and providing them with appropriate safety instructions.

Only staff members with knowledge of industrial process measurement and control, electrical engineering, electronics and the implementation of PES and ESD protective measures may modify or extend the system wiring.

1.3 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.
<i>Italics:</i>	For parameters and system variables
<code>Courier</code>	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.

1.3.1 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.
- Caution 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

1.3.2 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 Safety

The following safety information, notes and instructions must be strictly observed. The product may only be used if all guidelines and safety instructions are adhered to.

This product is operated in accordance with SELV or PELV. No imminent danger results from the module itself. The use in Ex-Zone is permitted if additional measures are taken.

2.1 Intended Use

This chapter describes the conditions for using HIMax systems.

2.1.1 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.

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.

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

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!

2.1.2 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.

2.1.3 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

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

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

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

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

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.

2.1.4 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.

2.2 Residual Risk

No imminent danger results from a HIMax system itself.

Residual risk may result from:

- Faults in the engineering
- Faults in the user program
- Faults in the wiring

2.3 Safety Precautions

Observe all local safety requirements and use the protective equipment required on site. Safety shoes are required while mounting the X-BASE PLATE.

2.4 Emergency Information

A HIMax controller is a part of the safety equipment of a system. If the controller fails, the system adopts the safe state.

In case of emergency, it is not permitted to access the safety equipment.

3 Product Description

HIMax is a safety-related control system and is intended for continuous operation and maximum availability.

HIMax is a modular system. Functions such as processing, input and output or communication are distributed on plug-in modules that can be inserted in one or several base plates. A controller specific to the application can be created by selecting appropriate modules.

Ethernet cables are used to interconnect the base plates.

The controller can be easily adapted to future extensions of the process to be controlled, e.g., by adding modules or base plates containing modules.

Figure 1 shows the structure of the HIMax system. The figure shows the base plates, both system busses, the system bus modules, the processor modules and the Connector Boards of the modules.

To increase availability, HIMax is intended for redundant operation. For more information, refer to Chapter 4.

The system can also be used as mono, non-redundant system. For more information, refer to Chapter 3.3.2, Variant 1, and Appendix.

In either case, safety-related operation up to SIL 3 is ensured.

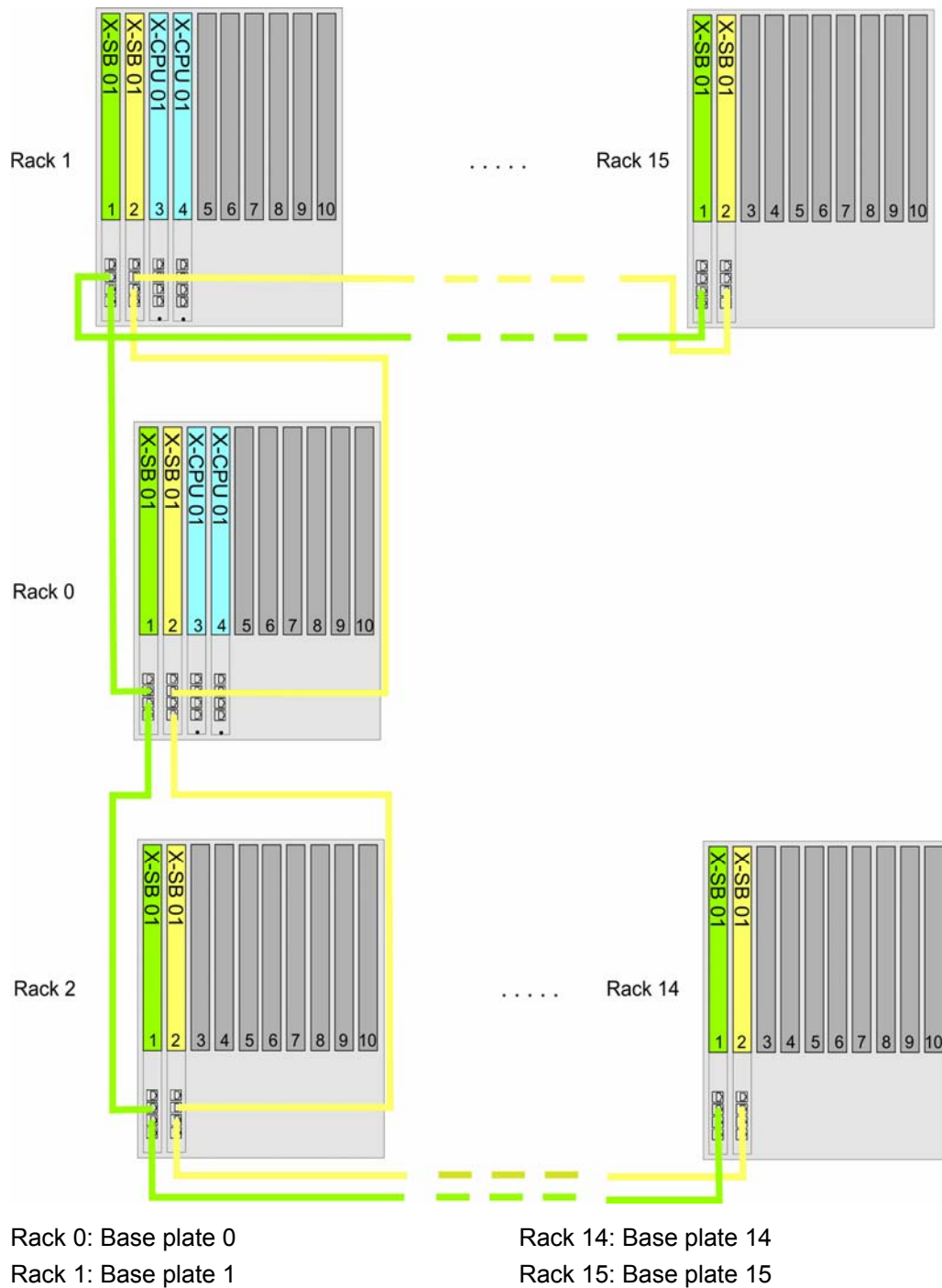


Figure 1: System Overview

A HIMax system is composed of at least one base plate, i.e., base plate 0. It has rack ID (base plate number) 0 and contains at least one processor module. All additional base plates are extension base plates. Among these, base plate 1 may contain one or two processor modules. The remaining base plates must contain no processor modules.

Base plate 0 can be extended with up to 15 extension base plates. Cables are used to interconnect the two system busses A and B on all of the base plates.

3.1 Base Plates and Base Plate Types

HIMax base plates differ in the number of slots.

Each base plate composing a HIMax controller can have 10, 15 or 18 slots.

Base plate types:

- With 10 slots: X-BASE PLATE 10 01
for mounting on a flat base, e.g., a mounting plate.
- With 15 slots: X-BASE PLATE 15 01
for mounting on a backplane
- With 15 slots: X-BASE PLATE 15 02
for 19" mounting
- With 18 slots: X-BASE PLATE 18 01
for mounting on a backplane

A total of one module and one Connector Board can be plugged in to each slot.

System cables are used to interconnect the base plates.

3.1.1 Base Plate Structure

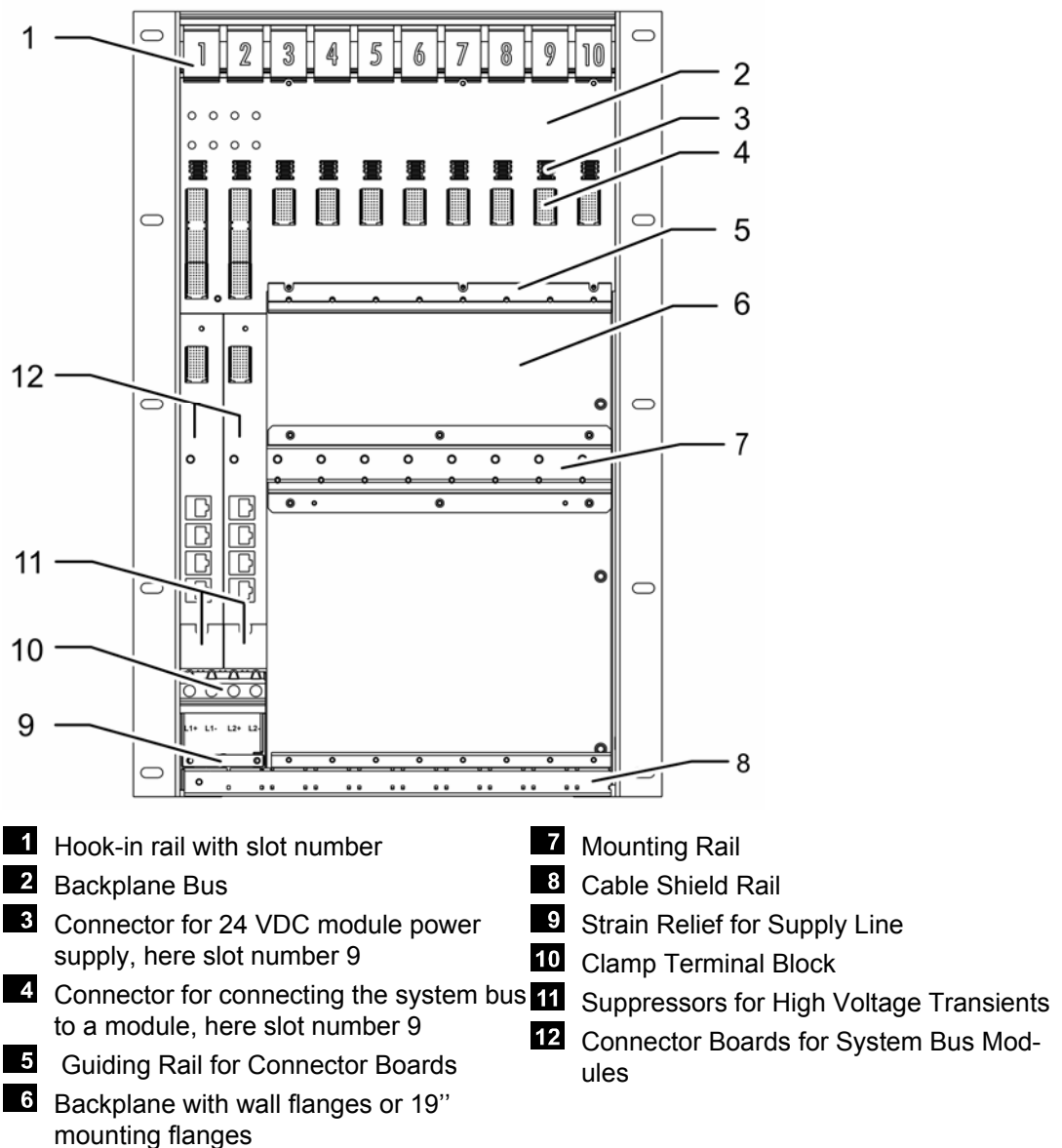


Figure 2: Base Plate Structure

Both left slots, slot 1 and slot 2, are reserved for system bus modules. The remaining slots can be used for other modules, but observe the restrictions for positioning processor modules, see Chapter 3.3.2.

Each module has a connector board to which external devices such as sensors, actuators and other controllers are connected. Both connector boards for the system bus modules are included within the scope of delivery.

The clamp terminal blocks of the base plate are used to connect the power supply. Two redundant 24 VDC power supply units can be connected.

3.1.2 Ventilation

A suitable fan rack located above the base plate ensures the ventilation.

The air flows from the fan rack downwards through the modules and through the connection space located in front of the connector boards. To ensure proper ventilation, insert blank modules in all the unused slots of the base plate!

NOTE

Controller damage due to overheating!
Overheating can destroy electronic components!
Only operate HIMax systems if ventilation is ensured!

A fan rack with suitable dimensions is available for each base plate type. Depending on the width, the fan racks are equipped with 2, 3 or 4 fans. For more information, refer to the X-FAN Manual (HI 801 033).

Additional dissipation of the generated hot air must be ensured, see Chapter 9.1.6.

3.1.3 Monitoring the Temperature

The modules monitor their own temperature. Use the SILworX programming tool to display the temperature level and evaluate it for programming reactions.

3.1.4 Power Supply

The HIMax system requires a power supply of 24 VDC.

The safe electrical isolation of the power supply must be ensured within the 24 V system supply. Use power supply units of type PELV or SELV only. When used in accordance with UL regulations, an adjustable power supply unit with a maximum voltage of 150 V and a maximum power of 10 kVA is allowed.

The power supply used must have a momentary interruption immunity value of < 1 ms. HIMA power supply units are appropriately equipped. Before using power supply units from other manufacturers, ensure their adequate testing.

Two redundant power supply units can be connected.

NOTE

Controller damage due to overvoltage!
Set the power supply unit so that the supply voltage cannot exceed 30 V!

NOTE

Controller damage due to overcurrent!
Protect each base plate pre-fusing it against voltage higher than 63 A!

The modules monitor both operating voltages. Use the SILworX programming tool to display the voltage level and evaluate it for programming reactions.

Estimating the Required Power

Use a rule of thumb to estimate the power required for the power supply.

$$P_{\text{Total}} = n_{\text{CPU}} * 35 + n_{\text{Modules}} * 20 + n_{\text{Fans}} * 20 + P_{\text{external}}$$

P_{Total} :	Total required power
n_{CPU} :	Number of processor modules in use
n_{Module} :	Number of modules used without processor modules
n_{Fans} :	Number of fans in use. Each fan rack contains 2, 3 or 4 fans.
P_{External} :	Power delivered from the output modules to the connected actuators.

The following reference values are used in this formula:

- Power consumption of a HIMax processor module: approx. 35 W
- Power consumption of another HIMax module (except for processor module): approx. 20 W
- Power consumption of a fan: approx. 20 W
- Power consumption of the actuators connected to and supplied by the output modules

The power in watts required for one HIMax system is the result of this rough calculation.

For an exact calculation of the power required, use the power consumption values of the individual modules and other consumer loads as specified in the corresponding data sheets or manuals.

3.2 System Bus

The HIMax system operates with two redundant system busses, system bus A and system bus B.

The system busses run within a base plate. The module is connected with the system busses by inserting it into the base plate. System busses A and B interconnect the modules via the system bus modules. The failure of one module does not affect the connections to remaining modules.

The system bus connection to the modules are electrically isolated from the base plate. An insulation voltage of at least 1 500 V is ensured between the processor module and each I/O module.

A system bus module is required to manage a system bus. The system bus module in slot 1 operates system bus A and the system bus module in slot 2 operates system bus B.

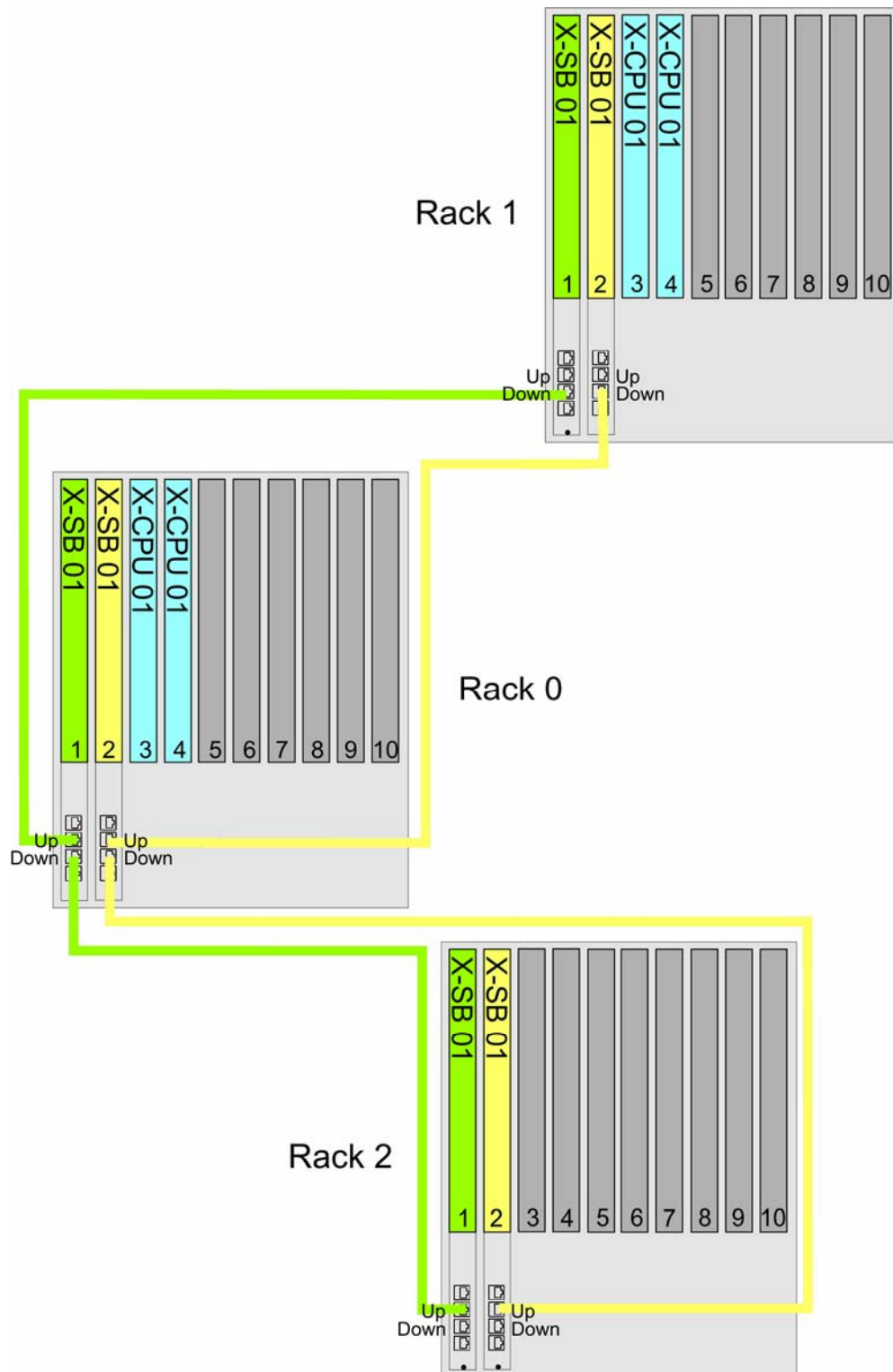
i

If only one system bus module is inserted in the base plate, only one system bus is available!

If both system bus modules are used to operate the HIMax system, communication runs on both system busses simultaneously.

If the HIMax system is composed of various base plates, use Ethernet patch cables to interconnect the system busses on the base plates. Insert the cables in the RJ-45 socket located on the connector boards of the system bus modules. To do this, connect the UP socket of one base plate to the DOWN socket of the next base plate. System bus A and system bus B must not be crossed.

Two adjacent base plates can be connected to one base plate.



Rack 0: Base plate 0

Rack 2: Base plate 2

Rack 1: Base plate 1

Figure 3: Arrangement of Base Plates on the System Bus

A base plate sequence results from interconnecting the base plates – see Figure 3.

- Start with the base plate having rack ID 0.
- The extension base plate connected to the UP socket of base plate 0 has rack ID 1.

- All additional base plates connected to base plate 0 through base plate 1 have un-even rack IDs up to 15.
- The extension base plate connected to the DOWN socket of base plate 0 has rack ID 2.
 - All additional base plates connected to base plate 0 through base plate 2 have even rack IDs up to 14.

NOTE**System malfunction possible!**

The system busses are not normal Ethernet connections. Therefore, the RJ-45 sockets UP and DOWN may only be used to connect to HIMax base plates.

Do not connect the sockets UP and DOWN to local networks or other devices with LAN connection such as the PADT!

Never interconnect system bus A and system bus B!

System bus cable characteristics

- Twisted pair Ethernet patch cables
- Cat. 5 for 100 Mbit/s and lower, Cat. 6 for 1 Gbit/s.
- RJ-45 connector on both sides.
- Auto-Crossover allows the use of crossover and straight through cables.
Exception: On the system bus module connector *PADT*, only crossover cables may be used.

3.2.1 Extending the System Bus

The system bus is based on Ethernet technology. For this reason, the system bus can be extended with Ethernet components. The HIMax system can thus stretch across an extensive production line.

Fiber optic cables are suitable for extending Ethernet over wider distances. The cable length limits the signal delay in the fiber optic cable and in the converters between Ethernet cable and fiber optic cable.

HIMax allows the following maximum delay times between modules:

- Among redundant processor modules: max. 10 μ s.
- Between a processor module and the furthestmost I/O module: max. 50 μ s.

The use of a fiber optic cable causes the following delays:

- Total delay due to converters "copper cable - fibre optic cable - copper cable": 1 μ s.
- Delay within the fiber optic cable: 1 μ s/200 m.

The delay due to the short copper cables between system bus modules and converters corresponds to the delay of the fiber optic cable. The length of these copper cables is included in the total length.

Two cases of maximum extensions exist:

- Maximum expansion of the HIMax system.
- Maximum distance between base plates with processor modules.

Maximum Expansion of the HIMax System

In this case, all processor modules are located close to one another, i.e., either in Base Plate 0 or distributed among base plate 0 and base plate 1, which are connected via one (short) copper cable. The two furthestmost base plates with I/O modules can then be located up to 9.8 km from the processor modules.

The HIMax system can have a maximum extension of up to 19.6 km (Figure 4).

