Manual

HIQuad® X

System Manual

All the current manuals can be obtained upon request by sending an e-mail to: documentation@hima.com.

Contact

HIMA Paul Hildebrandt GmbH

P.O. Box 1261

68777 Brühl

Phone: +49 6202 709-0

Fax: +49 6202 709-107

E-mail: info@hima.com

Document designation	Description
HI 803 210 D, Rev. 2.00 (1925)	German original document
HI 803 211 E, Rev. 2.00.00 (1927)	English translation of the German original document

Table of Contents

1 Introduction

The system manual describes the configuration and mode of operation of the safety-related HIQuad X controller system. The system is designed for safety-related applications up to SIL 3 (IEC 61508), PL e (EN ISO 13849) and for high availability.

HIQuad X can be used for various control tasks within the process and factory automation industry, in particular in process facilities.

1.1 Structure and Use of the Document

This system manual is composed of the following main chapters:

Introduction	Introduction to this manual.
Safety	Information on how to safely use the HIQuad X system.
Concept for HIQuad X	Concept of the innovative high-performance HIQuad X system.
Product description	Structure of the HIQuad X system.
Redundancy	Options for increasing availability.
Programming	Important instructions on how to create a user program.
Diagnostics	Summary of the diagnostic options.
Product data, dimensioning.	Specifications related to the entire system. Specifications for the individual components are included in the corresponding manual.
Lifecycle	Phases of a HIQuad X system lifecycle: ▪ Installation. ▪ Start-up. ▪ Maintenance and repairs.
HIQuad X Documentation	Overview of the documentation:
Appendix	▪ Glossary. ▪ Index of tables and index of figures. ▪ Index.

1.2 Target Audience

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

All specialist staff (planning, installation, start-up) must be instructed concerning the risks and the associated possible consequences which can arise as a result of modifications to a safety-related automation system.

Planners and configuration engineers must have additional knowledge about the selection and use of electrical and electronic safety systems in automated plants, 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 Writing Conventions

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

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

Safety notices and operating tips are specially marked.

1.3.1 Safety Notices

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

The safety notices are represented as described below.

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

The signal words have the following meanings:

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

⚠ SIGNAL WORD

Type and source of risk!

Consequences arising from non-observance.

Risk prevention.

NOTICE

Type and source of damage!

Damage prevention.

1.3.2 Operating Tips

Additional information is structured as presented in the following example:

i	The text giving additional information is located here.

Useful tips and tricks appear as follows:

TIP	The tip text is located here.

1.4 Safety Lifecycle Services

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

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

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

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

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

2 Safety

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

For further information on safety, observe the instructions provided in the HIQuad X safety manual (HI 803 209 E).

2.1 Intended Use

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

The automation system is designed for the industrial process market to control and regulate process plants as well as for factory automation plants. SILworX, HIMA's programming tool, is used for programming, configuring, monitoring, operating and documenting the HIQuad X system.

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

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

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

2.1.2 Application in Accordance with the Energize to Trip Principle

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

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

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

If the components are defective, the de-energized state is entered irrespective of whether the system is operating in accordance with the energized to trip or with the de-energized to trip principle.

2.1.3 Use in Fire Alarm Systems

The HIQuad X systems with analog inputs are tested and certified for use in fire alarm systems in accordance with DIN EN 54-2 and NFPA 72.

The conditions of use provided in the safety manual (HI 803 209 E) must be observed.

2.1.4 Explosion Protection

The HIQuad X automation system is suitable for mounting in zone 2.

The special conditions provided in the HIQuad X safety manual (HI 803 209 E) must be observed.

2.2 ESD Protective Measures

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

NOTICE

Damage to the HIQuad X system due to electrostatic discharge!

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

2.3 Residual Risk

No imminent risk results from a HIMA system itself.

Residual risk may result from:

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

2.4 Safety Precautions

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

2.5 Emergency Information

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

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

3 Concept for HIQuad X

HIQuad X is an innovative, high-performance automation system that is based on the existing HIQuad system. SILworX, HIMA's tried and tested programming tool, is used for programming, configuring, monitoring, operating and documenting HIQuad X. All HIMA programmable systems are thus future-proof and operated with just one programming tool. Additionally, HIQuad X supports already existing HIQuad I/O modules.

H51X and H41X, the HIQuad X system families, can be equipped with identical modules and have the following differences:

	HIQuad H51X	HIQuad H41X
Structure	Modular	Modular
Base rack	1 (without I/O modules)	1 (with a maximum of 12 I/O modules)
Extension rack	Max. 16	Max. 1
Processor module (F-CPU 01)	1 or 2	1 or 2
I/O processing module (F-IOP 01)	1 in each extension rack	1 in the base rack 1 in the extension rack
Communication module (F-COM 01)	A maximum of 10 in the base rack	A maximum of 2 in the base rack
I/O modules in each extension rack	16	16
Total number of I/O modules	Max. 256	Max. 28

Table 1: Differences of HIQuad H51X Compared to H41X

The H51X and H41X system families can be equipped with digital and analog I/O modules. For details, refer to Chapter 4.9.

The I/O processing module (F-IOP 01) uses the I/O bus to interconnect the I/O modules within one rack. The F-IOP 01 module safely communicates with the processor modules via one or two system buses, see Figure 2 and Figure 4.

3.1 Safety and Availability

The HIQuad X systems are designed for safety-related applications operating in accordance with the energize to trip and de-energize to trip principles up to SIL 3 in accordance with IEC 61508. Additionally, the HIQuad X system complies with the standards specified in the certificates.

Refer to the HIQuad X safety manual (HI 803 209 E) and certificates for the standards used to test the HIQuad X system.

For safety-related application up to SIL 3, the base racks must be equipped with the safety-related processor modules (F-CPU 01). The safety-related processor module (F-CPU 01) features a 1oo2 processor system. The 1oo2 processor system includes two microprocessors that continuously align their data.

Safety-related HIQuad I/O modules can be used to connect to the field level, see Chapter 4.9. I/O modules and processor modules exchange data via safety-related I/O processing modules (F-IOP 01) with integrated 1oo2 processor system.

Additional non-safety-related I/O modules can be used for non-safety-related applications.

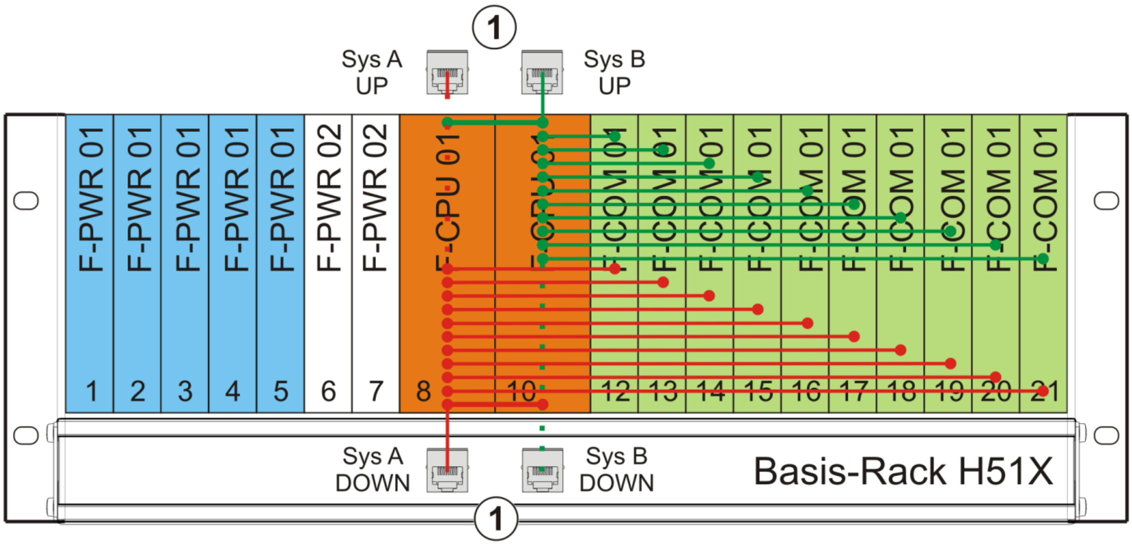
Depending on the required safety and availability, the H51X and H41X system families can be structured as mono or redundancy systems, see the following chapters.

3.2 Concept for HIQuad H51X

The H51X system family has a modular structure which includes an H51X base rack and up to 16 extension racks. The H51X base rack (F-BASE RACK 01) can be equipped as shown in Figure 1.

The communication modules are connected to the processor modules via two system buses (A and B) in a point-to-point connection. The processor module in slot 8 controls and monitors system bus A whereas the processor module in slot 10 controls and monitors system bus B. During redundant operation, the two processor modules align their data.

The RJ-45 interfaces on the rear side of the base rack are used to connect the extension racks to the processor modules. An I/O processing module (F-IOP 01) must be used in the extension rack to connect the system buses to the I/O bus, see Figure 2 and Figure 4.



1 System bus interfaces on the rear side of the base rack.

Figure 1: H51X Base Rack Completely Assembled

3.2.1 The H51X Mono System

Thanks to the use of safety-related modules (I/O modules, the I/O processing module and a processor module), the HIQuad H51X system can ensure safety-related signal processing in accordance with SIL 3 already when operating in a mono structure, see Figure 2.

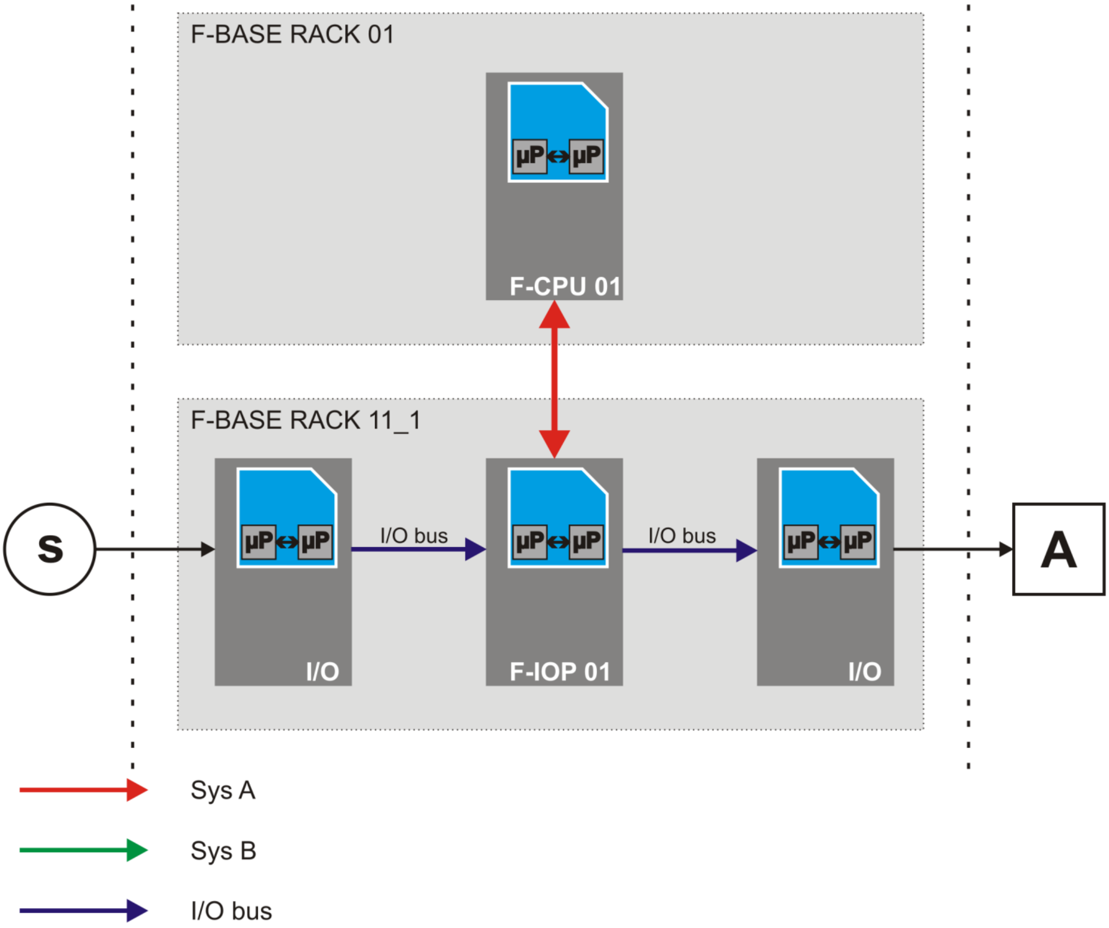


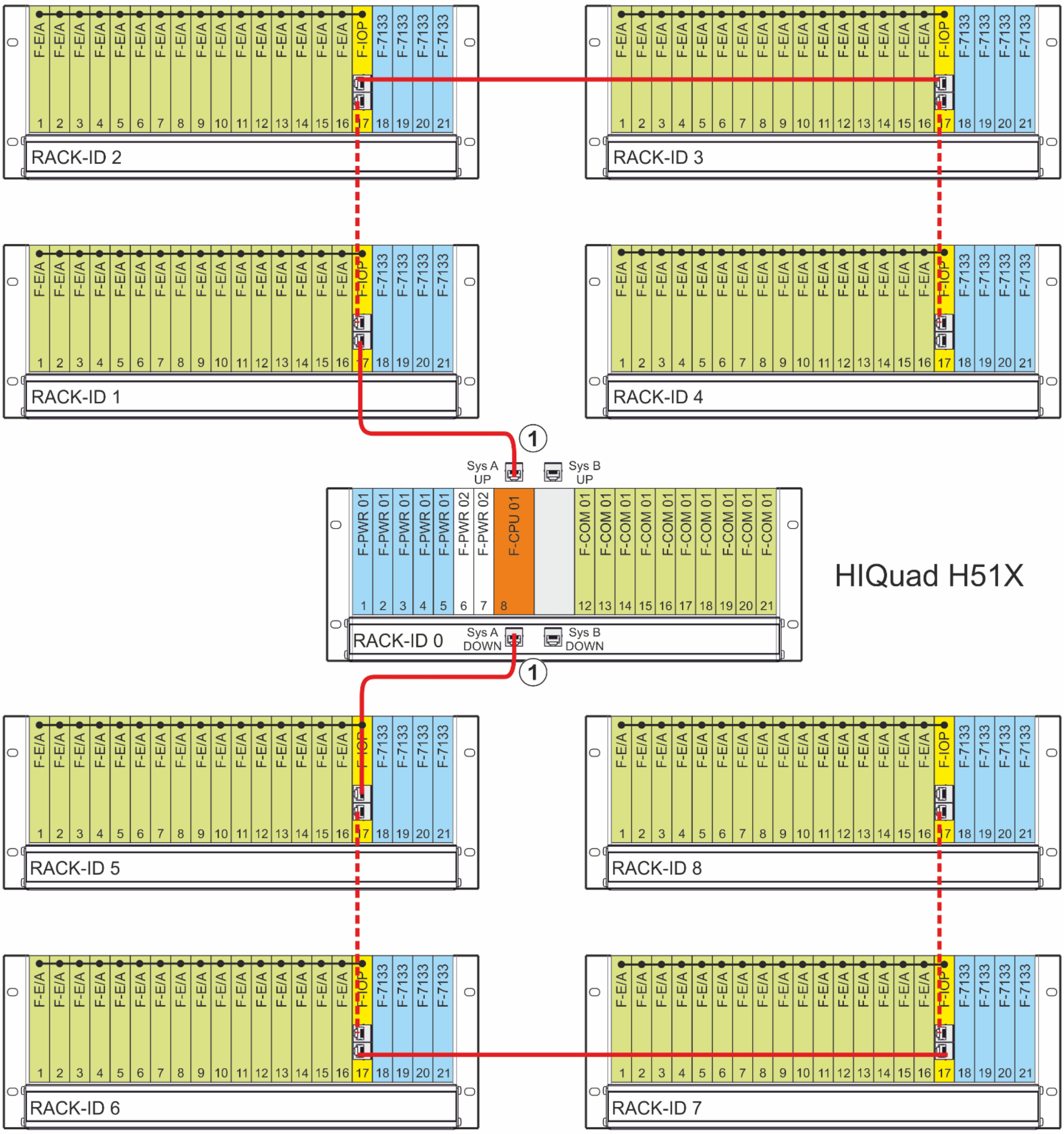
Figure 2: Example of Safe H51X Mono Operation (1oo2)

The input modules of the HIQuad H51X system safely record the values measured by sensors. Data is exchanged with the processor module via the I/O processing module. The measured values are cyclically queried by the processor module and processed by the user program. The user program's results are sent to the I/O processing module, which writes them to the output modules. The output modules thus control the field level, e.g., the actuators.

During mono operation, the signal is forwarded by the processor module in slot 8 via system bus A.

Figure 3 shows the example of an H51X mono system with system bus A. Up to 16 extension racks can be connected to the system bus in any UP loop, DOWN loop, or UP and DOWN loop. The extension racks are interconnected with system bus A via the I/O processing module, see the F-IOP 01 manual (HI 803 219 E).

If the system bus connection is interrupted in a mono system, all I/O modules located after the interruption point are no longer available. After the interruption point, all output modules enter the safety-related, de-energized state. As for the input modules, the failsafe initial values are processed in the respective processor module.



1 System bus interfaces on the rear side of the base rack.

Figure 3: Example of H51X Mono System

The rack IDs do not necessarily have to be arranged as described above, but they must be unique.

To ensure a clearer overview, HIMA recommends the following:

- Arrange the rack IDs in accordance with Figure 3.
- Use red patch cables for system bus A if only system bus A is used.

3.2.2 The H51X Redundancy System

During redundant operation with two processor modules, both system buses are used to process the signals. This variant with redundant processor modules and system buses increases the system's availability, see Figure 4. If a processor module fails, it automatically enters the safe state and the redundant processor module maintains safe operation. The faulty processor module must be replaced to ensure continued availability. The processor module can be replaced while the system is operating.

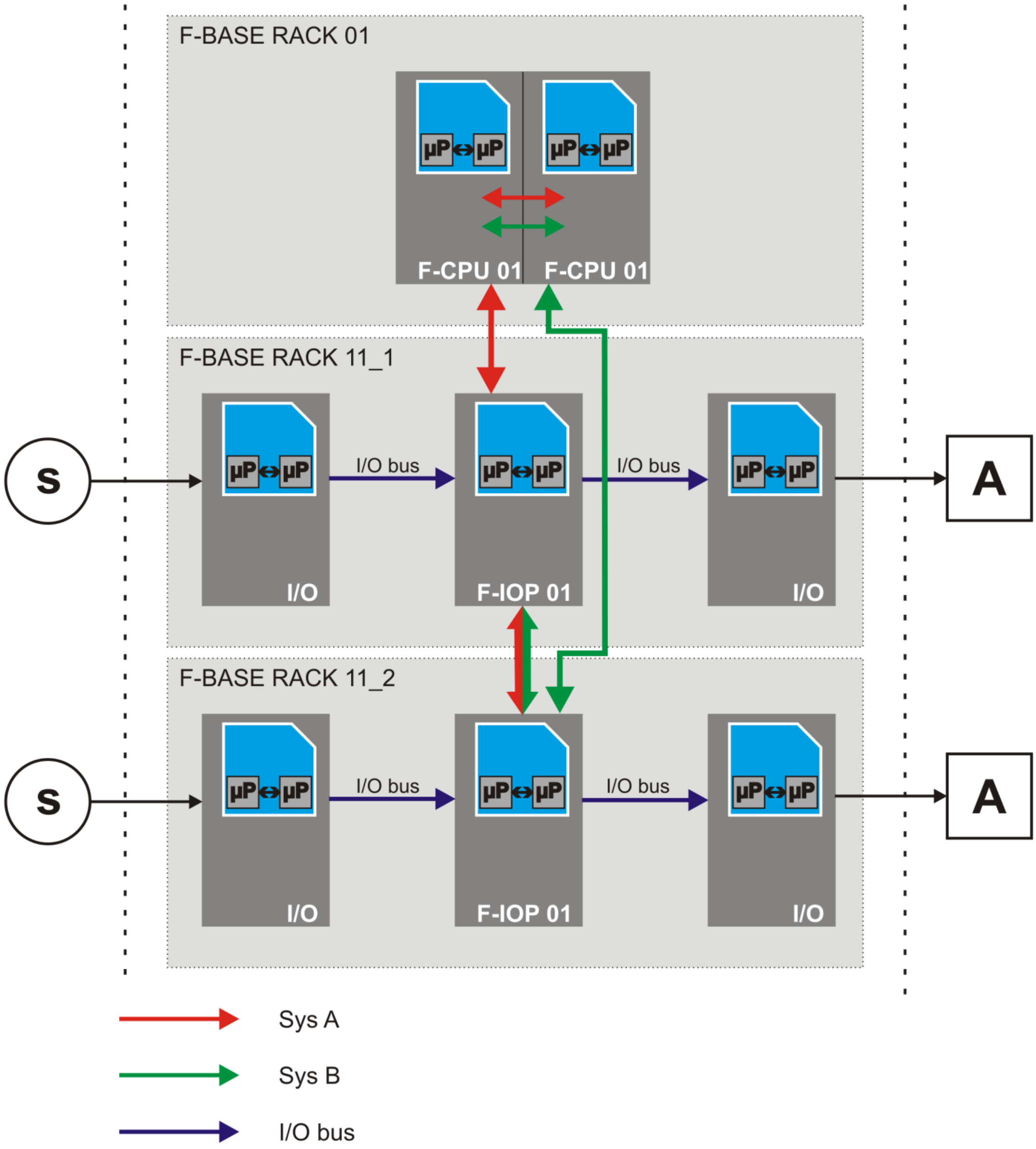


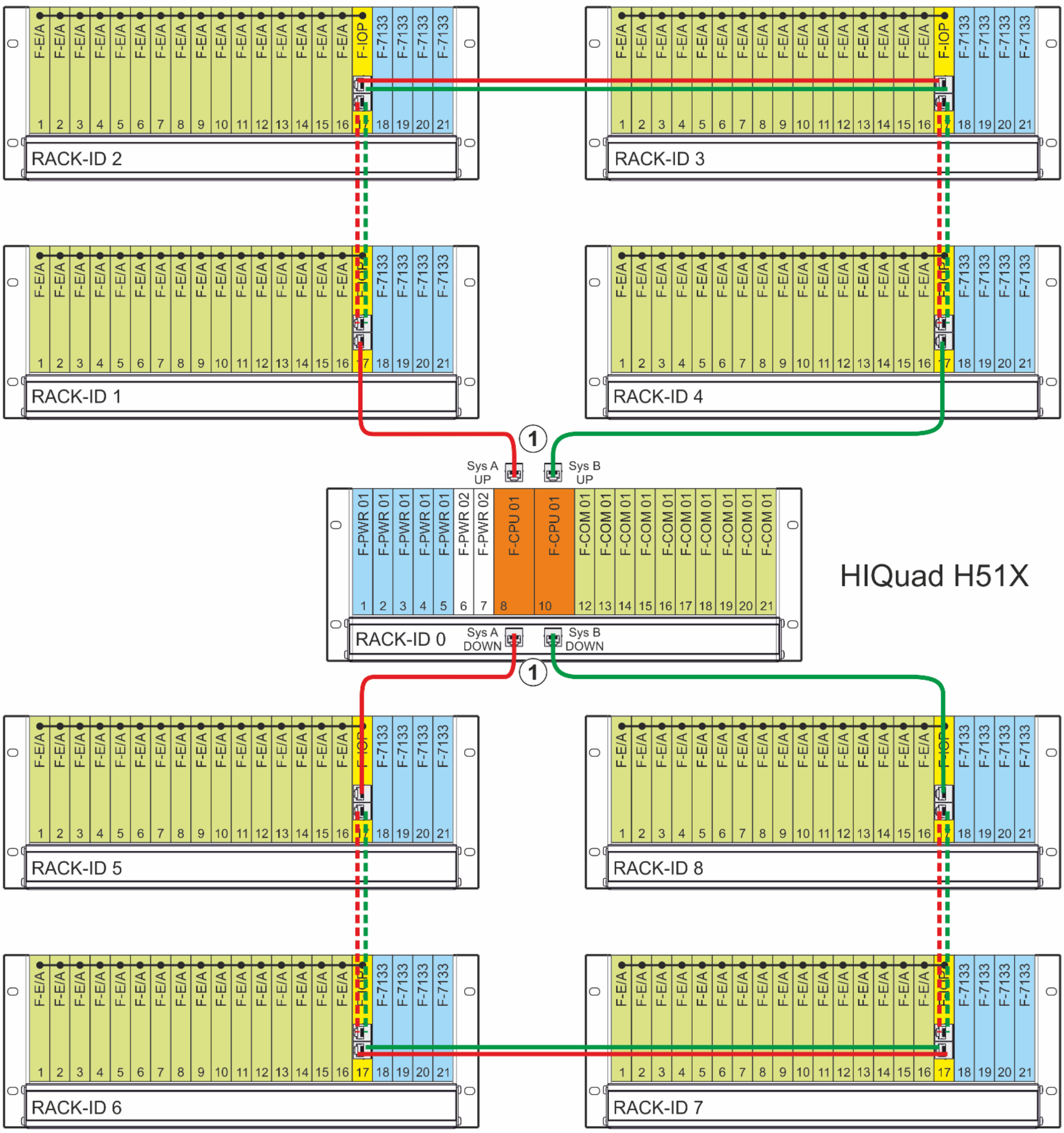
Figure 4: Example of Safe H51X Redundant Operation (1oo2)

In contrast to mono operation, the entire design of a redundant system is intended to ensure availability. Redundant input modules safely record the values measured by redundant sensors. They exchange data with the processor modules via the safety-related I/O processing modules. The measured values are cyclically queried and compared by the redundant processor modules, and then processed by the user program. The user program's results are sent to the I/O processing module, which writes them to the redundant output modules. The output modules thus control the field level, e.g., the actuators. The example in Figure 4 shows a redundant structure of field level and extension racks.

During redundant operation, the signal is processed via both system buses A and B. The system buses A and B between the I/O processing modules are implemented in a patch cable.

Figure 5 shows the example of an H51X redundancy system with system buses A and B. Up to 16 extension racks can be connected to the system buses in a UP loop, DOWN loop, or UP and DOWN loop. The extension racks are interconnected with system buses A and B via the I/O processing module, see F-IOP 01 manual (HI 803 219 E).