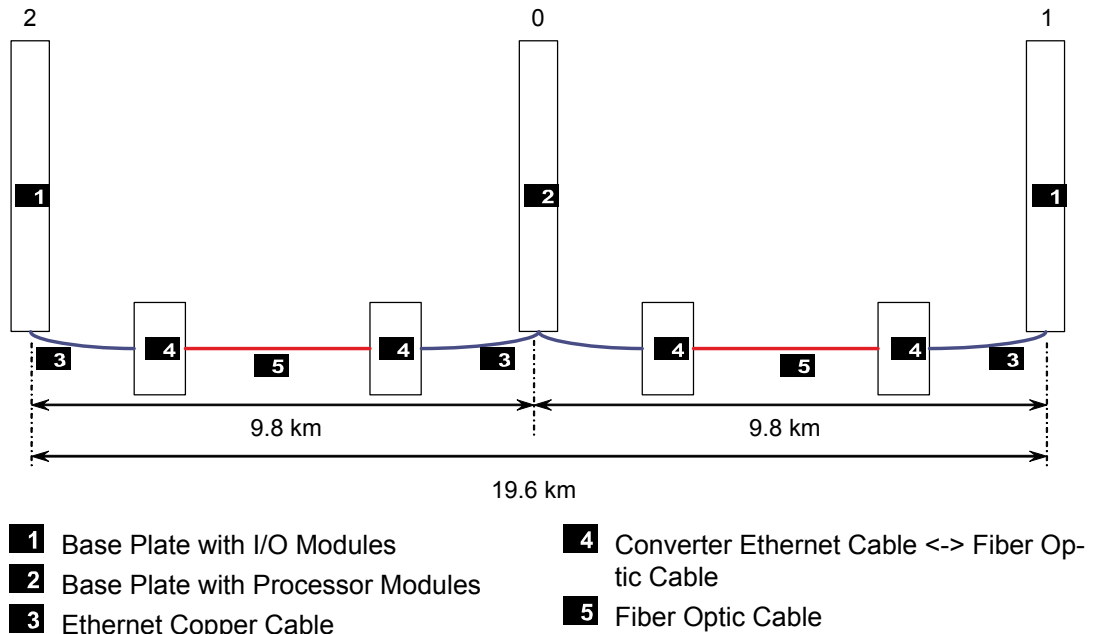


Figure 4: HIMax System in its Maximum Extension

The delay time between processor modules and, for instance, the left base plate with I/O modules is composed of the delay due to the converter (1 μ s) and the delay due to the fiber optic cable length (max. 50 μ s - 1 μ s). The following formula applies to the delay due to the fiber optic cable and its length:

$$49 \mu s \geq \text{length} * 1 \mu s / 200 \text{ m, i.e., length} \leq 9\,800 \text{ m}$$

The same formula applies to the length between processor modules and the right base plate with I/O modules, the maximum length of the fiber optic cable is also 9 800 m.

Maximum Distance between Processor Modules.

If the processor modules are distributed among base plate 0 and base plate 1, these base plates can be positioned far from one another and interconnected using fiber optic cables (Figure 5).

The two base plates with processor modules may be located up to 1.8 km far from one another.

In this case, the HIMax system can have a maximum extension of 17.4 km.

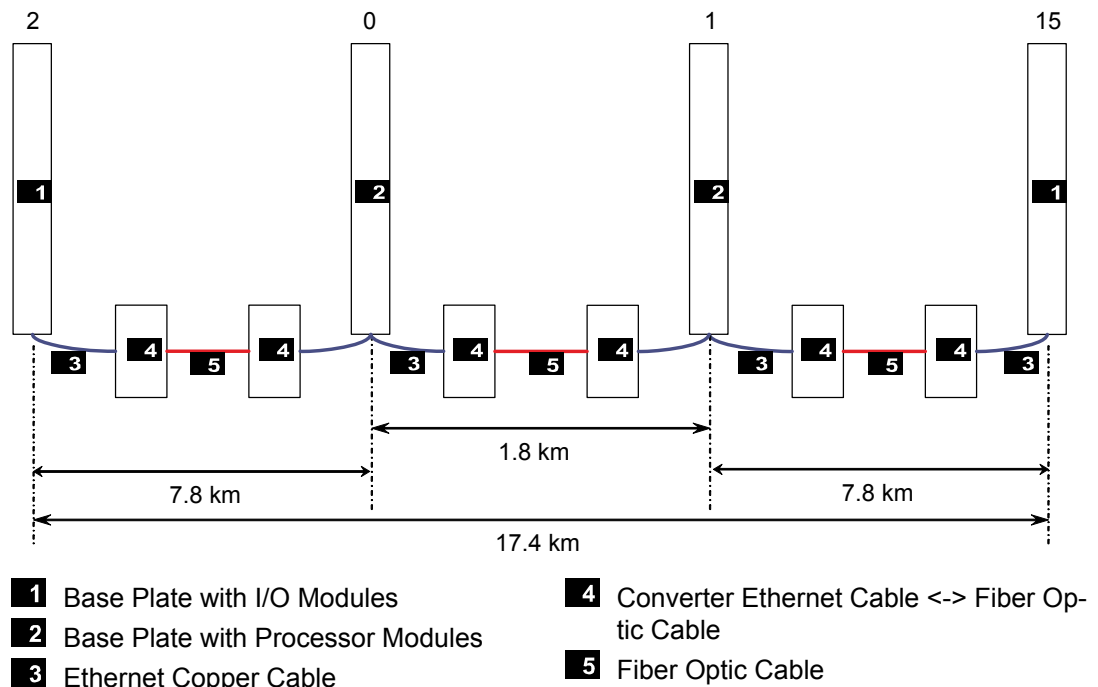


Figure 5: HIMax System with Maximum Distance between Processor Modules

- The delay time between base plate 0 and base plate 1 is composed of the delay due to both converters ($1 \mu\text{s}$) and the delay due to the fiber optic cable (max. $10 \mu\text{s} - 1 \mu\text{s}$). The following formula applies to the delay due to the fiber optic cable and its length:
 $9 \mu\text{s} \geq \text{length} * 1 \mu\text{s} / 200 \text{ m}$, i.e., $\text{length} \leq 1\,800 \text{ m}$
- The delay time between left base plate with I/O modules (in the example rack ID 2) and the right base plate with processor modules (rack ID 1) is composed of:
 - The delay on the distance between the base plate 0 and base plate 1 (see above).
 - The delay on the distance between base plate 0 and base plate 2, on the left. The maximum delay must be $50 \mu\text{s} - 10 \mu\text{s} = 40 \mu\text{s}$. It is composed of the delay due to both converters ($1 \mu\text{s}$) and the delay due to the fiber optic cable (max. $39 \mu\text{s}$). The following formula applies to the delay due to the fiber optic cable and its length:
 $39 \mu\text{s} \geq \text{length} * 1 \mu\text{s} / 200 \text{ m}$, i.e., $\text{length} \leq 7\,800 \text{ m}$

The same formula applies to the length of the fiber optic cable between base plate 1 and base plate 15, the maximum length of the fiber optic cable is also $7\,800 \text{ m}$.

3.3 Modules and Connector Boards

The following module types are available:

- Processor modules
for processing the user programs.
- System bus modules
for managing the system busses.
- Input modules
for measuring and preprocessing the process values.
- Output modules
for converting the results of the user program into control commands for actuators.
- Communication modules
 - for communicating with external devices or systems operating with standard data transfer protocols (e.g., Modbus, PROFIBUS).
 - Physical interfaces for safe**ethernet** for connecting to additional HIMA controllers.

A protective coating protects the electronic module components against corrosion and dust. Each module forms a functional unit with the connector board. A connector board establishes the connection between module and field zone or ensures communication to other controllers or devices. When replacing a module, the connector board remains in the base plate. In this way, the cables or wires connected to the connector board need not be released and reconnected.

Each module type is related to one or several types of connector boards.

The connector between I/O modules and the corresponding connector boards are mechanically coded. This ensures that a module of a certain type can only be plugged in to the corresponding connector board and prevents them from being equipped with improper modules. Coding is performed with wedges on the female connector located on the connector boards, see also the manual for I/O modules.

Two types of connector boards usually exist for I/O modules:

- Connector boards for directly connecting to the supply lines of the field devices.
- Connector boards for directly connecting field termination assemblies (FTAs)

FTAs are used to connect field devices. They are separated from the controller, e.g., in their own cabinet.

NOTE



Danger of short-circuit due to insulation damage!

According to UL regulations, only lines suitable for temperatures of at least 75 °C may be laid next to connector boards and field termination assemblies for X-DO12 01 modules!

For more information on connector boards and field termination assemblies, refer to the module manuals.

3.3.1 Identifying the Module via SRS

The HIMax system uses the parameters **S**ystem, **R**ack, **S**lot (SRS) to identify the modules.

Designation	Range of values	Description
System	1...65 535	Resource identification
Rack	0...15	Base plate identification
Slot	1...18	Slot identification

Table 9: Identifying a Module using the System.Rack.Slot



Assign a unique SRS to every device, e.g., remote I/O that can be reached in a network.

3.3.2 Permissible Slot Assignments

The slot assignment is defined as follows:

1. Slot 1 and slot 2 on each base plate are reserved for system bus modules. Do not insert any other module into these slots!
2. Only slots complying with the rules specified in the next section are permitted for processor modules.
3. Once the slots for the processor modules are determined, I/O and communication modules may be inserted into all remaining slots.

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 bus- ses
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.

3.4 Processor module

The CPU operating system controls the user programs running in a processor module.

3.4.1 Operating System

Tasks:

- Controlling the cyclic run of the user programs
- Performing the self-tests of the module
- Controlling safety-related communication via **safeethernet**
- Managing redundancy together with other processor modules

General CPU Cycle Sequence

Phases:

1. Reading of input data
2. Processing the user program,
3. Writing of output data
4. Other activities, e.g., reload processing

Operating System States

States that can be recognized by the user:

- LOCKED
- STOP/VALID CONFIGURATION
- STOP/INVALID CONFIGURATION
- STOP/OS_DOWNLOAD
- RUN
- RUN/UP STOP

Use the LEDs on the module to recognize the operating state. Here, several LEDs should be considered together. For more information, see Chapter 7.1. The operating states are also displayed in the SILworX online view.

Table 11 gives an overview of the operating system states and conditions under which they are adopted.

State	Description	The state is adopted
LOCKED	Emergency state: The processor module is reset to the factory settings (SRS, network settings, etc.).	Connecting the operating voltage to the processor module while the mode switch is set to Init.
STOP/VALID CONFIGURATION	Processor module stopped: A valid configuration is available in the memory.	Stopping the processor module using SILworX
		Connecting the operating voltage <ul style="list-style-type: none"> Autostart is disabled in the project configuration or Mode switch is set to Stop and the processor module starts by itself.
		From the LOCKED state: Setting the mode switch to Stop if one single processor module exists.
		A fault occurred
STOP/INVALID CONFIGURATION	Processor module stopped: No valid configuration is available in the memory.	Loading with error
		From the LOCKED state: Setting the mode switch to Stop if one single processor module exists.
STOP/OS_DOWNLOAD	Processor module stopped: The operating system is loaded in the non-volatile memory.	Loading the operating system using SILworx
RUN	The user program is running.	Using a SILworX command to enter the STOP/VALID CONFIGURATION state
		Connecting the operating voltage under the following conditions: <ul style="list-style-type: none"> A valid project configuration is loaded Autostart is enabled in the project configuration The mode switch is not set to Init The mode switch is set to Run if the processor module starts by itself
		From the LOCKED state: Setting the mode switch from Init to Stop or Run if an additional processor module is operating in the RUN.
RUN/UP STOP	The user program is not running. This state is used for testing the inputs/outputs and communication.	From the STOP/VALID CONFIGURATION state using a SILworX command

Table 11: Operating System States, Adopting the States

Table 12 specifies how the user may intervene during the corresponding states.

State	Possible user interventions
LOCKED	<ul style="list-style-type: none"> Changing the factory settings Setting the mode switch to Stop¹⁾ to enter the STOP state Setting the mode switch to Run to enter the RUN state Using a PADT command to enter the STOP state Using a PADT command to enter the RUN state
STOP/VALID CONFIGURATION	<ul style="list-style-type: none"> Loading the user program Starting the user program Loading the operating system Taking preliminary actions for forcing variables
STOP/INVALID CONFIGURATION	<ul style="list-style-type: none"> Loading the user program Loading the operating system
STOP/OS_DOWNLOAD	None. Once the loading process is completed, the processor module enters the STOP state.
RUN	<ul style="list-style-type: none"> Stopping the user program Forcing variables Online Test
RUN/UP STOP	<ul style="list-style-type: none"> Using a PADT command to enter the STOP state

¹⁾ STOP/VALID CONFIGURATION or STOP/INVALID CONFIGURATION, depending on whether the processor module has a valid configuration

Table 12: Operating System States, Possible User Interventions

3.4.2 Behavior in the Event of Faults

If faults occur, the processor module enters the error stop state and tries to restart. It performs a complete self-test which can also cause another error stop.

If the fault is still present, the module does not restart. Use the PADT to remove the cause of the fault, e.g., by loading a new application.

Once the processor module has normally run for approximately one minute, an error stop is considered again as *first error stop*, and the module tries to restart.

3.4.3 Noise Blanking

Noise blanking is used to improve the system availability by suppressing temporary faults occurred within the safety time. The safety time is configured in the system

The processor module performs noise blanking for the I/O modules and blanks interference on the channels and transmission paths between processor module and I/O module. In the same way, the output module blanks noises on the transmission path or processor modules.

Noise blanking for each I/O module channels can be activated or deactivated.

If the processor module or the output module detected an error on this signal path, the fault reaction is suppressed (■ 3), if another valid value still exists, which is not older than the fault tolerance time. The fault reaction to an interference is suppressed until the last user program cycle occurring within the fault tolerance time (■ 4). If the fault re-occurs during this cycle, fault handling is activated. Otherwise the the valid input value is processed by the user program

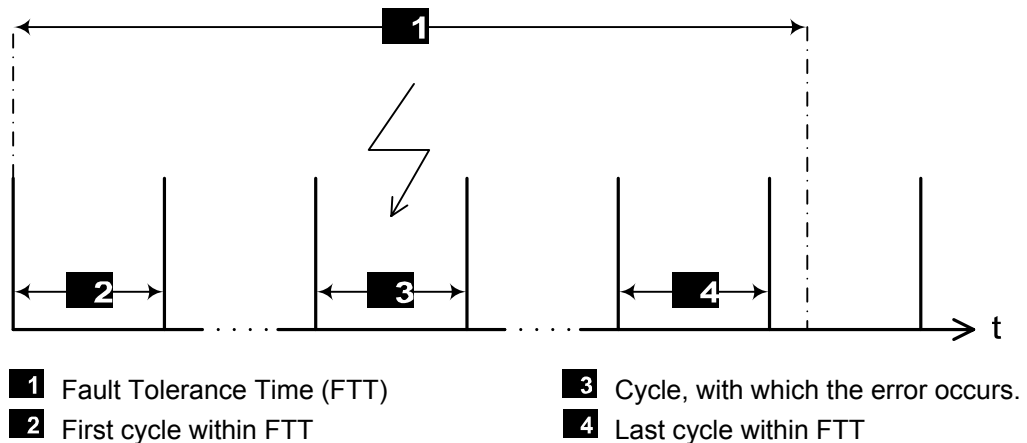


Figure 6: Noise Blanking Performed by Safety-Related Inputs

3.5 Alarm and Sequence of Events Recording

The HIMax system is able to record alarms and sequence of events (SOE) .

3.5.1 Alarm and Events

Events are state changes of the plant or controller and are provided with a timestamp.

Alarms are events that signalize an increasing risk potential.

The HIMax system records the state changes as events specifying the time point when they occurred. The X-OPC server transfers the events to other systems such as control systems, that display or evaluate the events.

HIMax differentiates between Boolean and scalar events.

Boolean Events:

- Changes of Boolean variables, e.g., of digital inputs.
- Alarm and normal state: They can be arbitrarily assigned to the variable states.

Scalar Events:

- Exceedance of the limit values defined for a scalar variable.
- Scalar variables have a numeric data type e.g., INT, REAL.
- Two upper limits and two lower limits are possible.
- For the limit values, the following condition must be met:
Highest limit \geq upper limit \geq normal area \geq lower limit \geq lowest limit.
- An hysteresis can be effective in the following cases:
 - If the value falls below the upper limit.
 - If the value exceeds the lower limit.

An hysteresis is defined to avoid a needless large number of events when a global variable strongly oscillate around a limit value.

HIMax can only create events if they are configured in SILworX, see Chapter 5.2.6.

3.5.2 Creating Events

Both the processor module and certain types of I/O modules are able to create events. In the following sections, these I/O modules are referred to as SOE modules.

Creating Events on the Processor Module

The processor module uses global variables to create the events and stores them in the buffer, see Chapter 3.5.3. The events are created in the user program cycle.

Creating Events on SOE Modules

SOE modules can create events using the input states. The events are created in the SOE module cycle.

The SOE module stores the events in the intermediate buffer that the processor modules use to read them. The intermediate buffer is part of the volatile memory so that the events are lost if the power is switched off.

Every event that has been read can be overwritten by a new event.

System Events

In addition to events, which records changes of global variables or input signals, processor and SOE modules create the following types of system events:

- Overflow: Some events were not stored due to buffer overflow. The timestamp of the overflow event corresponds to that of the event causing the overflow.
- Init: The event buffer was initialized.
- Operating mode 'Stop': A SOE module changed its operating mode to STOP.
- Operating mode 'Run': A SOE module changed its operating mode to RUN.
- Establishing communication: Communication between processor module and SOE module has started.
- Losing communication: Communication between processor module and SOE module was terminated.

System events contain the S.R.S identifier of the module causing the events.

Status Variables

Status variables provide the state of scalar events to the user program. A global variable of type BOOL can be assigned to each of the following status variables:

- Normal.
- Lower limit exceeded.
- Lowest limit exceeded.
- High limit exceeded.
- Highest limit exceeded.

The assigned status variable becomes TRUE when the corresponding state is achieved.

3.5.3 Recording Events

The processor module collects the events:

- Created by I/O modules
- Created by the processor module itself

The processor module stores all the events in its buffer. The buffer is part of the non-volatile memory and has a capacity of 5 000 events.

The processor module arranges the events from different sources by the time of their arrival and does not sort them by their timestamp.

If the event buffer is full, no new events can be stored as long as no further events are read and thus marked as to be overwritten.

Refer to Chapter 5.3.4 for more details on forcing and scalar events..

3.5.4 Transfer of Events

The X-OPC Server readout events from buffer and transfers this to a third-party system for evaluation and indication. Four X-OPC Servers can readout events simultaneously from a processor module.

3.6 Communication

Communication with other HIMA systems or third-party systems occurs via communication modules. The supported communication protocols are:

- **safeethernet** (safety-related)
- Standard protocols

safeethernet connections are also possible via the ethernet interfaces of the processor module.

Refer to the Communication Manual HI 801 101 for more details about communication.

ComUserTask (CUT)

Programs cyclically running on the communication module can be written in C programming language. This allows the users to implement their own communication protocols. This programs are not safety-related.

Licensing

Standard protocols and ComUserTask can only be run on the long term with a valid license. For some protocols, a software activation code is required.

To activate a protocol with an activation code

1. Create an activation code on the HIMA website at:
www.hima.com/Products/HIMax/SILworX_registration.php. To do this, the license number of the delivery note is required.
2. In SILworX, create a license management for the resource if not existing.
3. Create a license key in the license management and enter the activation code.
4. Compile the project and load it into the controller.

The protocol is activated.

4 Redundancy

The conceptual design of the HIMax system is characterized by high availability. To this end, almost all system components can be operated redundantly.

This following chapter describe redundancy aspects of the various system components.

i

Redundancy is not used to increase the Safety Integrity Level (SIL), but to increase availability!

4.1 Processor Module

A HIMax system can be configured as mono system with only one processor module or as highly available system with up to four redundant processor modules.

A system with redundant processor modules always requires a redundant system bus.

Processor modules can only be redundant if a user program with corresponding settings has been loaded.

4.1.1 Decreasing Redundancy

A HIMax system with double to fourfold redundancy of processor modules continues its safety-related operation even if one of the processor modules is no longer available, e.g., because a module failed or was removed. Safety-related operation is also ensured if several processor modules fail.

4.1.2 Upgrading Redundancy

If a new processor module is added to a running HIMax system, it automatically synchronizes itself with the configuration of the existing processor modules. Safety-related operation is ensured. Requirements:

- The user program run by the processor module is redundantly configured.
- One slot among 4, 5, 6 on Base Plate 0 or among 3, 4 on Base Plate 1 is still available.
- Both system busses are functional.
- The mode switch of the processor module that was added is not set to Init.

4.2 I/O Modules

The redundancy of input and output modules includes:

- I/O Module
- Channel redundancy

Define the module redundancy before the channel redundancy.

Twofold or threefold redundancy can be implemented.

4.2.1 I/O Module

Module redundancy: Two or three I/O modules of the same type are defined in the programming system as redundant to one another.

Spare Modules

In SILworX, module that are redundant to one another can have the attribute *Spare Module*. This avoids that an error message is issued if a module fails or is missing.

4.2.2 Channel Redundancy

Requirement: Two modules were configured redundant to one another.

Channels with the same number can be defined as redundant to one another.

In such a case, the programming system automatically assigns a global variable associated with a channel (channel number) to both channels of the redundant modules. For more information on the Hardware Editor in SILworX, refer to the online help.

With input channels, the user can specify how the controller should compose the signals of both redundant channels to a resulting value. The global variable adopts this value.

Not all channels of two redundant modules need to be redundant.

4.2.3 Connector Boards for Redundant Module

In several application cases, all channels of two redundant modules are redundant, but the connected transmitters or actuators are not.

Additional wiring effort can be saved by proceeding as follows:

- Use a connector board that is intended for this purpose and occupies two slots.
- Plug the two redundant modules in to adjacent slots.
- Field connections must be created on the connector board once only.

Specific connector boards can be also provided for the threefold redundant implementation of certain I/O modules.

For more information on these Connector Boards, refer to the module manuals.

4.3 System Bus

The HIMax system has two redundant system buses, system bus A and system bus B.

Requirements for redundant operation:

- Use of two system bus modules per base plate.
- Suitable configuration of the system bus modules.
- Connection of the base plates in a controller, see Chapter 3.2.

HIMA recommends using system bus A and system Bus B redundantly even if a non-redundant operation would be possible, see variant 1 in Chapter 3.3.2.

4.4 Communication

For more information, refer to the online help of SILworX and the Communication Manual HI 801 101.

4.4.1 safeethernet

Redundancy is configured in the SILworX **safeethernet** Editor. A communication connection is redundant if two identical physical transmission paths exist.

4.4.2 Standard Protocols

- Modbus
- PROFIBUS

If standard protocols are used, the user program must manage redundancy.

4.5 Power Supply

The HIMax system can be operated with redundant power supply units. The power supply units are connected to the terminal block, with terminals L1+/L1- used for the first power supply unit and L2+/L2- used for the redundant power supply unit. Each module supports internal decoupling of the operating voltage from the two terminals.

5 Programming

The user programs for the HIMax system are created using the programming system (PADT) which is composed of one PC with the programming tool SILworX. A user program is composed of standard function blocks in accordance with IEC 61131-3, of user-defined function blocks and of variables and connectors. The elements are placed in the SILworX program editor and graphically interconnected. From the resulting graphical representation, SILworX generates an executable program that can be loaded into the controller.

For more information on the programming tool, refer to the SILworX online help.

Until 32 user programs can be loaded into the controller. The controller processes the user programs simultaneously. The user programs can be processed with tunable priorities.

5.1 Connecting the Programming System

An Ethernet connection is used to connect the programming system to the HIMax system. The following interfaces are available:

- Ethernet interfaces on the processor module.
- Ethernet interfaces on the communication module.
- Ethernet interfaces "PADT" on the system bus module.

5.2 Using Variables in a Project

A variable is a placeholder for a value within the program logic. The variable name is used to symbolically address the storage space containing the stored value.

Two essential advantages result from using symbolic names instead of physical addresses:

- The names of inputs and outputs used in the process can be used in the user program.
- The modification of how the variables are assigned to the input and output channels does not affect the user program.

There are local and global variables. Local variables are valid in a delimited project area, in a user program or function block. Global variables can be used in several function blocks or programs and can exchange data between the function blocks.

Global variables can be created at different project tree levels. Global variables are valid for all sub-branches.

Example: If a project contains several resources, the global variables created under a resource are only valid for the branching under this resource.

Hierarchy of the levels within which global variables can be defined:

1. Project
2. Configuration
3. Resource

5.2.1 Types of Variables

Depending on the program organization unit (POU) – user program, function block or function –, different types of variables can be defined. The following table provides an overview:

Type of variable	User program	Function block	Function	Use
VAR	• (CONST, RETAIN)	• (CONST, RETAIN)	• (CONST)	Local variable
VAR_INPUT	•	•	•	Input Variable
VAR_OUTPUT	•	•	•	Output variable
VAR_EXTERNAL	• (CONST)	• (CONST)	•	External to / from other POU or higher global levels
VAR_GLOBAL	•			Global on a higher level (project, configuration, resource)
VAR_TEMP	•	•	•	Temporary variable

• Variable type supported for this program organization unit

CONST: constant that the user program cannot write (e.g., switch point)

RETAIN: With a warm start, a buffered value is taken, with a cold start, the initial value

Table 13: Types of Variables

5.2.2 Initial Value

An initial value can be assigned to any variable. The variable adopts this value if no other value was assigned by the program:

- While starting the program
- If a fault occurs, e.g., in a physical input or in a communication interface from which the variable assumes its value



Assign a safe value as initial value to all variables that receive their value from a physical input or from communication!

Variables to which no initial value was assigned, have initial value 0.

5.2.3 System Variables and System Parameters

System variables are pre-defined variables for processing properties or states of the HIMax system in the user program. To do so, the global variables used in the user program are assigned to system variables.

System parameters are used to configure the properties of the controller (possible with SILworX only). System parameters that can only have the values TRUE and FALSE are also called *switches*.

System variables and system parameters are defined at different project levels. The system variables and system parameters are configured in SILworX, either in the Properties dialog box of the corresponding structure tree branching or in the Detail View of the Hardware Editor.

Project level	Description of the system variables and system parameters
Resource	See Table 15
Hardware, in general	<ul style="list-style-type: none"> ▪ System variables for configuring the controller, see Table 16. ▪ System variables providing information, see Table 17.
Hardware: Modules	See the manual of the corresponding module type. The system variables and system parameters are configured in the module's Detail View of the Hardware Editor
User Program	See Table 19

Table 14: System Variables at Different Project Levels

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 <i>Target Cycle Time Mode</i> , 0...7500 ms	0 ms	Application-specific
safeethernet CRC	SILworX V.2.36.0 The CRC for safeethernet 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	Application-specific
	Current Version The CRC for safeethernet is created with the current algorithm		

Parameter / Switch	Description	Default value	Setting for safe operation
Multitasking Mode	<p>Mode 1 The duration of a CPU cycle is based on the required execution time of all user programs.</p> <p>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.</p> <p>Mode 3 The processor waits for the unneeded execution time of user programs to expire and thus increases the cycle.</p>	Mode 1	Application-specific
Sum of UP Max. Duration for Each Cycle [μ s]	Sum of the values indicated for <i>Max. Duration for each Cycle [μs]</i> in all the user programs; not changeable.		-
Target Cycle Time Mode	<p><i>Use of Target Cycle Time [ms]</i></p> <p>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.</p> <p>Dynamic HIMax maintains the target cycle time as well as possible, but it also executes the cycle as quickly as possible.</p>	Fixed	Application-specific
Minimum configuration version	<p>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.</p> <p>SILworX-V3 Code generation for HIMax version 3. This setting ensures the compatibility with future versions.</p>	SILworX-V2	Application-specific

Table 15: System Parameters of the Resource

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

These parameters are used to ensure that the resource cycle time (CPU cycle time) is constantly maintained on the *Target Cycle Time [ms]* value. To do this, this parameter must be set to a value $\neq 0$. HIMax reduces then tasks such as reload and synchronization of redundant modules to such an extent that the target cycle time can be maintained.

The parameter *Target Cycle Time Mode* determines how accurately the target cycle time is maintained:

- If *Fixed* is set, the target cycle time is maintained accurately. The target cycle time must be set such that sufficient reserve is ensured for performing reload and synchronizing the redundant processor modules. If the CPU cycle is shorter than the target cycle time, HIMax prolongs the cycle to the target cycle time.
- If *dynamic* is set, HIMax completes the cycle as quickly as possible.

Hardware System Variables for Setting the Parameters

These system variables can be accessed in the SILworX Hardware Editor. Select the dark-gray background outside the base plate symbols and double-click or use the context menu to open the Detail View.

Variable	Description	Data type
Force Deactivation	ON: Forcing is deactivated. OFF: Forcing is possible. Switching from OFF to ON, all forcing procedures are immediately deactivated. Default value: OFF	BOOL
Spare 0...Spare 16	Reserved	USINT
Emergency Stop 1 ... Emergency Stop 4	ON: It sets the controller to the STOP state OFF: The controller is normally running Default value: OFF	BOOL
Read-only in RUN	ON: It locks the operator actions: Stop, Start, Download (but not Force and Reload). OFF: The operator actions: Stop, Start, Download are not locked. Default value: OFF	BOOL
Reload Deactivation	ON: It prevents the controller from being by performing a reload. OFF: Loading by performing a reload is permitted. Default value: OFF	BOOL

Table 16: Hardware System Variables for Setting the Parameters



The system variables Force Deactivation, Read-only in Run and Reload Deactivation can be activated by a key switch for authorized persons.
In this way, the user with the proper key switch can interrupt e.g. forcing.

To make one of the system variables Force Deactivation, Read-only in Run or Reload Deactivation operable with the key switch:

1. Assign a global variable to a system variable.
2. Assign the same global variable to a digital input.
3. Connect a key switch to the digital input.

The position of the key switch defines the the system variable value.

One key switch can be used to control several system variables.

Hardware System Variables for Reading the Parameters

These system variables can be accessed in the SILworX Hardware Editor. Select the dark-gray background outside the base plate symbols and double-click or use the context menu to open the Detail View.

Variable	Description	Data type
IO Error Counter	Number of current I/O errors	UDINT
IO Error Historic Count	Counted number of I/O errors (counter resettable)	UDINT
IO Warning Count	Number of current I/O warnings	UDINT
IO Warning Historic Count	Counted number of I/O warnings (counter resettable)	UDINT
Communication Error Count	Number of current communication errors	UDINT
Communication Error Historic Count	Counted number of communication errors (counter resettable)	UDINT
Communication Warning Count	Number of current communication warnings	UDINT
Communication Warnings Historic Count	Counted number of communication warnings (counter resettable)	UDINT
System Error Count	Number of current system errors	UDINT
System Error Historic Count	Counted number of system errors (counter resettable)	UDINT
System Warning Count	Number of current system warnings	UDINT
System Warning Historic Count	Counted number of system warnings (counter resettable)	UDINT
Autostart CPU Release	ON: When the processor module is connected to the supply voltage, it automatically starts the user program OFF: When the voltage supply is connected, the processor module enter the STOP state	BOOL
OS Major [1]...OS Major [4]	Operating system version contained in processor module 1...4	UINT
OS Minor [1]...OS Minor [4]		UINT
CRC	Project configuration checksum	UDINT
Date/time [ms part]	System date and time in s and ms since 1970-01-01	UDINT
Date/time [sec. part]		UDINT
Force Deactivation	ON: Forcing is deactivated. OFF: Forcing is possible.	BOOL
Forcing Active	ON: Global or local forcing is active. OFF: Global and local forcing are not active.	BOOL
Force Count	Number of force switches set	UDINT
Global Forcing Started	ON: Global forcing is active. OFF: Global forcing is not active.	BOOL
Spare 0 ... Spare 1 6	Reserved	USINT
Spare in17		BOOL
Last IO Warning [ms]	Date and time of the last I/O warning in s and ms since 1970-01-01	UDINT
Last IO Warning [s]		UDINT
Last Communication Warning [ms]	Date and time of the last communication warning in s and ms since 1970-01-01	UDINT
Last Communication Warn-		UDINT

Variable	Description	Data type										
ing [s]												
Last System Warning [ms]	Date and time of the last system warning in s and ms since 1970-01-01	UDINT										
Last System Warning [s]		UDINT										
Last IO Error [ms]	Date and time of the last I/O error in s and ms since 1970-01-01	UDINT										
Last IO Error [s]		UDINT										
Last Communication Error [ms]	Date and time of the last communication error in s and ms since 1970-01-01	UDINT										
Last Communication Error [s]		UDINT										
Last System Error [ms]	Date and time of the last system error in s and ms since 1970-01-01	UDINT										
Last System Fault [s]		UDINT										
Fan State	Reserved: it always provides the value 0xFF for <i>not available</i> .	BYTE										
Major CPU Release	Main enable switch of the processor module ON: The subordinate enable switches can be changed. OFF: The subordinate enable switches cannot be changed.	BOOL										
Mono Startup Release	Enable for non-redundant operation. ON: One single processor module in rack 0, slot 3 may also start with one system bus only. OFF: Both system buses are also required for one single processor module.	BOOL										
Read-only in RUN	ON: The operator actions: Stop, Start, Download are locked. OFF: The operator actions: Stop, Start, Download are not locked.	BOOL										
Redundancy Info	Bit-coded redundancy state of the processor modules: <table><tr><th>Bit no.</th><th>Processor module</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>2</td></tr><tr><td>2</td><td>3</td></tr><tr><td>3</td><td>4</td></tr></table> Bit = 0: Processor module not operating redundantly Bit = 1: Processor module operating redundantly All the remaining bits have the value 0.	Bit no.	Processor module	0	1	1	2	2	3	3	4	UDINT
Bit no.	Processor module											
0	1											
1	2											
2	3											
3	4											
Reload Release	ON: Controller can be loaded by performing a reload OFF: Controller cannot be loaded by performing a reload	BOOL										
Reload Deactivation	ON: Loading by performing a reload is locked OFF: Loading by performing a reload is possible	BOOL										
Reload Cycle	TRUE during the first cycle after a reload, otherwise FALSE	BOOL										
CPU Safety Time [ms]	Safety time set for the controller in ms	UDINT										
Start CPU Release	ON: Start of processor module using PADT is allowed OFF: Start of processor module using PADT is not allowed	BOOL										
Start Cycle	TRUE during the first cycle after starting, otherwise FALSE	BOOL										
Power Supply State [1]...[4]	Bit-coded state of the power supply units in the processor modules 1...4 <table><tr><th>Bit no.</th><th>State when bit is set</th></tr></table>	Bit no.	State when bit is set	BYTE								
Bit no.	State when bit is set											

Variable	Description	Data type
	0 Power supply rail 1, faulty	
	1 Power supply rail 2, faulty	
	2 Internally generated voltage too high or too low (overvoltage/low voltage)	
	3 Invalid adjustment data for the internally generated voltages	
System ID	System ID of the controller, 1...65535	UINT
Systemtick HIGH	Circular millisecond counter (64 bit)	UDINT
Systemtick LOW		UDINT
Temperature State [1] ...[4]	Bit-coded temperature state of processor modules 1...4	BYTE
	Bit no. State when bit is set	
	0 Temperature threshold 1 exceeded	
	1 Temperature threshold 2 exceeded	
	2 Incorrect temperature value	
Remaining Global Force Duration [ms]	Time in ms until the time limit set for global forcing expires.	DINT
CPU Watchdog Time [ms]	Maximum permissible duration of a RUN cycle in ms.	UDINT
Cycle Time, last [ms]	Current cycle time in ms	UDINT
Cycle Time, max [ms]	Maximum cycle time in ms	UDINT
Cycle Time, min [ms]	Minimum cycle time in ms	UDINT
Cycle Time, average [ms]	Average cycle time in ms	UDINT

Table 17: Hardware System Variables for Reading the Parameters

The following system variables taking from Table 17 are arrays. Their index is the processor module number:

- OS Major, OS Minor
- Redundancy Info (bit bar)
- Power Supply State
- Temperature State

The processor module index used in these fields is mapped onto the slots of the processor modules in the base plates as specified below:

- In base plate 0, the index is counted in ascending order starting with slot 3.
- In base plate 1, the index is counted in descending order down to slot 3.

The following assignment results:

	Slots			
	3	4	5	6
Base plate 1	4	3		
Base pPlate 0	1	2	3	4

Table18: Assigning the Index to Processor Module Slots

Processor modules with indexes 3 and 4 can either be located in base plate 0 or base plate 1!

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 Time-out 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 V3	Application-specific
	SILworX V2		

Table 19: System Parameters of the User Program

5.2.4 Assignment to I/O Channels

In the Hardware Editor of SILworX, a global variable can be assigned to an I/O channel. In the Detail View of an I/O module, drag a global variable from the Object Panel to the channel list of the I/O module.

In doing so, the channel's value and status information are available in the user program.

Use of Digital Input

Perform the following steps to use the value of a digital input in the user program

1. Define a global variable of type BOOL.
2. When defining the global variable, enter the initial value as safe value.
3. Assign the global variable to the channel value of the input.

The global variable provides the safe value to the user program.

For digital input modules for proximity switch internally operating in analog mode, the raw value can also be used and the safe value can be calculated in the user program. For more information, see below.

To get additional options for diagnosing the external wiring and programming fault reactions in the user program, assign global variable to *Channel OK* and to further diagnostic statuses. For more information on the individual diagnostic statuses such as line short-circuits and line breaks, refer to the manual of the corresponding module.

Use of Analog Inputs

Analog input channels convert the measured input currents into a value of type DINT (double integer). This value is made available to the user program as a "raw value". Here, 1 mA corresponds to a value of 10 000 and the range of values is 0...240 000.

As an easier alternative, the "process value" of the REAL data type can be often used instead of the "raw value". HIMax calculated the process value based on the raw value and the scale value on the parameters 4 and 20 mA. Refer to the module manual for more details.

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.

There are two possibilities to use the values of analog inputs in the user program.

- Use of the process value
If an analog input is configured correctly, its process value provides the value including the safe fault reaction.
- Using the raw value
the raw value is the measuring value without the safe fault reaction. You must program a safe fault reaction as appropriate for the project.

Perform the following steps to use the process value

1. Define a global variable of type REAL.
2. When defining the global variable, enter the initial value as safe value.
3. Assign the global variable to the process value of the input.
4. Specify the measuring range of the channel by giving a REAL value for 4 mA and for 20 mA.

The global variable provides the safe value to the user program.

Perform the following steps to use the raw value:

1. Define a global variable of type DINT.
2. In the user program, define a global variable of the type needed.
3. In the user program, program a suitable conversion function to convert the raw value into a used type and consider the measurement range.
4. In the user program, program a safety-related fault reaction using the statuses *Channel OK*, *LS*, *LB* (if necessary others).

The user program can process the measuring in a safety-related manner.

If the value 0 for a channel is *within the valid measuring range*, the user program must, at a minimum, evaluate the parameter *Channel OK* in addition to the process value.

To get additional options for diagnosing the external wiring and programming fault reactions in the user program, assign global variables to *Channel OK*, *Submodule OK*, *Module OK* and to further diagnostic statuses. For more information on the individual diagnostic statuses such as line short-circuits and line breaks, refer to the manual of the corresponding module.

Use of Safety-Related Counter Inputs

The counter reading or the rotation speed/frequency can be used as an integer value or as a scaled floating-point value.

Perform the following steps to use the integer value:

1. Define a global variable of type UDINT.
2. When defining the global variable, enter the initial value as safe value.
3. Assign the global variable to the integer value of the input..

The global variable provides the safe value to the user program.

Perform the following steps to use the scaled floating point value:

1. Define a global variable of type REAL.
2. When defining the global variable, enter the initial value as safe value.
3. Assign the global variable to the scaled floating point value of the input..
4. Specify the scaling value of the channel by giving a REAL value.

The global variable provides the safe value to the user program.

Use of Digital Input

Perform the following steps to write a value in the user program to a digital output:

1. Define a global variable of type BOOL.
2. When defining the global variable, enter the initial value as safe value.
3. Assign the global variable to the channel value of the output.

The global variable provides the safe value to the digital output.

To get additional options for diagnosing the external wiring and programming fault reactions in the user program, assign global variable to *Channel OK* and to further diagnostic statuses. For more information on the individual diagnostic statuses such as line short-circuits and line breaks, refer to the manual of the corresponding module.

Use of Digital Input

Perform the following steps to write a value in the user program to a digital output:

1. Define a global variable of type BOOL.
2. When defining the global variable, enter the initial value as safe value.
3. Assign the global variable to the channel value of the output.

The global variable provides the safe value to the digital output.

To get additional options for diagnosing the external wiring and programming fault reactions in the user program, assign global variable to *Channel OK* and to further diagnostic statuses. For more information on the individual diagnostic statuses such as line short-circuits and line breaks, refer to the manual of the corresponding module.

Use of Analog Outputs

Perform the following steps to write a value in the user program to a digital output:

1. Define a global variable of type REAL.
2. When defining the global variable, enter the initial value as safe value.
3. Assign the global variable to the channel value of the output..
4. With the output channel parameters *4 mA* and *20 mA*, set the corresponding REAL values according to the range used with the global variables.

The global variable provides the safe value to the digital output.

i

If output channels are not (or no longer) used, the parameters *4 mA* and *20 mA* must be set to the default settings 4.0 and 20.0, respectively.

To get additional options for diagnosing the external wiring and programming fault reactions in the user program, assign global variable to *Channel OK* and to further diagnostic statuses. For more information on the individual diagnostic statuses such as line short-circuits and line breaks, refer to the manual of the corresponding module.

5.2.5 Assignment to Communication Connections

Communication connections can be used to send or receive the values of global variables. To do this, open the editor for the communication protocol in use and drag the global variable from the Object Panel to the workspace.

For more information on the communication protocols, refer to the Communication Manual HI 801 101. For more information on how to use the editor for the communication protocols, refer to the SILworX online help.

5.2.6 Configuring the Sequence of Events Recording

Event Definition

1. Define a global variable for each event. Generally use global variables that have already been defined for the program.
2. Below the resource, create a new **Alarm & Events** branch, if not existing.
3. Define events in the Alarm & Event Editor.
 - Drag global variables into the event window for Boolean or scalar events.
 - Define the details of the events, see Table 20 and Table 21.

The events are defined.

For further information, refer to the SILworX online help.