The advantage of a redundancy system is that, if one system bus is disconnected, the system can continue to operate via the redundant system bus. If an I/O processing module fails, all output modules in the affected rack enter the safety-related, de-energized state. The failsafe default values are transmitted for the input modules.



1 System bus interfaces on the rear side of the base rack.

Figure 5: Example of H51X Redundancy System

The rack IDs do not necessarily have to be arranged as described above, but they must be unique.

To ensure a clearer overview, HIMA recommends the following:

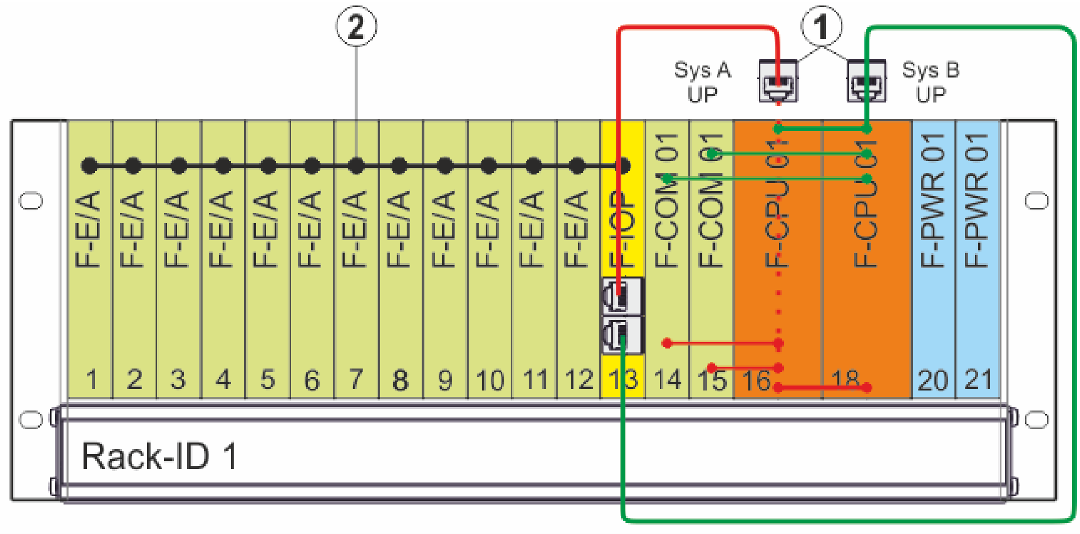
- Arrange the rack IDs in accordance with Figure 5.
- Between the base rack and the first F-IOP module, use red patch cables for system bus A.
- Between the base rack and the first F-IOP module, use green patch cables for system bus B.

3.3 Concept for HIQuad H41X

The H41X system family has a modular structure which includes an H41X base rack and can be additionally equipped with an extension rack. The additional extension rack can be used to create a redundant I/O structure. The H41X base rack (F-BASE RACK 02) can be equipped as shown in Figure 6.

The communication modules are connected to the processor modules via two system buses (A and B) in a point-to-point connection. The processor module in slot 16 controls and monitors system bus A whereas the processor module in slot 18 controls and monitors system bus B. During redundant operation, the two processor modules align their data.

The RJ-45 system bus interfaces on the rear side of the base rack are used to connect the I/O modules in the H41X base rack to the processor modules. An I/O processing module (F-IOP 01) must be used in the H41X base rack to connect the I/O bus to the system buses. An I/O level redundant to that in the base rack can be set up by integrating an additional extension rack in the system bus connection of the H41X base rack.



1 System bus connection on the rear side of the base rack. **2** I/O bus.

Figure 6: H41X Base Rack Completely Assembled

The rack IDs for the HIQuad H41X system are fixed.

To ensure a clearer overview, HIMA recommends the following:

- Between the base rack and the first F-IOP module, use red patch cables for system bus A.
- Between the base rack and the first F-IOP module, use green patch cables for system bus B.

3.3.1 The H41X Mono System

Thanks to the use of safety-related modules (I/O modules, the I/O processing module and a processor module), the HIQuad H41X system can ensure safety-related signal processing in accordance with SIL 3 already when operating in a mono structure, see Figure 7.

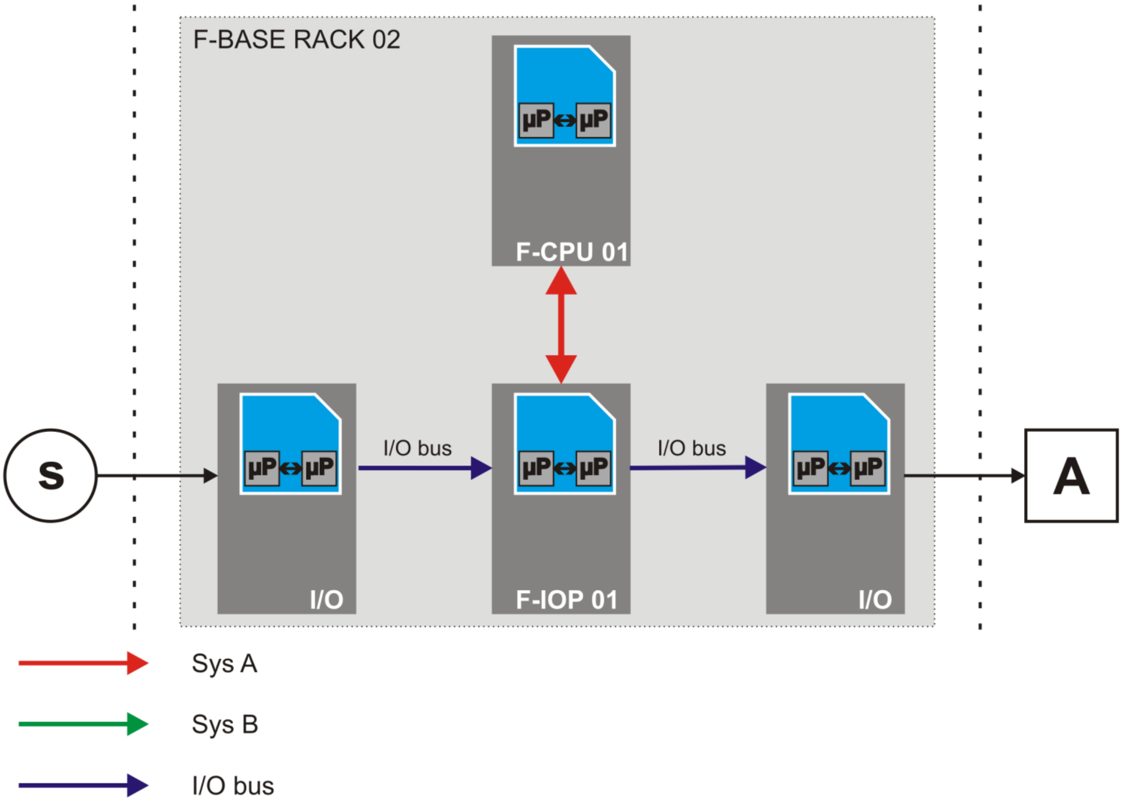


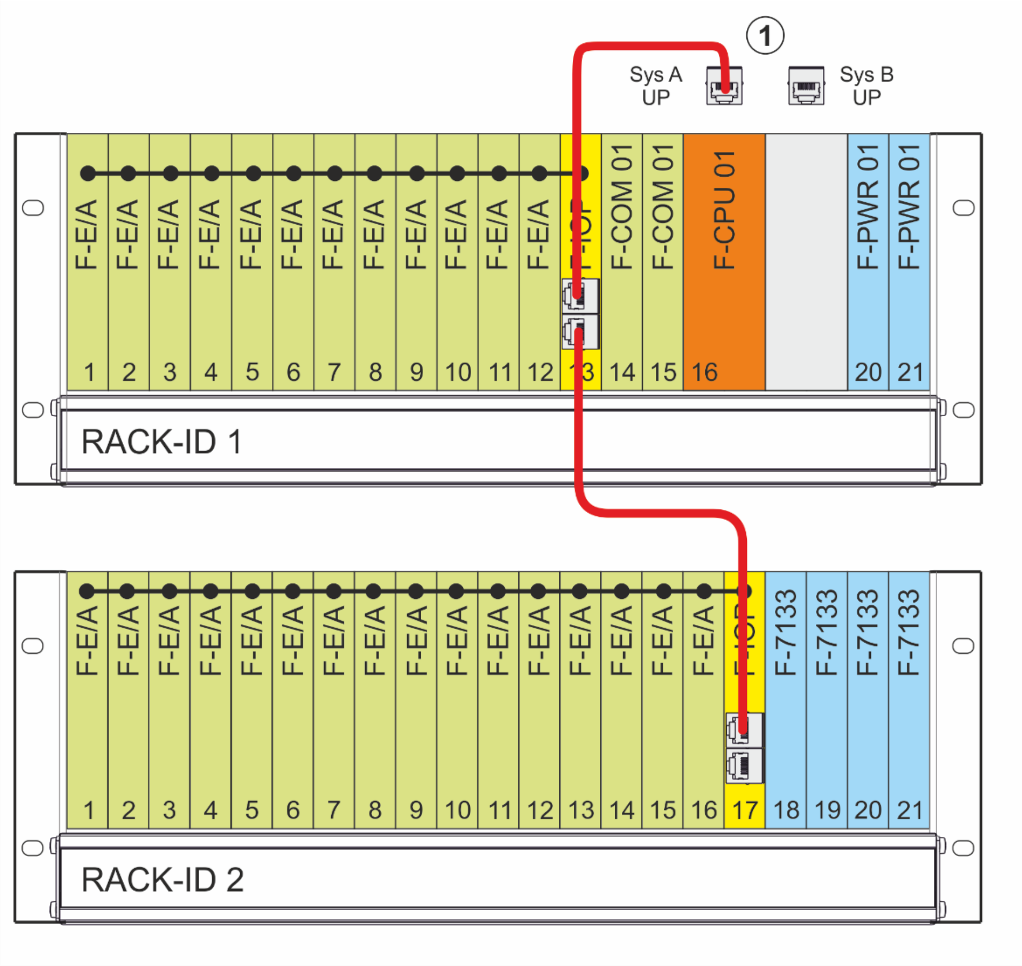
Figure 7: Example of Safe H41X Mono Operation (1oo2)

The input modules of the HIQuad H41X system safely record the values measured by sensors. Data is exchanged with the processor module via the I/O processing module. The measured values are cyclically queried by the processor module and processed by the user program. The user program's results are sent to the I/O processing module, which writes them to the output modules. The output modules thus control the field level, e.g., the actuators.

During mono operation, the signal is forwarded by the processor module in slot 16 via system bus A.

Figure 8 shows the example of an H41X mono system with system bus A. An additional extension rack can be connected to the system bus A. The extension rack is connected to the H41X base rack via the I/O processing module and the system bus A, see F-IOP 01 manual (HI 803 219 E).

If the system bus connection is interrupted in a mono system, all I/O modules located after the interruption point are no longer available. After the interruption point, all output modules enter the safety-related, de-energized state. As for the input modules, the failsafe initial values are processed in the respective processor module.



1 System bus interfaces on the rear side of the base rack.

Figure 8: Example of H41X Mono System

The rack IDs for the HIQuad H41X system are fixed.

To ensure a clearer overview, HIMA recommends the following:

- Use red patch cables for system bus A if only system bus A is used.

3.3.2 The H41X Redundancy System

During redundant operation with two processor modules, both system buses are used to process the signals. This variant with redundant processor modules and system buses increases the system's availability, see Figure 9. If a processor module fails, it automatically enters the safe state and the redundant processor module maintains safe operation. The faulty processor module must be replaced to ensure continued availability. The processor module can be replaced while the system is operating.

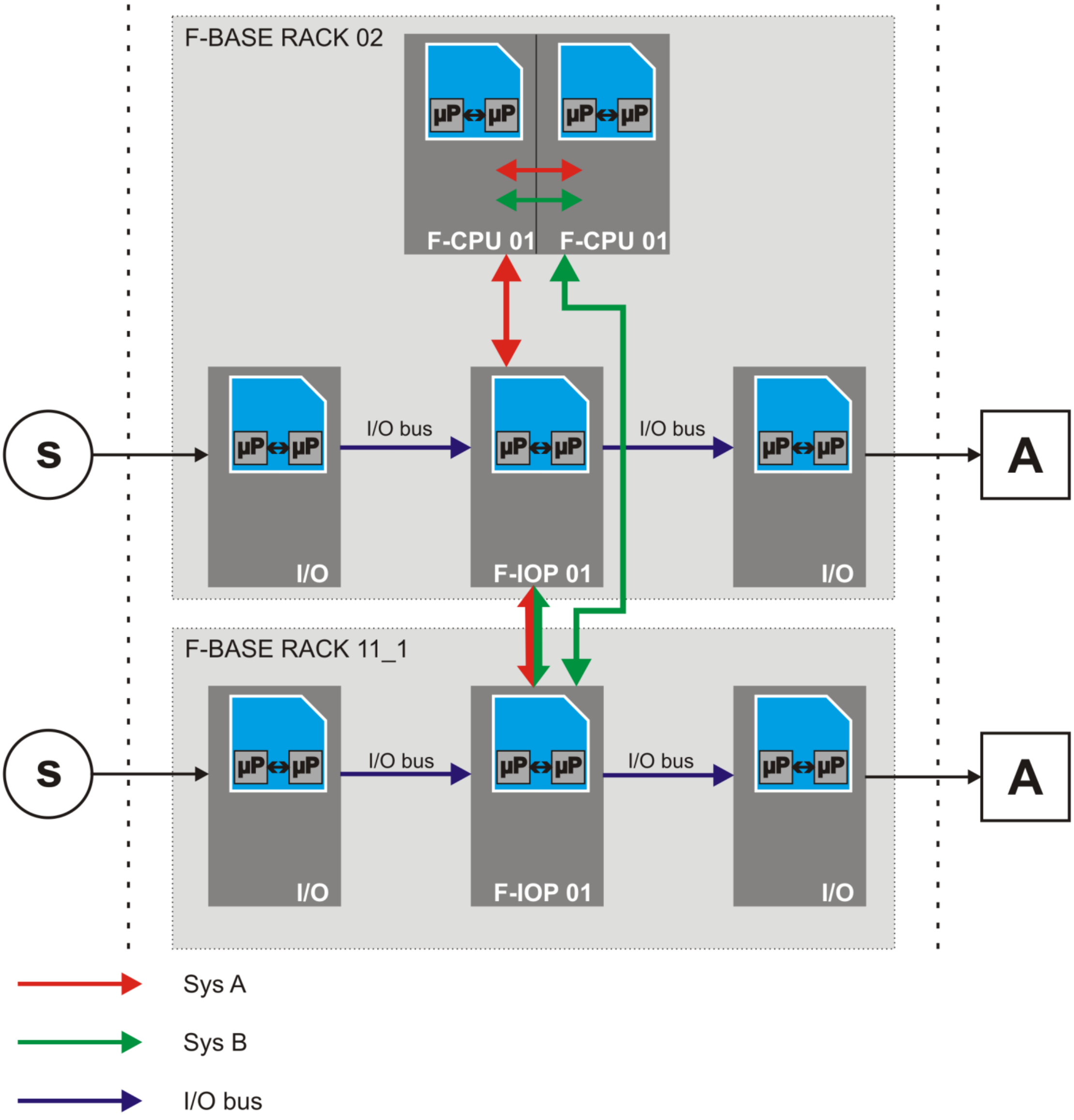


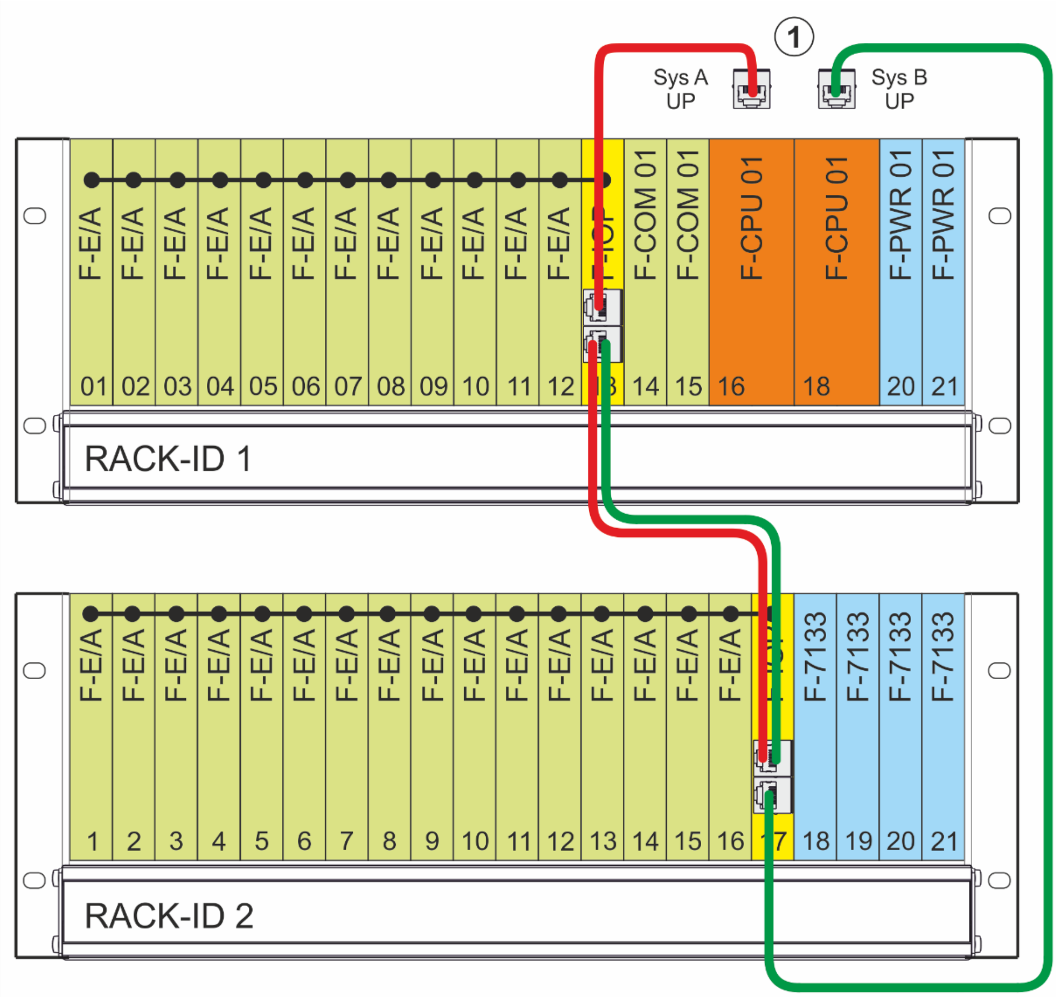
Figure 9: Example of Safe H41X Redundant Operation (1oo2)

In contrast to mono operation, the entire design of a redundant system is intended to ensure availability. Redundant input modules safely record the values measured by redundant sensors. They exchange data with the processor modules via the safety-related I/O processing modules. The measured values are cyclically queried and compared by the redundant processor modules, and then processed by the user program. The user program's results are sent to the I/O processing module, which writes them to the redundant output modules. The output modules thus control the field level, e.g., the actuators. The example in Figure 9 shows a redundant structure of field level and extension rack.

During redundant operation, the signal is processed via both system buses A and B. The system buses A and B between the I/O processing modules are implemented in a patch cable.

Figure 10 shows the example of an H41X redundancy system with system buses A and B. An additional extension rack can be connected to the system buses. The extension rack is connected to the H41X base rack via the I/O processing module and the system buses A and B, see F-IOP 01 manual (HI 803 219 E).

The advantage of a redundancy system is that, if one system bus is disconnected, the system can continue to operate via the redundant system bus. If one I/O processing module fails, the I/O modules in the affected rack enter the safe state while the other rack is not impaired by this failure.



1 System bus interfaces on the rear side of the base rack.

Figure 10: Example of H41X Redundancy System

The rack IDs for the HIQuad H41X system are fixed.

To ensure a clearer overview, HIMA recommends the following:

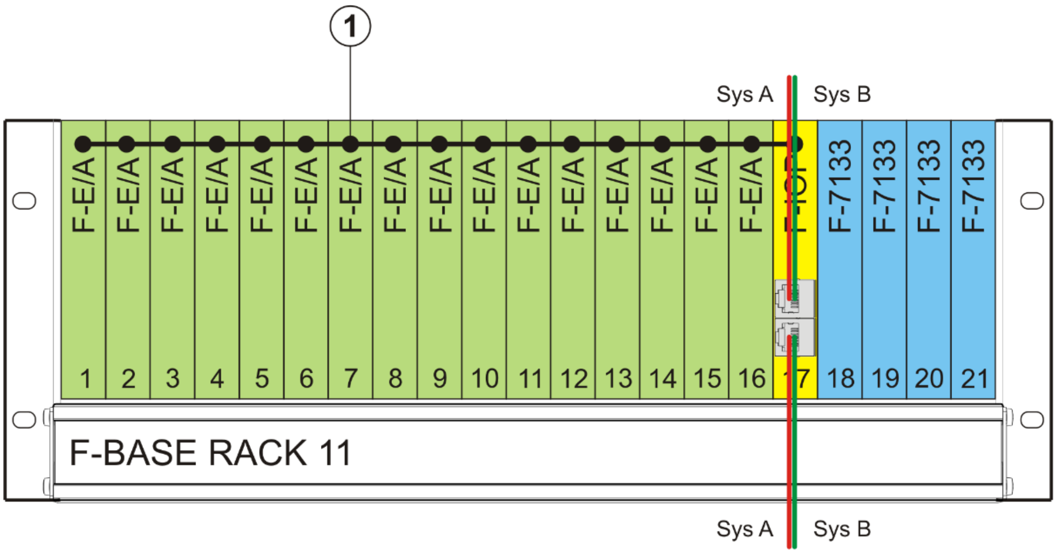
- Between Sys A UP and the F-IOP module, use red patch cables for system bus A.
- Between Sys B UP and the F-IOP module, use green patch cables for system bus B.

3.4 Extension Rack

The extension racks (F-BASE RACK 11) allow the HIQuad X system to be equipped with up to 265 I/O modules. The extension racks can be equipped with a maximum of 16 I/O modules to be inserted in slots 1...16. The I/O processing module is used to connect the system buses to the I/O bus.

The power distribution modules (F 7133) are used to fuse and distribute L+ and L- for the I/O modules. The power distribution modules are interference-free. They are provided with fuse monitoring and signal a failed fuse via contact and LED.

Figure 11 shows a fully equipped extension rack.



1 I/O bus.

Figure 11: Extension Rack

3.5 Ventilation Concept

The high integration level of electronic components causes heat loss, which also depends on the external load of the HIQuad X system. For this reason, ensure proper ventilation within the control cabinet. Low ambient temperature increases the product life and the reliability of the electronic components within the system.

3.5.1 Measures for Reducing the Temperature

Low ambient temperature increases the product life and the reliability of the electronic components within the system. To reduce the temperature within the control cabinets, HIMA's standard control cabinets are equipped with the following components:

- SK 3162 S air intake filters ensuring air supply within the control cabinet through the door. The air intake filter should be used to prevent contaminants from entering the control cabinets.
- Air exhaust through the cut-outs on top of the control cabinet.
- HIMA K 9202B cabinet fan for mounting on the top internal section of the control cabinet.
- M 7200/M 7202 ventilation tray for the air duct between the individual racks.
- K 9203A rack fan for forced cooling of the air from the racks.

3.5.2 Engineering Support

The power dissipation of the equipment within the cabinet is decisive for determining the fan components. Uniform distribution of the heat load is assumed; the maximum temperature increase is 25 °C.

The average heat dissipation of a HIMA standard cabinet achieved by convection, i.e., without additional aids, is 300 W. This assumes installation of several cabinets next to each other and with their rear side to the wall so that the heat can only be dissipated via the top of the cabinet.

Using a K 9202B cabinet fan, an air flow rate of 200 m³ per hour can be achieved by means of forced cooling. The following total power dissipation has thus to be purged:

Type of standard control cabinet	Power dissipation
M 1511	1000 W
M 1512	1000 W
M 1513	1000 W
M 1514	800 W

Table 2: Power Dissipation of Standard Control Cabinets

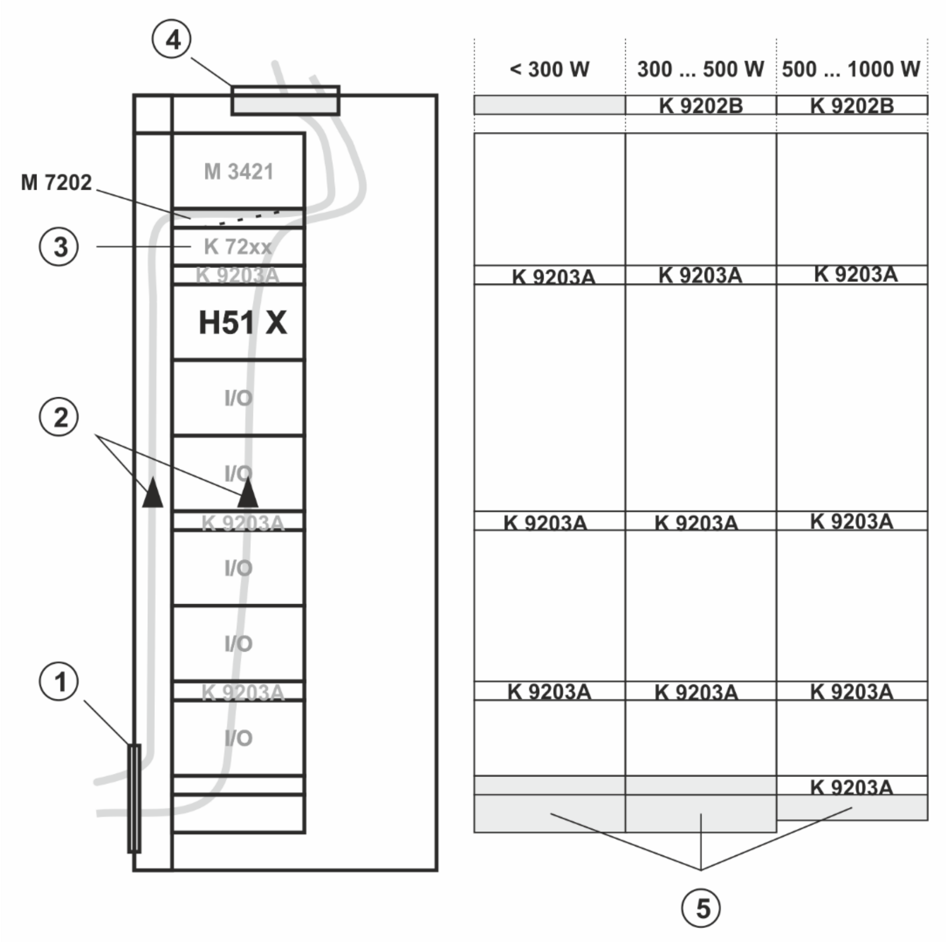
The air enters the control cabinet through the air intake filter located in the control cabinet door. For optimal air discharge, ensure 2 RU free space at the bottom of the pivoting frame. If a K 9203A fan rack is the lowest element being installed, only 1 RU free space is needed at the bottom.