The parameters of the Boolean events are entered in a table with the following columns:

Column	Description	Range of Values
Name	Name for the event definition; it must be unique within the resource.	Text, max. 32 characters.
Global Variable	Name of the assigned global variable (added e.g. using a drag&drop operation)	
Data type	Data type of the global variable; it cannot be modified.	BOOL
Event Source	<p>CPU event The processor module creates the time-stamp. It creates all the events in each of its cycle.</p> <p>IO event A suitable I/O module (e.g., AI 32 02) creates the timestamp.</p> <p>Auto event The timestamp is created by a suitable I/O module, if assigned, otherwise by the processor module.</p> <p>Default value: Auto</p>	CPU, I/O, Auto
Alarm at FALSE	<p>Activated If the global variable value changes from TRUE to FALSE, an event is triggered.</p> <p>Deactivated If the global variable value changes from FALSE to TRUE, an event is triggered.</p> <p>Default value: Deactivated</p>	Checkbox activated, deactivated
Alarm Text	Text specifying the alarm state	Text
Alarm Severity	Priority of the alarm state Default value: 0	0...1000
Alarm Ack Required	<p>Activated The alarm state must be confirmed by the user (acknowledgement)</p> <p>Deactivated The alarm state may not be confirmed by the user</p> <p>Default value: Deactivated</p>	Checkbox activated, deactivated
Return to Normal Text	Text specifying the alarm state	Text
Return to Normal Severity	Priority of the normal state	0...1000
Return to Normal Ack Required	<p>The normal state must be confirmed by the user (acknowledgement)</p> <p>Default value: Deactivated</p>	Checkbox activated, deactivated

Table 20: Parameters for Boolean Events

The parameters of the scalar events are entered in a table with the following columns:

Column	Description	Range of Values
Name	Name for the event definition; it must be unique within the resource.	Text, max. 32 characters
Global Variable	Name of the assigned global variable (added using a drag&drop operation)	
Data type	Data type of the global variable; it cannot be modified.	depending on the global variable type
Event Source	<p>CPU event The processor module creates the timestamp. It creates all the events in each of its cycle.</p> <p>IO event A suitable I/O module (e.g., AI 32 02) creates the timestamp.</p> <p>Auto event The timestamp is created by a suitable I/O module, if assigned, otherwise by the processor module.</p> <p>Default value: CPU Event</p>	CPU Event, IO Event, Auto Event
HH Alarm Text	Text specifying the alarm state of the highest limit value.	Text
HH Alarm Value	Highest limit value triggering an event. Condition: (HH Alarm Value - Hysteresis) > H Alarm Value or HH Alarm Value = H Alarm Value	depending on the global variable type
HH Alarm Severity	Priority of the highest limit value; default value: 0	0...1000
HH Alarm Ack Required	<p>Activated The user must confirm that the highest limit value has been exceeded (acknowledgment).</p> <p>Deactivated The user may not confirm that the highest limit value has been exceeded.</p> <p>Default value: Deactivated</p>	Checkbox activated, deactivated
H Alarm Text	Text specifying the alarm state of the upper limit value.	Text
H Alarm Value	Upper limit value triggering an event. Condition: (H Alarm Value - Hysteresis) > (L Alarm Value + Hysteresis) or H Alarm Value = L Alarm Value	depending on the global variable type
H Alarm Severity	Priority of the upper limit value; default value: 0	0...1000
H Alarm Ack Required	<p>Activated The user must confirm that the upper limit value has been exceeded (acknowledgment).</p> <p>Deactivated The user may not confirm that the upper limit value has been exceeded.</p> <p>Default value: Deactivated</p>	Checkbox activated, deactivated
Return to Normal Text	Text specifying the normal state	Text
Return to Normal Severity	Priority of the normal state; default value: 0	0...1000
Return to Normal Ack Required	The normal state must be confirmed by the user (acknowledgment); default value: Deactivated	Checkbox activated, deactivated
L Alarm Text	Text specifying the alarm state of the lower limit value.	Text
L Alarm Value	Lower limit value triggering an event. Condition: (L Alarm Value + Hysteresis) < (H Alarm Value - Hysteresis) or L Alarm Value = H Alarm Value	depending on the global variable type
L Alarm Severity	Priority of the lower limit value; default value: 0	0...1000
L Alarm Ack Required	<p>Activated The user must confirm that the lower limit value has been exceeded (acknowledgment).</p> <p>Deactivated The user may not confirm that the lower limit value has been exceeded.</p> <p>Default value: Deactivated</p>	Checkbox activated, deactivated
LL Alarm Text	Text specifying the alarm state of the lowest limit value.	Text

Column	Description	Range of Values
LL Alarm Value	Lowest limit value triggering an event. Condition: (LL Alarm Value + Hysteresis) < (L Alarm Value) or LL Alarm Value = L Alarm Value	depending on the global variable type
LL Alarm Severity	Priority of the lowest limit value; default value: 0	0...1000
LL Alarm Ack Required	Activated The user must confirm that the lowest limit value has been exceeded (acknowledgment). Deactivated The user may not confirm that the lowest limit value has been exceeded. Default value: Deactivated	Checkbox activated, deactivated
Alarm Hysteresis	The hysteresis avoids that many events are continuously created when the process value often oscillate around a limit.	depending on the global variable type

Table 21: Parameters for Scalar Events

NOTICE

Faulty event recording due wrong parameter settings possible!

Setting the parameters *L Alarm Value* and *H Alarm Value* to the same value can cause an unexpected behavior of the event recording since no normal range exists in such a case.

For this reason, make sure that *L Alarm Value* and *H Alarm Value* are set to different values.

5.3 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 5.3.1.

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.

5.3.1 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!

5.3.2 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.

5.3.3 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>.

5.3.4 Forcing and Scalar Events

When a global variable used to create scalar events - see Chapter 3.5.1, is being forced, observe the following points:

- The events are created according to the force value.
- The values of these variable-dependent status variables are not tracked in accordance with the force value!

In such cases, the corresponding status variables must also be forced!

5.4 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 7 shows both scenarios in a project containing two user programs.

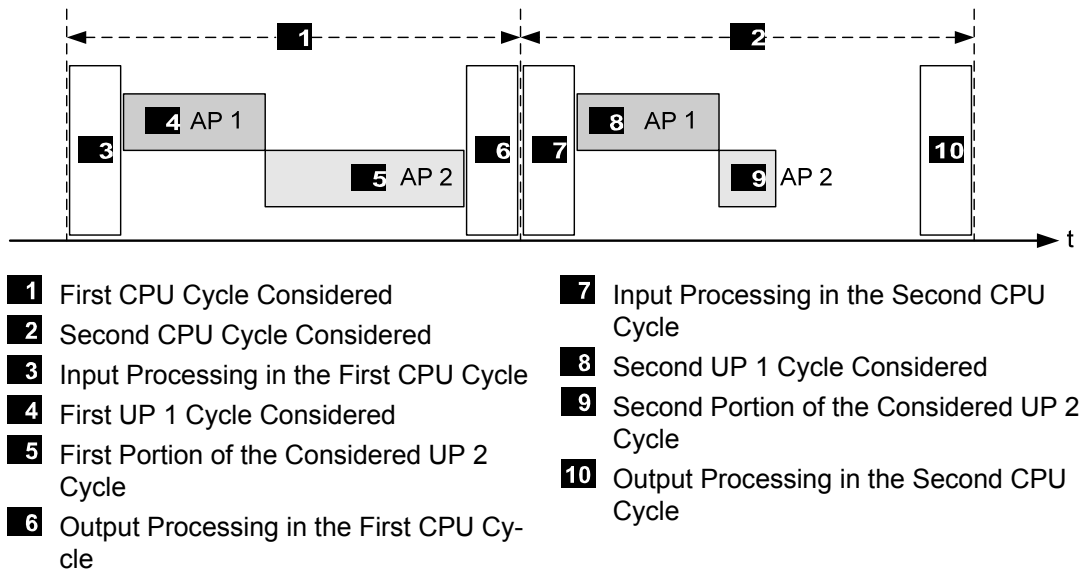


Figure 7: CPU Cycle Sequence with Multitasking

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 **1**. 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 [μ s]	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 <i>Max. Duration for Each Cycle [μs]</i> . 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 <i>Target Cycle Time [ms]</i>	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 22: Parameters Configurable for Multitasking

Observe the following rules when setting the parameters:

- If *Max. Duration for Each Cycle [μ s]* 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 [μ s]* 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 [μ s]* parameters in all user programs must be large enough to ensure that sufficient reserve is available to maintain the target cycle time.
- The *Program IDs* 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 = *watchdog time* * maximum number of cycles

i

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

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

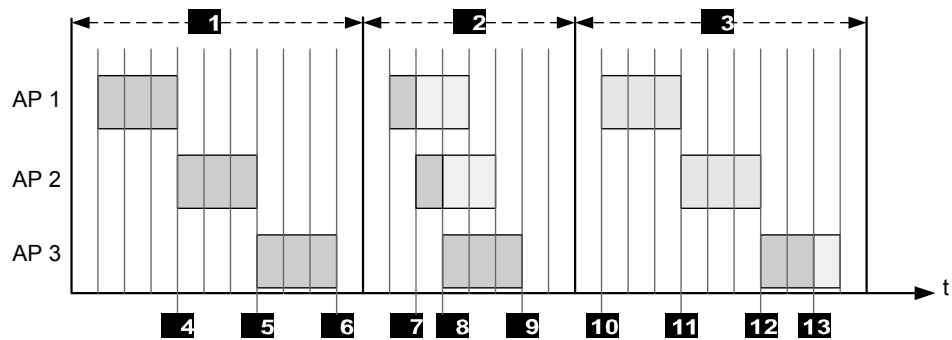
HIMA recommends to set the *Max. Duration for each Cycle [μs]* parameter to an appropriate value $\neq 0$. This ensures that a user program with an excessively long runtime is stopped during the current CPU cycle and resumed in the next CPU cycle without affecting the other user programs.

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.

5.4.1 Multitasking Mode

For every resource, one of three operation modes can be selected for multitasking. These modes differ in how the time that is not needed for executing the CPU cycle of the user programs is used:

1. **Multitasking Mode 1** uses the unneeded time to reduce the CPU cycle. If the user program is completely processed, processing of the next user program begins immediately. In total, this results in a shorter cycle.
Example: 3 user programs (UP 1, UP 2 and UP 3) that allow a user program cycle to take up to 3 CPU cycles -

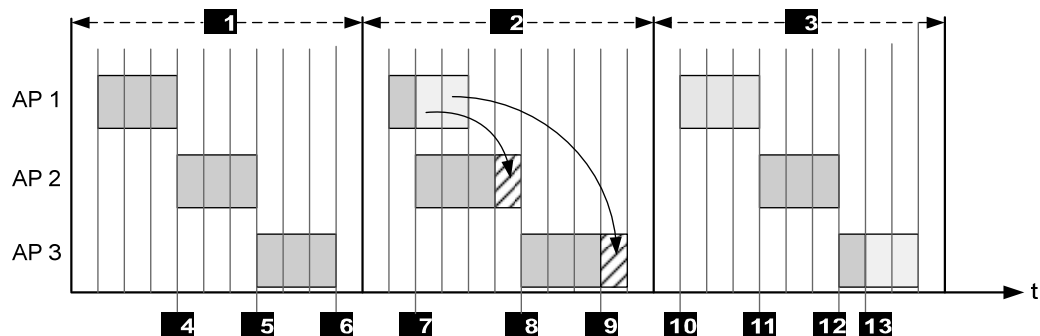


- | | |
|---|---|
| <p>1 First CPU Cycle Considered.</p> <p>2 Second CPU Cycle Considered.</p> <p>3 Third CPU Cycle Considered.</p> <p>4 The UP 1 <i>Max. Duration for Each Cycle</i> [μs] has Expired, UP 2 Starts.</p> <p>5 The <i>Max. Duration for Each Cycle</i> [μs] of UP 2 has Expired, UP 3 Starts..</p> <p>6 The UP 3 <i>Max. Duration for Each Cycle</i> [μs] has Expired, Completion of the First CPU Cycle.</p> | <p>7 Completion of the UP 1 Cycle, UP 2 Resumes.</p> <p>8 Completion of the UP 2 Cycle, UP 3 Resumes.</p> <p>9 The UP 3 <i>Max. Duration for Each Cycle</i> [μs] has Expired, Completion of the Second CPU Cycle.</p> <p>10 Completion of UP 3 Cycle (User Programs Can Resume on Redundant Processor Modules)</p> |
|---|---|

Figure 8: Multitasking Mode 1

2. In **multitasking mode 2**, the unneeded duration of lower-priority user programs is distributed among higher-priority user programs. In addition to the specified *Max. Duration for Each Cycle [μs]*, these user programs can use the portions of unneeded duration. This procedure ensures high availability.

Example:



- 1 First CPU Cycle Considered.
- 2 Second CPU Cycle Considered.
- 3 Third CPU Cycle Considered.
- 4 The UP 1 *Max. Duration for Each Cycle [μs]* has Expired, UP 2 Starts.
- 5 The *Max. Duration for Each Cycle [μs]* of UP 2 has Expired, UP 3 Starts.
- 6 The UP 3 *Max. Duration for Each Cycle [μs]* has Expired, Completion of the First CPU Cycle.
- 7 Completion of the UP 1 Cycle, UP 2 Resumes. The Remaining Duration is Distributed Among the UP 2 and UP 3 *Max. Duration for Each Cycle [μs]* (Arrows).
- 8 UP 2 *Max. Duration for Each Cycle [μs]* + Proportional Remaining Duration of UP 1 have Expired, UP 3 Resumes.
- 9 UP 3 *Max. Duration for Each Cycle [μs]* + Proportional Remaining Duration of UP 1 have Expired, Completion of the Second CPU Cycle.
- 10 Completion of the UP 2 Cycle, UP 3 Resumes.
- 11 Completion of the UP 3 Cycle, the Remaining Duration is Possibly Used for Synchronization with Redundant Processor Modules

Figure 9: Multitasking Mode 2

i

The unused execution time of user programs that were not run cannot be exploited as residual time by other user programs. User programs are not run if they are in one of the following states:

- STOP
- ERROR
- TEST_MODE

As a consequence, the number of CPU cycles required to process another user program cycle could increase.

In such a case, if the value set for *Maximum Cycle Count* is too low, the maximum time for processing a user program can be exceeded and result in an error stop!
Maximum processing time = *Max. Duration for Each Cycle [μs]* * *Maximum Number of Cycles*

3. **Multitasking mode 3** does not use the unneeded duration for running the user programs, rather, it waits until the *Max. Duration for Each Cycle* [μ s] of the user program is reached and then starts processing the next user program. This behavior results in CPU cycles of the same duration.

Example:

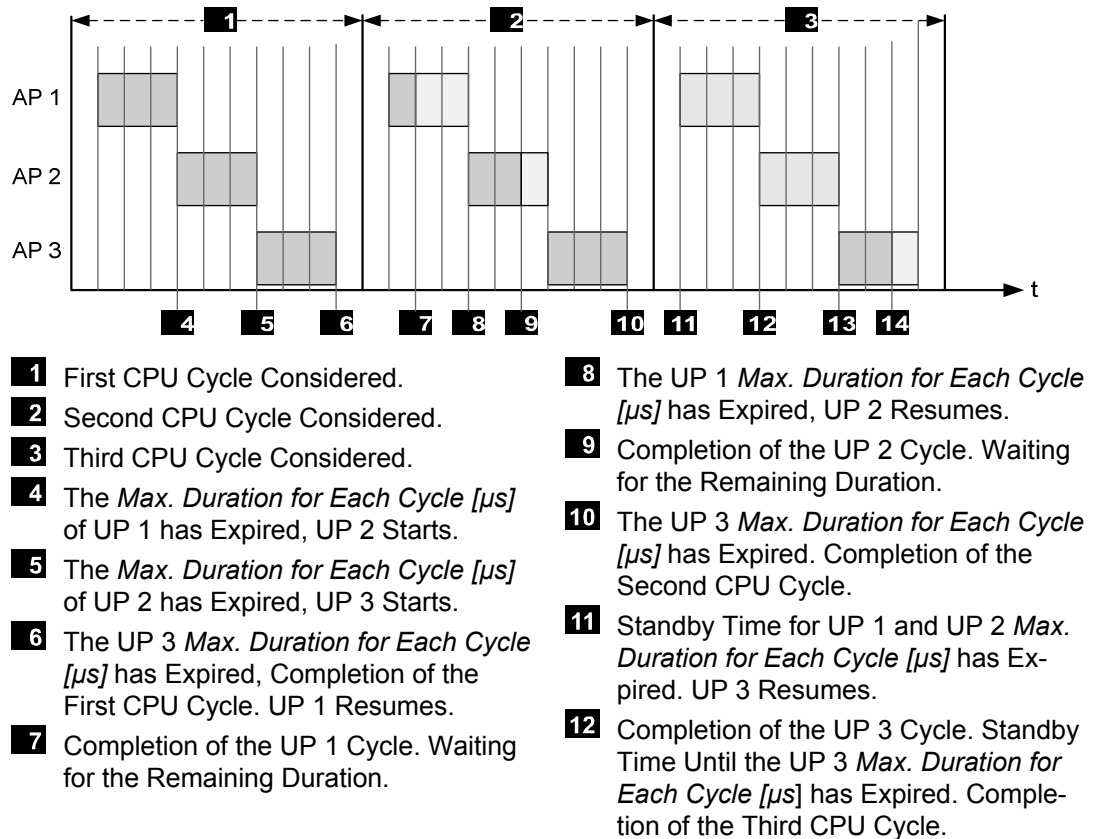


Figure 10: Multitasking Mode 3

i

In the examples illustrating the multitasking modes, input and output processing are represented as empty spaces at the beginning and the end of each CPU cycle.

The multitasking mode can be set using the resource parameter *Multitasking Mode*, see Table 15.

5.5 Loading User Programs

Use SILworX to load the project configuration with the user programs into the controller. Two load variants exist:

- Download
Load of a new project configuration with interruption of safety-related operation
- Reload
Load of a modified project configuration without interruption of safety-related operation

i

HIMA recommends backing up the project configuration, e.g., on a data storage medium, after loading a user program into the controller.

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 user program load.

5.5.1 Download

Requirements for the download:

- Controller in STOP
- Resource enable switch set to Load Allowed

After a download, the user program must be launched in SILworX to start safety-related operation.

Use the download function to load a new program into the controller or if one of the conditions mentioned in the next section prevents using the reload function.

5.5.2 Reload

Requirements:

- Controller in RUN
- Enable switch Reload Allowed is set to ON.
- System variable Reload Deactivation is set to OFF.

i

- A reload can also be performed if the controller only contains one processor module.
- During a reload, the user cannot use the PADT to operate on the controller!

Exceptions:

It is possible to abort the reload procedure and to modify the watchdog and target cycle times in order to allow the reload.

If a user program already running in a controller is modified, the modified version can be loaded into the controller by performing a reload. While the previous version of the user program is still running, the new version is stored in the controller memory, tested and provided with the variable values. Once the preparation steps are completed, the controller adopts the new user program version and continues safety-related operation seamlessly.

The following factors limit the possibility to load a modified program into the controller by performing a reload:

- The changes described in Chapter "[Conditions for Using the Reload Function](#)".
- Time required to perform a reload.

The execution cycle takes longer due to the time required by the additional reload tasks. To prevent that the watchdog triggers and the controller enters the error stop state, both SILworX and the controller verify the additional time required to perform a reload. If the time required is too high, a reload is rejected.

i

In the watchdog time, plan sufficient time reserve to perform the reload.

HIMA recommends the procedure in safety manual HI 801 003 to evaluate the watchdog time..

The watchdog and target cycle times can be increased for the duration of the reload, refer to the SILworX online help for more details. This can be necessary if the defined time reserve is too short and the reload procedure blocks in the Cleanup phase.

The online function only allows one to increase the watchdog and target cycle times, and not to reduce them below the value set in the project.

Conditions for Using the Reload Function

The following project modifications can be transferred to the controller by performing a reload:

- Changes to the user program parameters.
- Changes to the logic of the program, function blocks and functions.
- Changes that allow a reload in accordance with Table 23.

Changes to	Type of change			
	Add	Delete	Change of the initial value	Assignment of other variables
Assigning global variables to				
User programs	•	•	•	•
System Variables	•	•	•	•
I/O channels	•	•	•	•
Communication protocols	-	-	-	-
safeethernet	-	-	•	-
Base plate with system bus and I/O modules	•	•	n.a.	n.a.
Modules (I/O, system bus, and processor modules)	•	•*	n.a.	n.a.
Communication protocols	-	-	n.a.	n.a.
User programs	•	•**	n.a.	n.a.

• Reload possible

- Reload impossible

* Reload possible, except for system bus modules in which the *Responsible* attribute is activated

** Reload possible, but the controller must still contain at least one user program

n.a. non-applicable

Table 23: Reloading after Changes

A reload may only be performed in accordance with the conditions mentioned in the previous section. In all the other cases, stop the controller and perform a download.

TIP

Proceed as described below to be able to perform a reload even if global variable assignments have been added:

- While creating the user program, assign unused global variables to communication protocols.
- Assign safe value as initial value to unused global variables.

To a later time point, this assignment must only be changed and not added ensuring the possibility to perform a reload.

5.6 Loading Operating Systems

All HIMax system modules contain one processor system and one operating system that controls the module. The operating system is delivered with the module. HIMA is continuously improving the operating systems. The improved versions can be loaded into the module using SILworX.

5.6.1 Load Process

Update the operating system in the following module order:

No.	Modules	File name beginning with	PADT connected to
1	I/O modules	himaxio_...	Processor module
2	Communication modules	himaxcom_...	Processor module
3	System bus modules	himaxsb_...	Processor module, if the connection is possible, otherwise system bus module
4	Processor modules	himaxcpu_...	System bus module, if the connection is possible, otherwise processor module

Table 24: Module Order while Loading the Operating System

NOTE



Service interruption possible during the loading procedure!