3.5.2.1 Installing the HIQuad X System in the Control Cabinet

When installing a HIQuad X system in the control cabinet (pivoting frame), note the following points for the fan concept:

- A K 9203A rack fan must be used above a base rack.
- A maximum of 2 extension racks may be directly positioned one below the other. To install a K 9203A rack fan, 1 RU clearance must be free between 2 successive extension rack blocks.
- From 300 W power dissipation, a roof-mounted fan must be used in the control cabinet.

The following figure shows the side view of a control cabinet with built-in components. The figure shows the air flow course within the control cabinet and the relationship between maximum power dissipation and fan components to be used.



1 Air intake filter mounted in the control cabinet door.

2 Air flow.

3 Fuse and power distribution modules K 7205, K 7206, K 7212, K 7213 or K 7214.

4 Air exhaust through the HIMA K 9202B cabinet fan for mounting on the top internal section of the control cabinet.

5 Required clearance for air intake and air exhaust.

Figure 12: Fan Concept within the Control Cabinet

Total power dissipation	Maximum power dissipation per extension rack	Fan components
< 300 W	< 50 W	3 x K 9203A
300…500 W	< 50 W	3 x K 9203A + K 9202B
500…1000 W	< 100 W	3 x K 9203A + K 9202B

Table 3: Fan Ccomponents as a Function of Power Loss

i When installing I/O modules, always observe the special instructions specified in the corresponding data sheets, e.g., additional fans may be required depending on the module type.

If processor modules, I/O processing modules or communication modules report that the temperature limits have been exceeded for a longer period of time, the existing ventilation concept must be reviewed.

3.5.2.2 Heat Dissipation

The following consideration can also be used to determine the power dissipation. Uniform distribution of the heat load and unhindered natural convection are assumed.

Magnitude	Description	Unit
P_V	Power dissipation (heat capacity) of the electronic components within the device	W
A	Effective enclosure surface (see below)	m²
B	Enclosure width	m
H	Enclosure height	m
T	Enclosure depth	m
K	Coefficient of heat transfer of the enclosure	W/m² K
	Example: Steel plate	Approx. 5.5 W/m² K

Table 4: Definitions for Calculating the Power Dissipation

3.5.2.3 Installation Type

The effective enclosure surface area A as a function of the mounting or installation type is determined as follows:

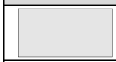
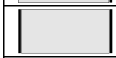
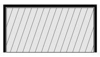
Enclosure installation type in accordance with VDE 0660, Part 5		Calculation of the enclosure surface A
	Individual enclosure, free-standing on all sides	$A = 1.8 \times H \times (W + D) + 1.4 \times W \times D$
	Individual enclosure for wall mounting	$A = 1.4 \times W \times (H + D) + 1.8 \times H \times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

Table 5: Installation Types for Control Cabinets

3.5.2.4 Natural Convection

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

The maximum temperature increase ΔT_{max} of all electronic devices within the enclosure is calculated as follows:

$\Delta T_{max} = P_V / k \times A$

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.

3.5.2.5 Note on the Standard

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

i Considerations about heat must take every component within a cabinet or enclosure into account, including components that are not directly part of the HIQuad X system!

4 Product Description

HIQuad X is a 19-inch system which includes a base rack and one or multiple extension racks.

4.1 Backplane

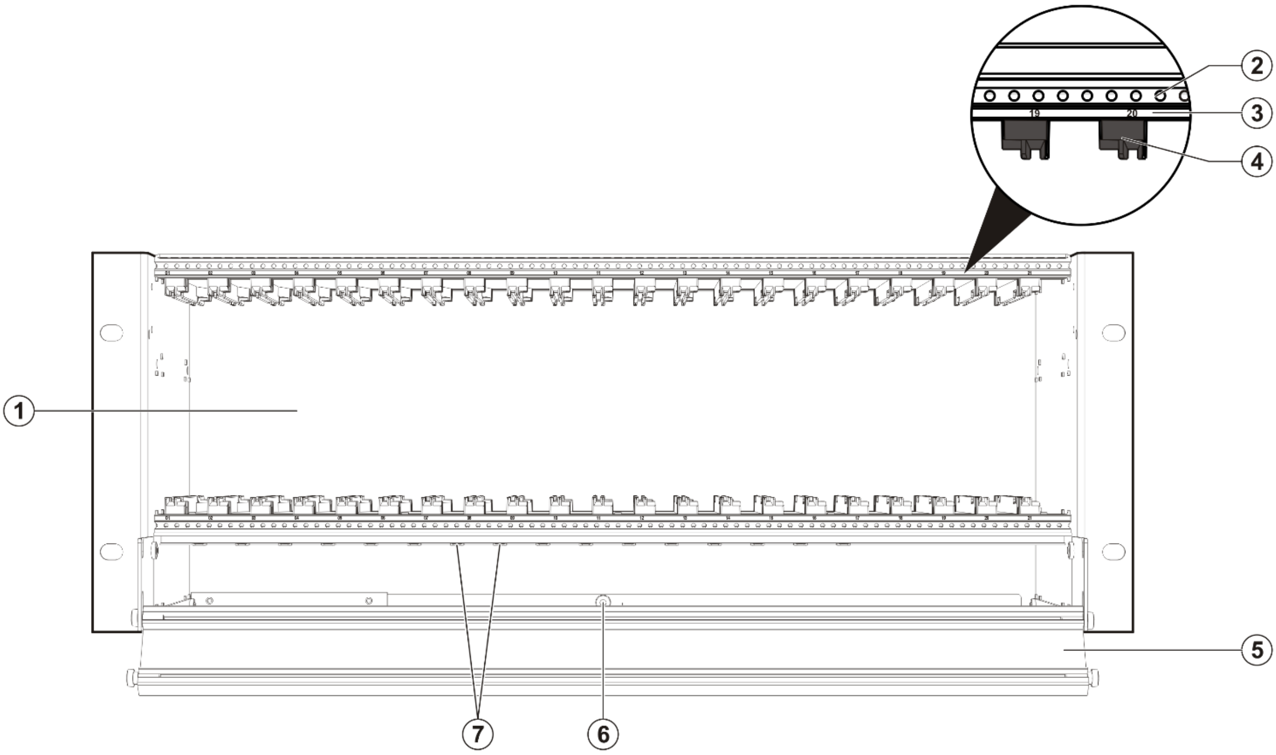
The different backplanes are firmly screwed to the 19-inch frame, creating the following racks:

H51X base rack	H41X base rack	Extension rack
H51X Backplane	H41X Backplane	Extension rack backplane

Table 6: Rack Backplanes

4.2 19-Inch Frame

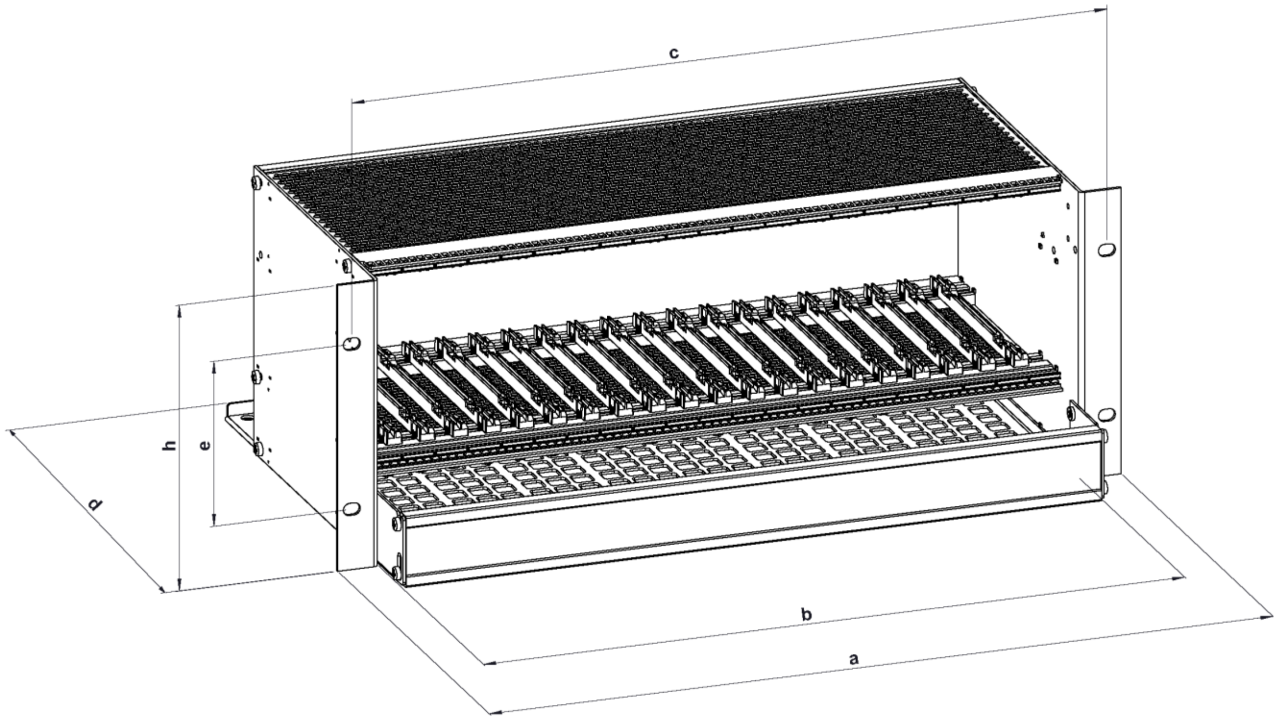
The 19-inch frame is the basic mechanical structure of the HIQuad X system. The following figure shows the structure of the 19-inch frame:



1 19-inch frame with integrated cable tray.
2 Threaded inserts.
3 Numbering of slots.
4 Insertion guide for modules.
5 Folding labeling profile.
6 General M4 ground terminal on the rear side.
7 Grounding bar (functional ground), 1 Faston flat connector (6.3 x 0.8 mm) each for slot 1…slot 16.

Figure 13: 19-Inch Frame

The following figure shows the dimensions of the 19-inch frame:

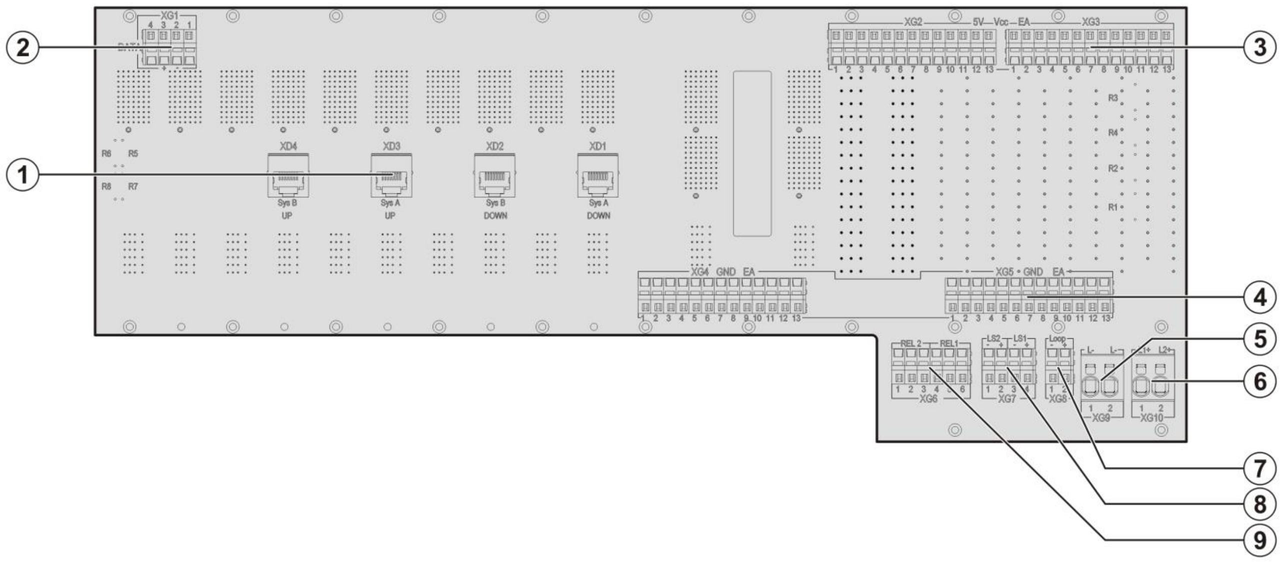


a	External dimensions = 482.6 mm	c	Mounting hole distance = 101.6 mm
b	Mounting space (84 HP) = 84 x 5.08 mm	f	---
c	Mounting hole distance = 465 mm	g	---
d	Mounting depth = 263 mm	h	Rack unit (4 RU) = 4 x 44.45 mm

Figure 14: Dimensions of the 19-Inch Frame

4.2.1 H51X Backplane

The following figure shows the rear side of the H51X backplane:



1	System bus connections Sys. A DOWN (XD1), Sys. B DOWN (XD2) Sys. A UP (XD3), Sys. B UP (XD4).
2	XG1 (DATA) Not applicable (for future use)!
3	5 V power supplies for extension racks, XG2 and XG3.
4	Reference potential (GND) for extension racks, XG4 and XG5.
5	Connection to L- reference potential (24 V power supply), XG9.
6	Clamp terminal block for redundant 24 V power supply, XG10.
7	Control cabinet diagnostics for future applications, XG8 (loop) Not applicable (for future use)!
8	24 V power supply (LS1, LS2) for the F-IOP 01 modules in the extension racks, XG7 (Buffer module in slot 6 supplies LS1 and buffer module in slot 7 supplies LS2).
9	Signaling relay contacts for F-PWR 02, XG6.

Figure 15: Rear View of H51X Backplane

4.2.1.1 Supply of the H51X Base Rack

For supply and power distribution, HIMA recommends using the following components:

- K 7205: Redundant supply up to a maximum of 63 A total current with fuse protection of up to 18 individual circuits with circuit breakers.
- K 7212: Redundant supply up to a maximum of 35 A total current with 2 decoupling diodes and 2 mains filters, with fuse protection of up to 12 individual circuits with circuit breakers.
- K 7213: Redundant supply up to a maximum of 35 A total current with fuse protection of up to 12 individual circuits with circuit breakers.
- K 7214: Redundant supply up to a maximum of 150 A total current with fuse protection of up to 18 individual circuits with circuit breakers.

The 24 V power supply is connected to the following terminals:

Spring terminal	Cross-section and color	Fuse
XG10.1/.2 (L1+, L2+)	2.5 mm² RD	Maximum 16 A gL
XG9.1/.2 (L-)	2.5 mm² BK	

Table 7: Connection to the 24 V Power Supply

4.2.1.2 Buffered Voltage for LS1+ and LS2+ in F-PWR 02 Buffer Modules

The buffered voltage (LS1+ or LS2+) for extension racks is connected to the following terminals:

Spring terminal	Cross-section and color
XG7.1 (LS2-)	2.5 mm² BK
XG7.2 (LS2+)	2.5 mm² RD
XG7.3 (LS1-)	2.5 mm² BK
XG7.4 (LS1+)	2.5 mm² RD

Table 8: Spring Terminals for Buffered Voltage

4.2.1.3 5 V Power Supply for Extension Racks

The 5 V power supply for extension racks is connected to the following terminals:

Spring terminal	Cross-section and color
XG2.1…XG2.13 (Vcc)	2.5 mm² YE
XG3.1…XG3.13 (Vcc)	2.5 mm² YE
XG4.1…XG4.13 (GND)	2.5 mm² GN
XG5.1…XG5.13 (GND)	2.5 mm² GN

Table 9: Spring Terminals for 5 V Power Supply

4.2.1.4 Signaling Relay for F-PWR 02 Buffer Module, XG6

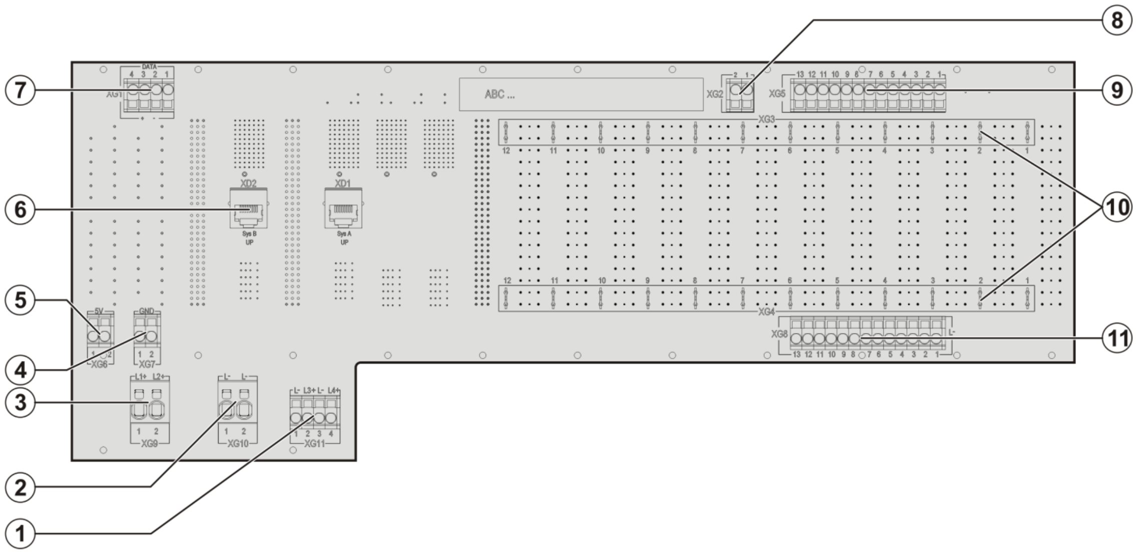
The signaling relay in the buffer modules is connected to the following terminals:

Spring terminal	Cross-section and color
XG6.1…XG6.3 (REL2)	2.5 mm² GR
XG6.4…XG6.6 (REL1)	2.5 mm² GR

Table 10: Spring Terminals in 5 Signaling Relays for Buffer Module

4.2.2 H41X Backplane

The following figure shows the rear side of the H41X backplane:



1 24 V power supply for slots 14…18, XG11. Not applicable (for future use)!

2 Connection to L- reference potential (24 V supply), XG10.

3 Connection to redundant 24 V power supply for F-PWR 01, XG9.

4 Reference potential GND for extension racks, XG7.

5 5 V power supply for extension racks, XG6.

6 System bus connections, Sys. A (XD1) and Sys. B (XD2).

7 XG1 (DATA) Not applicable (for future use)!

8 Watchdog signal supply, XG2 Not applicable (for future use)!

9 Connection of 24 VDC auxiliary voltage for slot 1…slot 3, XG5.

10 Cable plug supply
LS1+…LS12+, slot 1…slot 12; XG3
LS-, slot 1…slot 12; XG4.

11 LS- reference potential for auxiliary voltage (24 VDC), XG8.

Figure 16: Rear View of H41X Backplane

4.2.2.1 Supply of the H41X Base Rack

For supply and power distribution, HIMA recommends using the following components:

- K 7205: Redundant supply up to a maximum of 63 A total current with fuse protection of up to 18 individual circuits with circuit breakers.
- K 7212: Redundant supply up to a maximum of 35 A total current with 2 decoupling diodes and 2 mains filters, with fuse protection of up to 12 individual circuits with circuit breakers.
- K 7213: Redundant supply up to a maximum of 35 A total current with fuse protection of up to 12 individual circuits with circuit breakers.
- K 7214: Redundant supply up to a maximum of 150 A total current with fuse protection of up to 18 individual circuits with circuit breakers.

The 24 V power supply is connected to the following terminals:

Spring terminal	Cross-section and color	Fuse
XG9.1/.2 (L1+, L2+)	2.5 mm² RD	Maximum 16 A gL
XG.10.1/.2 (L-)	2.5 mm² BK	

Table 11: Connection to the 24 V Power Supply

4.2.2.2 5 V Power Supply for Extension Racks

The 5 V power supply for the extension rack is connected to the following terminals:

Spring terminal	Cross-section and color
XG6.1/.2 (5 V)	2.5 mm² YE
XG7.1/.2 (GND)	2.5 mm² GN

Table 12: Spring Terminals for 5 V Power Supply

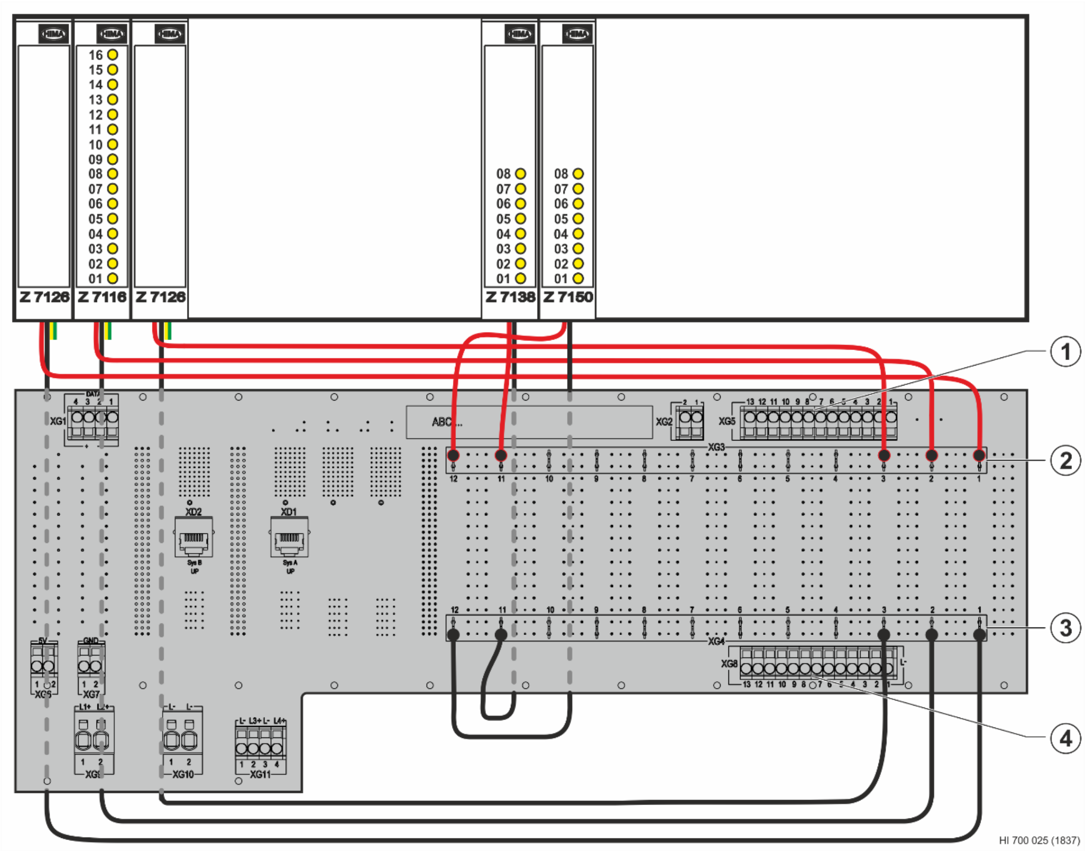
4.2.2.3 24 V Auxiliary Voltages for I/O Modules and I/O Processing Module

The 24 V auxiliary voltage for the cable plugs of the I/O modules and the I/O processing module is connected at the following terminals.

Spring terminal	Cross-section and color
XG5.1…XG5.13	2.5 mm² RD
XG8.1…XG8.13 (L-)	2.5 mm² BK

Table 13: Spring Terminals for 24 V Auxiliary Voltages in I/O Modules

The connection to the 24 V power supply of the cable plugs is performed as shown in Figure 17. The Faston flat connectors XG3 and XG4 are supplied via field terminals XG5 and XG8 in accordance with the corresponding slot number.



1 XG5: Connection to 24 V auxiliary voltage for slot 1…slot 13; assignment based on slot number.

2 XG3: Faston flat connectors for supplying the I/O cable plugs.

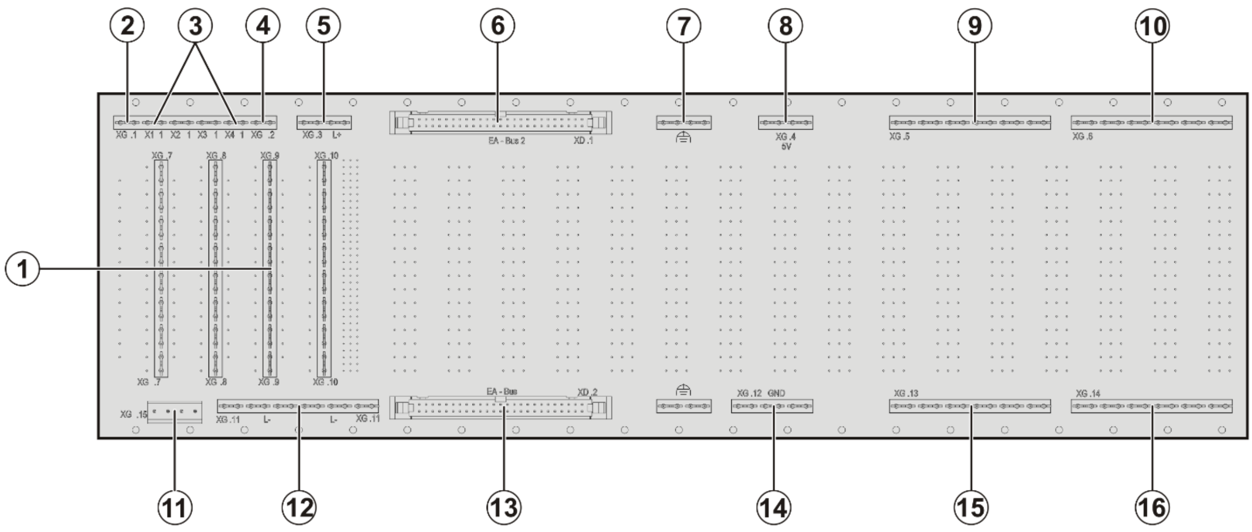
3 XG4: Faston flat connectors for GND of I/O cable plugs.

4 XG8: LS- reference potential for auxiliary voltage (24 VDC).

Figure 17: Connection to the 24 V Power Supply of the Cable Plugs (H41X)

4.2.3 Extension Rack Backplane

The following figure shows the rear side of the extension rack backplane:



1 24 VDC supply (L+) for power distribution modules in slots 18…21, XG.7…XG.10.
2 Fuse monitoring for power distribution modules, XG.1.
3 Jumpers X1…X4 (fuse monitoring).
4 Fuse monitoring for power distribution modules, XG.2.
5 24 VDC supply for I/O processing module, XG.3.
6 Do not use it for HIQuad X!
7 PE connection.
8 5 VDC voltage, XG.4.
9 Potential distributor, for free use, XG5.
10 Potential distributor, for free use, XG6.
11 Do not use it for HIQuad X!
12 L- (24 VDC) XG.11.
13 Do not use it for HIQuad X!
14 GND (+ 5 VDC), XG.12.
15 Potential distributor, for free use, XG13.
16 Potential distributor, for free use, XG14.

Figure 18: Rear View of Extension Rack Backplane

⚠ WARNING

In HIQuad X, connectors in pos. 11 XG15, pos. 6 XD1 and pos. 13 XD2 must not be connected. The connectors must be provided with blind covers (within the scope of delivery of the F-IOP module).

Failure to comply with this measure may lead to critical system states.

A HIQuad X controller tailored to the concrete application can be created by selecting appropriate modules.

The controller can be easily adapted to future extensions of the process to be controlled, e.g., by adding modules or extension racks with modules for H51X.

4.2.4 Temperature Monitoring

The HIQuad X system is intended for operation up to a maximum ambient temperature of 60 °C. Sensors located at specific temperature-relevant positions on the modules record the temperature state of processor modules, I/O processing modules and communication modules. The temperature state of these modules is centrally monitored and evaluated by the processor modules (F-CPU 01).

System parameters *Temperature State [1]* and *Temperature State [2]* in the user program can be used to evaluate the temperature state, see Chapter 6.2.3.

The system parameters *Temperature State [1] and Temperature State [2]* signal the measured temperatures as follows:

Temperature threshold	Temperature state	*Temperature State [X]* [BYTE]
≤ 40 °C	Normal	0x00
> 40 °C	**Warning**: Threshold 1 exceeded.	0x01
> 60 °C	**Error**: Threshold 2 exceeded.	0x03

Table 14: Thresholds of the Temperature States

If a value exceeds or falls below one of the temperature thresholds, the temperature state changes.

The transition to the state *Threshold 1* exceeded or *Threshold 2* exceeded does **not** indicate an impairment of the system safety.

The user must implement suitable measures to ensure that the ambient temperature limits specified for the system are met.

i The temperature can be used in the user program, e.g., as additional shutdown condition; however, the temperature is not recorded in a safety-related manner.

Temperature State may be used as an additional shutdown condition.

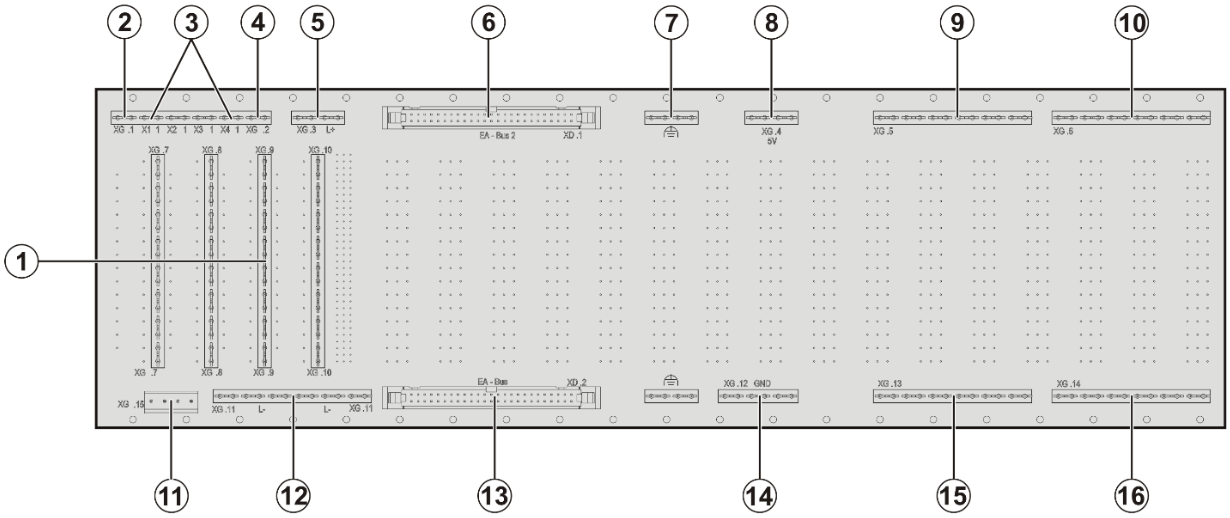
In the SILworX Hardware Editor, the *Temperature State* system parameter can be used to define if exceeding the temperature threshold should cause a message to be issued.

4.3 Power Supply

HIQuad X requires a 24 V power supply that can be connected as follows:

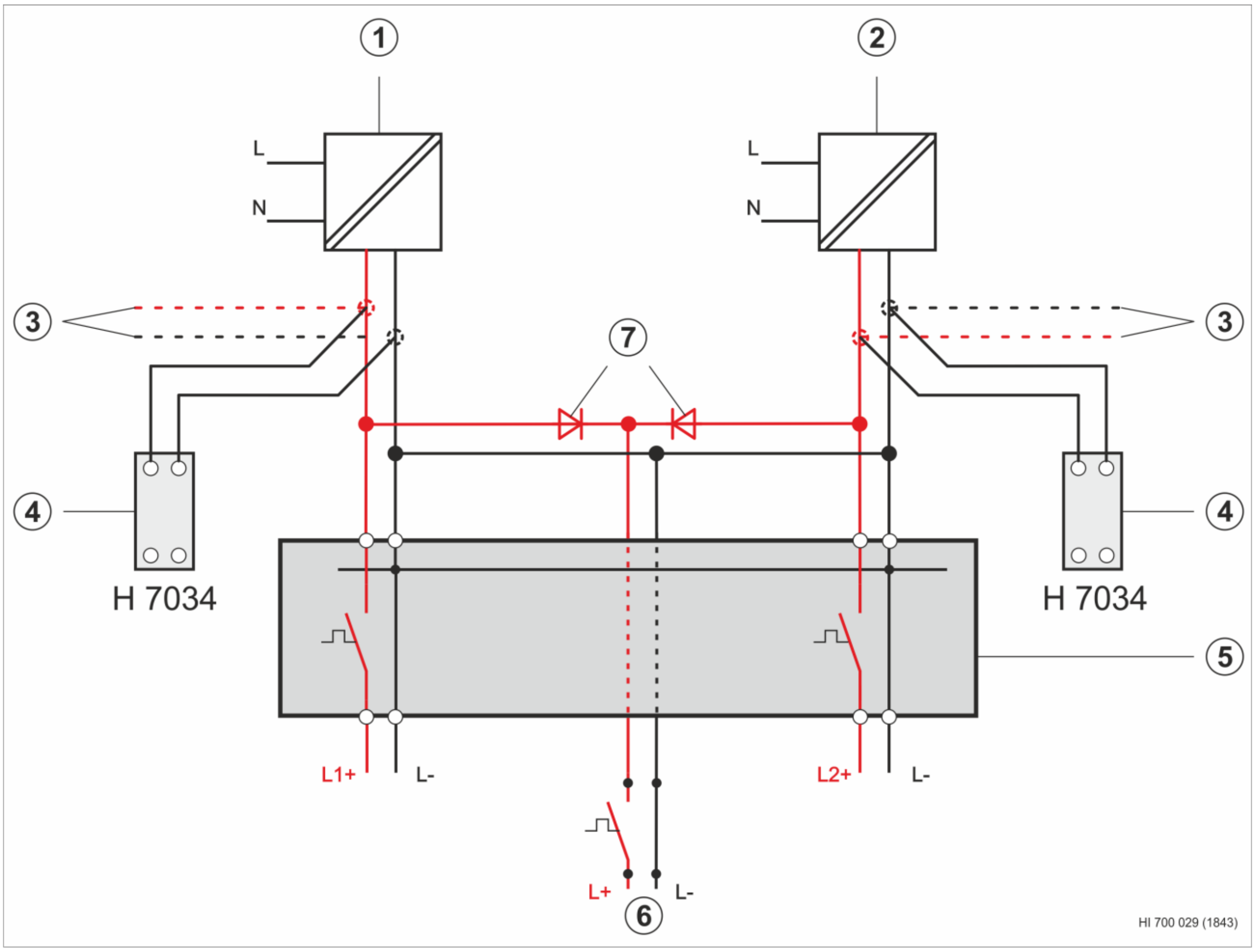
- Mono connection to one or redundant power supply units, see Figure 19.
- Redundant with redundant power supply units, see Figure 20.

HIMA uses red cables for positive potentials (L+) and black cables for negative potentials (L-).



1 Decoupled SELV/PELV power supply unit.
2 Decoupled redundant SELV/PELV power supply unit.
3 Alternative: 24 VDC power nets.
4 H 7034 mains filter.
5 Fuse and power distribution module, see Table 41.

Figure 19: Mono 24 V Power Supply



1 SELV/PELV power supply unit.
2 Redundant SELV/PELV power supply unit.
3 Alternative: 24 VDC power net.
4 H 7034 mains filter.
5 Fuse and power distribution module, see Table 41.
6 Alternatively: Connection for mono components, observe decoupling!
7 Decoupling diodes if not included in the power supply units.

Figure 20: Redundant 24 V Power Supply

The power supply units must meet the requirements in accordance with SELV or PELV. The power supply units must bridge voltage dropouts of up to 20 ms. HIMA power supply units of the PS 1000 series are designed and suitable for a mean time to failure (MTTF) of 30 years. Power supply units from other manufacturers must be checked to ensure that they meet the mentioned requirements. The requirements for a 24 VDC power net are the same as those applying to power supply units.

HIMA recommends using a H 7034 mains filter to protect the 24 V power supplies transient interference. The filter must be installed close to the 24 V supply to suppress interferences directly at the supply point.

The base racks are equipped with redundant L1+ and L2+ terminals to connect to redundant power supply units, see Figure 15 and Figure 16. In doing so, protective separation of the power supplie units must be ensured.

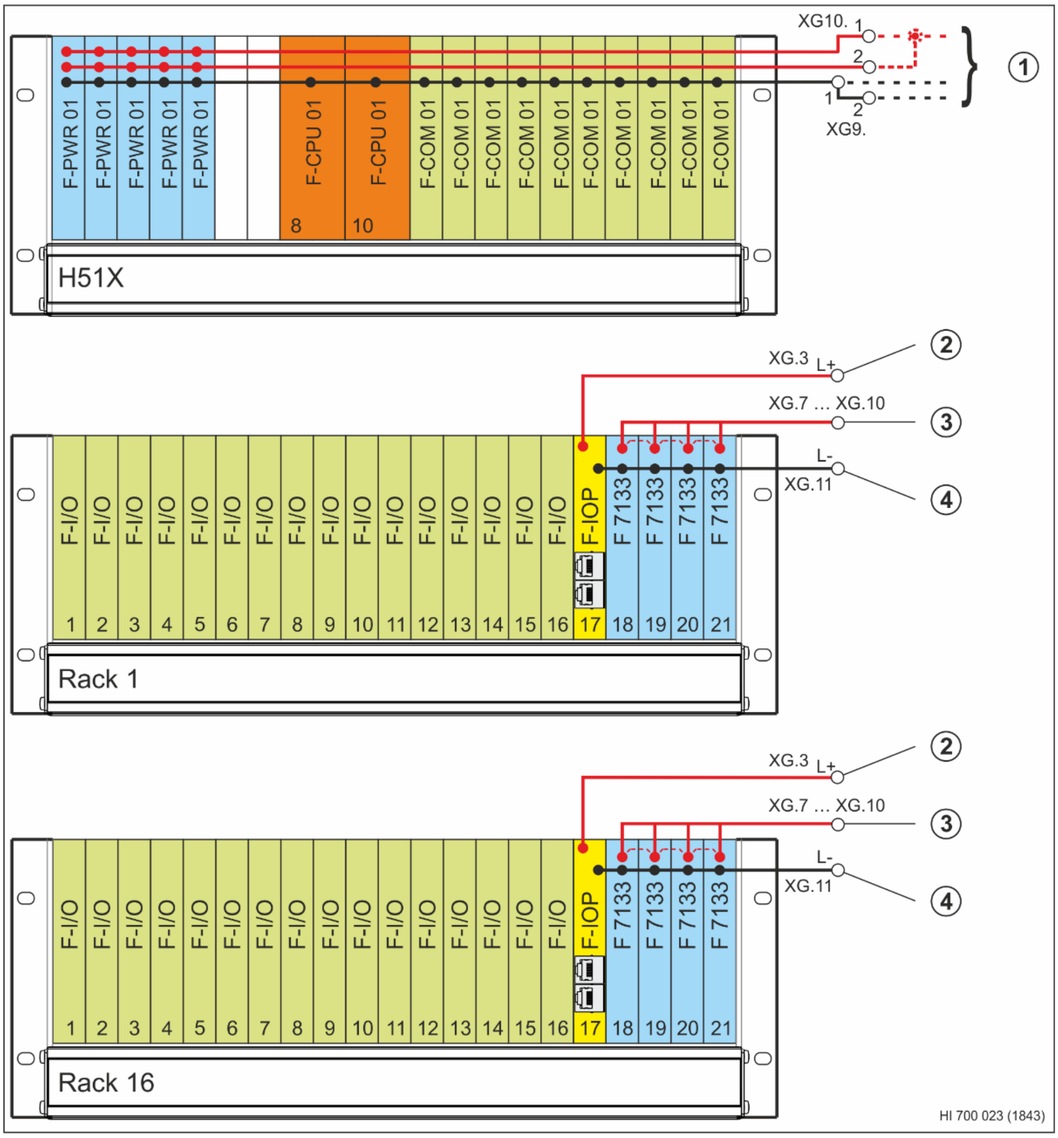
To ensure high availability, operate the HIQuad X systems as follows:

- Use redundant power supplies.
- The power supply units or the 24 VDC power nets must ensure the output voltage never exceeds 31 V.
- Use suitable fuses in the fuse and power distribution module to limit the maximum current input in each base rack to 16 A.
- Users must implement external measures to ensure that the power supply does not fall below 0.8 x U_N (= 19.2 VDC). If no redundant power supply is available, the system responds with failure of individual components or the entire system.

The power distribution modules, K 7205, K 7212, K 7213 and K 7214, include all components required to secure up to 18 individual circuits with circuit breakers. The K 7212 set is additionally equipped with decoupling diodes and mains filters with monitoring relays.

4.3.1 Mono H51X Base Rack (24 VDC)

The 24 V mono power supply is performed for the H51X base rack and the I/O processing modules by connecting to one or redundant power supply units, see Figure 19.



1 Connection to one or redundant power supply units, see Figure 19.

2 Connection to 24 VDC for the I/O processing modules from the same source as the H51X base rack.

3 Redundant supply of the F 7133 power distribution modules, insert jumpers in accordance with the application.

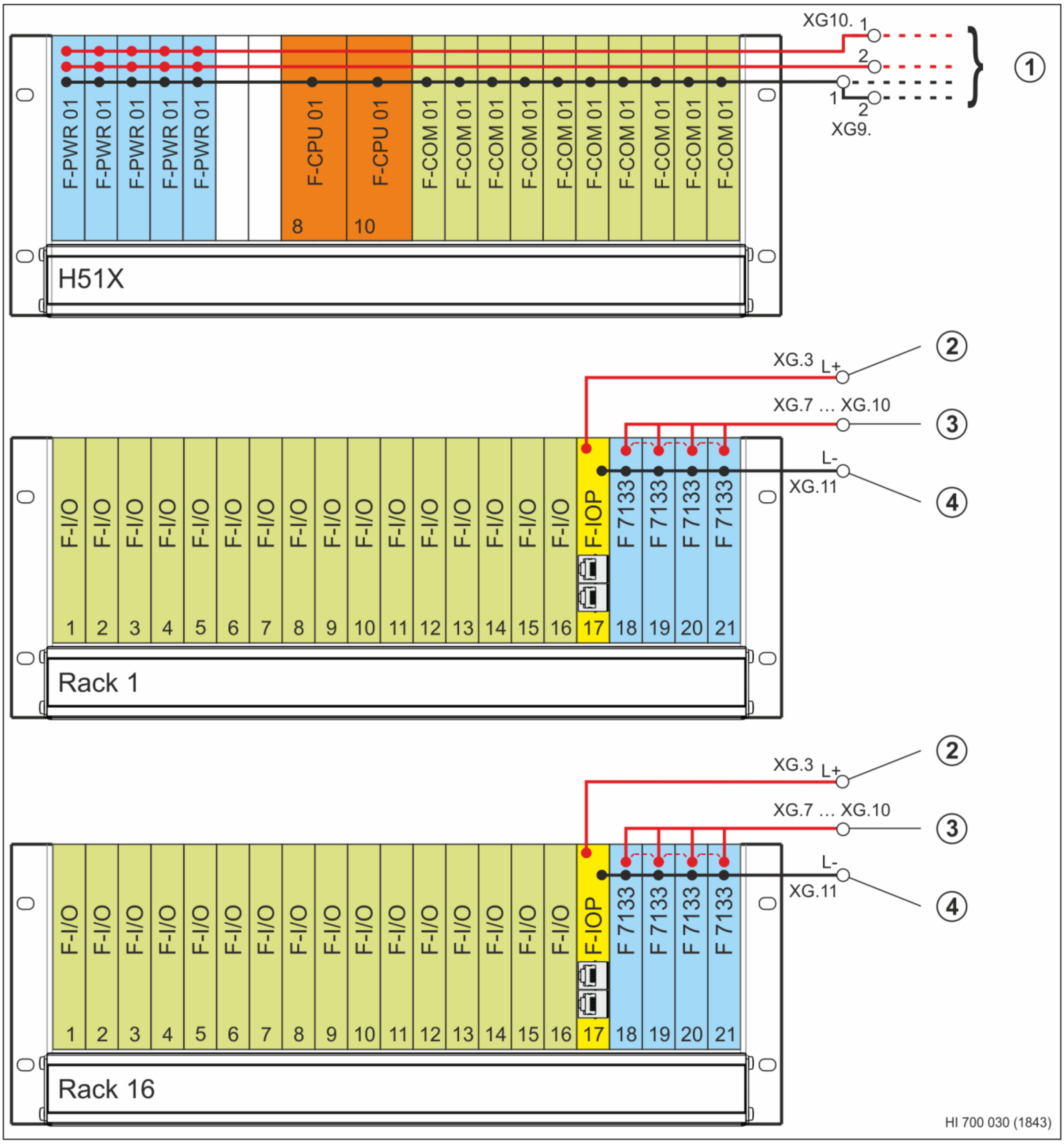
4 Reference potential L-.

Figure 21: Mono Connection to H51X Base Rack (24 VDC)

i For redundant power supply, HIMA recommends using the K 7212 power distribution module with decoupling diodes.

4.3.2 Redundant H51X Base Rack (24 VDC)

The redundant 24 V power supply is performed for the H51X base rack by using redundant power supply units and for the I/O level at the connection for mono components, see Figure 20.



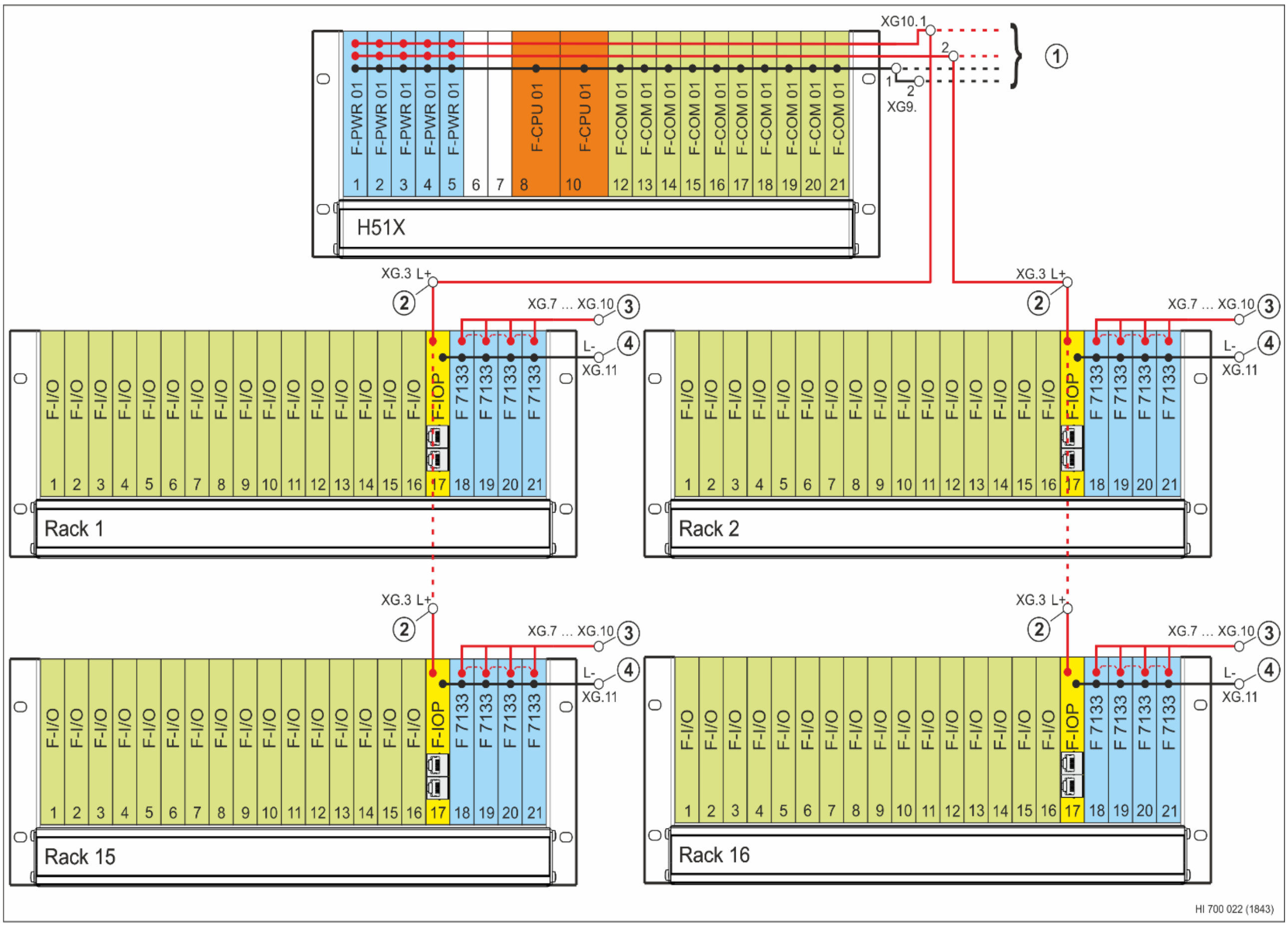
1. Connection to redundant power supply units, see Figure 20.
2. Attach to the connector for mono components, see Figure 20.
3. Redundant supply of the F 7133 power distribution modules, insert jumpers in accordance with the application.
4. Reference potential L-.

Figure 22: Redundant Connection to H51X Base Rack (24 VDC)

4.3.3 Redundant H51X Base Rack and I/O Level (24 VDC)

To ensure increased availability, the user can individually structure the redundancy of the I/O processing modules, and thus of the I/O level. The 24 V power supply is performed for the H51X base rack by using redundant power supply units, see Figure 20. The I/O level is supplied redundantly depending on the application, such as shown in Figure 22.

In the example below, one power supply unit (L1+) powers the extension racks with odd rack IDs while a redundant power supply unit (L2+) powers the racks with even IDs. In this example, the redundancy of the I/O level is portioned based on the rack IDs so that operation is ensured even if a power supply unit fails.



1. Connection to redundant power supply units, see Figure 20.
2. XG.3 terminal (L+) for connecting to 24 VDC in accordance with redundant application.
3. Redundant supply of the F 7133 power distribution modules, insert jumpers in accordance with the application.
4. Reference potential L-.