Ensure the operation of a functional, redundant module! The redundant module maintains operation during the loading procedure.

To load a new operating system into all modules

1. Extract the zip file delivered by HIMA in a folder.
2. Connect the PADT with the processor module via Ethernet.
3. Start *Online* mode in the SILworX Hardware Editor.
Log in to the system using the IP address of the processor module.
4. Stop the new module to be loaded using the context menu.
5. Load the operating system.
Use the file specified in Table 24 from the folder created in step 1.
☒ After being loaded, the modules automatically adopt the RUN state.

Perform these steps for all the I/O modules, for the communication modules, and for the system bus modules.

6. Stop the processor module's system operation prior to loading to a processor module.
If a second processor module exists, it adopts system operation.
 7. Load the operating system using the context menu.
☒ After loading the operating system of the processor module connected to the PADT, the PADT closes the connection. Log in again to the system.
 8. Only load the second processor module, if the first one is in RUN.
- All modules operate with the new operating system.

5.6.2 Updating and Downgrading Operating Systems

In seldom cases, it can make sense to load a previous operating system version into the module:

If a controller has run for a long time without modification and an individual module must be replaced, it may be better to import the existing operating system version into the new module. The existing operating system version may be better suited for use with that of the remaining modules.

6 User Management

SILworX can set up and maintain an own user management scheme for each project and controller.

6.1 User Management for SILworX Projects

A PADT user management scheme for administering the access to the project can be added to every SILworX project.

If no PADT user management scheme exists, any user can open and modify the project. If a user management scheme has been defined for a project, only authorized users can open the project. Only users with the corresponding rights can modify the projects. The following authorization types exist.

Type	Description
Safety Administrator (Sec Adm)	Safety administrators can modify the user management scheme: setting up, deleting, changing the PADT user management scheme, and the user accounts and user groups, setting up the default user account. Furthermore, they can perform all SILworX functions.
Read and Write (R/W)	All SILworX functions, except for the user management
Read only (RO)	Read-only access, i.e., the users may not change or archive the projects.

Table 25: Authorization Types for the PADT User Management Scheme

The user management scheme allocates the rights to the user groups. The user groups allocates the rights to the user accounts assigned to it.

Characteristics of user groups:

- The name must be unique within the project and must contain 1...31 characters.
- A user group is assigned an authorization type.
- A user group may be assigned an arbitrary number of user accounts.
- A project may contain up to 100 user groups.
- If the name of a user group is changed, it might happen that the controllers can no longer be loaded by performing a reload.

Characteristics of user accounts:

- The name must be unique within the project and must contain 1...31 characters.
- A user account is assigned a user group.
- A project may contain up to 1000 user accounts.
- A user account can be the project default user

6.2 User Management for the Controller

The user management for a controller (PES user management) is used to protect the HI-Max controller against unauthorized access and actions. The user and its access rights are part of the project; they are defined with SILworX and loaded into the processor module.

The user management is used to set and manage the access rights to a controller for up to ten users. The access rights are stored in the controller and remain valid also after switching off the operating voltage.

Each user account is composed of name, password and access right. The user data can be used to log in once a download has been performed to load the project into the controller. The user accounts of a controller can also be used for the corresponding remote I/Os.

Users log in to a controller using their user name and password.

Creating user accounts is not required, but is a contribution to a safe operation. If a user management scheme is defined for a resource, it must contain at least one user with administrator rights.

6.2.1 Default User

The factory user settings apply as long as no user accounts have been created for a resource. The factory settings also apply after starting a processor module using the mode switch set to *Init*.

Factory settings

Number of users:	1
User ID:	Administrator
Password:	None
Access right:	Administrator



Note that the default settings cannot be retained if new user accounts are defined.

6.2.2 Parameters for User Accounts

Define the following parameters to create user accounts:

Parameter	Description
Username	User name or ID to log in to a controller. The user name must not contain more than 32 characters (recommended: a maximum of 16 characters) and may only be composed of letters (A ... Z, a ... z), numbers (0 ... 9) and the special characters underscore «_» and hyphen «-». The user name is case sensitive.
Password	Password assigned to a user name required for the log-in. The password must not contain more than 32 characters and may only be composed of letters (A ... Z, a ... z), numbers (0 ... 9) and the special characters underscore «_» and hyphen «-» The password is case sensitive.
Confirm Password	Repeat the password to confirm the entry.
Access Mode	The access modes define the privileges that a user may have. The following access types are available: <ul style="list-style-type: none"> Read: Users may only read information but they cannot modify the controller. Read + Write: Similar to "Read", but users may also: <ul style="list-style-type: none"> Create programs Translate programs Load programs into the controller Test programs Administrator: Similar to "Read + Write", but users may also: <ul style="list-style-type: none"> Load operating systems. Modify the main enable switch setting Change the SRS Set the system bus modules to "responsible" Change the IP settings <p>At least one user must have administrator rights, otherwise the controller settings are not accepted. The administrator can cancel the access to a controller to a later time point by completely deleting the user from the list.</p>

Table 26: Parameters for User Accounts in the PES User Management Scheme

6.2.3 Setting Up User Accounts

A user with administrator rights can access to all user accounts.

Observe the following points when setting up user accounts:

- Make sure that at least one user account is assigned with administrator rights. Define a password for the user account with administrator rights.
- If a user account was created in the user management and should be edited, its password must be used to be able to access it.
- In SILworX, use the Verification function to check the created user account.
- The new user accounts are valid once the code has been generated and a download has been performed to load the project into the controller. All the user accounts previously saved, e.g., the default settings, are no longer valid.

7 Diagnosis

The diagnostic LEDs are used to give a first quick overview of the system state. The diagnostic history in SILworX provides detailed information.

7.1 Light Emitting Diodes

Light emitting diodes (LEDs) on the front plate indicate the module state. All LEDs should be considered together. The state of one single LED is not sufficient to assess the module state.

The LEDs on the modules are divided into the following groups:

- Module status indicators
- Redundancy indicators
- Rack connection indicators
- System bus indicators
- Slot indicators
- Maintenance indicators
- Fault indicators
- I/O indicators
- Fieldbus indicators
- Ethernet indicators
- Communication indicators

When the voltage is connected, the module performs a test of the LEDs.

When the supply voltage is switched on, a LED test is performed and all LEDs briefly flash simultaneously.

Definition of Blinking Frequencies

The following table defines the blinking frequencies of the LEDs:

Name	Blinking Frequencies
Blinking1	Long (approx. 600 ms) on, long (approx. 600 ms) off
Blinking2	Short (approx. 200 ms) on, short (approx. 200 ms) off, long (approx. 600 ms) on, long (approx. 600 ms) off
Blinking-x	Ethernet communication: Flashing in sync with data transfer

Table 27: Blinking Frequencies of LEDs

Assignment of the LED groups to the types of modules:

Groups	Module types
Module Status Indicators	All
Redundancy Indicators	Processor module, system bus module
System Bus Indicators	All, except for system bus module
Rack connection indicators	System bus module
Slot Indicators	System bus module
Fault Indicators	Processor module
Maintenance Indicators	Processor module
I/O Indicators	I/O modules
Fieldbus Indicators	Communication module
Ethernet Indicators	Processor module, communication module
Communication Indicators	System bus module

Table 28: Assignment of the LED Groups to the Types of Modules

7.1.1 Module Status Indicators

These LEDs are located on the front plate, on the upper part of the module.

LED	Color	Status	Description
Run	Green	On	Module in RUN, normal operation
		Blinking1	Module state: STOP/OS_DOWNLOAD or OPERATE (only with processor modules)
		Off	Module not in RUN, observe the other status LEDs
Error	Red	On/Blinking1	Internal module faults detected by self-tests, e.g., hardware, software or voltage supply. Fault while loading the operating system
		Off	Normal operation
Stop	Yellow	On	Module state: STOP / VALID CONFIGURATION
		Blinking1	Module state: STOP / INVALID CONFIGURATION or STOP / OS_DOWNLOAD
		Off	Module not in STOP, observe the other status LEDs
Init	Yellow	On	Module state: INIT, observe the other status LEDs
		Blinking1	Module state: LOCKED, observe to the other status LEDs
		Off	Module state: neither INIT nor LOCKED, observe the other status LEDs

Table 29: Module Status Indicators

7.1.2 Redundancy Indicators

LED	Color	Status	Description
Ess	Yellow	On	Do not remove the module! The module is absolutely required for operating the HIMax system. Only one module is configured.
		Blinking1	Do not remove the module! The module is absolutely required for operating the HIMax system. Multiple redundant modules are configured.
		Off	The module is not absolutely required for operation. It may be removed, if required.
Red	Yellow	On	The module is operating redundantly with at least one additional module.
		Blinking1	At least one processor module starts system operation or less modules than planned are operating redundantly.
		Off	The module is not operating redundantly.

Table 30: Redundancy Indicators

7.1.3 System Bus Indicators

The system bus LEDs are labeled *Sys Bus*.

LED	Color	Status	Description
A	Green	On	Physical and logical connection to the system bus module in slot 1.
		Blinking1	Fault on the system bus
	Yellow	Blinking1	The physical connection to the system bus module in slot 1 has been established. No connection to a (redundant) processor module running in system operation.
		Off	No physical connection to the system bus module in slot 1.
B	Green	On	Physical and logical connection to the system bus module in slot 2.
		Blinking1	Fault on the system bus
	Yellow	Blinking1	The physical connection to the system bus module in slot 2 has been established. No connection to a (redundant) processor module running in system operation.
		Off	No physical connection to the system bus module in slot 2.

Table 31: System Bus Indicators

7.1.4 Rack connection indicators

The rack connection and slot LEDs are labeled *Sys Bus*.

LED	Color	Status	Description
Up	Green	On	Physical and logical connection to the system bus module in another base plate.
	Yellow	Blinking1	Only a physical connection to the system bus module in another base plate.
	Off	Off	No connection to another system bus module.
Down	Green	On	Physical and logical connection to the system bus module in another base plate.
	Yellow	Blinking1	Only a physical connection to the system bus module in another base plate.
	Off	Off	No connection to another system bus module.

Table 32: Rack Connection Indicators

7.1.5 Slot Indicators

LED	Color	Status	Description
3...18	Green	On	Module inserted in slot X, logical connection established.
	Yellow	On	Not activated!
		Blinking1	Module inserted in slot X, logical connection not established.
	Off	Off	Slot X not used

Table 33: Slot Indicators

7.1.6 Fault Indicators

The fault LEDs are labeled *Fault*.

LED	Color	Status	Description
System	Red	Blinking1	Faults detected in a HIMax system module, e.g., hardware, software, over temperature or power supply. The module or base plate is missing or does not match the configuration or cannot be operated as intended.
		Off	No module fault displayed for a HIMax system module
Field	Red	Blinking1	Field faults in an I/O module of the HIMax system
		Off	No field faults displayed for an I/O module in the HIMax system
Com	Red	Blinking1	Fault in the external process data communication
		Off	No faults displayed for the external process data communication

Table 34: Fault Indicators

7.1.7 Maintenance Indicators

The maintenance LEDs are labeled *Maint*.

LED	Color	Status	Description
Force	Yellow	On	Forcing prepared, processor module in STOP, RUN or OPERATE
		Blinking1	Forcing active, processor module in RUN or OPERATE
		Off	Forcing inactive
Test	Yellow	On	Connection to the PADT with write permission
		Blinking1	One user program is in RUN_FREEZE, (single step operation)
		Off	No connection to the PADT with write permission
Prog	Yellow	On	Download (processor module in STOP), the configuration is being loaded
		Blinking1	Reload procedure active or exchange of configuration data between processor modules
		Off	No loading procedure active and no configuration data exchange on the processor module

Table 35: Maintenance Indicators

7.1.8 I/O Indicators

The LEDs of the I/O indicators are labeled *Channel* and *Field*.

LED	Color	Status	Description
Channel 1 - n	Yellow	On	The related channel is active (energized).
		Blinking2	The related channel is faulty.
		Off	The related channel is inactive (de-energized).
Field	Red	On	LED test while booting
		Blinking1	Field faults in at least one channel (line break, line short-circuit, over-current, etc.
		Off	No field faults

Table 36: I/O Indicators LEDs

The number of channels and thus the number of *Channel* LEDs depends on the type of input or output module.

With modules that (internally) operate in analog, the signal value of the *Channel* LEDs is based on thresholds set during the planning phase:

- The *Channel* LED is lit if the switching point set for HIGH (*SP HIGH*) has been exceeded.
- The *Channel* LED is no longer lit if the switching point set for LOW (*SP LOW*) has been under-run.
- The *Channel* LED state remains unchanged as long as one of the conditions previously mentioned modifies it.

Depending on the module, the *Field* LED also indicates overvoltage, low voltage or overcurrent of transmitter supply.

For more information on the I/O indicators for a specific module, refer to the corresponding module manual.

7.1.9 Fieldbus Indicators

The fieldbus LEDs are labeled *Fieldbus*.

LED	Color	Status	Description
1, 2	Yellow	On	Fieldbus operating
		Off	No activity, fieldbus not operating
Fault	Red	Blinking1	Fieldbus fault of the bus (e.g., the slave is not present or faulty response), depending on the fieldbus protocol (minimum blinking duration 5 s).
		Off	No fieldbus faults

Table 37: Fieldbus Indicators

7.1.10 Ethernet Indicators

The Ethernet LEDs are labeled *Ethernet*.

LED	Color	Status	Description
Eth 1...4	Green	On	Communication partner connected No communication detected on interface
		Blinking-x	Communication detected on interface.
		Blinking1	IP address conflict detected All Ethernet LEDs are blinking
		Off	No communication partner connected
H/F/Col 1...4	Yellow	On	Full duplex operation on Ethernet line "F".
		Blinking-x	Collisions detected on Ethernet line "Col"
		Blinking1	IP address conflict detected All Ethernet LEDs are blinking
		Off	Half duplex operation on Ethernet line "H".

Table 38: Ethernet Indicators

7.1.11 Communication Indicators

The communication LEDs are labeled *Ethernet*.

LED	Color	Status	Description
PADT	Green	Blinking-x	Communication detected on interface.
		Blinking1	IP address conflict detected. LEDs adjacent to one another, PADT and H/F/Col blinking
		Off	PADT not connected.
H/F/Col (PADT)	Yellow	On	Speed = 100 Mbit/s
		Blinking-x	not defined!
		Blinking1	IP address conflict detected. LEDs adjacent to one another, PADT and H/F/Col blinking
		Off	Speed = 10 Mbit/s or no connection.
Up	Green	On	System bus module connected, physical connection established.
		Off	No system bus module connected.
Down	Green	On	System bus module connected, physical connection established.
		Off	No system bus module connected.
Diag	Green	On	Diagnostic device connected, physical connection established
		Off	No diagnostic device connected.
H/F/Col (Up, Down, Diag)	Yellow	On	Full duplex operation on Ethernet line "F".
		Blinking-x	Collision detected on Ethernet line "Col".
		Off	Half duplex operation on Ethernet line "H".

Table 39: Communication Indicators

7.2 Diagnostic History

Each HIMax module maintains a diagnostic history about the occurred faults or other events. The events in the history are stored in chronological order. The history is organized as a ring buffer.

The diagnostic history is composed of short term diagnosis and long term diagnosis.

- Short term diagnosis
If the maximum number of entries has been reached, each new entry deletes the oldest entry.
- Long term diagnosis
The long term diagnosis essentially stores actions and configuration changes performed by the user.
If the maximum number of entries has been reached, each new entry deletes the oldest entry if this is older than three days.
The new entry is rejected if the existing entries are not older than three days. The rejection is marked by a special entry.

The number of events that can be stored depends on the type of module.

Module Type	Max. number of events long term diagnosis	Max. number of events short term diagnosis
X-CPU 01	2500	1500
X-COM 01	300	700
I/O modules	400	500
X-SB 01	400	500

Table 40: Maximum Number of Entries Stored in the Diagnostic History per Module Type

i

The diagnostic entries can be lost if a power outage occurs just before they could be saved into non-volatile memory.

SILworX can be used to read the histories of the individual modules and represent them so that the information required to analyze a problem is available. Example

- Mixing the histories from various sources
- Filtering them according to the time period
- Printing out the edited history
- Saving the edited history

For additional functions, see the SILworX online help.

i

If a module is plugged in to a base plate, it generates diagnostic messages during its initialization phase indicating faults such as incorrect voltage values.

If these messages only occur before the entire system is operated, they do not indicate a fault in the module.

7.3

Online Diagnosis

The Online View in the SILworX Hardware Editor is used to diagnose failures in the HIMax modules. Failed modules are signalized by a color change:

- Red indicates severe failures, e.g., that the module is not inserted.
- Yellow indicates less severe failures, e.g., that the temperature threshold has been exceeded.

Point to a module to display a tooltip providing the following state information about the module:

Information	Representa- tion	Range of values	Description																
S.R.S	Three deci- mals	0...65535, 0...15, 1...18	Module identification																
Module state	Text	e.g., STOP, RUN	State text indicating the operating state of the module.																
Inserted module	Text	Permissible module types	Type of the module actually inserted in the base plate.																
Configured module	Text	Permissible module types	Type of the module planned in the project currently loaded.																
Module type in project	Text	Permissible module types	Type of the module planned in SILworX.																
Connection status	Hexadeci- mal value	16#00...0F	Status of the connection between each of the processor modules (max. 4) and the module. Each of the bits 0..3 shows the connection to the processor module with the corresponding index. The value 1 of the bit means "connected" while the value 0 means "not connected ".																
Send status	Hexadeci- mal value	16#0000...F FFF	Every set of two bits represents the state of the interface with an index. Bits 0 and 1 apply to inter- face 0, and so on.																
Receive status			<table><tr><th>Value</th><th>Description</th></tr><tr><td>00</td><td>No message received/sent yet, unknown status</td></tr><tr><td>01</td><td>OK, no faults</td></tr><tr><td>10</td><td>Las reception/transmission was defective</td></tr><tr><td>11</td><td>No faults during last recep- tion/transmission, one fault oc- curred before</td></tr></table>	Value	Description	00	No message received/sent yet, unknown status	01	OK, no faults	10	Las reception/transmission was defective	11	No faults during last recep- tion/transmission, one fault oc- curred before						
Value			Description																
00			No message received/sent yet, unknown status																
01			OK, no faults																
10	Las reception/transmission was defective																		
11	No faults during last recep- tion/transmission, one fault oc- curred before																		
Module status	Hexadeci- mal value	16#00...3F	Bit-coded module status																
			<table><tr><th>Bit</th><th>Meaning with value = 1</th></tr><tr><td>0</td><td>Warning related to external communi- cation.</td></tr><tr><td>1</td><td>Warning related to field connection</td></tr><tr><td>2</td><td>System warning</td></tr><tr><td>3</td><td>External communication error</td></tr><tr><td>4</td><td>Field connection error</td></tr><tr><td>5</td><td>System error</td></tr><tr><td>6- 7</td><td>Not used</td></tr></table>	Bit	Meaning with value = 1	0	Warning related to external communi- cation.	1	Warning related to field connection	2	System warning	3	External communication error	4	Field connection error	5	System error	6- 7	Not used
Bit	Meaning with value = 1																		
0	Warning related to external communi- cation.																		
1	Warning related to field connection																		
2	System warning																		
3	External communication error																		
4	Field connection error																		
5	System error																		
6- 7	Not used																		
Status of system bus A	Hexadeci- mal value	16#0...3	Status of the interface to system bus A/B:																
Status of system bus B			<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>The interface is OK</td></tr><tr><td>1</td><td>The interface detected an error dur- ing last reception, now it is OK.</td></tr><tr><td>2</td><td>An error occurred on the interface.</td></tr><tr><td>3</td><td>The interface is switched off.</td></tr></table>	Value	Description	0	The interface is OK	1	The interface detected an error dur- ing last reception, now it is OK.	2	An error occurred on the interface.	3	The interface is switched off.						
Value			Description																
0			The interface is OK																
1	The interface detected an error dur- ing last reception, now it is OK.																		
2	An error occurred on the interface.																		
3	The interface is switched off.																		

Table 41: Diagnostic Information Displayed in the Online View for the Hardware Editor

TIP To convert hexadecimal values to bit strings a scientific calculator such as the Windows calculator with the corresponding view can be used.

8 Specifications, Dimensioning

Applicable per project	Value from ... to
Number of resources (controllers)	1...65 534
Per ressource	Value from ... to
Number of base plates	1...16
Number of I/O modules	0...200
Number of I/O elements (sensors, actuators)	0...6400
Maximum length of a system cable to FTA	30 m
Number of processor modules	1...4
Program memory for each user program	1023 kB
Data memory for each user program	1023 kB
Total program and data memory for all user programs	10 MB less 4 kB for CRCs
Memory for retain variable	
per user program	2 kB
A total for all user programs	32 kB
Number of variables	Depending on the variable type
Example type INTEGER (16 bits):	
Number of simple variables	523 776
Number of retain variables	1 024
Number of system bus modules, per base plate	1...2
Maximum length of system busses	100 m
Using Fiber optic cables (see Chapter 3.2.1)	19.6 km
Number of communication modules	0...20
Count of safe ^e thernet connections	0...255
Buffer of safe ^e thernet	
Connection to another HIMax controller.	1100 bytes
Connection to HIMatrix controller.	900 bytes
Connection buffer size to a X-OPC server.	16 kBytes
Number of user accounts	1...10
Number of user programs	1...32
Number of event definitions	0...20 000
Size of the non-volatile event buffer.	5000 Events
Parameter	Value from ... to
Length of the names defined by the user	1...31 characters
<ul style="list-style-type: none"> ▪ Username ▪ Password ▪ Project ▪ Resource ▪ Configuration 	

Table 42: Dimensioning of a HIMax Controller

Detailed specifications are provided in the manuals for the individual components and in the communication manual HI 801 101.