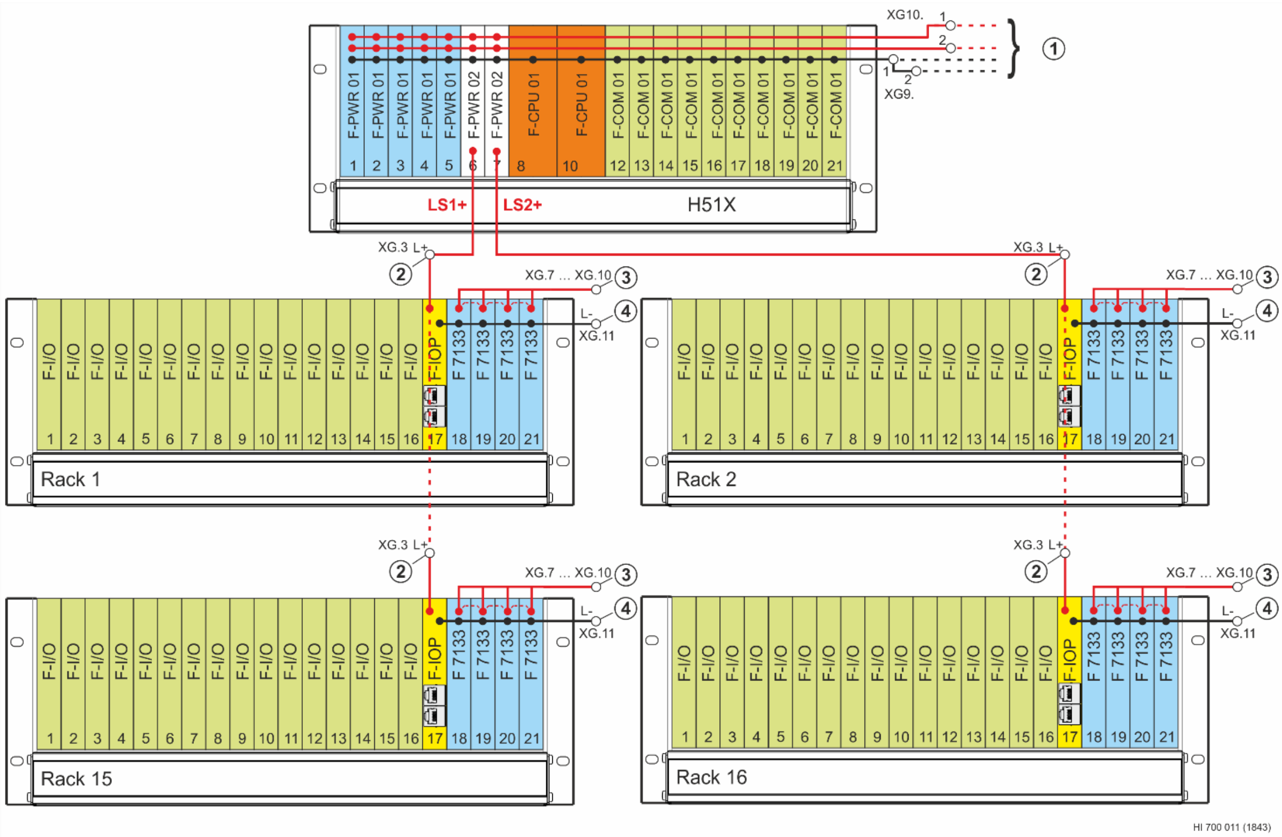
Figure 23: Redundant Connection to H51X Base Rack (24 VDC) and Redundant I/O Level

i For redundant power supply, HIMA recommends using the K 7212 power distribution module with decoupling diodes.

4.3.4 H51X Base Rack (24 VDC) I/O Level via F-PWR 02 Buffer Modules (Optional)

If the power supply units do not meet the requirements for protective separation and for compensating voltage failures of up to 20 ms such as specified in Chapter 4.3, or the requirements are > 20 ms, the buffer modules (F-PWR 02) can be used as an option to supply the I/O processing modules with 24 VDC.

In the following example, the buffer module in slot 6 compensates for voltage dropouts of I/O processing modules in the expansion racks with odd rack IDs. The buffer module in slot 7 compensates for voltage dropouts of the expansion racks with even rack IDs. In doing so, redundant I/O levels can be assembled based on the rack IDs. If one buffer module fails, operation of redundant racks is ensured via the remaining module. The failed buffer module must be replaced immediately to restore the original availability. This structure corresponds to that of the HIQuad HRS system.



1 Connection to redundant power supply units, see Figure 20.

2 Connection to the LS1+/LS2+ power supply of the F-PWR 02 buffer modules in accordance with the desired redundant structure of the I/O level.

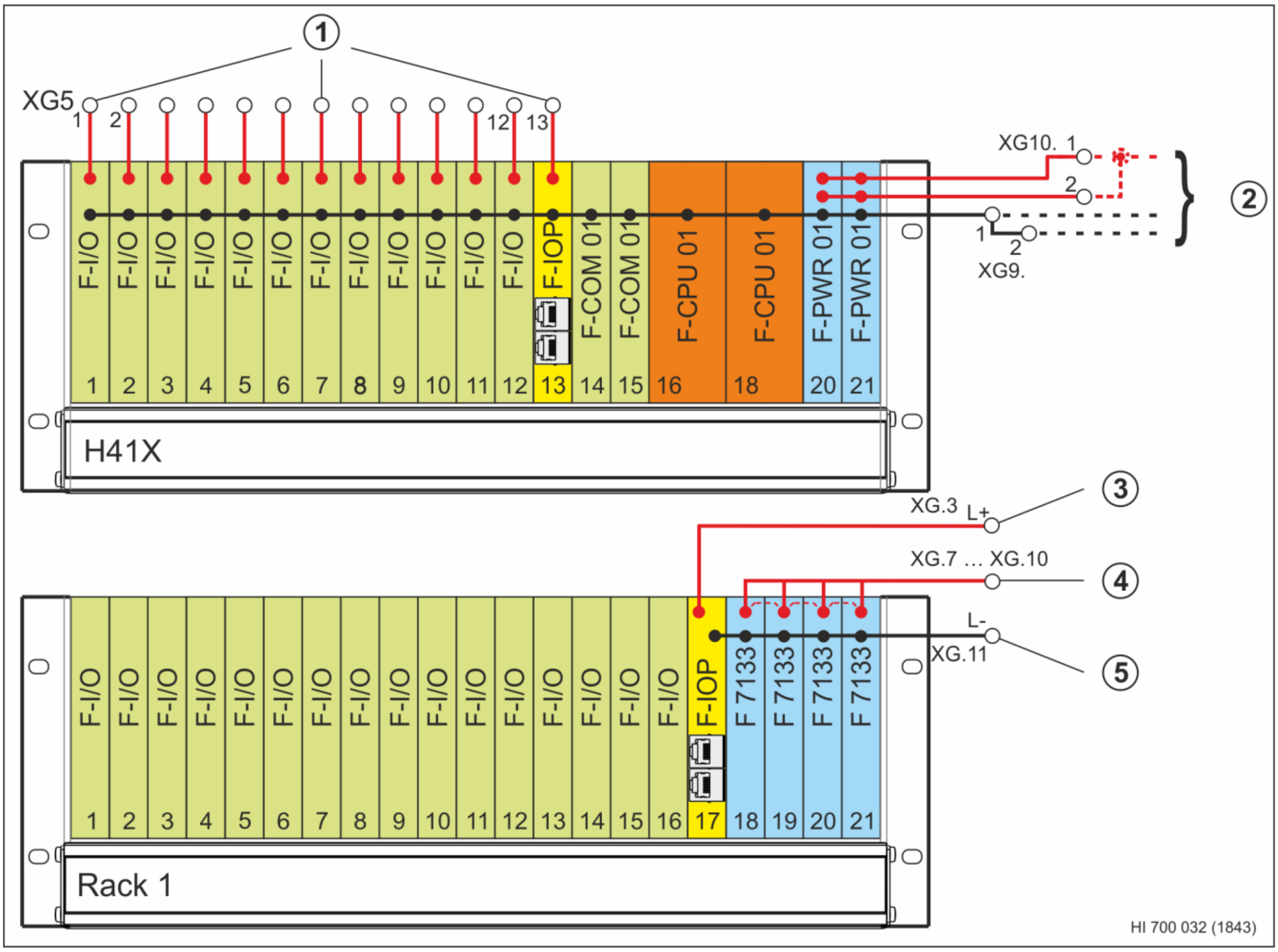
3 Redundant supply of the F 7133 power distribution modules, insert jumpers in accordance with the application.

4 Reference potential L-.

Figure 24: Mono Connection to H51X Base Rack (24 VDC)

4.3.5 Mono H41X Base Rack (24 VDC)

The 24 V mono power supply is performed for the H41X base rack and the I/O processing modules by connecting to one or redundant power supply units, see Figure 19.



1 Inline terminals XG5.1…XG5.13 to 24 V power supply, XG5.13 for connecting to the I/O processing modules.

2 Connection to one or redundant power supply units, see Figure 19.

3 Connection to 24 VDC for the I/O processing module from the same source as the H41X base rack.

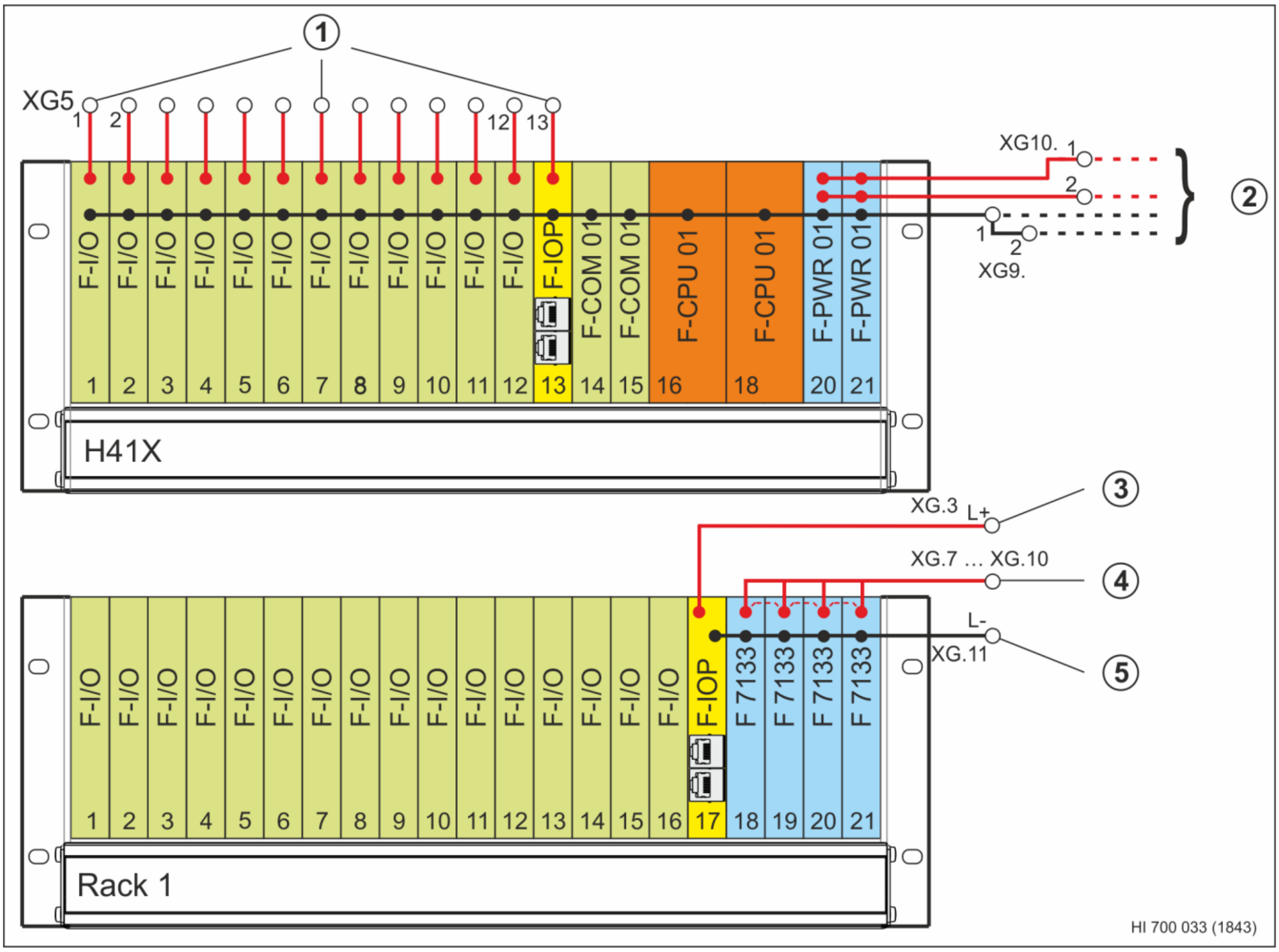
4 Redundant supply of the F 7133 power distribution modules, insert jumpers in accordance with the application.

5 Reference potential L-.

Figure 25: Mono Connection to H41X Base Rack

4.3.6 Redundant H41X Base Rack (24 VDC)

The I/O processing module in the H41X base rack and that in the extension rack must be powered from different power supply units to assemble redundant I/O levels in the HIQuad H41X system. To this end, e.g., the I/O processing module in the H41X base rack can be connected to L1+ (terminal XG5.13) and the I/O processing module in the extension rack (terminals XG.3, L+), see Figure 20. The power supply units must be able to bridge voltage dropouts of up to 20 ms.



1 Inline terminals XG5.1…XG5.13 to 24 V power supply, XG5.13 for connecting to the I/O processing module in the H41X base rack.
2 Connection to redundant power supply units, see Figure 20.
3 Terminal XG.3 (L+) for connecting to 24 VDC for the I/O processing module in extension rack 1.
4 Redundant power supply of the F 7133 power distribution modules, insert jumpers in accordance with the application.
5 Reference potential L-.

Figure 26: Redundant Connection to H41X Base Rack and Extension Rack 1

4.3.7 24 V Distribution for HIQuad X

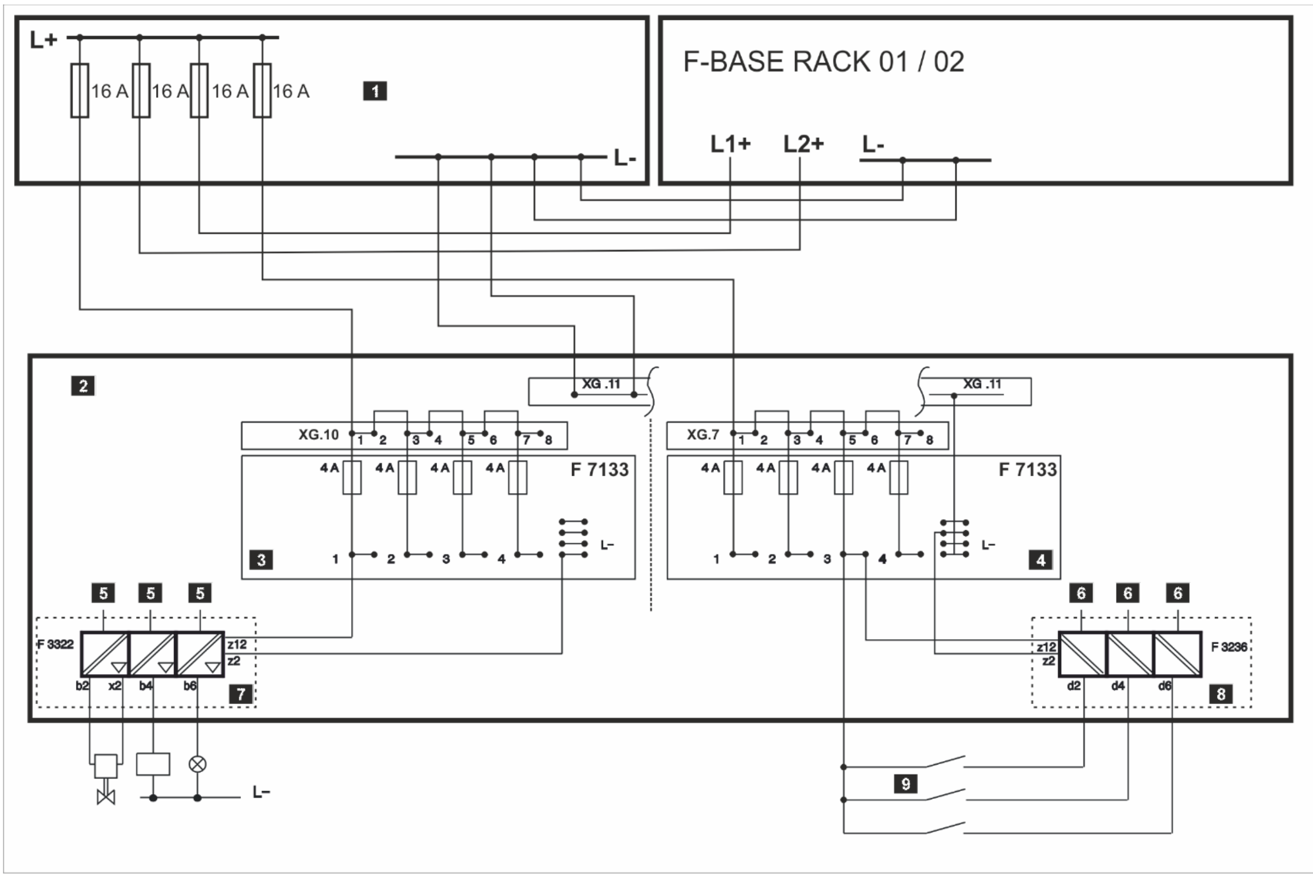
The 24 V power supply is distributed via a fuse and current distribution module connected to the base racks and extension racks.

Each extension rack may be equipped with a maximum of 4 F 7133 power distribution modules. For each F 7133, a back-up fuse (16 A) must be used in the fuse and power distribution module.

Each I/O module in the extension rack is secured by one fuse of the F 7133 power distribution module. Each F 7133 protects 4 slots with 4 A per slot. The assignment of the power distribution modules to the slots of the I/O modules is as follows:

F 7133 power distribution module	Supplies the I/O modules in
Slot 18	Slot 1…4
Slot 19	Slot 5…8
Slot 20	Slot 9…12
Slot 21	Slot 13…16

Table 15: Assignment of F 7133 Power Distribution Modules to I/O Module Slots



1 Fuse and power distribution module, see Table 41.
2 Extension rack (F-BASE RACK 11).
3 F 7133 power distribution module, slot 18.
4 F 7133 power distribution module, slot 21.
5 Output signals.
6 Input signals.
7 Output module in slot 1 (example).
8 Input module in slot 15 (example).
9 Transmitter 1…3.

Figure 27: 24 VDC Distribution for HIQuad X

The I/O modules are either supplied via the front cable plug or via the backplane PCB. The XG.11 potential distributor is connected to the L- of the fuse and power distribution module. All F 7133 power distribution modules are internally connected to the L- of the potential distributor. The L- is connected to the I/O modules through the front panel of the power distribution module via the cable plugs.

In Figure 27, the power supply of the transmitter circuits is tapped at the front of the F 7133 power distribution module. The transmitters are protected by the same fuse as the input module **8**.

4.3.8 5 V Distribution for HIQuad X

To generate 5 V power supply, a base rack can be equipped with up to 5 (H41X: 2) F-PWR 01 power supply units that are connected in parallel. The 5 V power supply is distributed to each slot via the backplane PCB. The 5 V power supply is monitored by the power supply units and its status is transmitted to the processor modules. In the user program, system variables can be used to evaluate the status of the power supply units.

4.3.8.1 5 VDC Distribution for H51X

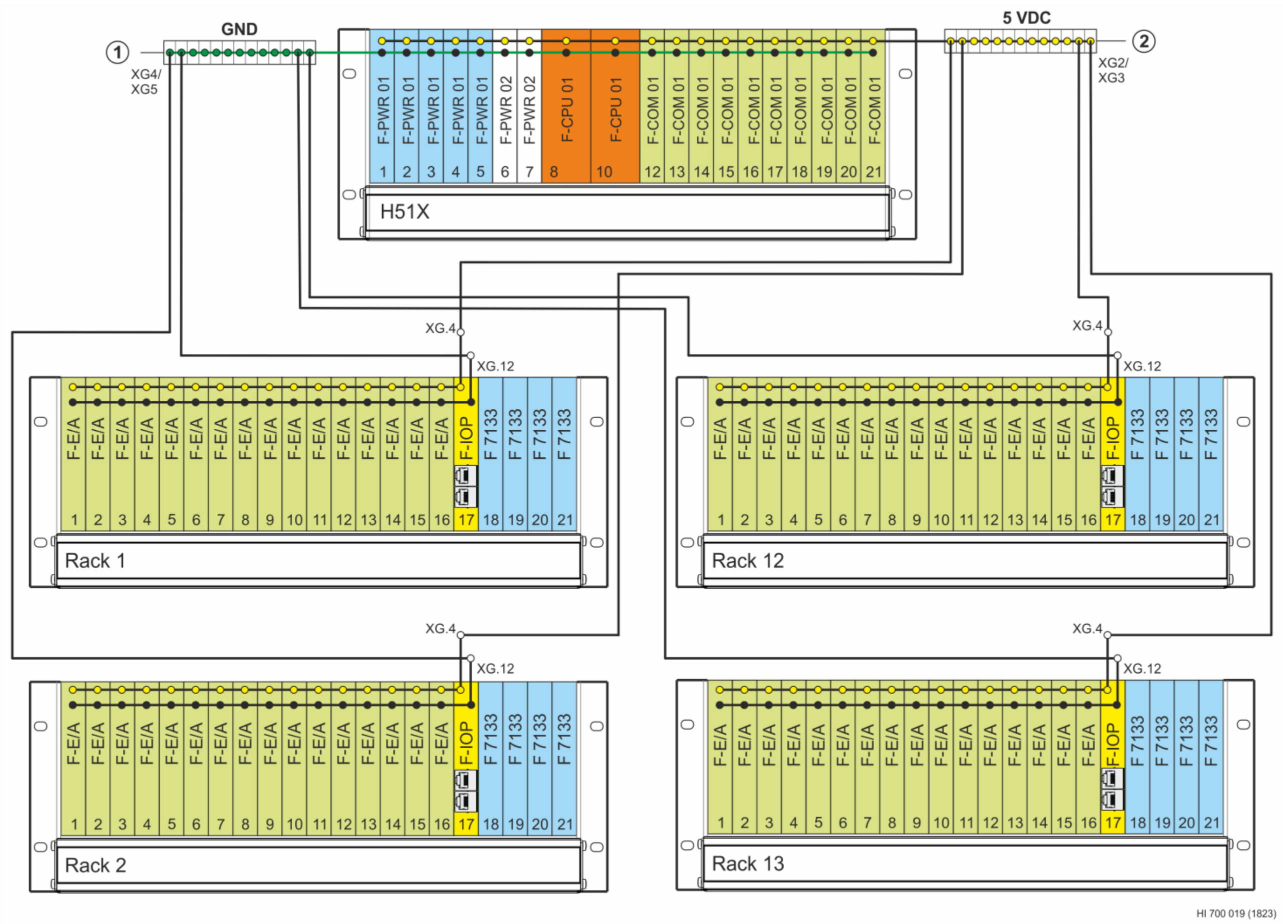
26 connection points can be used to distribute the 5 V power supply through terminal blocks XG2 and XG3 for 5 V, or XG4 and XG5 for GND. The supply voltage is distributed in a star configuration, see Figure 25. The resistance of a 5 V supply line with a maximum length between H51X base rack and extension rack of 12 m must be ≤ 40 mΩ. If cables longer than 3 m are used, HIMA recommends shielding the cables to protect them against transient interference (LIY-CY), and applying the shield at both sides as flat as possible.

To connect cables with a cross-section larger than 2.5 mm², pin terminals with a pin diameter < 2 mm or other suitable transfer terminals can be used.

The I/O processing modules (F-IOP 01) monitor the 5 V power supply of the racks on which they are installed. If the minimum voltage is underrun, I/O processing modules switch off the I/O level of their rack.

HIMA uses yellow wires for 5 V and green wires for GND. If the H51X system is distributed among several control cabinets, separate power supply units may be necessary to supply 5 V to the control cabinets without base rack, see Chapter 4.3.9 for details.

The wires on the extension racks are connected to the flat connectors XG.4 (5 V) and XG.12 (GND), and the shield to the PE connector. The voltage is distributed to the I/O modules via the backplane PCB.



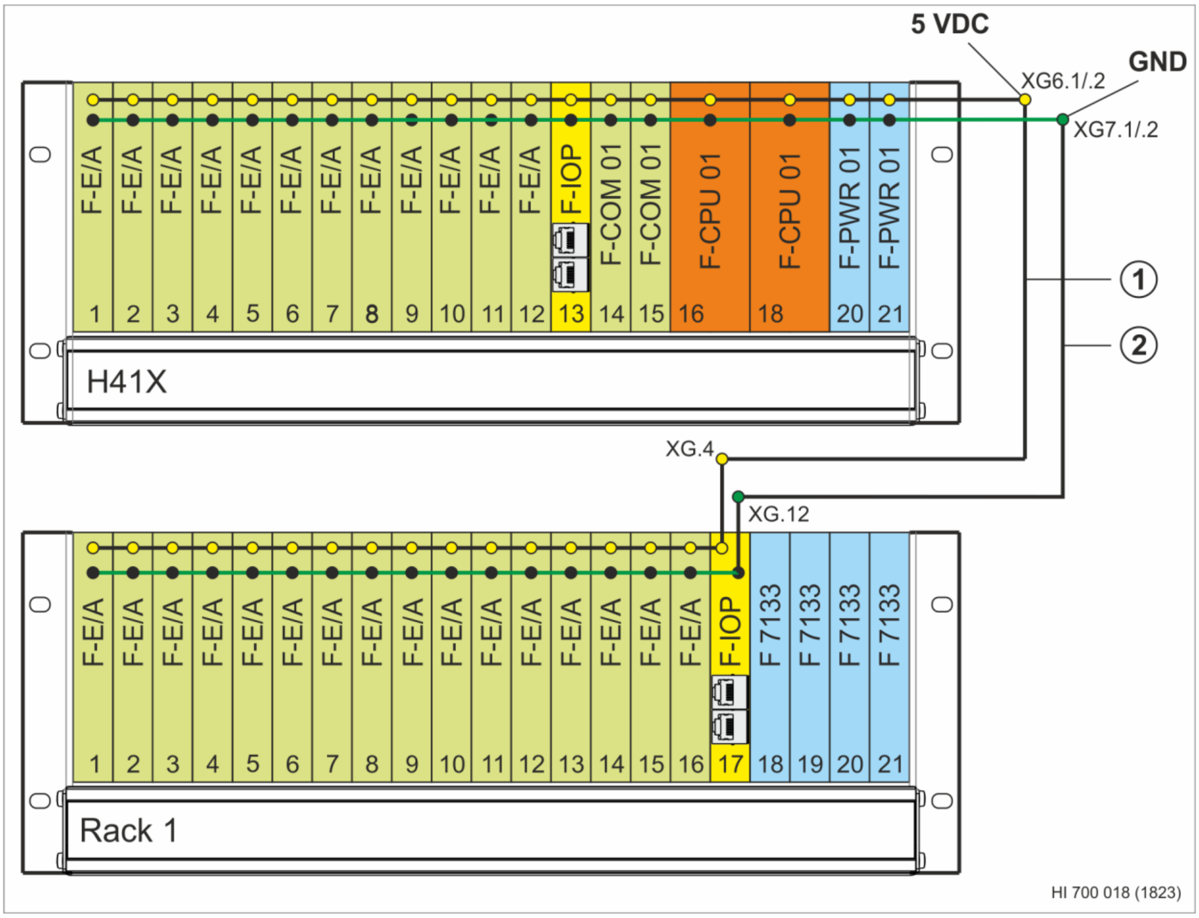
1 GND connectors (XG4 and XG5) on the rack rear side.
2 5 VDC connectors (XG2 and XG3) on the rack rear side.

Figure 28: Extension Rack Connected to a 5 VDC (H51X)

4.3.8.2 5 VDC Distribution for H41X

The 5 V power supply is distributed to the extension rack through terminal blocks XG6 for 5 V and XG7 for GND. The extension rack must be connected to 2 parallel wires (2.5 mm²) for 5 V and GND in a star configuration so that the 5 V power supply is applied to the I/O processing modules at sufficiently high voltage. HIMA uses yellow wires for 5 V and green wires for GND. The wire length within the control cabinet is limited to 3 m. If the H41X system is distributed among two control cabinets, the 5 V power supply must be provided in the control cabinet without base rack by a separate power supply unit, see Chapter 4.3.9 for details.

The wires on the extension rack are connected to the flat connectors XG.4 (5 V) and XG.12 (GND). The voltage is distributed to the I/O modules via the backplane PCB.



1 5 VDC connector XG6.1.2 (base rack), XG.4 (extension rack).
2 GND connector XG7.1.2 (base rack), XG.12 (extension rack).

Figure 29: Extension Rack Connected to a 5 VDC (H41X)

When designing the 5 V voltage supply, the current consumption of all I/O modules and the modules in the base rack must be taken into account. For details on the power consumption of the individual modules, refer to the module-specific manuals.

For HIQuad H51X:

Number of F-PWR 01 power supply units	Maximum permissible current consumption	Availability design (with a failed power supply unit tolerated!)
1	10 A	---
2	20 A	10 A
3	30 A	20 A
4	40 A	30 A
5	40 A	40 A

Table 16: Allowed Power Consumption in Relation to the Number of Power Supply Units

For HIQuad H41X:

Number of F-PWR 01 power supply units	Maximum permissible current consumption	Availability design (with a failed power supply unit tolerated!)
1	10 A	---
2	10 A	10 A

Table 17: Allowed Power Consumption in Relation to the Number of Power Supply Units

4.3.9 5 VDC Additional Power Supply (H51X)

The 5 VDC power supply can be extended by a H51X additional power supply consisting of the B 9361 set and at least an F 7126 power supply unit.

4.4 System Bus

The HIQuad X system is based on the redundant system buses A and B. Each system bus is controlled and monitored by one processor module located in the base rack. For redundant operation, the system must be operated with two processor modules. In redundant operation, communication runs on both system buses simultaneously. If only one processor module is inserted in the base rack, the system runs in mono operation with only one system bus.

Redundant operation ensures that, if one processor module fails, communication is maintained by the redundant processor module via one system bus. To ensure redundant operation again, the defective processor module must be replaced immediately.

It is not allowed to interconnect the system buses of several HIQuad X systems!

No active elements such as switches may be connected to the system bus.

NOTICE

System malfunction possible!

Using system bus connectors XD1...XD4 on the back of the backplane PCB as normal Ethernet connections may cause the system to malfunction.

- **Only use the system bus connectors XD1...XD4 to connect to the I/O processing modules (F-IOP 01).**
- **Do not interconnect or cross system bus A and system bus B.**

The system buses connect the I/O level to the processor modules via the I/O processing modules (F-IOP 01). To do so, the RJ-45 interfaces on the rear side of the base racks must be connected to the I/O processing modules, see Chapter 3.2. The maximum length of the patch cable between two system bus subscribers is 50 m. The cable diameter must be selected in relation to the cable length.

The patch cables for the system buses must meet the following requirements:

- At least Cat. 5e (in accordance with IEEE 802.3) for 1 Gbit/s, for industrial applications.
- Industrial RJ-45 connectors on both sides.
- The cable shielding must comply with at least Class D in accordance with ISO/IEC 11801.
- Autocrossover allows the use of both crossover and straight through cables.

Suitable patch cables with industrial connector are available from HIMA in standard lengths.

NOTICE

Communication interference possible!

Use patch cables compliant with industrial standard Cat. 5e or better!

In harsh environments (e.g., subject to temperature changes, electromagnetic interference), low-quality patch cables may cause communication to fail.

The maximum system bus latency can be set to System Defaults or 100 µs using the *Maximum System Bus Latency [µs]* system parameter located in the resource properties. When the Maximum System Bus Latency [µs] is set to System Defaults, the maximum system bus latency is determined by the system. For the 100 µs setting, the maximum system bus latency is set to this value!

For system bus connections running within a control cabinet, the minimum cross-section of patch cables must be 0.2 mm².

For system bus connections running outside a control cabinet, the minimum cross-section of patch cables must be 0.5 mm². If necessary, installation cables with rigid cores must be used instead of patch cables with flexible cores.

4.5 I/O Bus

All I/O modules are connected to the I/O processing module via the I/O bus. The I/O processing module in the H41X base rack (slot 13) and in the extension rack (slot 17) connect the I/O bus to the system buses.

4.6 I/O Watchdog (WD)

A second independent shutdown option is required in safety-related systems. This is ensured by an I/O watchdog signal (24 V). The I/O watchdog is controlled, monitored, and applied to the output modules by the I/O processing modules. The output modules only operate when the watchdog signal is present (high level). If the I/O watchdog signal is switched off, the output modules safely enter the de-energized state.

4.7 Modules

The HIQuad X system is a modular system that can be equipped with various modules. The following modules are available for the system:

- F-CPU 01 processor module.
- F-IOP 01 I/O processing module.
- F-COM 01 communication module.
- I/O modules, see Chapter 4.11.
- F-PWR 01 power supply unit (24/5 V).
- F-PWR 02 buffer module.

4.8 F-CPU 01 Processor Module

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

4.8.1 Operating system

Tasks:

- Controlling the cyclic run of the user programs.
- Performing the self-tests of the module.
- Controlling safety-related communication via safe**ethernet.**
- Managing the processor modules' redundancy (synchronization).

4.8.1.1 General Cycle Sequence

Phases:

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

4.8.1.2 Operating System States

States that can be recognized by the user:

- LOCKED
- STOP/VALID CONFIGURATION
- STOP/INVALID CONFIGURATION
- STOP/LOADING OS
- RUN
- RUN/UP STOP

Use the LEDs on the module to recognize the operating state. All LEDs must be taken into account, see the module-specific manuals.
SILworX displays the operating states in the online view.

Table 18 provides an overview of the operating system states and indicates the conditions for entering them.

State	Description	The state is entered:
LOCKED	The processor module is reset to the factory settings (SRS, network settings, etc.).	Connecting the supply 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.
		Applying the supply voltage ▪ Autostart is disabled in the project configuration. ▪ Mode switch is set to *Stop* and the processor module starts by itself.
		A fault occurred.
STOP/INVALID CONFIGURATION	Processor module stopped: No valid configuration is available in the memory.	Loading with error.
STOP/LOADING OS	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.	From the STOP/VALID CONFIGURATION state: SILworX command.
		Applying the supply voltage, the following conditions must be met: ▪ 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.
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 through a SILworX command.

Table 18: Operating System States, States Entered

Table 19 specifies how the user may intervene during the corresponding states.

State	Possible user interventions:
LOCKED	▪ Changing the factory settings. ▪ Using a PADT command to stop (STOP state). ▪ Using a PADT command to start (RUN state).
STOP/VALID CONFIGURATION	▪ Loading the user program. ▪ Starting the user program. ▪ Loading the operating system. ▪ Taking preliminary actions for forcing variables.
STOP / INVALID CONFIGURATION	▪ Loading the user program. ▪ Loading the operating system.
STOP/LOADING OS	None. Once the loading process is completed, the processor module stops (STOP state).
RUN	▪ Stopping the user program. ▪ Forcing variables. ▪ Performing the test.
RUN/UP STOP	▪ Using a PADT command to stop (STOP state).

Table 19: Operating System States, User Interventions

i The cycle time increases by the number of modules used in the system. This applies irrespective of whether or not the modules are included in the configuration.

- **Connecting additional extension racks with several modules during operation can cause the watchdog time to be exceeded!**

4.8.2 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 a fault is still present, the module restarts with reduced functionality to prevent a reboot loop.

Once the processor module has properly run for one minute, the next error stop to occur is considered the first *error stop* attempting a restart.

i Use the PADT for troubleshooting and removing the cause of the fault, e.g., by loading a new application.

4.9 F-IOP 01 I/O Processing Module

The I/O processing module manages the I/O bus of the H41X base rack and that of the extension racks. The I/O bus is used to exchange process data between I/O modules and the I/O processing module. The module's tasks include exchanging data with the processor modules and providing the watchdog signal to the output modules via system bus A and system bus B.

4.10 F-COM 01 Communication Module

The communication module is equipped with 2 Ethernet interfaces and 1 fieldbus interface allowing the HIQuad X system to communicate with external systems. The module is approved for use in the safety-related HIQuad X system and can be employed to transport safety-related protocols.

4.11 I/O Modules

The following table shows the I/O modules that can be used for HIQuad X:

Module	Cable plug	Channels	SIL	Type	HI number Data Sheet
F 3221	Z 7116 / 3221	16	---	DI	HI 803 174 E
F 3224A	Z 7114 / 3224	4	---	DI; (Ex)i	HI 803 175 E
F 3236	Z 7116 / 3236	16	3	DI	HI 803 176 E
F 3237	Z 7108 / 3237	8	3	DI	HI 803 177 E
F 3238	Z 7008 / 3238	8	3	DI; (Ex)i	HI 803 178 E
F 3240	Z 7130 / 3240	16	3	DI 110 VDC	HI 803 179 E
F 3248	Z 7130 / 3248	16	3	DI 48 VDC	HI 803 180 E
F 3322	Z 7136 / 3322	16	---	DO 0.5 A	HI 803 181 E
F 3325	Z 7025 / 3325	6	---	Supply module	HI 803 182 E
F 3330	Z 7138 / 3330	8	3	DO 0.5 A	HI 803 183 E
F 3331	Z 7138 / 3331	8	3	DO 0.5 A	HI 803 184 E
F 3333	Z 7134 / 3333	4	3	DO 2 A	HI 803 185 E
F 3334	Z 7134 / 3334	4	3	DO 2 A	HI 803 186 E
F 3335	Z 7035 / 3335	4	3	DO; (Ex)i	HI 803 187 E
F 3349	Z 7150 / 3349	8	3	DO 0.5 A	HI 803 188 E
F 3422	Z 7139 / 3422	8	---	Relay 60 VDC	HI 803 189 E
F 3430	Z 7149 / 3430	4	3	Relay 110 VDC	HI 803 190 E
F 5220	Z 7152 / 5220	2	3	Counter	HI 803 191 E
F 6215	Z 7127 / 6215	8	---	AI	HI 803 192 E
F 6217	Z 7127 / 6217	8	3	AI	HI 803 193 E
F 6220	Z 7062 / 6220	8	3	Thermocouple; (Ex)i	HI 803 194 E
F 6221	Z 7063 / 6221	8	3	AI; (Ex)i	HI 803 196 E
F 6705	Z 7126 / 6705	2	3	AO	HI 803 197 E
F 6707	Z 7126 / 6706	2	---	AO	HI 803 198 E

Table 20: Possible I/O Modules to Be Used in HIQuad X

4.11.1 Scope of Application of the I/O Modules

Refer to the safety manual (HI 803 209 E) for more information on the standards used to certify the I/O modules.

4.11.2 Mounting Position

The I/O modules must be mounted vertically. The vertical mounting position automatically results from the horizontal position of the rack within a control cabinet.

4.12 Noise Blanking

This chapter describes how noise blanking of I/O modules operates in the HIQuad X system.

4.12.1 Effects of Noise Blanking

Noise blanking suppresses transient interference to increase the system availability. It ensures that the system triggers a safety-related response to existing interferences within the configured time.

Noise blanking can be activated for I/O modules. For details, refer to the SILworX Hardware Editor and the module-specific manuals.

If an interference is blanked out, the system automatically processes the last valid input and output values instead of the currently disturbed values. The time in which noise can be blanked out is limited by the safety time, watchdog time and the cycle time.

The maximum noise blanking time can be calculated using the following equation:

Maximum noise blanking time = safety time - (2 x watchdog time)

The greater the noise blanking time value, the longer the interference can be blanked out. Since an interference can be present for up to one cycle before it is detected while reading in the values, the minimum noise blanking time can be determined by subtracting a cycle from the maximum noise blanking time value.

Minimum noise blanking time = maximum noise blanking time - cycle time

Noise blanking is effective if the cycle time value is less than the noise blanking time.

4.12.2 Configuring Noise Blanking

To blank out as many cycles as possible, the safety time must be set as large as possible taking the process safety time into account. At the same time, the value set for the watchdog time should be as low as possible, but sufficiently large to allow reload and synchronization of an additional processor module. Refer to the safety manual (HI 803 209 E) for further details on the various time parameters and their application.

Configure noise blanking in accordance with the following examples:

Example	1 [1)]	2	3 [2)]
Safety time [ms]	600	2000	1000
Watchdog time [ms]	200	500	500
Target cycle time [ms]	100	200	200
Maximum noise blanking time [ms]	200	1000	0
Minimum noise blanking time [ms]	100	800	0

[1)] Default setting in SILworX.

[2)] No noise blanking is possible in example 3 since the noise blanking time is less than the cycle time.

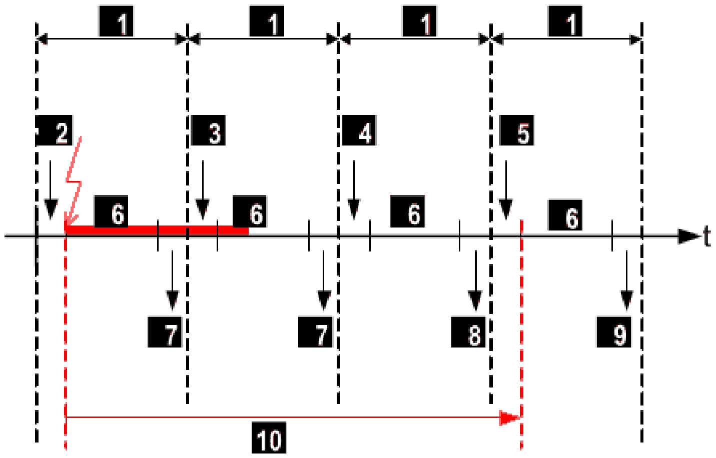
Table 21: Examples of Calculating the Min. and Max. Noise Blanking Time

4.12.3 Noise Blanking Sequence

The following examples illustrate the sequence of noise blanking:

- A transient interference is blanked out.
- An interference present for longer than the maximum noise blanking time triggers the safe response.

Example 1: Transient interference is successfully blanked out



1 Cycle, duration = watchdog time.	6 Processing (in all cycles).
2 Reading in cycle 1.	7 Output process in cycle 1 and 2.
3 Reading in cycle 2.	8 Output process in cycle 3.
4 Reading in cycle 3.	9 Output process in cycle 4.
5 Reading in cycle 4.	10 Duration of safety time.

Figure 30: Transient Interference

In example 1, valid input values 2 are read within one cycle. For this cycle, the system processes the valid input values, even though an interference occurred directly upon completion of the read-in process.

If the interference is still present in the following cycle during the read-in process 3, the module detects the interference and the system decides if noise blanking can be performed at this point in time based on the following rule:

Safety time - elapsed time - (2 x watchdog time) > 0

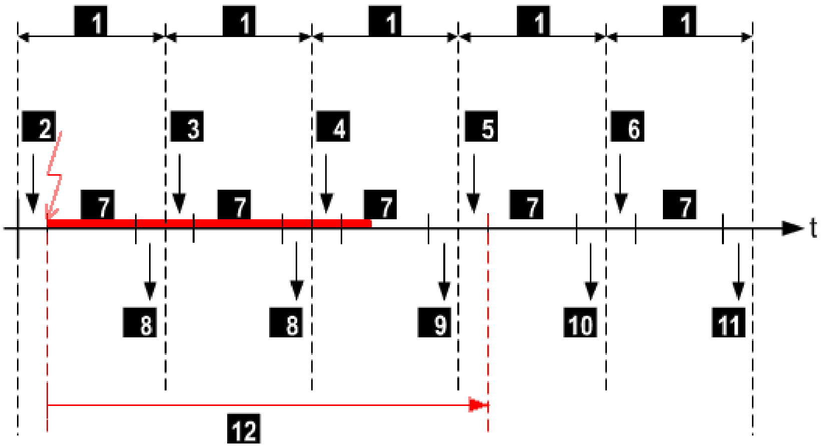
Elapsed time = Time interval between the moment, in which the last valid values were read in, and the moment, in which the interference was detected.

In this example, noise blanking is possible since the interference is present for less than a cycle (= elapsed time) and two additional cycles (2 x watchdog time) are available for triggering a safe response. For this cycle, the system processes the last valid input values of 2 and no fault response is triggered. The transient interference was successfully blanked out.

If the interference is no longer present in 4, new valid values are read in and processed.

If noise blanking is not active, the system immediately triggers the defined fault response during the read-in process 3.

Example 2: Triggering a safety-related response when interference occurs



1	Cycle, duration = watchdog time.	7	Processing (in all cycles).
2	Reading in cycle 1.	8	Output process in cycle 1 and 2.
3	Reading in cycle 2.	9	Output process in cycle 3.
4	Reading in cycle 3.	10	Output process in cycle 4.
5	Reading in cycle 4.	11	Output process in cycle 5.
6	Reading in cycle 5.	12	Duration of safety time.

Figure 31: Interference Triggers a Safe Response

In example 2, valid input values 2 are read within one cycle. For this cycle, the system processes the valid input values, even though an interference occurred directly upon completion of the read-in process.

If the interference is still present in the following cycle during the read-in process 3, the module detects the interference and the system decides if noise blanking can be performed at this point in time based on the following rule:

Safety time - elapsed time - (2 x watchdog time) > 0

Noise blanking is possible in the 1st and 2nd cycle since the interference is present for less than a cycle (= elapsed time) and two additional cycles (2 x watchdog time) are available for triggering a safe response. For this cycle, the system processes the last valid input values of 2 and no defined fault response is triggered. The transient interference was successfully blanked out.

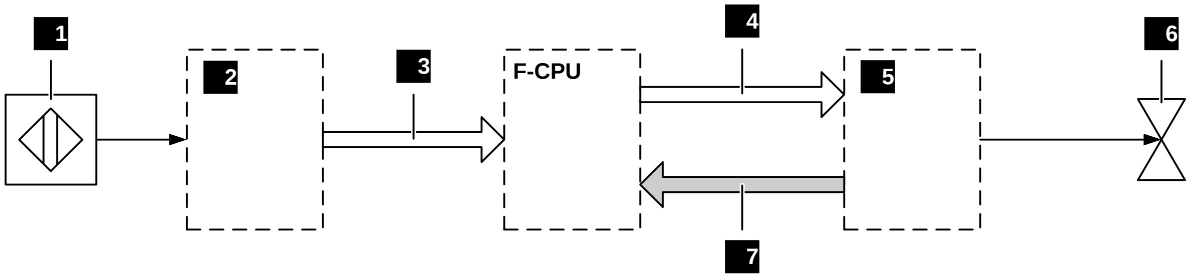
In case of a ratio of safety time/watchdog time = 3/1, as in example 2, 2 cycles are still available for the safe response

If the interference is still present in the next read-in process 4, the fault response must be triggered in that cycle. The fault response must be triggered no later than when the outputs are written to 9. At the next output moment 10, the safety time has already expired.

If noise blanking is not active, the system immediately triggers the defined fault response during the read-in process 3.

4.12.4 Effective Direction of Noise Blanking

The effective direction must be observed when considering noise blanking and output noise blanking, see Figure 32 and the following chapters.



1 Sensor.
2 Input module.
3 Effective direction from the input module to the processor module.
4 Effective direction from the processor module to the output module.
5 Output module.
6 Actuator.
7 Effective direction from the output module to the processor module.

Figure 32: Effective Direction Associated with Noise Blanking and Output Noise Blanking

4.12.4.1 Effective Direction from the Input Module to the Processor Module (3)

Noise blanking with effective direction from the input module to the processor module is performed by the processor module. Noise blanking suppresses the transient interference on the input module and on the buses (system bus and I/O bus). Noise blanking on the input module can be deactivated in the properties (SILworX) (default = Activated), see the input module manuals. Noise blanking on the buses is always active and cannot be deactivated in SILworX.

4.12.4.2 Effective Direction from the Processor Module to the Output Module (4)

Noise blanking with effective direction from the processor module to the output module is performed by the output module and is always active. Noise blanking suppresses the transient interference on the bus.

4.12.4.3 Effective Direction from the Processor Module to the Output Module (7)

Noise blanking with effective direction from the output module to the processor module on the system bus is performed by the processor module. Noise blanking suppresses status acknowledgments of the output module such as SC/OC detection. Noise blanking on the output module can be deactivated in the properties (SILworX) (default = Activated), see the output module manuals.

4.13 Communication

Communication with other HIMA systems or third-party systems occurs via communication modules. HIQuad X supports the following communication protocols:

- safe**ethernet** (safety-related)
- Standard protocols

For details on communications and supported standard protocols, refer to the communication manual (HI 801 101 E).

4.13.1 Licensing Protocols

Standard protocols can only be run in the long term with a valid license. For some protocols, a software activation code is required. For activation, see Chapter 4.15.

i Order a software activation code on time!

After 5000 operating hours, communication continues until the controller is stopped. Afterwards, the user program cannot be started without a valid software activation code for the protocols used in the project (invalid configuration).

4.14 Connecting the PADT to the System

The physical connection between the PADT and a HIQuad X system is established by connecting the Ethernet interface of a PC to an RJ-45 socket (X1, X2) of a processor module. A PADT (programming and debugging tool) is a personal computer (PC) that is running the SILworX programming tool.

The connection requires a patch cable that complies with Cat. 5e or better and is connected to a free PC network card.

A HIQuad X system can simultaneously communicate with up to 5 PADTs. If this is the case, only one programming tool can access the controller with write permission. The remaining PADTs can only read information. If they try to establish a writing connection, the controller only allows them a read-only access.

4.15 Licensing

A license is required for using some communication protocols of the HIQuad X system, refer to the communication manual (HI 801 101 E).

The licenses can be obtained from HIMA for a fee. To activate the function, HIMA provides an activation code which can be entered with the PADT in the configuration. The activation code is bound to the system ID of the PES.

The activation code is generated on the HIMA website at: `www.hima.com/en` under Products & Services -> Product Registration. Refer to the corresponding page for more details.

To activate a function with an activation code

1. Generate the software activation code on the HIMA website `www.hima.com/en` using the system ID of the controller (e.g., 10 000) and the license numbers received from HIMA. To do so, follow the instructions provided on the HIMA website:

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

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 function is activated.

5 Redundancy

The conceptual design of the HIQuad X system is characterized by high availability. To this end, all system components can be operated redundantly. The following chapter describes redundancy aspects for the various system components.

i A redundancy system only increases the system availability, but not its safety integrity level (SIL)!

5.1 Processor Module Redundancy

A HIQuad X system can be configured as a mono system with only 1 processor module, or as a high-availability redundancy system with 2 redundant processor modules.

Processor modules only operate redundantly, if there is a digital depiction of the processor modules in the Hardware Editor (SILworX) and the configuration is compiled with these settings.

5.1.1 Reducing Redundancy

A HIQuad X system with redundant 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 still ensured even if one of the two redundant processor modules fail.

5.1.2 Increasing Redundancy

If a new processor module is added to a running HIQuad X system, it automatically synchronizes with the configuration of the existing processor module. Safety-related operation is ensured.

Requirements:

- The user program run by the processor module is redundantly configured (default setting).
- The redundant slot for the processor module is not in use.
- At least one system bus is operating.
- The position of the mode switch on the added processor module is *Stop* or *Run*.
- The operating system of the added processor module has either the same version as the existing processor module or a higher one.

5.2 Redundancy of I/O Modules

In terms of redundancy, two cases can be distinguished for input and output modules:

- Module redundancy.
- Channel redundancy.

HIMA recommends defining module redundancy before channel redundancy. HIQuad X supports dual redundancy of input and output modules. Higher redundancy levels are possible using the corresponding programming logic. For redundant connection of a sensor or actuator to several I/O modules, observe the input and output values permitted for I/O modules.

5.2.1 Module Redundancy

Two I/O modules of the same type can be combined in the SILworX Hardware Editor to form a redundancy group. To ensure that availability of HIQuad X is increased as required, the I/O modules of the redundancy group must be inserted in different racks:

i To increase availability of HIQuad X, redundant I/O modules must be inserted in different racks:

5.2.2 Channel Redundancy

Channel redundancy is only possible within a redundancy group. Only channels with the same channel number can be defined as redundant. In such cases, the programming tool automatically allocates a global variable, which is assigned to a channel (channel number), to both channels of the redundant modules. For details, refer to the Hardware Editor section of the SILworX online help.

5.3 System Bus Redundancy

The HIQuad X system can be operated with redundant system bus A and system bus B, see Chapter 3.

Requirements for redundant operation:

- Use of 2 processor modules per base rack.
- Suitable configuration in the programming tool.
- Connection of the racks in a controller, see Chapter 3.

5.4 Communication Redundancy

For further details, refer to the SILworX online help and communication manual (HI 801 101 E).

5.4.1 safe**ethernet**

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

5.4.2 Standard Protocols

If standard protocols are used, the user program must manage redundancy, except for Modbus slaves.

6 Programming

The user programs for the HIQuad X system must be created using a 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 FBD editor and graphically interconnected. Based on the resulting graphical representation, SILworX generates an executable program that can be loaded into the controller.

For further details on the programming tool, refer to the SILworX online help.

6.1 Connection of the Programming Tool

The PC with the SILworX programming tool (PADT) is connected to the HIQuad X system via Ethernet. The following interfaces are available:

- Ethernet interfaces of the communication modules (F-COM).
- Ethernet interfaces of the processor modules (F-CPU).

6.1.1 Use of Ethernet Interfaces

Depending on the system structure, the Ethernet interfaces must be used in the following order for reasons of automation security:

1. Ethernet interface to an F-COM.
2. Ethernet interface to an F-CPU.

6.2 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 also 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.

Local and global variables exist. 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 user programs and can exchange data between the function blocks.

Global variables can be created at different project tree levels. The global variables apply to all subordinate levels within the scope.

Example: If a project contains several resources, the global variables created under a resource are only valid for the branches subordinated to that resource.

Hierarchy of the levels at which global variables can be defined.

1. Project.
2. Configuration.
3. Resource.

Values may only be written to global data in one program location! The possible sources are:

- Logic in a user program.
- Inputs (safety-related).
- System variables.
- Communication protocols (safety-related).