9 Life Cycle

This chapter describes the following life cycle phases:

- Installation
- Start-up
- Service and maintenance

Instructions for a correct decommissioning and disposal of the products are provided in the manuals for the individual components.

9.1 Installation

This chapter describes how to install and connect the HIMax controllers.

9.1.1 Mechanical Structure

To ensure a faultless operation, choose a suitable mounting location for the HIMax system in accordance with the operating requirements, see Chapter 2.1.3.

Observe the instructions for the installation of Base Plates and other components in the respective manuals.

9.1.2 Connecting the Field Zone to the I/O Module

HIMax is a flexible system designed for continuous operation. It allows the I/O modules to be connected to the field zone:

- directly, via the connector boards.
- indirectly, via the field termination assemblies.

The following section describes the four recommended wiring variants.

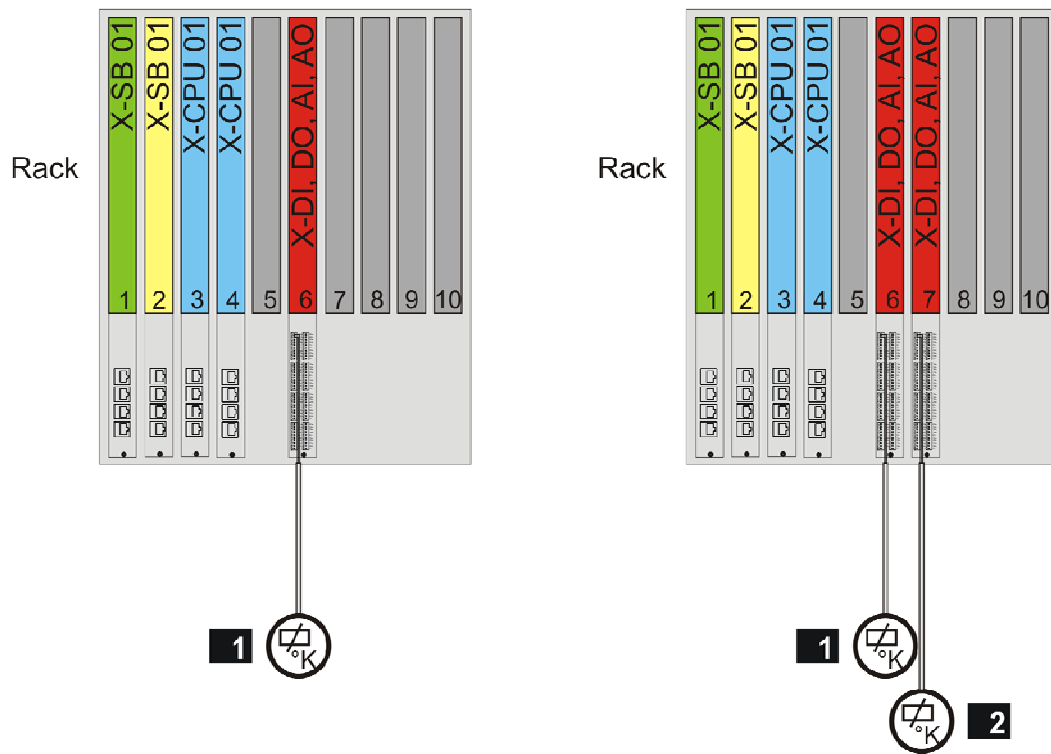
1. Connection to single connector boards with screw terminals
2. Connection to redundant connector boards with screw terminals
3. Connection to single connector boards via field termination assemblies and system cables.
4. Connection to redundant connector boards via field termination assemblies and system cables.

Additional wiring options require higher planning efforts and are not described in the manuals. If required, please contact HIMA Project Management & Engineering division.

Wiring 1

For an individual I/O module, connect the sensors or actuators to a non-redundant connector board with screw terminals.

- Connect individual sensors or actuators to an individual I/O module on a per channel basis (non-redundant).
- Connect two or more redundant sensors or actuators to two or more redundant modules on a per channel basis. The number of redundant sensors or actuators must be identical with the number of redundant modules (e.g., two sensors/two modules).



1 Sensor or Actuator

2 Redundant Sensor or Actuator

Figure 11: Wiring 1 - Single Connector Board with Screw Terminals

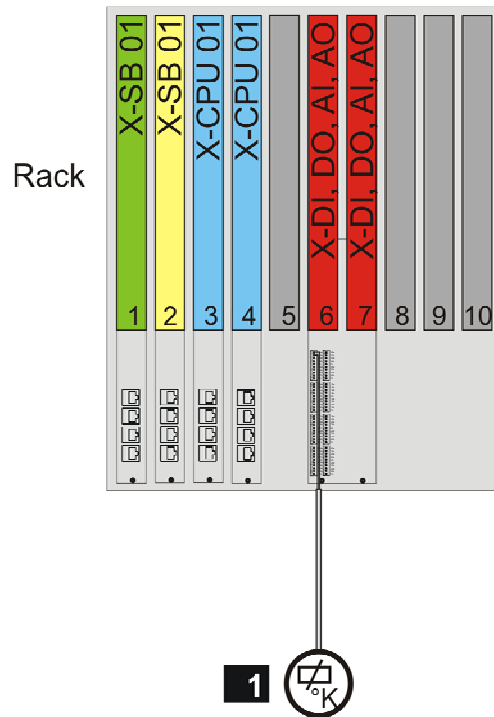
With wiring 1, connector boards of type 01, e.g., X-CB 008 01, are required in the base plate.

Wiring 2

Connect the sensors or actuators to a redundant connector board with screw terminals. The connector board distributes the signals from one sensor to two redundant modules or merges the signals from two redundant modules to one actuator.

For this wiring, the redundant system bus and the redundant power supply must be ensured.

- Connect the individual sensors or actuators on a per channel basis to a redundant connector board on which the I/O modules are mounted adjacently.



1 Sensor or Actuator

2

Figure 12: Wiring 2 - Redundant Connector Board with Screw Terminals

With wiring 2, connector boards of type 02, e.g., X-CB 008 02, are required in the base plate.

Wiring 3

Connect the sensors or actuators to a single connector board with cable plug via field termination assembly.

- Connect the individual sensors or actuators to a field termination assembly on a per channel basis.
- Connect two or more redundant sensors or actuators to two or more redundant field termination assembly on a per channel basis. Connect the field termination assembly to a single connector board via field termination assembly. The number of redundant sensors or actuators must be identical with the number of redundant modules (e.g., two sensors/two modules).

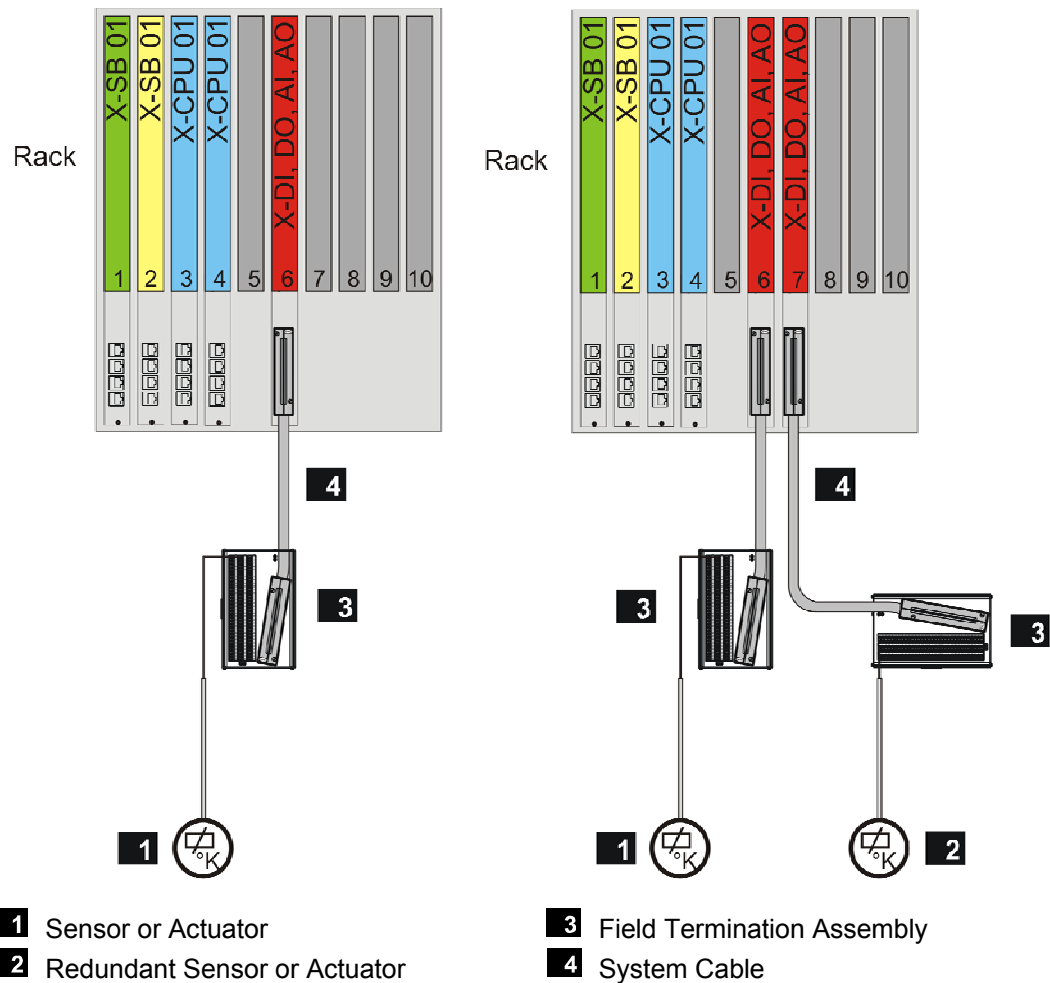


Figure 13: Wiring 3 - Single Connector Board with System Cable

With wiring 3, connector boards of type 03, e.g., X-CB 008 03, are required in the base plate.

Wiring 4

Connect the sensors or actuators to a redundant connector board with cable plug via field termination assembly and system cable. The connector board distributes the signal from one sensor to two redundant modules or merges the signals from two redundant modules to one actuator.

For this wiring, the redundant system bus and the redundant power supply must be ensured.

Connect the individual sensors or actuators to a redundant connector board with cable plug via field termination assembly. In doing so, insert the I/O modules into adjacent slots.

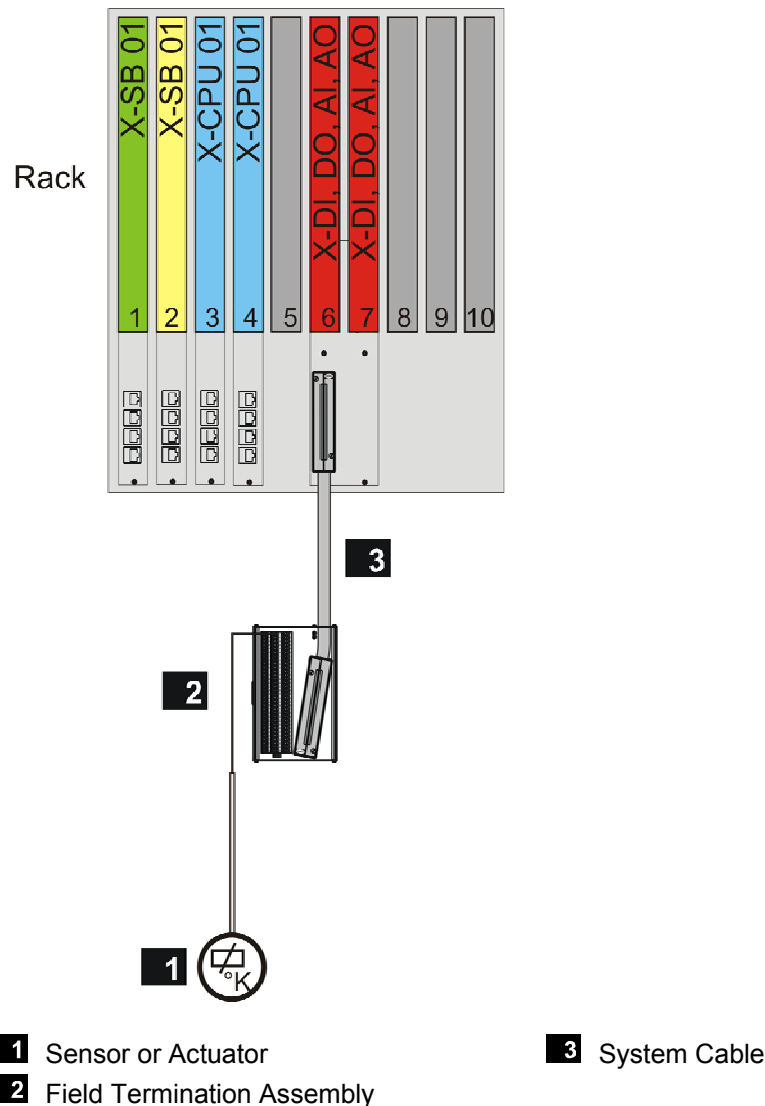


Figure 14: Wiring 4 - Redundant Connector Board with System Cable

For wiring 4, connector boards of 'type 04 (e.g., X-CB 008 04) are required on the base plate.

9.1.3 Earthing

Observe the requirements specified in the low voltage directives SELV (Safety Extra Low Voltage) or PELV (Protective Extra Low Voltage). A functional earth is prescribed to improve the electromagnetic compatibility (EMC). Perform this functional earth in the control cabinet so that it meets the requirements for protective earth.

All HIMax systems can be operated with earthed L- or unearthed.

Unearthed Operation

In unearthed operation, one single earth fault does not affect the safety and availability of the controller.

If several undetected earth faults occur, faulty control signals can be triggered. For this reason, earth fault monitoring must be used in unearthed operation (see also e.g., VDE 0116). Only use earth fault monitoring devices approved by HIMA.

Earthed Operation

Requirements for earthed operation are proper earth conditions and possibly separate earth connection in which no external power supply flows. Only earthing of negative pole L-

is permitted. Earthing of positive pole L+ is not permitted since a potential earth fault on the transmitter line would bypass the affected transmitter.

L- can only be earthed on one place within the system. L- is usually earthed directly behind the power supply unit (e.g., on the busbar). The earthing should be easily accessible and well separable. The earthing resistance must be $\leq 2 \Omega$.

Measures for Installing the Control Cabinet in Conformity with the CE Label

In accordance with the EU Council Directive 89/336/EEC, converted in the EMC law for the Federal Republic of Germany, from the 1st January 1996, all electric equipment within the European Union must be labeled with the CE conformity marking for electromagnetic compatibility (EMC).

All modules of the HIMA system family "HIMax" are labeled with the CE conformity marking.

When installing controllers in control cabinets and support frames, ensure proper and interference-free electrical installation in the vicinity of the controllers to prevent EMC problems. For instance, do not lay power lines together with 24 V feed lines.

Earthing the HIMA Controllers

While also taking the EMC aspects into account, implement the following earthing measures to ensure the safe function of HIMA controllers.

Earthing Connections

All tangible surfaces of the HIMax components (e.g., base plate), except for pluggable modules, are electrically conductive (ESD protection, ESD = electrostatic discharge). Use cage nuts with claw fasteners to ensure the safe electrical connection of components such as base plates and the control cabinet. The claw fasteners penetrate the pivoting frame surface **1** and ensure safe contact making. The screws and flat washers used are made of stainless steel to prevent electric corrosion **2**.

The components of the cabinet frame **3** are welded together and function as electrical conductive construction element. Short earthing straps with cross-sections of 16 mm² or 25 mm² are used to conductively connect the pivoting frames, door, mounting rails and, optionally, mounting plates to the cabinet frame. The earthing straps are covered with a yellow-green marking hose **5**. The earthing straps are covered with a yellow-green marking hose.

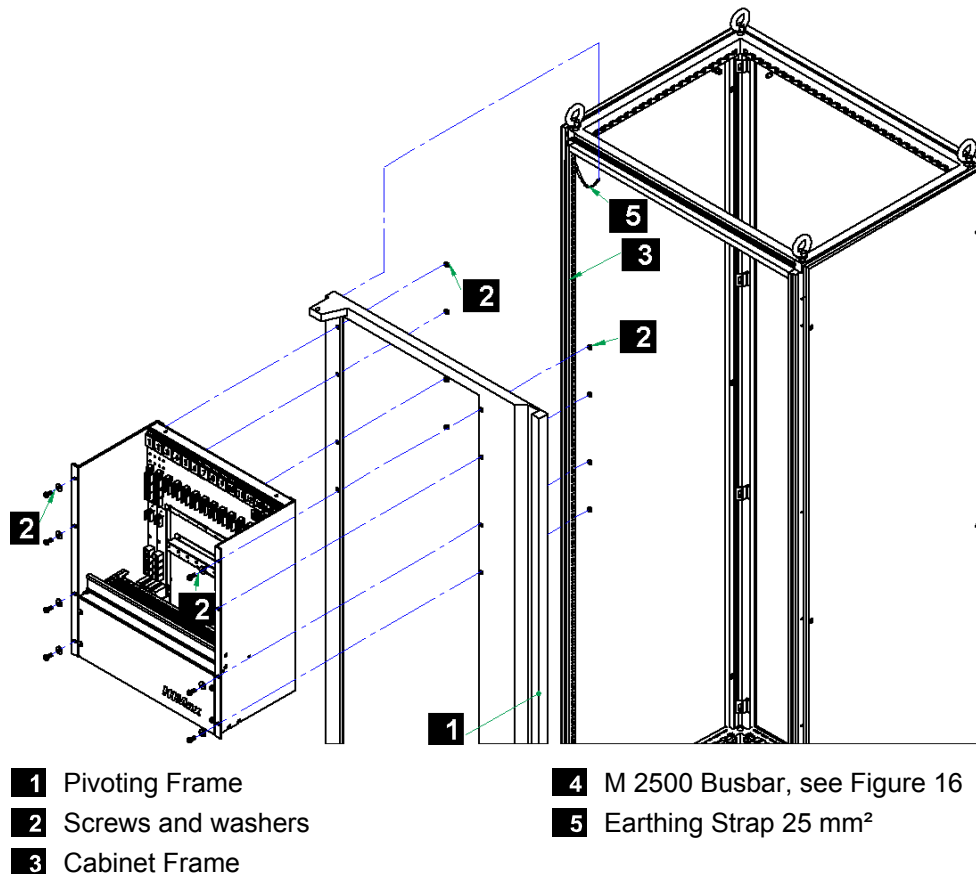
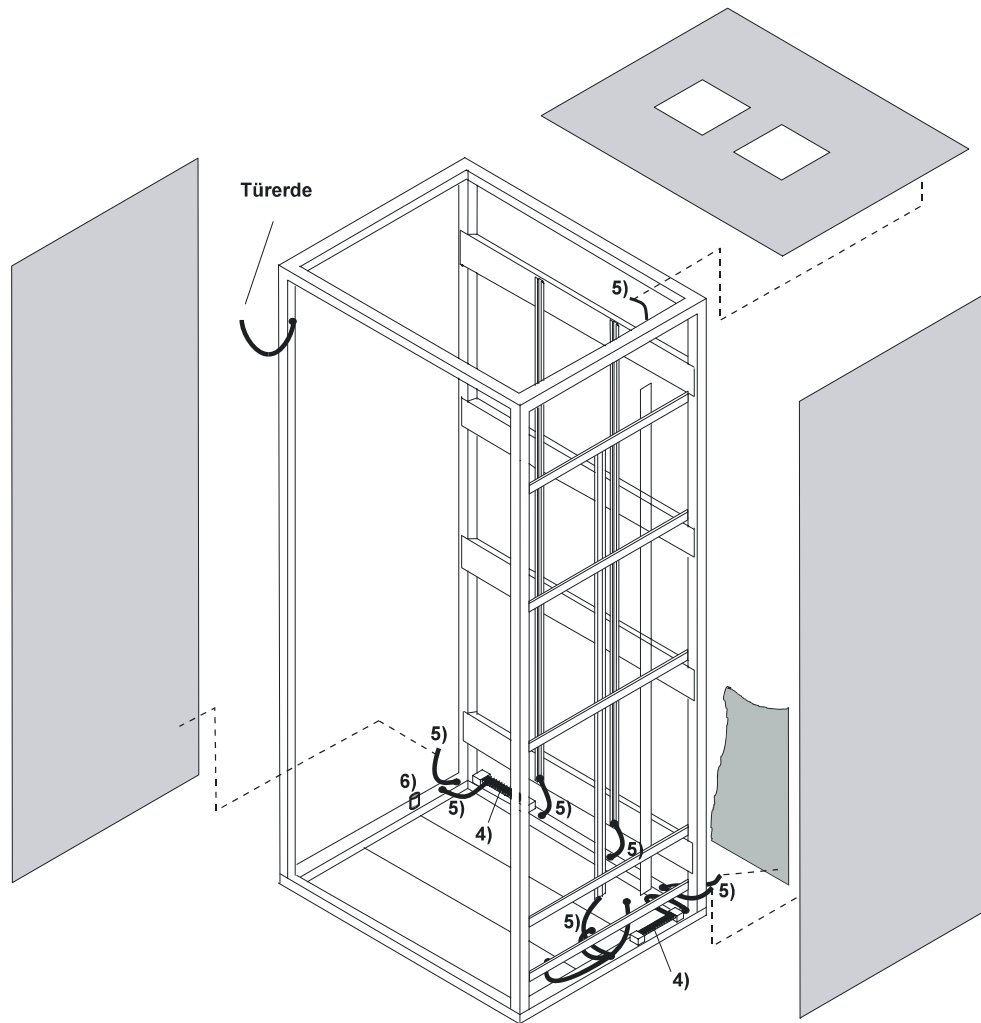


Figure 15: Earth Connections for Base Plate

The roof sheeting is secured to the cabinet frame with four lifting eyes **8** (see Figure 16). Earth claw fasteners **7** are used to conductively connect the side panels and the back-plane to the cabinet, screws are employed to conductively secure the floor panel to the cabinet frame.