Writing to global variables at multiple positions within the program can result in unintended effects!

In the Global Variable Editor, check the usage of global data with the *Cross-Reference in Column* function.

6.2.1 Variable Types

SILworX supports the following variable types:

- VAR, a variable within a logic (read and write).
- VAR with the CONST attribute, a variable that was defined as constant and cannot be modified.
- VAR with the RETAIN attribute, a variable that retains its value after a power outage.
- VAR_EXTERNAL, reference to global variables (read and write).
- VAR_GLOBAL, global variable (read and write) to exchange values between programs and subordinated functions and function blocks.
- VAR_INPUT, input variable (read) of a POU. It is also displayed in the interface viewer.
- VAR_OUTPUT, output variable (write) of a POU. It is also displayed in the interface viewer.
- VAR_TEMP, temporary variable (read and write).
- VAR_ACTION, action declaration (read and write).

Which type can be assigned to a variable depends on the hierarchy of the variables in the structure tree. The following table shows the permissible variable types in connection with the structure tree nodes.

Type	Project	Configuration	Resource	Program type	Function block type	Function type
VAR				X		
VAR_EXTERNAL				X		
VAR_GLOBAL	X	X	X	(1)		
VAR_INPUT				(1)	X	X
VAR_OUTPUT				(1)	X	X
VAR_TEMP				(2)	X	
VAR_ACTION				(2)	X	
(1) Contrary to the standard, this function is not supported. (2) Contrary to the standard, VAR_ACTION is supported.						

Table 22: Supported Variable Types

6.2.2 Initial Value

An initial value can be allocated 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 in one of the following sources from which the variable derived its value, for example:
  - Physical input.
  - Communication interface.
  - User program in the STOP state.

Initial values are specified as follows:

- The initial value that is in accordance with the data type declaration is used. For elementary data types, the initial values in accordance with IEC 61131-3, Table *Ranges of Values and Initial Values for the Data Types* apply. This initial value has the lowest priority.
- The initial value specified by the user for a derived data type, applies. The initial value of the basis type is no longer inherited.
- The initial value explicitly specified by the user for a variable in the variable declaration, applies.
- The instance-specific initial value that was already defined at the time of initialization, applies. This initial value has the highest priority.

The value that the connected variable should adopt can be set for safe**ethernet** and communication protocols.

i In safety-related applications, a safe value must be assigned as initial value to all variables that receive their value from a physical input or from the communication.

6.2.3 System Variables and System Parameters

System variables are pre-defined variables for processing properties or states of the HIQuad X system in the user program. To define them, they are assigned global variables used in the user program.

The system parameters are used to configure properties of the controller (only possible in SILworX). System parameters that can only have the values TRUE and FALSE are also referred to as switches.

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

Project level	Description of the system variables and parameters
Resource	See Table 24.
Hardware, in general	▪ System variables for configuring the controller, see Table 24. ▪ System variables providing information, see Table 26 and Table 27.
Hardware: Modules	Refer to 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 Chapter 6.2.6.

Table 23: System Variables at Different Project Levels

6.2.4 Resource System Parameters

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

Parameters	S 1)	Description	Setting for safe operation
Name	N	Name of the resource.	Any
System ID [SRS]	Y	System ID of the resource. Range of values: 1…65 535 Default value: 60 000 The value assigned to the system ID must differ from the default value, otherwise the project is not able to run!	Unique value within the controller network. This network includes all controllers that can potentially be interconnected.
Safety Time [ms]	Y	For details on the safety time of the resource (in milliseconds), refer to the safety manual (HI 801 003 E). Range of values: 20…22 500 ms Default value: 600 ms (can be changed online)	Application-specific
Watchdog Time [ms]	Y	For details on the watchdog time of the resource (in milliseconds), refer to the safety manual (HI 801 003 E). Range of values: 6…7500 ms Default value: 200 ms (can be changed online)	Application-specific
Target Cycle Time [ms]	N	Target or maximum cycle time, see *Target Cycle Time Mode*. Range of values: 0…7500 ms Default value: 0 ms (can be changed online) The maximum target cycle time value may not exceed the configured *Watchdog Time [ms]* minus the minimum value that can be set for *Watchdog Time [ms]* (6 ms, see above); otherwise the entry is rejected. If the default value is set to 0 ms, the target cycle time is not taken into account. For further details, refer to the following chapters.	Application-specific
Target Cycle Time Mode	N	For details on the use of the *Target Cycle Time [ms],* see the following chapters. The default setting is *Fixed-tolerant* (can only be changed online).	Application-specific
Multitasking Mode	N	Mode 1 The duration of a CPU cycle is based on the required execution time for all user programs. Mode 2 The processor provides the execution time portion not needed by lower priority user programs to higher priority user programs. Operation mode for high availability. Mode 3 The processor waits until the execution time not needed by the user programs has expired, thus increasing the cycle. Default value: Mode 1	Application-specific
Max. Com.Time Slice [ms]	N	Highest value in ms for the time slice used for communication during a resource cycle, see the communication manual (HI 801 101 E). Range of values: 2…5000 ms Default value: 60 ms	Application-specific

<table>
<tr><th>Parameters</th><th>S 1)</th><th>Description</th><th>Setting for safe operation</th></tr>
<tr><td>Optimized Use of Com. Time Slice</td><td>N</td><td>The system parameter reduces the response times for communications via processor module(s).

i This can affect the temporal utilization of Max.Com. Time Slice ASYNC [ms] and the system parameter Max. Duration of Configuration Connections [ms] such that these two times can be subject to more demands (e.g., during reload).</td><td>---</td></tr>
<tr><td>Max. Duration of Configuration Connections [ms]</td><td>N</td><td>This defines how much time within a CPU cycle is available for configuration connections.
Range of values: 2...3500 ms
Default value: 20 ms
For further details, refer to the following chapters.</td><td>Application-specific</td></tr>
<tr><td>Maximum System Bus Latency [µs]</td><td>N</td><td>Maximum delay of a message between an I/O processing module and a processor module.
Settings: System defaults or 100…50,000 µs
Default value: System Defaults

i A license is required for setting the maximum system bus latency to a value ≠ System Defaults.</td><td>---</td></tr>
<tr><td>Allow Online Settings</td><td>Y</td><td>TRUE: All the switches/parameters listed under FALSE can be changed online using the PADT. This is only valid if the system variable Read-only in RUN has the value FALSE.
Default value: TRUE.

FALSE: The following parameters cannot be changed online:
▪ System ID
▪ Autostart
▪ Global Forcing Allowed
▪ Global MultiForcing Allowed
▪ Global Force Timeout Reaction
▪ Load Allowed
▪ Reload Allowed
▪ Start Allowed
The following parameters can be changed online if Reload Allowed is TRUE.
▪ Watchdog Time (for the resource)
▪ Safety Time
▪ Target Cycle Time
▪ Target Cycle Time Mode

Allow Online Settings can only be TRUE when the controller is stopped or by performing a reload.</td><td>HIMA recommends using the FALSE setting.</td></tr>
</table>

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

Parameters	S [1)]	Description	Setting for safe operation
Minimum Configuration Version	N	The default setting is based on the current SILworX version. It ensures compatibility with future SILworX versions. Code is generated in accordance with SILworX V10 conventions, since HIQuad X is supported as of SILworX V10. Any setting to a SILworX version prior to V10 is rejected for HIQuad X. An error message is displayed in the logbook! For further details, refer to the chapter on the *Minimum Configuration Version* parameter.	Application-specific
Fast Start-Up	Y	Not applicable to HIQuad X.	---
[1)] The operating system handles the system parameter in a safety-related manner, yes (Y) or no (N).			

Table 24: Resource System Parameters

6.2.4.1 Notices on the *Minimum Configuration Version* Parameter

The following notices apply to the *Minimum Configuration Version* parameter:

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

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

  The safety-related SILworX version comparison can be used to determine and prove changes performed to the current project version compared to a previous one.
- For HIQuad X, *Minimum Configuration Version* must be set to *SILworX V10* or higher.

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

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

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

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

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

Table 25: Settings for Target Cycle Time Mode

6.2.4.3 Maximum Communication Time Slice

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

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

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

6.2.4.4 Determining the Maximum Duration of the Communication Time Slice

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

- For each communication module (F-COM 01): 3 ms.
- For each redundant safe**ethernet** connection: 1 ms.
- For non-redundant safe**ethernet** connection: 0.5 ms.
- For each kilobyte user data of non-safety-related protocols, e.g., Modbus: 1 ms.

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

To determine the actual duration of the maximum communication time slice

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

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

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

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

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

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

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

$\mathbf{t}_{Config}$ for F-CPU 01 processor modules is calculated as follows:

F-CPU 01: $\mathbf{t}_{Config} = (\mathbf{n}_{Com} + \mathbf{n}_{PADT}) * 1\text{ ms} + \mathbf{n}_{RIO} * 0.25\text{ ms} + 4\text{ ms} + 4 * (\mathbf{t}_{Latency} * 2 + 0.8\text{ ms})$

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

If *System Defaults* is selected for the *Maximum System Bus Latency [µs]* parameter, the value 2.2 ms must be used in the upper formula. If a value of 100…50 µs was manually entered for $t_{Latency}$, then this value must be used in the upper formula as $t_{Latency}$.

TIP The current system bus latency is displayed in the Control Panel.

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

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

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

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

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

6.2.5 Rack System Variables

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

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

Table 26: Rack System Variables

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

6.2.5.1 Input variables for reading out parameters.

The input variables can be read out and assigned to a global variable in the *System* tab located in the rack detailed view of the SILworX Hardware Editor.

The input variables are selected in the *Input Variables* column through a tick.

System variables	S [1)]	Description	Data type
Number of Field Errors	N	Number of current field faults.	UDINT
Number of Field Errors - Historic Count	N	Counted number of field faults (counter resettable).	UDINT
Number of Field Warnings	N	Number of current field warnings.	UDINT
Number of Field Warnings - Historic Count	N	Counted number of field warnings (counter resettable).	UDINT
Number of Communication Errors	N	Number of current communication errors.	UDINT
Number of Communication Errors - Historic Count	N	Counted number of communication errors (counter resettable).	UDINT
Number of Communication Warnings	N	Number of current communication warnings.	UDINT
Number of Communication Warnings - Historic Count	N	Counted number of communication warnings (counter resettable).	UDINT
Number of System Errors	N	Number of current system errors.	UDINT
Number of System Errors - Historic Count	N	Counted number of system errors (counter resettable).	UDINT
Number of System Warnings	N	Number of current system warnings.	UDINT
Number of System Warnings - Historic Count	N	Counted number of system warnings (counter resettable).	UDINT
Autostart	Y	TRUE: When the processor module is connected to the supply voltage, it automatically starts the user program. FALSE: When the supply voltage is connected, the processor module enters the STOP state.	BOOL
OS Major [1]…[2] OS Minor [1]…[2]	Y	Version of the operating system for every processor module. The number of redundant processor modules and the values depend on the controller type.	UINT
CRC	Y	Resource configuration checksum.	UDINT
Date/time [ms portion] Date/time [s portion]	N	System date and system time in milliseconds and seconds since 1970-01-01: 0…999 ms, 0…4 294 967 295 s.	UDINT
Force Deactivation	Y	TRUE: Forcing is deactivated. FALSE: Forcing is possible.	BOOL
Forcing Active	Y	TRUE: Global or local forcing is active. FALSE: Global and local forcing are not active.	BOOL
Force Switch State	N	Information about the selected force switch. 0xFFFF FFFE: No force switch set. 0xFFFF FFFF: At least one force switch set.	UDINT
Global Forcing Started	Y	TRUE: Global forcing is active. FALSE: Global forcing is not active.	BOOL

System variables	S [1)]	Description	Data type
Last Field Warning [ms] Last Field Warning [s]	N	Date and time of the last field warning in milliseconds and seconds since 1970-01-01: 0…999 ms, 0…4 294 967 295 s.	UDINT
Last Communication Warning [ms] Last Communication Warning [s]	N	Date and time of the last communication warning in milliseconds and seconds since 1970-01-01: 0…999 ms, 0…4 294 967 295 s.	UDINT
Last System Warning [ms] Last System Warning [s]	N	Date and time of the last system warning in milliseconds and seconds since 1970-01-01: 0…999 ms, 0…4 294 967 295 s.	UDINT
Last Field Error [ms] Last Field Error [s]	N	Date and time of the last field error in milliseconds and seconds since 1970-01-01: 0…999 ms, 0…4 294 967 295 s.	UDINT
Last Communication Error [ms] Last Communication Error [s]	N	Date and time of the last communication error in milliseconds and seconds since 1970-01-01: 0…999 ms, 0…4 294 967 295 s.	UDINT
Last System Error [ms] Last System Error [s]	N	Date and time of the last system error in milliseconds and seconds since 1970-01-01: 0…999 ms, 0…4 294 967 295 s.	UDINT
Fan State	N	Depends on the controller type; see the documentation. 0xFF: Not available.	BYTE
Mono Startup Release	Y	Enable for non-redundant operation. The system variable exists depending on the controller family TRUE: A single processor module in rack 0, slot 10/12 (H51X) may also start with one system bus only. A single processor module in rack 0, slot 16/18 (H41X) may also start with one system bus only. FALSE: Both system buses are also necessary for a single processor module.	BOOL
MultiForcing Denied	Y	TRUE: MultiForcing is denied. FALSE: MultiForcing is possible.	BOOL
Power supply input voltage 1…5	N	Value of the input voltage in the respective F-PWR 01 power supply unit expressed in mV.	UINT
Power Supply Status	N	Status of the F-PWR 01 power supply units. 3 adjacent bits are coded for each power supply unit. Bit 0…2: Status of power supply unit 1 Bit 3…5: Status of power supply unit 2 Bit 6…8: Status of power supply unit 3 Bit 9…11: Status of power supply unit 4 Bit 12…14: Status of power supply unit 5 The following applies to each power supply unit: ▪ Once the power supply unit has been detected, the least significant bit is 1. Otherwise, it is bit 0. ▪ If at least one warning was issued for the power supply unit, the middle bit is 1. Otherwise, it is bit 0. ▪ If at least on fault occurred in the power supply unit, the most significant bit is 1. Otherwise, it is bit 0.	WORD
Allow Online Settings	Y	Main enable switch of the processor module TRUE: The subordinate enable switches may be changed. FALSE: The subordinate enable switches may not be changed.	BOOL

System variables	S [1)]	Description	Data type
Read-only in RUN	Y	TRUE: The following operator functions are locked: Stop, Start, Download. FALSE: The following operator functions are unlocked: Stop, Start, Download.	BOOL
Redundancy Info	Y	Bit-coded redundancy state of the processor modules. The variable exists depending on the controller family.	
Reload Allowed	Y	TRUE: Reload is allowed to load a controller. FALSE: Reload is not allowed to load a controller.	BOOL
Reload Deactivation	Y	TRUE: Reload is locked. FALSE: Reload is not locked.	BOOL
Reload Cycle	Y	TRUE: The current cycle is the first cycle after a reload. FALSE: Otherwise.	BOOL
Responsible Module Essential	Y	Essential state of redundant processor modules. The variable exists depending on the controller family.	BYTE
Safety Time [ms]	Y	Safety time set for the resource in ms.	UDINT
Start Allowed	Y	TRUE: The processor module may be started through the PADT. FALSE: The processor module may not be started through the PADT.	BOOL
Start Cycle	Y	TRUE: The current cycle is the first cycle after the start. FALSE: Otherwise.	BOOL
Power Supply State [1]…[2]	N	Bit-coded state of the power supply of processor modules 1…2. The following properties differ depending on the controller family: - Possible number of processor modules. - State bits suitable for safety functions!	BYTE
System ID [SRS]	Y	System ID of the controller, 1…65,535.	UINT
Systemtick HIGH Systemtick LOW	Y	Revolving millisecond counter (64-bit).	UDINT
Temperature State [1]…[2]	N	Bit-coded temperature state of processor modules 1…2: Bit no. – State when the bit is set. 0 – Temperature threshold 1 exceeded. 1 – Temperature threshold 2 exceeded. 2 – Incorrect temperature value.	BYTE
Remaining Global Force Duration [ms]	Y	Time in ms until the time limit set for global forcing expires.	DINT
Watchdog Time [ms]	Y	Maximum permissible duration of a RUN cycle in ms (dependent on the controller family).	UDINT
Cycle Time, last [ms]	Y	Current cycle time.	UDINT
Cycle Time, max [ms]	N	Maximum cycle time in milliseconds.	UDINT
Cycle Time, min [ms]	N	Minimum cycle time in milliseconds.	UDINT
Cycle Time, average [ms]	N	Average cycle time in milliseconds.	UDINT

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

Table 27: Input Variables

6.2.5.2 Locking and Unlocking the Resource

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

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

The system variables *Read-Only in RUN*, *Reload Deactivation*, Force Deactivation and MultiForcing Denied are used to lock the controller, see Table 26.

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

To make a controller lockable

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

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

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

6.2.6 User Program System Parameters

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

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

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

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

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

Table 28: System Parameters of the User Program

6.2.7 Notes on the *Code Generation Compatibility* Parameter

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

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

6.2.8 Local User Program System Variables

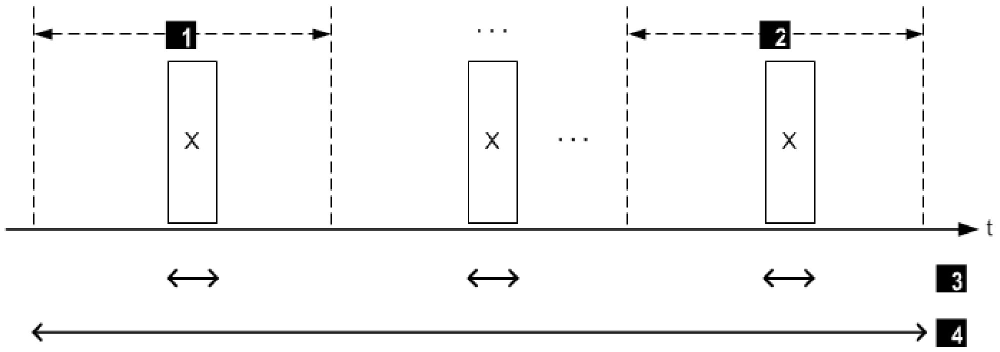
The local system variables provide information at runtime about the operating conditions of the user program. Not all local system variables may be used for programming safety-related responses.

Variable	S [1)]	Description	Data type
Program_CRC	X	User program checksum used to detect potential corruptions	LWORD
Program_CycleDuration		Duration of all processor module cycles expressed in µs and required for executing a user program cycle, see Figure 33. The value is determined based on the duration of the previous user program cycle. In the first cycle after starting up the user program *Program_CycleDuration* is set to 0.	UDINT
Program_ExecutionCycles		Number of processor module cycles necessary to completely execute a user program cycle. The number is determined based on the number of processor module cycles required in the previous user program cycle.	UDINT
Program_ExecutionDuration		Time in µs necessary for executing a user program cycle, see Figure 33. The value is determined based on the processing time required in the previous cycle. In the first cycle after starting up the user program *Program_ExecutionDuration* is set to 0.	UDINT
Program_ForceSwitch	X	TRUE The conditions for local forcing are met. FALSE The conditions for local forcing are not met.	BOOL
Program_ID		Program ID assigned by the user. Program_ID identifies the user program in display functions and in the Control Panel.	UDINT
Program_ReloadCycle	X	TRUE During the first cycle after a reload Exception: Reload only changes user program parameters FALSE In all other cycles, even during a reload, which only changes user program parameters	BOOL
Program_StartCycle	X	TRUE During the first cycle after starting up the user program FALSE In all other cycles	BOOL
[1)] Only system variables that are marked in the S column may be used for safety functions! The other system variables must **not** be used for programming safety functions!			

Table 29: Local User Program System Variables

Program_CycleDuration and *Program_ExecutionDuration*

The following figure shows the progression of a cycle of user program X lasting over multiple processor module cycles. The first processor module cycle considered includes the beginning of the user program cycle, and the last processor module cycle considered includes the end of the user program cycle.



1 First processor module cycle considered.
2 Last processor module cycle considered.
3 *Program_ExecutionDuration.*
4 *Program_CycleDuration.*

Figure 33: *Program_CycleDuration* and *Program_ExecutionDuration*

Program_CycleDuration indicates the duration of all the processor module cycles needed for a user program cycle; in Figure 33, it corresponds to the large double arrow 4.

Program_ExecutionDuration is the processing time portion of user program X in all the processor module cycles considered. In Figure 33, *Program_ExecutionDuration* corresponds to the sum of all small double arrows 3.

Program_ExecutionCycles is the number of cycles from the first processor module cycle 1 to the last 2.

6.2.9 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.

6.2.9.1 Use of Digital Inputs

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

To get additional options for diagnosing the external wiring and programming fault responses in the user program, assign global variables to *Channel OK* and to further diagnostic statuses. For further details on the individual diagnostic statuses, such as short-circuits and open-circuits, refer to the module-specific manual.

6.2.9.2 Use of Analog Inputs

Analog input channels convert the measured input currents into a value of type DINT (double integer). This value is then provided 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 often be used instead of the raw value. HIQuad X calculates the process value based on the raw value and the scale value on the parameters 4 and 20 mA. Refer to the module-specific manual for details.

The values of analog inputs can be used in the user program in two ways:

- Using the process value
  If an analog input is configured correctly, its process value includes the safe fault response.
- Using the raw value
  The raw value is the measured value without the safe fault response. The safe fault response must be programmed in line with the project requirements.

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.

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. Program a suitable conversion function to convert the raw value into a type used in the user program. In doing so, take the measuring range of the data type into account.
4. In the user program, program a safety-related fault response using the statuses *Channel OK, SC, OC* (and others, if necessary).

► The user program can safely process the measured value.

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

To get additional options for diagnosing the external wiring and programming fault responses in the user program, assign global variables to *Channel OK*, *Submodule OK*, *Module OK* and to further diagnostic statuses. For further details on the individual diagnostic statuses, such as short-circuits and open-circuits, refer to the module-specific manual.

6.2.9.3 Use of Safety-Related Counter Inputs

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

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.

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.

6.2.9.4 Use of Digital Outputs

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 responses in the user program, assign global variables to *Channel OK* and to further diagnostic statuses. For further details on the individual diagnostic statuses, such as short-circuits and open-circuits, refer to the module-specific manual.

6.2.9.5 Use of Analog Outputs

To write a value in the user program to an analog 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. In connection with the output channel parameters *4 mA* and *20 mA*, set the REAL values in accordance with the range used for global variables.

► The global variable provides the safe value to the analog 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 responses in the user program, assign global variables to *Channel OK* and to further diagnostic statuses. For further details on the individual diagnostic statuses, such as short-circuits and open-circuits, refer to the module-specific manual.

6.2.10 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 onto the workspace.

For further details on the communication protocols, refer to the communication manual (HI 801 101 E). For further details on how to use the editor for the communication protocols, refer to the SILworX online help.

6.3 Forcing

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

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

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

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

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

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

6.3.1 Use of Forcing

Forcing supports users during the following tasks:

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

WARNING

Physical injury due to forced values is possible!

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

When forcing values, the person in charge must take further technical and organizational measures to ensure that the process is sufficiently monitored in terms of safety. HIMA recommends setting a time limit for the forcing procedure, refer to Chapter 6.3.3 for details.

WARNING

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

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

Forcing can operate at two levels:

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

6.3.2 Assigning a Data Source Changed through Reload

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

- Hardware.
- Communication protocols.
- System variables.

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

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

Consequence

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

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

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

6.3.3 Time Limits

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

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

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

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

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

6.3.4 Restricting the Use of Forcing

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

- Configuring different user profiles with or without forcing permissions.
- Explicitly allowing forcing for a resource (PES).
- Setting up MultiForcing user accounts in PES User Management.
- Explicitly allowing local forcing for a user program.
- Forcing can be stopped immediately via the *Force Deactivation* system variable using the key switch.
- Additionally, the *MultiForcing Denied* system variable can be used to disable MultiForcing.

6.3.5 Force Editor

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

For each variable, the following can be set:

- Forcing values.
- One individual force switch to prepare for forcing variables.

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

Forcing can be started for a predefined time limit or for an indefinite time period. After forcing has started, all variables with an active force switch are set to their force values.

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

If forcing is started again, the configured force values will replace the values from the process!

For further details on the Force Editor and forcing, refer to the SILworX online help.

Copying the current data from the Force Editor to the clipboard only gathers the data visible in the Force Editor. Data that are not visible are not periodically refreshed and can thus have an obsolete value! Press `Ctrl+A` to select and copy all the data, even if they are not in the visible area.

6.3.6 Automatic Forcing Reset

The operating system resets forcing in the following cases:

- When the resource is restarted, e.g., after connecting the supply voltage.
- When the resource is stopped.
- When a new configuration is loaded by performing a download.
- When a user program is stopped: Reset of local forcing for this user program

During a reload, local and global force values, individual force switches, force times and force timeout responses are still valid.

WARNING

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

- **Global force values and individual force switches can be set when a resource is stopped. The configured values become valid after restarting the resource and forcing.**
- **Local force values and individual force switches can be set when the user program is stopped. The configured values become valid after restarting the user program and forcing.**

6.3.7 Forcing and Scalar Events

When forcing a global variable used to create scalar events, observe the following points:

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

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

6.3.8 MultiForcing

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

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

WARNING

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

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

WARNING

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

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

6.3.8.1 Objectives of MultiForcing

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

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

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

6.3.8.2 Global MultiForcing

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

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

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

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

6.4 Cycle Sequence

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

1. Processing of the input data.
2. Processing of the user program.
3. Provision of the output data.

These phases do not include special tasks such as reload, which might be executed within a CPU cycle.

Global variables, results from function blocks, and other data are processed in the first phase and represent the input data for the second phase. The first phase need not start at the beginning of the cycle, but may be delayed. For this reason, inaccurate cycle times, potentially exceeding the watchdog time, may result if timer function blocks are used to determine the cycle time in the user program.

In the third phase, the user program results are forwarded for being processed in the following cycles and supplied to the output channels.

6.5 Loading User Programs

SILworX can be used 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 changed project configuration without interruption of safety-related operation.

i HIMA recommends performing a data backup of the project configuration, e.g., on a removable medium, after loading a user program into the controller.

This will ensure that the project data matching the configuration in the controller remains available even if the PADT fails.

HIMA recommends performing a data backup on a regular basis, independently from loading the user program.