Two M 2500 busbars **4** are installed in the cabinet by default and connected to the cabinet frame with 25 mm² earthing straps **5**. After removing the earthing straps, use the busbars **4** for a earth-isolated potential (e.g., for connecting the field cable shielding).

An M 8 bolt is located on the cabinet frame **6** to allow customers to connect the protective earth cable.



- | | |
|--|-------------------------|
| 4 M 2500 Busbar | 7 Earth Fastener |
| 5 Earthing Strap 25 mm ² | 8 Lifting Eye |
| 6 Cabinet Frame | 9 Door Earthing |

Figure 16: Earthing Connections in the Control Cabinet

Figure 17 shows the concept of earthing and shielding the 19" control cabinet.

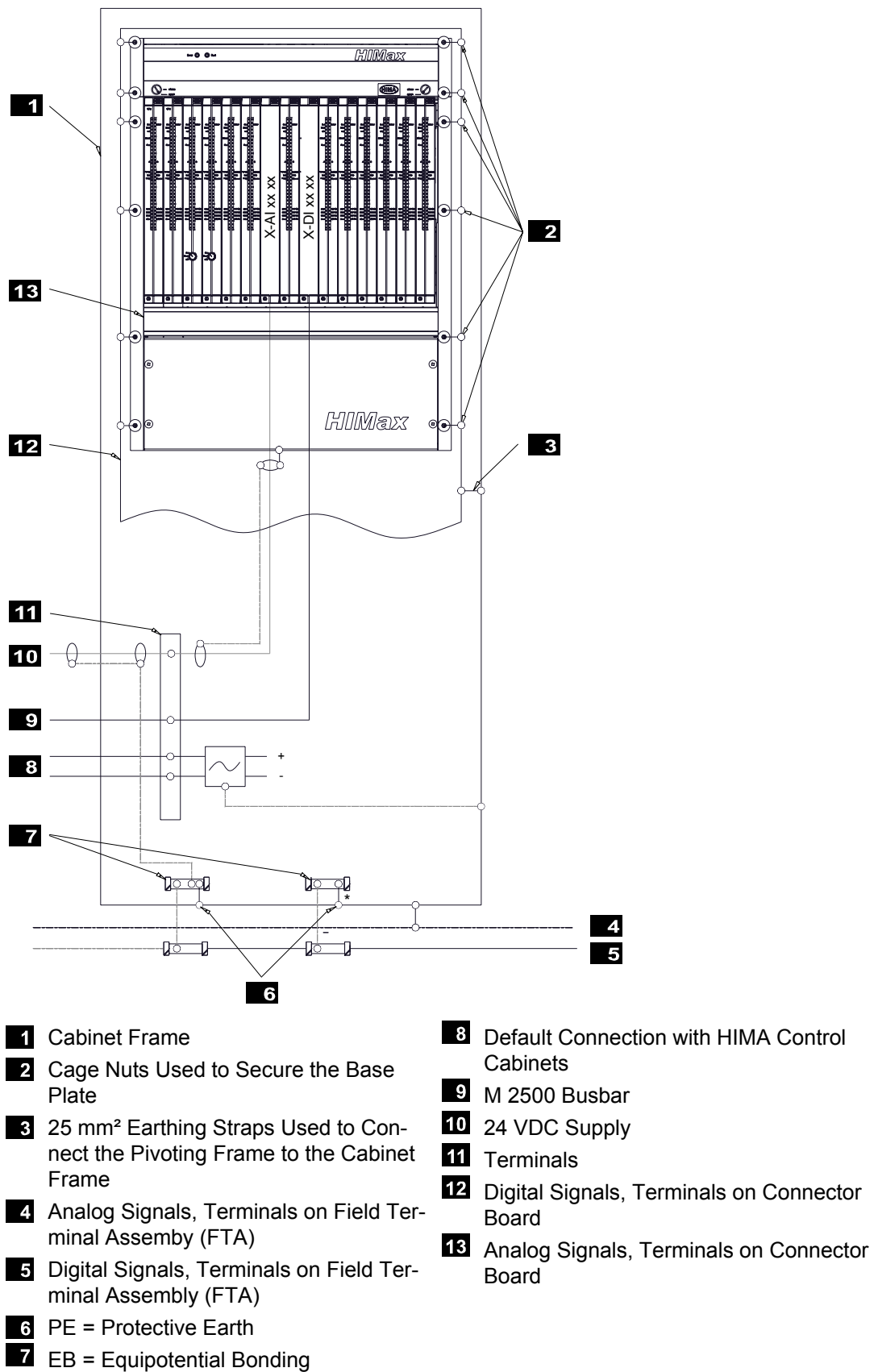


Figure 17: Earthing and Shielding the 19" Control Cabinet

Earthing Straps

The following table provides an overview of the dimensions of earthing straps:

Place of installation	Position in the figure	Cross-section	Length
Mounting rail (one-way with wire end fer- rule)	5 in Figure 16	16 mm ²	300 mm
Door	9 in Figure 16	16 mm ²	300 mm
Pivoting frame	1 in Figure 15	25 mm ²	300 mm
M 2500 busbar (one-way with wire end fer- rule)	4 in Figure 16	25 mm ²	300 mm

Table 43: Earthing Straps

The following elements are relevant for the earthing procedure:

- Fastener terminals (position **7** in Figure 16)
used on the side panels, backplane and floor panel
- Central earth point (position **6** in Figure 16)
- Lifting eyes (position **8** in Figure 16)
The roof sheeting is secured to the cabinet frame with four lifting eyes. Contact disks are used to perform the electrical connection.

Ensure proper connection of earthing straps!

Connecting the Earth Terminals of Several Control Cabinets

Central earth should possibly be free of interference voltage. If this cannot be ensured, provide the controller with its own earthing.

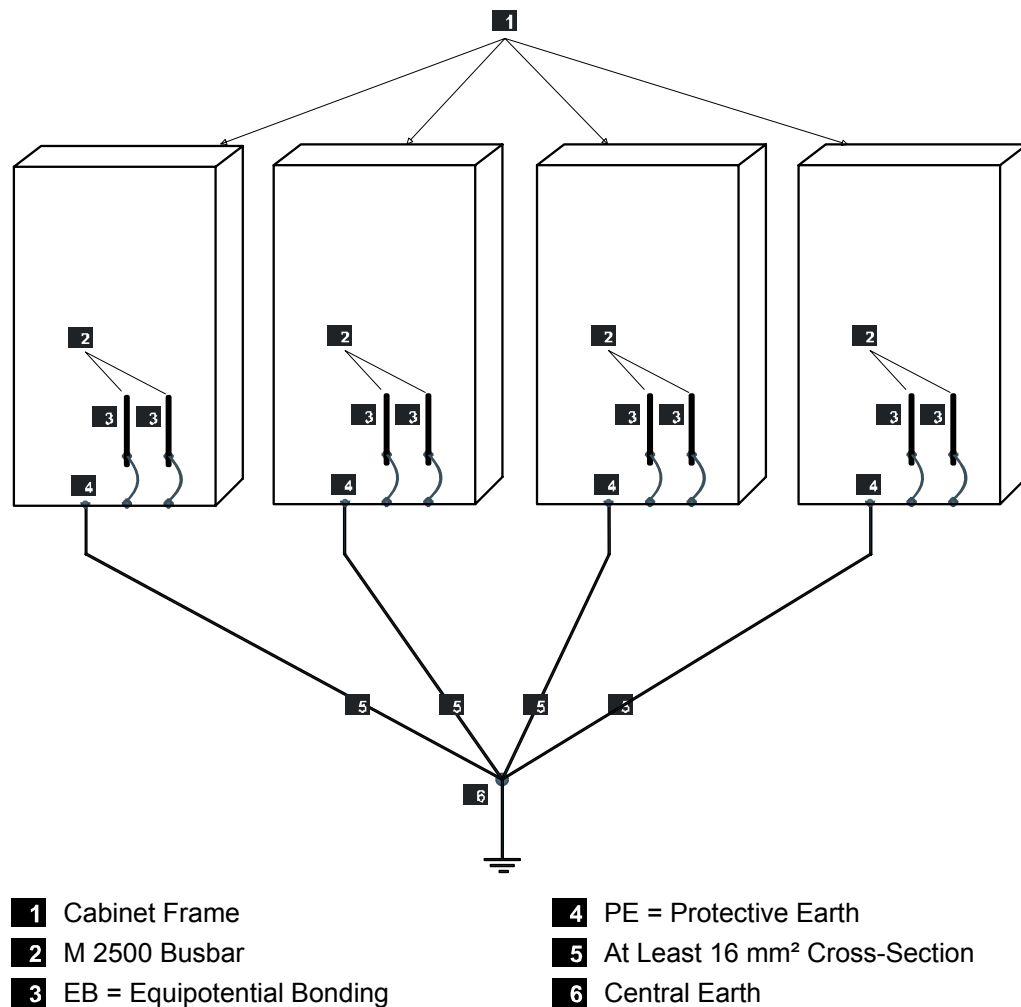


Figure 18: Earth Terminals of Various Control Cabinets

9.1.4 Electrical Connections

Shielding within the Input and Output Areas

Lay field cables for sensors and actuators separately from the power supply lines and sufficiently distant from electromagnetic active devices (electric motors, transformers).

Ensure that the cables to the input modules of the HIMax systems are laid as interference-free as possible, e.g., as shielded cables. This particularly applies to cables with analog signals and proximity switches.

For more information on shielding and earthing requirements, refer to the module manuals.

Lightning Protection for Data Lines in HIMA Communication Systems

To minimize problems due to lightning:

- Completely shield the field wiring of the HIMA communication systems
- Properly earth the system

Install lightning protection devices in places outside of buildings and exposed to lightning.

Cable Colors

The cable colors used in the HIMax devices comply with international standards.

Notwithstanding HIMA standard, other cable colors can be used for wiring due to national standard requirements. In such a case, please document and verify the deviations.

Connecting the Operating Voltage

Connect the operating voltage supply lines to the screw terminal connector blocks X1 and X2. When tightening the screw, make sure that the maximum locked torque of 2 Nm / 18 lbf in is not exceeded.

Attach the operating voltage supply lines of the system fan to the screw terminal connector blocks. When tightening the screw, make sure that the maximum locked torque of 4.52 Nm / 40 lbf in is not exceeded.

Connecting Field Devices and Shielding

With I/O modules, either attach the supply lines for the field devices to the screw terminal connector block of the Connector Boards or of the FTA. In doing so, observe the maximum locked torque of the screws specified in Table 44.

Module	Location	Locked torque [Nm]	Locked torque [lbf in]
X-BASE PLATE	X1	2.0	18
X-AI 32 01, X-AI 32 02	Connector board, screw couplings	0.26	2.25
X-DI 32 01, X-DI 32 04	Connector board, screw couplings	0.26	2.25
X-DI 32 02, X_DI 32 05	Connector board, screw couplings	0.26	2.25
X-DO 12 01	Connector board, screw couplings	0.51	4.5
X-DO 24 01	Connector board, screw couplings	0.26	2.25
X-FAN	X102	0.26	2.25
H 7201	XG13	4.5	40

Table 44: Locked Torque of the Screws for Connection to the Field Devices

To connect the field devices via FTAs, use the system cables intended for this use. Use the system cables to connect the FTAs and the corresponding connector boards.

i

The correct wiring depends on the application. Observe the following points when laying the wires:

- Correct wiring
- Cable/line bending radius
- Strain relief
- Cable/line load capacity

Connecting the Base Plates

To establish a - redundant - connection between the system busses of two base plates

1. Plug a RJ-45 connector of a patch cable in to the "UP socket located on the connector board of the left system bus module within the first base plate.
2. Plug the second RJ-45 connector of the same patch cable in to the DOWN socket located on the connector board of the left system bus module within the second base plate.
 - ☒ A non-redundant connection is established
3. Plug a RJ-45 connector of a second patch cable in to the UP socket located on the connector board of the right system bus module within the first base plate.

4. Plug the second RJ-45 connector of the same patch cable in to the "DOWN" socket located on the Connector Board of the right system bus module within the second base plate.

The two base plates are redundantly connected.

i

Patch cables colored or marked in a different way help avoiding mixing up cables, e.g., red cables for system bus A, green cables for system bus B

9.1.5 Mounting a Connector Board

Tools and utilities

- Screwdriver, slotted 0.8 x 4.0 mm
- Matching connector board

To install the connector board

1. Insert the connector board into the guiding rail with the groove facing upwards (see following figure). Fit the groove into the guiding rail pin.
2. Place the connector board on the cable shield rail.
3. Secure the two captive screws to the base plate. First screw in the lower than the upper screw.

To remove the connector board

1. Release the captive screws from the base plate.
2. Lift the lower section of the connector board from the cable shield rail.
3. Remove the connector board from the guiding rail.

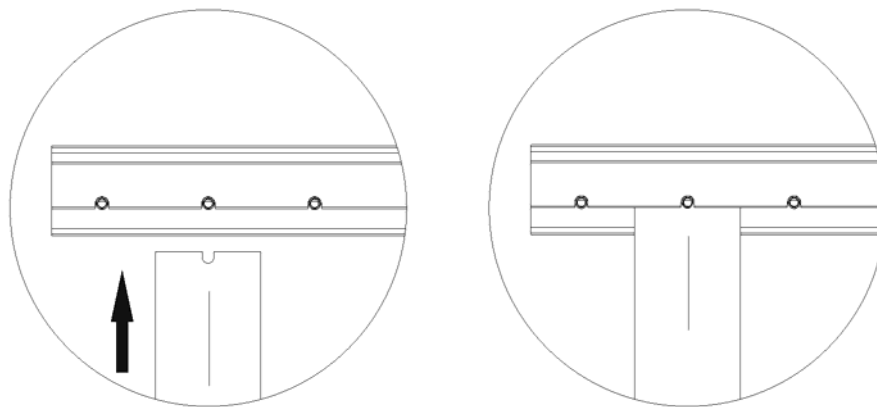


Figure 19: Inserting the Connector Board

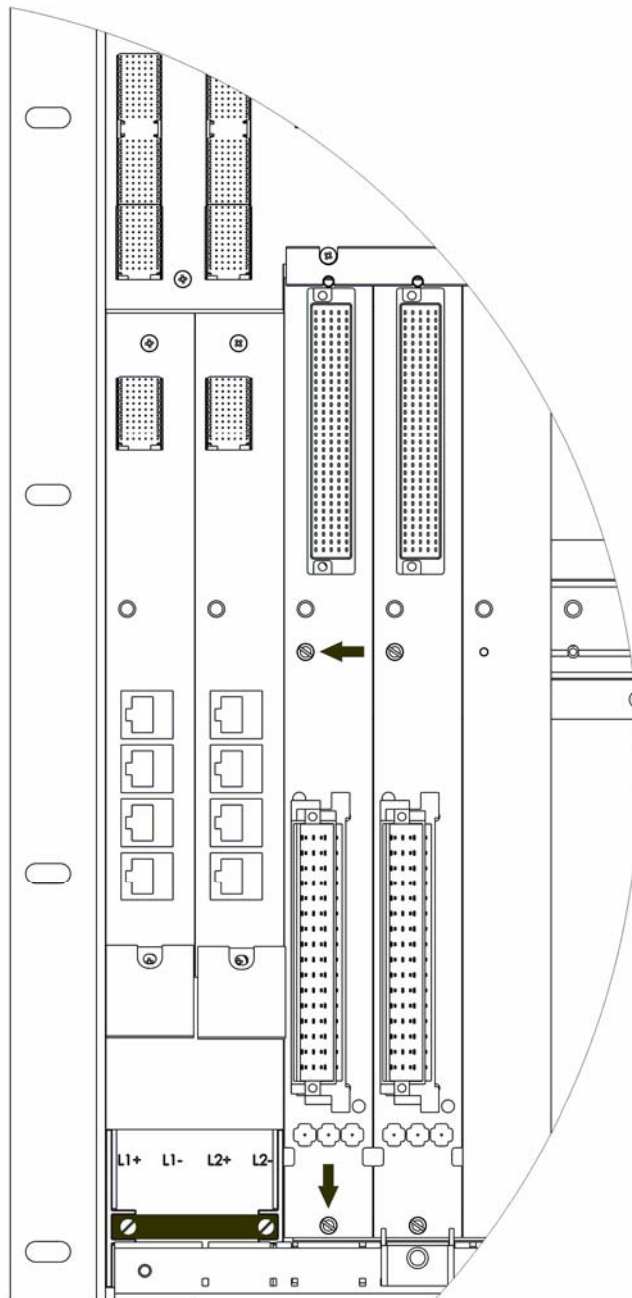


Figure 20: Securing the Connector Board with Captive Screws

9.1.6 Considerations about Heat

The increased integration level of electronic components causes a corresponding lost heat. This depends on the external load of HIMax modules. Depending on the structure, the device installation and ventilation are thus of importance.

Observe the environmental requirements when mounting the devices. Low operating temperature increases the product life and the reliability of the electronic components within the systems.

Heat Dissipation

A closed enclosure or a closed cabinet must be designed such that the heat generated inside can be dissipated through the surface.

Choose the mounting type and position such that heat dissipation is ensured.

The power dissipation of the installed equipment is decisive for determining the fan components. It is assumed that heat load and unhindered natural convection are uniformly distributed.

Definitions

Size	Description	SI-Unit	Engl. Unit
P_V	Power dissipation (heat capacity) of the electronic components within the device	W	Btu*ft
A	Effective enclosure surface (see below)	m ²	ft ²
B	Enclosure width	m	ft
H	Enclosure height	m	ft
T	Enclosure depth	m	ft
k	Heat transmission coefficient of the enclosure	W/m ² K	Btu ft/(h ft °F)
	Example of steel plate	~ 5.5 W/m ² K	~ 9.5 Btu ft/(h ft °F)

Table 45: Definitions for Calculating the Power Dissipation

Installation Type

In accordance with the mounting or installation type, the effective enclosure surface A is determined as follows:







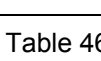
Enclosure installation type in accordance with VDE 0660, Part 5		Calculation of the enclosure surface A
	Individual enclosure, free-standing on all sides	$A = 1.8 \times H \times (W + D) + 1.4 \times W \times D$
	Individual enclosure for wall mounting	$A = 1.4 \times W \times (H + D) + 1.8 \times H \times D$
	Final enclosure, free-standing	$A = 1.4 \times D \times (W + H) + 1.8 \times W \times H$
	Final enclosure for wall mounting	$A = 1.4 \times H \times (W + D) + 1.4 \times W \times D$
	Central enclosure, free-standing	$A = 1.8 \times W + H + 1.4 \times W \times D + H \times D$
	Central enclosure for wall mounting	$A = 1.4 \times W \times (H + D) + H \times D$
	Central enclosure for wall mounting With covered roof area	$A = 1.4 \times W + H + 0.7 \times W \times D + H \times D$

Table 46: Installation Types

Natural Convection

When the natural convection is used, the lost heat is dissipated through the enclosure walls. Requirement: The ambient temperature must be lower than the temperature within the enclosure.

The maximum increase of temperature $(\Delta T)_{\max}$ of all electronic devices within the enclosure is calculated as follows:

$$(\Delta T)_{\max} = \frac{P_V}{k \cdot A}$$

The power dissipation P_V can be calculated based on the specifications for the electric power rating of the controller and its inputs and outputs.

Remark to the Standards

The temperature within a housing can also be calculated in accordance with VDE 0660, Part 507 (HD 528 S2).

i

All considerations about heat must take every component within a cabinet or enclosure into account, also components that are not directly part of the HIMax system!

Temperature State/Operating Temperature

The controllers are intended for operation up to a maximum temperature of 60 °C. The temperature states of the individual modules or controllers are centrally evaluated by the processor module.

A temperature sensor located on a specific temperature-relevant position independently detects and continuously monitors the temperature state on the corresponding module.

Use the *Temperature State* system variable in SILworX to evaluate the temperature states.

The *Temperature State* system variable indicates the operating temperatures measured in the following temperature ranges:

Ambient temperature, approx.	Temperature State	System variable values <i>Temperature State</i> [BYTE]
< 40 °C	Normal	0x00
40...60 °C	Threshold 1 exceeded	0x01
> 60 °C	Threshold 2 exceeded	0x03
Back to 54...44 °C	Threshold 1 exceeded	0x01
Back to 44 °C	Normal	0x00

Table 47: Temperature States

If a temperature sensor detects that the temperature exceeds a specific threshold or falls below it, the temperature state changes.

The temperature state is displayed with a temperature hysteresis of 6 K.

i

Under unfavorable operating conditions, the *Temperature State* system variable can even enter the "High Temperature" or "Very High Temperature" state at lower temperatures than those specified in in Table 47.

Example after a fan failure.

For each base plate, it is possible to define the temperature threshold that should cause a message when it is exceeded. In the SILworX Hardware Editor, use the Detail View for the base plate to configure this setting.

9.2 Start-Up

Only power up the system after the hardware is completely mounted and all the cables are connected. First start up the control cabinet, then the PES itself.

9.2.1 Starting-up the Control Cabinet

Prior to connecting the operating voltage, check if the connection can be performed without damaging the controller and system.

Testing all Inputs and Outputs for External Voltage

Impermissible external voltages (in particular with 230 V ~ against earth or L-) can be measured using an universal measuring instrument. HIMA recommends testing every single terminal for impermissible external voltages.

Testing all Inputs and Outputs for Earth Faults

When checking external cables for leakage resistance, potential short-circuits or breakage, the cables must not be connected on any end to prevent potential damage or destruction of modules caused by high voltages.

Unplug the voltage connection plugs from the power distributor and disconnect the supply voltages for sensors and the negative pole of actuators.

If the negative pole is earthed during operation, interrupt the earth connection for the duration of the earth fault check. The same applies to the earth connection of earth fault measuring equipment, which may be connected to the system. Use a megohmmeter or a special measuring facility to check each connection can against earth.

Voltage Connection

Requirement: The I/O module is inserted and the corresponding cable is connected. Check proper polarity, voltage and ripple prior to connecting the operating voltage 24 VDC.

9.2.2 Starting-up the PES

Requirements for start-up:

- The hardware is installed.
- The hardware is correctly configured - it is sufficient that base plates, system bus modules and processor modules are configured.
- The base plates have not been interconnected yet.
- The mode switches on all the processor modules are set to Init.
- The supply voltage is switched on (only switch on after the mode switches on the processor modules are set to Init).
- All remaining modules are in STOP.
- The PADT network connection is configured such that the modules of the HIMax base plate can be reached: if required, enter routing for the used interface card.

To start up the controller

1. Set the IP address and the SRS on the system bus module:
 - Establish a direct physical connection between PADT and system bus module.

i

The Ethernet interface *PADT* of the system bus module cannot perform an Auto-Cross-Over.

Use therefore a crossover cable to connect to the system bus module.

- System log-in to the resource branching "Hardware".
Abort the login window!
The *Online Hardware* tab opens.
 - In the *Online Hardware*, log in to the system bus module (double-click the system bus module, the module log-in window appears).
Use the MAC address (see the label on the module) to read the IP address and the SRS (button **Browse** in the log-in window).
 - Select **Set SRS** and **Module Network Settings** on the *Online -> Commissioning* menu to first set the SRS and then the IP address on the system bus module.
2. Repeat step 1 for all the system bus modules on all existing base plates.
 3. If the system contains more than one base plate, set the system bus module, slot 2, on base plate 0 or on base plate 1 to "responsible". If the system does not contain base plate 1, set the system bus module on base plate 0, slot 2 to "responsible".
 - Establish the direct physical connection between PADT and the system bus module in base plate 1 or base plate 0, slot 2.
 - Use the IP address and the SRS to log in to the system bus module

- Click **Set Responsible** on the *Online->Commissioning* menu to set this system bus module to "responsible".
4. Prepare the processor module in base plate 0, slot 3:
 - Establish a direct physical connection between PADT and processor module. Log in to the processor module: Double click the processor module symbol in the online figure.

i

If a valid configuration is loaded into a processor module and the conditions for system operation are met, all settings such as SRS and IP address from the valid configuration become operative. This is particularly important during the initial operation of a processor module that was previously used.

HIMA recommends: Reset to the factory settings (master reset) when using processor modules with an unknown past.

- Set the IP address and the SRS on the processor module.
With a mono system (one processor module and at least one system bus module), set the mono operation. To do so, click **Set Mono/Redundancy Operation** on the *Online->Start-up* menu.
This setting is only operative if a mono project is loaded. Otherwise, the system automatically resets the switch.
 - Set the mode switch of the processor module to Stop.
5. If required, interconnect the base plates.
 6. Log in to the system.
 - Establish a direct physical connection between PADT and processor module or system bus module.
 - Right click the resource , and then click **Online**.
 - Select the IP address of the module on the drop-down menu.
 - Enter the user name and password. Use **CTRL+A** to enter the default value Administrator with empty password.
 - Click **Log-in**.
 7. Set the mode switches of all remaining processor modules one after another to Stop
 8. Load the project into the processor module
 - Load the project configuration into the system: (menu **Online -> Resource Download**)
 - ☒ All processor modules enter the STOP/VALID CONFIGURATION state.
 9. Set the mode switches of all the processor modules to Run
 10. Start the system

The system, i.e., all modules, are in RUN (or in RUN / UP STOP, if the user program was not started).

For more information on how to start up the system, refer to the First Step Manual HI 801 103.

Faults

- A processor module starts the redundant operation or quits it, in case of malfunction.
- The system enters the STOP/INVALID CONFIGURATION state if the project in SILworX does not fit to the hardware.

9.2.3 Assigning the Rack ID

The identification number must be assigned to the base plates or modified if already existing, when assembling or extending the hardware.

The rack ID is stored in the connector board of the system bus module and must be modified using the system bus module. The system bus module distributes the rack IDs among the remaining modules on the base plate.

Whether the base plate and its modules are uniquely identified depends on the rack ID. On that, in turn, depends the identification of the inputs and outputs.

Observe this procedure since the rack ID is a safety-critical parameter

To assign the rack ID

1. Create the preconditions:
 - ☒ Ensure that all modules on the base plate are in STOP (to prevent the modules from exchanging old rack IDs).
 - ☒ Ensure that no connection exist between the PADT and processor.
 - ☒ Establish a direct connection between PADT and system bus module.
2. Change the rack ID:
 - Change the rack ID of one system bus module through the direct connection.
 - Also change the rack ID of a second system bus module (if existing) through the direct connection.

The new rack ID is valid. The configuration is consistent.

NOTE



Controller malfunction due to inconsistent rack IDs!

Since the rack ID is a safety-critical parameter, it may only be changed using the described procedure!

9.3 Maintenance and Repairs



For a safety-related application, the controller must be subjected to a proof test at regular intervals. For more information, refer to the Safety Manual HI 801 003.

HIMA recommends replacing the fans of the controllers at regular intervals.

NOTE



Malfunction due to electrostatic discharge!

Damage of the controller or electronic devices connected to it!

Only qualified personnel may perform maintenance actions to supply, signal and data lines. Implement ESD protection measures. Personnel must be electrostatically discharged prior to any contact with the supply or signal lines!

NOTE



In Ex applications, danger of explosion due to spark formation!

Spark formation possible by unplugging on-load connectors.

Do not unplug on-load connectors!

9.3.2 Disturbances

Disturbances in the processor modules cause the redundant processor module to assume the control task. If no redundant processor module is configured, the entire controller is shut-down.

The *Error* LED on the processor module indicates the existence of disturbances.

For possible reasons for the *Error* LED to light, refer to the X-CPU manual. In the Control Panel, execute the Resource command on the Online menu to switch off the Error LED.

All the modules automatically detect disturbances during operation and use the *Error* LED to indicate them on the module front plate.

Use SILworX to diagnose faults (except for communication faults) even if the controller is in the STOP state.

Prior to replacing an I/O module, check if disturbances exist on the external line and if the corresponding sensor or actuator is properly functioning.

9.3.3 Connecting the redundant Power Supply

Because of potential high currents, act with particular caution when connecting a redundant power supply during operation.

WARNING



Physical injury due to overheating possible when connecting a power supply unit!

Check proper polarity, prior to connecting connecting a redundant power supply unit during operation!

9.3.4 Repair

NOTE



Malfunction of the controller due to insufficient repair!

Only HIMA is authorized to repair a safety-related HIMax system or the modules contained in it.

In case of unauthorized intervention in the device, functional safety cannot be ensured and the warranty or certification lapses.

10 HIMax Documentation and Support

10.1 HIMax Documentation

The following documents are available:

Document	Document number	Content	File format
System manual	HI 801 001	This document!	PDF
Safety manual	HI 801 003	Safe use of the HIMax system	PDF
X-BASE PLATE	HI 801 025	Base plates	PDF
X-FAN	HI 801 033	System fan	PDF
X-CPU 01	HI 801 009	Processor module	PDF
X-COM 01	HI 801 011	Communication module	PDF
X-SB 01	HI 801 007	System bus module	PDF
X-AI 32 01	HI 801 021	Analog input module, 32 channels	PDF
X-AI 32 02 SOE	HI 801 055	Analog input module, 32 channels, sequence of events recording	PDF
X-AO 16 01	HI 801 111	Analog output module, 16 channels	PDF
X-CI 24 01	HI 801 113	Counter input module, 24 channels	PDF
X-DI 16 01	HI 801 057	Digital input module, 16 channels	PDF
X-DI 32 01	HI 801 015	Digital input module, 32 channels	PDF
X-DI 32 02	HI 801 017	Digital input module, 32 channels for proximity switches	PDF
X-DI 32 03	HI 801 059	Digital input module, 32 channels	PDF
X-DI 32 04 SOE	HI 801 051	Digital input module, 32 channels, sequence of events recording	PDF
X-DI 32 05 SOE	HI 801 053	Digital input module, 32 channels for proximity switches, sequence of events recording	PDF
X-DI 64 01	HI 801 093	Digital i module, 64 channels	PDF
X-DO 12 01	HI 801 023	Digital relay output module, 12 channels	PDF
X-DO 12 02	HI 801 099	Digital output module, 12 channels	PDF
X-DO 24 01	HI 801 019	Digital output module, 24 channels	PDF
X-DO 24 02	HI 801 095	Digital output module, 24 channels	PDF
X-DO 32 01	HI 801 097	Digital output module, 32 channels	PDF
X-FTA AI 32 01 01	HI 801 041	Field termination assemblies for the various modules	PDF
X-FTA DI 32 01 01	HI 801 035		PDF
X-FTA DI 32 02 01	HI 801 037		PDF
X-FTA DO 12 01 01	HI 801 047		PDF
X-FTA DO 24 01 01	HI 801 039		PDF
X-FTA 001 01	HI 801 115		PDF
X-FTA 001 02	HI 801 131		PDF
X-FTA 002 01	HI 801 117		PDF
X-FTA 002 02	HI 801 119		PDF
X-FTA 003 02	HI 801 121		PDF
X-FTA 005 02	HI 801 125		PDF
X-FTA 006 01	HI 801 127		PDF
X-FTA 006 02	HI 801 129		PDF
X-FTA 007 02	HI 801 133		PDF
X-FTA 008 02	HI 801 135		PDF
X-FTA 009 02	HI 801 137		PDF
First steps manual	HI 801 103	Introduction for planning HIMax controllers using SILworX	PDF
SILworX online help	-		CHM
Communication manual	HI 801 101	Communication protocols and their application	PDF

Table 48: Overview of the HIMax Documentation

10.2 HIMA Service, Training and Hotline

Deadlines and the extent of actions for commissioning, testing and modifying programs and HIMA control cabinets can be agreed upon with the service department.

HIMA holds training in accordance with the current seminar program for its software programs and PES hardware. The training are usually in-house at HIMA. Refer to the Internet page www.hima.com or contact HIMA for details about the current program and dates of the HIMA internal training.

Additionally, end customer training can be offered on-site. Special training can be tailored to customer-specific topics and provided upon request.

Important telephone numbers and e-mail addresses

HIMA Reception	Telephone	(+49) 06202 709 - 0
	Fax	(+49) 06202 709 - 107
	E-Mail	info@hima.com
HIMA Hotline	Telephone	(+49) 06202 709 - 255 (or 258)
	Fax	(+49) 06202 709 - 199
	E-Mail	hotline@hima.com

For questions about specific topics or to locate the appropriate HIMA contact person, please use the contact form provided on our website www.hima.com.

Appendix

Application Examples

This chapter provides examples of how to mount the HIMax systems. I/O modules and communication modules were not taken into account. They are plugged in to the remaining slots, if required.

If required, base plates with 15 or 18 slots can also be used instead of base plates with 10 slots as presented in the examples.

Small System

This redundant system is composed of one base plate and two processor modules. The base plate has rack ID 0.

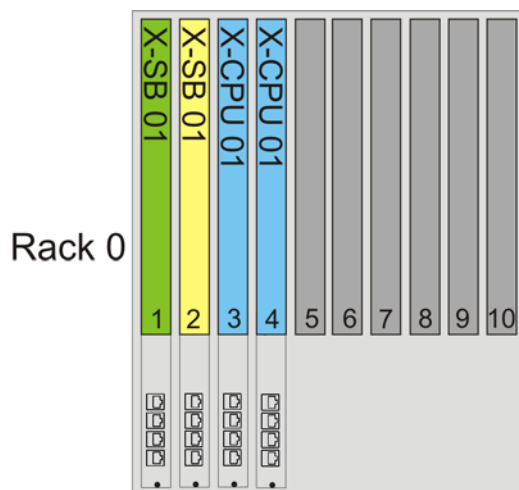


Figure 21: Small HIMax System: One Base Plate, Two Processor Modules

Minimum System

This system without redundancy represents the absolute modicum: One base plate 0, one processor module, one system bus module. Only one system bus A is used.

To ensure proper ventilation, a blank module must be used in slot 2. Slot 2 may not be used for I/O modules.

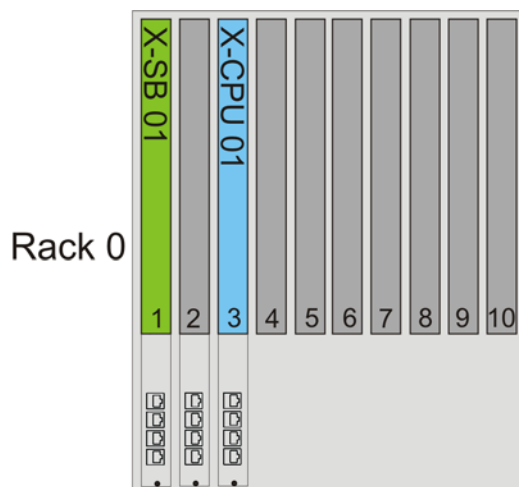


Figure 22: Minimum System without Redundancy

i

HIMA recommends using both system bus modules.

Distributed Redundancy

This system contains four redundant processor modules distributed on base plate 0 and base plate 1.

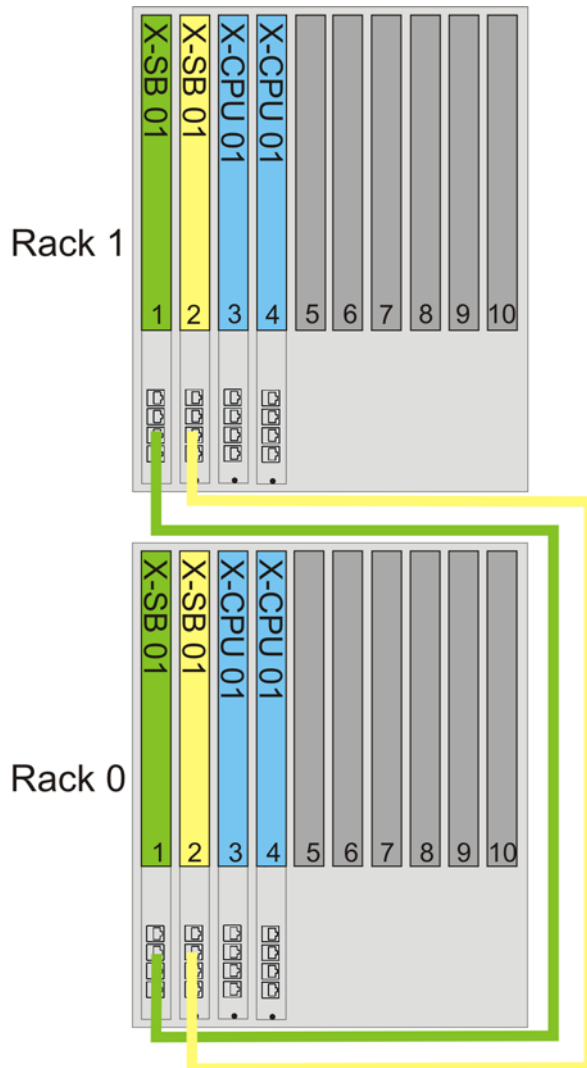


Figure 23: HIMax System with Distributed Redundancy

Glossary

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

Figure 1: System Overview	16
Figure 2: Base Plate Structure	18
Figure 3: Arrangement of Base Plates on the System Bus	21
Figure 4: HIMax System in its Maximum Extension	23
Figure 5: HIMax System with Maximum Distance between Processor Modules	24
Figure 6: Noise Blanking Performed by Safety-Related Inputs	30
Figure 7: CPU Cycle Sequence with Multitasking	52
Figure 8: Multitasking Mode 1	55
Figure 9: Multitasking Mode 2	56
Figure 10: Multitasking Mode 3	57
Figure 11: Wiring 1 - Single Connector Board with Screw Terminals	75
Figure 12: Wiring 2 - Redundant Connector Board with Screw Terminals	76
Figure 13: Wiring 3 - Single Connector Board with System Cable	77
Figure 14: Wiring 4 - Redundant Connector Board with System Cable	78
Figure 15: Earth Connections for Base Plate	80
Figure 16: Earthing Connections in the Control Cabinet	81
Figure 17: Earthing and Shielding the 19" Control Cabinet	82
Figure 18: Earth Terminals of Various Control Cabinets	84
Figure 19: Inserting the Connector Board	86
Figure 20: Securing the Connector Board with Captive Screws	87
Figure 21: Small HIMax System: One Base Plate, Two Processor Modules	97
Figure 22: Minimum System without Redundancy	97
Figure 23: HIMax System with Distributed Redundancy	98

Index of Tables

Table 1: Standards for EMC, Climatic and Environmental Requirements	11
Table 2: General requirements	11
Table 3: Climatic Requirements	11
Table 4: Mechanical Tests	12
Table 5: Interference Immunity Tests	12
Table 6: Interference Immunity Tests	12
Table 7: Noise Emission Tests	12
Table 8: Review of the DC Supply Characteristics	13
Table 9: Identifying a Module using the System.Rack.Slot	25
Table 10: Slot Positions Recommended for Processor Modules	26
Table 11: Operating System States, Adopting the States	28
Table 12: Operating System States, Possible User Interventions	29
Table 13: Types of Variables	36
Table 14: System Variables at Different Project Levels	36
Table 15: System Parameters of the Resource	38
Table 16: Hardware System Variables for Setting the Parameters	39
Table 17: Hardware System Variables for Reading the Parameters	42
Table 18: Assigning the Index to Processor Module Slots	42
Table 19: System Parameters of the User Program	43
Table 20: Parameters for Boolean Events	47
Table 21: Parameters for Scalar Events	49
Table 22: Parameters Configurable for Multitasking	53
Table 23: Reloading after Changes	59
Table 24: Module Order while Loading the Operating System	60
Table 25: Authorization Types for the PADT User Management Scheme	61
Table 26: Parameters for User Accounts in the PES User Management Scheme	63
Table 27: Blinking Frequencies of LEDs	64
Table 28: Assignment of the LED Groups to the Types of Modules	64
Table 29: Module Status Indicators	65
Table 30: Redundancy Indicators	65
Table 31: System Bus Indicators	66
Table 32: Rack Connection Indicators	66
Table 33: Slot Indicators	66
Table 34: Fault Indicators	67
Table 35: Maintenance Indicators	67
Table 36: I/O Indicators LEDs	67
Table 37: Fieldbus Indicators	68
Table 38: Ethernet Indicators	68

Table 39: Communication Indicators	69
Table 40: Maximum Number of Entries Stored in the Diagnostic History per Module Type	70
Table 41: Diagnostic Information Displayed in the Online View for the Hardware Editor	71
Table 41: Dimensioning of a HIMax Controller	73
Table 43: Earthing Straps	83
Table 44: Locked Torque of the Screws for Connection to the Field Devices	85
Table 45: Definitions for Calculating the Power Dissipation	88
Table 46: Installation Types	88
Table 47: Temperature States	89
Table 48: Overview of the HIMax Documentation	95

Index

alarm (see event).....	30	protocols	32
analog inputs		lightning protection	85
use	45	loading the configuration	
analog outputs		download	59
use	46	reload.....	59
base plate types	17	loading the operating system	60
blank module	18	maintenance.....	93
counter inputs		module status indicators	66
use	46	noise blanking	29
'de-energize to trip' principle'	10	operating requirements	
diagnosis	65	climatic.....	11
Ethernet indicators	69	EMC.....	12
fault indicators.....	68	ESD protection	13
fieldbus indicators	69	mechanical	12
history	70	power supply	13
maintenance indicators	68	PADT user management.....	62
rack connection indicators	67	PES user management.....	62
redundancy indicators.....	66	programming	35
slot indicators.....	67	rack ID	
system bus indicators	67	assigning	92
digital inputs		redundancy	33
use	44	communication	34
digital outputs		I/O modules	33
use	46	power supply	34
disturbances	94	processor module.....	33
earthing.....	79	system bus	34
'energize to trip' principle'	10	SILworX.....	35
event		spare module	33
definition.....	47	start-up	
in general	30	control cabinet	90
recording.....	31	system bus	20
forcing.....	50	extension	22
heat dissipation.....	89	training	97
initial value.....	36	user account.....	62
installation.....	75	user group.....	62
licensing		user management	62

HI 801 001 E

© 2009 HIMA Paul Hildebrandt GmbH + Co KG

HIMax and SILworX are registered trademark of:

HIMA Paul Hildebrandt GmbH + Co KG

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