6.5.1 Download

Requirements for download:

- The controller is in the STOP state.
- The resource enable switch is set to Load Allowed.

After the download, start the user program using SILworX to launch 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 precludes the use of the reload function.

6.5.2 Reload

Requirements for reload:

- The controller is in the RUN state.
- The enable switch Reload Allowed is set to TRUE.
- The system variable *Reload Deactivation* is set to FALSE.

i Reload can be performed with one or several processor modules.

During a reload, the PADT cannot be used to access the controller!

Exceptions:

Abort of the reload is possible as well as the change of the watchdog time and the target cycle time to enable reload.

If a user program already running in a controller is modified, a reload can be performed to load the modified version into the controller. While the previous version of the user program is still running, the new version is stored in the controller memory, checked 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.

During a reload, the global and local variables obtain the values of the corresponding variables from the previous project version. The names of local variables contain the POU instance names.

If names are changed and loaded into the controller by performing a reload, this procedure has the following consequences:

- Renaming a variable has the same effect as deleting the variable and creating a new one, i.e., it results in an initialization process. This also applies to retain variables. The variables lose their current value.
- Renaming a function block instance results in initializing all the variables, including retain variables, and all the function block instances.
- Renaming a program results in initializing all its variables and function block instances.

Similarly, moving parts of the program logic is like deleting them from a user program and adding them to another location.

This behavior may have unintended effects on one or multiple user programs and therefore on the plant to be controlled!

The following factors restrict the potential that a changed program is loaded into the controller:

- The restrictions described in Chapter 6.5.2.1.
- The time required to perform a reload.

Since time is required to process the additional reload tasks, the cycle takes longer. 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 required time is too long, a reload is rejected.

i When configuring the watchdog and target cycle times, plan sufficient time reserve for the reload.

HIMA recommends following the procedure for determining the watchdog time described in the safety manual (HI 801 003 E).

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

The online function can only be used to the increase the watchdog and target cycle times, and not to reduce them below the values configured in the project.

i **Observe the following points when reloading sequence chains:**

The reload information for sequence chains does not take the current sequence status into account. The sequence can therefore be changed and set to an undefined state by performing a reload. The user is responsible for this action.

Examples:

- Deleting the active step. As a result, no chain step has the active state.
- Renaming the initial step while another step is active. As a result, a sequence has two active steps!

i

Observe the following points when reloading actions:

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

Examples:

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

Removing a *P0* action qualifier set to TRUE actuates the trigger function.

6.5.2.1 Conditions for Using the Reload Function

A reload can be performed to load the following project changes to the controller:

- 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 30.

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 [3]	•	•	•	•
safe**ethernet** [1]	•	•	•	•
Rack with I/O processing module and I/O modules	•	•	n. a.	n. a.
Modules (I/O processing module, I/O modules and processor modules)	•	•*	n. a.	n. a.
Communication protocols [3]	•	•	n. a.	n. a.
User programs	•	•**	n. a.	n. a.
Event definitions [2]	•	•	n. a.	• (Event states)
Changes to	**Changes**			
Rack names	• [3]			
Module names	•, System bus modules and communication modules: • [3]			
System ID, rack ID	-			
safe**ethernet** target addresses (IP addresses)	• [1]			
User accounts and licenses	•			
Processor modules: ▪ IP configuration ▪ Routings ▪ Switch configuration ▪ Adding, changing, deleting communication protocols	• [3]			
I/O processing modules ▪ IP configuration ▪ Routings ▪ Power supply and temperature monitoring ▪ Setting *Minimum Configuration Version* ▪ Module name	• [3]			
Communication modules	Refer to the communication manual (HI 801 101 E)			

- • Reload possible
- \- Reload not possible
- * 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.: not applicable

[1] For details on how to load changes via reload in connection with safe**ethernet**, refer to the communication manual (HI 801 101 E)

[2] The event source of an event definition may not be modified by performing a reload, i.e., the identification number cannot be used again through a reload.

[3] By performing a cold reload, i.e., restarting the module

Table 30: Reload 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.

6.5.2.2 Cold Reload

In certain cases, a reload cannot be executed for an individual module:

- The conditions designated with [3)] in Table 30.
- Configuration changes within communication and I/O processing modules.
- The communication module is operating with standard protocols that are not capable of reload.

For these cases, a cold reload must be performed. The concerned module must be in STOP (cold)while the reload process is running and must be restarted upon completion of the process.

- Communication and I/O processing modules are stopped and restarted by the reload process.
- When processor modules are being reloaded, a message appears, asking the user to stop or start the module.

Either way, the user maintains the control of the reload process and is informed about its progress. If necessary, the user can abort the reload process.

Prior to performing a cold reload, consider the following points:

- Which modules might be stopped?
- Are redundant modules configured for the modules potentially concerned, or which of the functions they perform can be temporarily renounced?

TIP Proceed as described below to avoid a cold reload in situations in which global variable assignments are added:

- When creating the configuration, already assign unused global variables to communication protocols.
- Assign safe value as initial value to unused global variables.

In doing so, there is no need later to add variables, but only to change their names, which allows a reload to be performed.

6.5.2.3 Restrictions for Reload

The reload of a user program may become impossible if the following conditions apply:

- The number of function and function block instances (POUs) in SILworX is limited to 21 845. Loading of a user program containing more instances cannot be performed through a reload.
- Large arrays with user-defined structure elements can no longer be loaded through the reload when the structure is changed.

  Example: Adding/deleting a structure element in/from a structure data type.

In both cases, SILworX indicates that the maximum number of transfer operations (65 536) was exceeded.

Workaround

To reduce the number of transfer operations, the following is possible:

- Avoid excessively structured programs, e.g., reduce the number of POUs.
- Avoid large structure data types and large arrays of structures.
- A simple workaround is to avoid using VAR in POUs with many instances.

  Replacing all POU VAR with VAR_OUTPUT reduces the number of transfer operations, but also has the following effects:

  - Number of POU outputs is increased.
  - Forcing the variables is not possible.

6.6 User Management

The user management is used for project-specific management of user groups, user accounts, access permissions for the PADT and access permissions for the controllers (PES).

A user group consists of one or more user accounts. Opening a project in SILworX may require a login with the data of a user account. In addition, the user group can be allowed to access resources.

Exactly one user management can be created in a project. The user management applies for this project and is saved in encrypted form in the project. The user management name is assigned by SILworX and is language-dependent.

A user group is a collection of users with identical access permissions for operation of the PADT or for access to the controllers.

PADT User Management	The access to the SILworX project are managed in the PADT user management.
PES User Management	The PES-related access permissions are managed in the PES user management. The PES user management is based on the data in the PADT user management.

WARNING

If no user management scheme is configured, any kinds of manipulations of the project or the resource are possible!

Safe operation of a system calls for a project to employ a user management with several different user groups and different access modes. There must be a number of user accounts defined in the user groups.
Otherwise, anyone can open the project in SILworX and log themselves in to resources as the default user.

If no user management is available, a project is opened without a login dialog. Otherwise, a dialog box for a PADT user query is displayed and there is a prompt to enter a user name and a password.

If the project was created with an earlier version of SILworX and a project conversion is required, the project is only converted if the logged-in user has at least Write access. Otherwise, the project is not converted and is not opened.

When the user management is created, an initial security administrator is also automatically added to the PADT user management and the parameters are configured. After this, changes can only be made to the user management by this or another security administrator.

6.6.1 Standard Access Permissions

As long as no user management has been set up for a project, no *User Management* element is displayed in the structure tree. The factory-specified access permissions apply for both the project opened in SILworX and the resources.

	PADT	Resource (controller)
Number of users	1	1
User ID	Security administrator	Administrator
Password	None	None
Access permission	Security administrator	Administrator

Table 31: Factory Access Permissions for PADT and PES

i The factory-specified access permissions are disabled when a user management is created in the project.

6.6.1.1 The Security Administrators User Group

Security administrators have all access permissions. SILworX automatically creates a user group with the *Security Administrator* access mode as soon as a user management is created. By default this user group contains one (1) user account, and additional user accounts can be added.

This ensures that there is always at least one security administrator present in the user administration.

The user account created by default and all other user accounts added to this user group later on have the *Security Administrator* access mode.

This access mode is only valid for editing the project in SILworX. The security administrators user group can also have a different access mode with reduced privileges in the *PES User Management*.

i Security administrators have full access to all user accounts. For example, they can change the passwords of these user accounts without knowing the user account's current password. This can become necessary if users forget their password.

6.6.2 Access Modes and Permissions

Depending on the defined access mode and the licenses available, a user will be able to use more or fewer permissions and functions in SILworX or with a resource.

Depending on the user currently logged in, the access permissions can also be restricted via licenses. To do this, the *Maintenance* license option can be used to downgrade the operative permissions for every user to *Read*.

6.6.3 PADT User Management Access Modes

The following table summarizes the access modes which can be used in the PADT user management.

Access Mode	Description
Security administrator	The *Security Administrator* access mode allows the user to operate all functions in SILworX which are available with the existing licenses.
Read and Write	The *Read and Write* access mode allows users to execute all functions in SILworX that are are covered by the available licenses and not explicitly bound to the *Security Administrator* privilege.
Read	Users with this access mode can access functions in reading mode only. This means that Archiving is not permitted, for instance. Some types of information are not displayed (e.g., details concerning user management).

Table 32: Access Modes for the PADT User Management

6.6.4 PES User Management Access Modes

The following table summarizes the access modes which can be used in the PES user management. A maximum of 5 PES users can simultaneously be logged into a controller.

Access Mode	Description
Administrator	**Activity**: Putting a new system into operation. Within the scope of application of the resource, administrators have all of the permissions which are available with the existing licenses. - Actions required to put modules or systems into operation. - Setting up a replaced controller. - Start-up of safety devices.
Read and Write	**Activity**: Modifying an existing system. The *Read and Write* access mode allows users to execute all functions on a resource that are required to make modifications and are covered by the available licenses. Users can change the online settings for which modification has been enabled, but not the setting of the *Allow Online Settings* parameter itself.
Read and Operator	**Activity**: Maintaining a running system in operation. The *Read and Operator* access mode allows users to carry out all actions which are required to maintain a running system. No modifications to the programmed functions are possible. The permissions here are downgraded once again as compared to *Read and Write* and are by and large restricted to the following activities: - All actions which are necessary for diagnostic purposes. - Actions which are required to start and to force a system.
MultiForcing	**Activity**: Forcing by several users simultaneously. Users with *MultiForcing* access can write force data (force values and individual force switches) for global variables in a resource if the required higher-order conditions have been met and the force permissions have been granted. The use of preconfigured watchpages is also possible. To all other functions of a resource, users have Read-Only access. Starting, stopping or resetting a force process is not possible. To modify watchpages and save them in the project, users require the corresponding access permissions in the PADT (*Read and Write* or *Security Administrator*).
Read	**Activity**: Reading the diagnostics. Users with this access mode can access functions in reading mode only. The *Read* access mode is assigned automatically during a system login to a resource when there is already another user logged in with Write access permissions.
No Access	**Activity**: None, login is locked. This is the default setting for all user groups for which no access mode has been explicitly set in the assignment table. A blank table cell is displayed for *No Access*.

Table 33: Access Modes for the PES User Management

6.6.5 Creating the PADT User Management

Refer to the SILworX online help for furhter details on how to create the PADT user management.

6.6.6 Creating the PES User Management

The PES user management automatically displays the user groups created in the PADT user management. Access permissions for the created user groups can be assigned to the resources in the PES user management. Refer to the SILworX online help for details.

User accounts for up to 10 user groups can be managed in the PES user management of each controller. The user data is available for login once the project has been loaded in the controller through a download. The user accounts are stored in the controller and still apply after switching off the operating voltage. The user accounts of a controller also apply to the connected remote I/Os.

Users log in to a controller using the user group name and password. The values for PES user group name and password are not those currently entered in the project, but those transferred to the PES during the last loading process!

7 Diagnostics

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(LEDs)

LEDs on the front plate indicate the module state. All LEDs should be considered together. A single LED is not sufficient to assess the module state.

A description of the LEDs is provided in the module-specific manuals.

After connecting the supply voltage, an LED test is performed and all the LEDs are lit for at least 2 s. The color of two-color LEDs changes once during the test.

Definition of blinking frequencies

The following table defines the blinking frequencies:

Definition	Blinking frequencies
Blinking1	Long (600 ms) on, long (600 ms) off.
Blinking2	Short (200 ms) on, short (200 ms) off, short (200 ms) on, long (600 ms) off.
Blinking-x	Ethernet communication: Blinking synchronously with data transmission.

Table 34: Blinking Frequencies of the LEDs

Some LEDs can report warnings (On) and faults or errors (Blinking1). The indication of errors or faults has priority over the indication of warnings. Warnings cannot be reported if errors or faults are being signaled.

7.2 Diagnostic History

The modules F-CPU, F-IOP and F-COM keep a chronological history of the faults and events that have occurred. The history is organized as a ring buffer.

The diagnostic history is composed of short-term and long-term diagnostics.

- Short-term diagnostics
  If the maximum number of entries has been reached, each new entry deletes the oldest entry.
- Long term diagnostics
  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 diagnostics
F-CPU 01	2500	1500
F-IOP 01	400	500
F-COM 01	300	700

Table 35: 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 by time period.
- Printing out the edited history
- Saving the edited history

For additional functions, see the SILworX online help.

7.2.1 Diagnostic Messages

A diagnostic message for I/O modules is structured as follows:

IO ERROR >> slot `S` I/O module type: `MMMM` status[Mod: `mm` OUT: `AAAA` IN: `EEEE`] channel[OUT:`aaaa` IN:`eeee`] <<

The following table describes the data fields in the message.

Data field	Format	Description
`S`	Decimal	Slot number of the I/O module
`MMMM`	Hexadecimal	Module type, see Table 37.
`mm`	Hexadecimal	Status of the module.
`AAAA`	Hexadecimal	Code for faults in the module's outputs
`EEEE`	Hexadecimal	Code for faults in the module's inputs
`aaaa`	Hexadecimal	Code for channel faults in the output channels
`eeee`	Hexadecimal	Code for channel faults in the input channels

Table 36: Data Field of Diagnostic Message

For further details on the error codes, refer to the corresponding manuals. If multiple channels are faulty, the data field `aaaa / eeee` contains an OR gate with 0x8000, e.g., the most significant bit is set to 1 in addition to the error code.

The I/O module type can be determined in the following table or in the Hardware Editor.

Value	Format	I/O module type
0x01	Hexadecimal	F 3221
0x03	Hexadecimal	F 3224A
0x04	Hexadecimal	F 3236
0x05	Hexadecimal	F 3237
0x06	Hexadecimal	F 3238
0x07	Hexadecimal	F 3240
0x08	Hexadecimal	F 3248
0x09	Hexadecimal	F 3322
0x0B	Hexadecimal	F 3330
0x0C	Hexadecimal	F 3331
0x0D	Hexadecimal	F 3333
0x0E	Hexadecimal	F 3334
0x0F	Hexadecimal	F 3335
0x11	Hexadecimal	F 3349
0x12	Hexadecimal	F 3422
0x13	Hexadecimal	F 3430
0x14	Hexadecimal	F 5220
0x15	Hexadecimal	F 6215
0x16	Hexadecimal	F 6217
0x17	Hexadecimal	F 6220
0x18	Hexadecimal	F 6221
0x19	Hexadecimal	F 6705
0x1A	Hexadecimal	F 6706

Table 37: I/O Module ID

7.3 Online Diagnosis

The online view in the SILworX Hardware Editor is used to diagnose failures in the HIQuad X 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 limit has been exceeded.

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

<table>
<tr><th>Information</th><th>Representation</th><th>Range of values</th><th>Description</th></tr>
<tr><td>SRS</td><td>Decimal numbers</td><td>System: 0…65 535
Rack: 0…16
Slot: 1…18</td><td>Module identification: System, Rack, Slot</td></tr>
<tr><td>Name</td><td>Text</td><td></td><td>Designation of the information , here always: Online MODULE Information.</td></tr>
<tr><td>Module state</td><td>Text</td><td>RUN, STOP, NOT CONNECTED, Unknown, ...</td><td>State in which the I/O processing module is operating.</td></tr>
<tr><td>Plugged Module Type</td><td>Text</td><td>Permissible module types</td><td>Type of module which is plugged into the rack.</td></tr>
<tr><td>Configured Module Type</td><td>Text</td><td>Permissible module types</td><td>Type of module which is configured and loaded in the controller.</td></tr>
<tr><td>Module Type in Project</td><td>Text</td><td>Permissible module types</td><td>Type of module which is being projected as the digital depiction.</td></tr>
<tr><td>Connection Status of the Protocol</td><td>Hexadecimal value</td><td>16#00…0F</td><td>Status of the connection between each of the processor modules (max. of 2) and the I/O processing module. Each of the bits 0...3 shows the connection to the processor module with the corresponding index.
Bit x = 0: Not connected.
Bit x = 1: Connected.</td></tr>
<tr><td>Interface Send Status</td><td>Hexadecimal value</td><td>16#0000…FFFF</td><td>Every two bits represent the state of one interface identified with an index 0...16. Bits 0 and 1 apply to interface 0, and so on.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>00</td><td>No message has been received or sent yet, unknown status.</td></tr>
<tr><td>01</td><td>OK, no faults.</td></tr>
<tr><td>10</td><td>Last data received or sent was defective.</td></tr>
<tr><td>11</td><td>No faults during last reception/transmission, one fault occurred before.</td></tr>
</table></td></tr>
<tr><td>Interface Receive Status</td><td>Hexadecimal value</td><td>16#0000…FFFF</td><td>Every two bits represent the state of one interface identified with an index 0...16. Bits 0 and 1 apply to interface 0, and so on.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>00</td><td>No message has been received or sent yet, unknown status.</td></tr>
<tr><td>01</td><td>OK, no faults.</td></tr>
<tr><td>10</td><td>Last data received or sent was defective.</td></tr>
<tr><td>11</td><td>No faults during last reception/transmission, one fault occurred before.</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Information</th><th>Representation</th><th>Range of values</th><th>Description</th></tr>
<tr><td>Module Error Status</td><td>Hexadecimal value</td><td>16#00…3F</td><td>Bit-coded status of the I/O processing module:
<table>
<tr><th>Bit</th><th>Description for value = 1</th></tr>
<tr><td>0</td><td>Warning related to external communication.</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</td><td rowspan="2">Not used.</td></tr>
<tr><td>7</td></tr>
</table></td></tr>
<tr><td>Connection status of system bus A</td><td>Hexadecimal value</td><td>16#0…3</td><td>Status of the interface to system buses A:
<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 during 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></td></tr>
<tr><td>Connection status of system bus A</td><td>Hexadecimal value</td><td>16#0…3</td><td>Status of the interface to system buses A:
<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 during 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></td></tr>
</table>

Table 38: Diagnostic Information Displayed in the Online View for the Hardware Editor

Due to the timing behavior of the operating system, it is possible that the safety parameters are not displayed in the diagnostics. To allow the safety parameters to be displayed in the diagnostics, proceed as follows when opening the diagnostics:

1. Open the Control Panel and wait for all the fields to be refreshed.
2. In the Hardware Editor, open the diagnostics using the context menu of the online view and not of the detail view!

Prior to opening the diagnostics, do not open the detail view and open as few online views as possible (e.g., Force Editor, online test)!

8 Product Data, Dimensioning

This chapter specifies the environmental requirements and the dimensions to be set in the SILworX programming tool.

8.1 Environmental Requirements

Exposing the HIQuad X system to environmental conditions other than those indicated can cause it to malfunction. Additionally, the instructions provided in the module-specific manuals must be observed.

General	
Protection class	Protection class II in accordance with IEC/EN 61131-2
Ambient temperature	0…+60 °C
Transport and storage temperature	-40…+70 °C
Pollution	Pollution degree II in accordance with IEC/EN 60664-1
Installation height	< 2000 m
Enclosure	Standard: IP20 If required by the relevant application standards (e.g., EN 60204), the system must be installed in an enclosure with the specified degree of protection (e.g., IP54).
Power Supply Input Voltage	24 VDC, -15...+20 %, $r_p \leq 5$ % SELV, PELV

Table 39: Environmental Requirements

Exposing the HIQuad X system to environmental conditions other than those indicated can cause it to malfunction.

8.2 Dimensioning

For details, refer to the component-specific manuals and communication manual (HI 801 101 E).

For each resource	Value
Number of racks	▪ H51X: 1 base rack and a maximum of 16 extension racks. ▪ H41X: 1 base rack and a maximum of 1 extension rack.
Number of I/O modules	▪ H51X: 256 ▪ H41X: 28 (with extension rack)
Number of I/O elements	Depending on the module type ▪ H51X: 4096 ▪ H41X: 224
Number of processor modules	2
Total program and data memory for all user programs	5 MB less 64 kB for CRCs
Memory for retain variables	32
Number of I/O processing modules for each rack	1 (none in the H51X base rack)
Maximum length of the system buses	50 m between two subscribers.
Number of communication modules	▪ H51X: 0…10 ▪ H41X: 0…2
Buffer size for connections to the OPC Server	
Numer of PES user groups	1…10
Numer of user programs	1…32
Number of event definitions	0…5000
Size of the non-volatile event buffer	1000 Events

Table 40: Dimensioning of a HIQuad X Controller

9 Lifecycle

This chapter describes the following lifecycle phases:

- Installation.
- Start-up.
- Maintenance and repairs.

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 structure and connect the HIQuad X system.

9.1.1 Mechanical Structure

To ensure proper operation when structuring the HIQuad X system, observe the conditions of use specified in Chapter 2.1.

Observe the instructions for installing base racks and other components specified in the corresponding manuals.

9.1.2 Connecting the Field Level

The field level is connected to the cable plugs of the I/O modules. The cable plug wires must be connected to terminals.

9.1.3 Grounding

Observe the requirements specified in the low voltage directives SELV (Safety Extra Low Voltage) or PELV (Protective Extra Low Voltage).

Functional ground is provided to improve electromagnetic compatibility (EMC). This functional ground must be connected over a large area in the control cabinet.

HIQuad X systems must be grounded as described in the follow chapters.

9.1.3.1 CE-Compliant Structure of the Control Cabinet

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

All modules included in the HIQuad X system bear the CE marking.

To prevent EMC issues when installing controllers in control cabinets and support frames, the following measures must be implemented:

- The H 7034 filter must be installed close to the 24 V supply to suppress interferences directly at the supply point.
- Ensure proper and interference-free electrical installation in the vicinity of the controllers, e.g., do not lay power lines together with field lines or 24 VDC supply lines. For further information, see Chapter 9.1.3.2.
- Observe the instructions provided in this manual related to grounding, shielding and cable routing to sensors and actuators.

9.1.3.2 Surges on Digital Inputs

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

To prevent malfunctions, take one of the following measures for the application:

- Install shielded input wires to prevent surges within the system.
- Noise blanking in the user program: a signal must be present for at least two cycles before it is evaluated.

Caution: This measure increases the system response time!

---

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

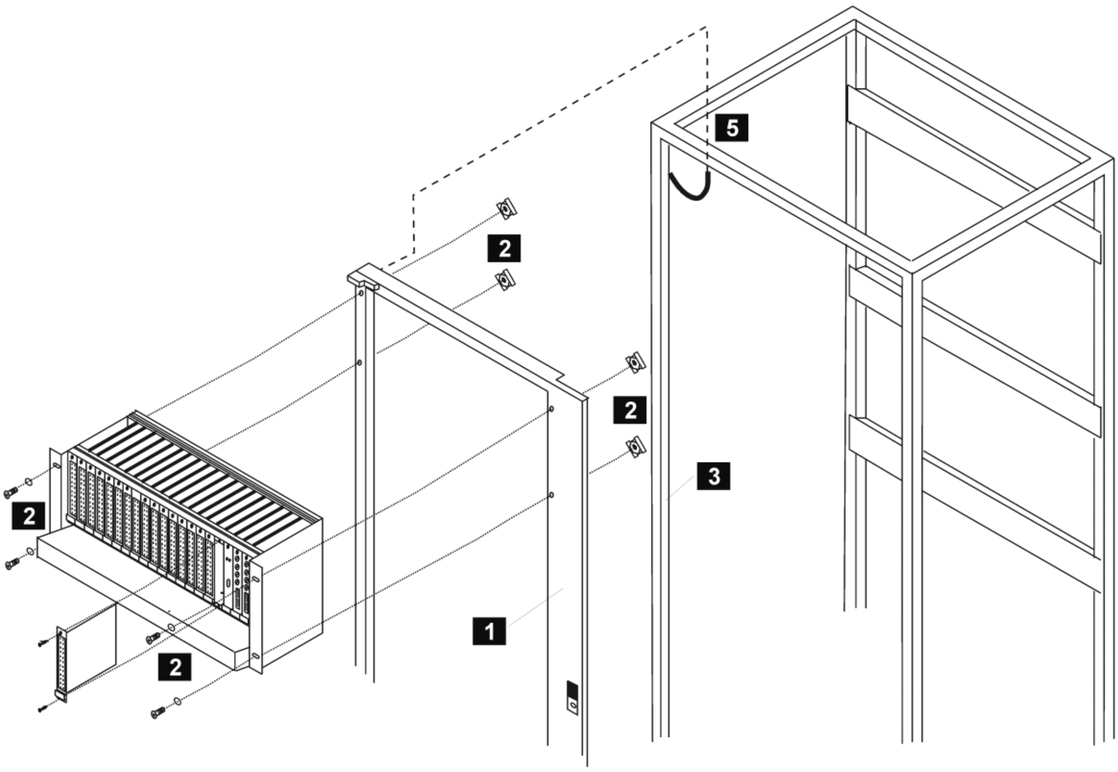
In particular, the design must include protective measures with respect to overvoltage, lightning, grounding and plant wiring in accordance with the relevant standards and the manufacturer's specifications.

9.1.4 Grounding Connectors

All tangible surfaces of the 19-inch HIMA components (e.g., base racks, extension racks and dummy front plates) are chromized and electrically conductive for ESD protection reasons.

Safe electrical connection between components and the control cabinet is ensured by cage nuts with claw fasteners. The claw fasteners penetrate the surface of the pivoting frame [1], ensuring safe electrical contact. Stainless steel screws and flat washers are used to prevent electrical corrosion [2].

The components of the cabinet frame [3] are welded together and are considered an electrical conductive construction element. Short grounding straps [5] with cross-sections of 16 mm^2 or 25 mm^2 are used to conductively connect the pivoting frames, door, mounting rails and mounting plates to the cabinet frame.



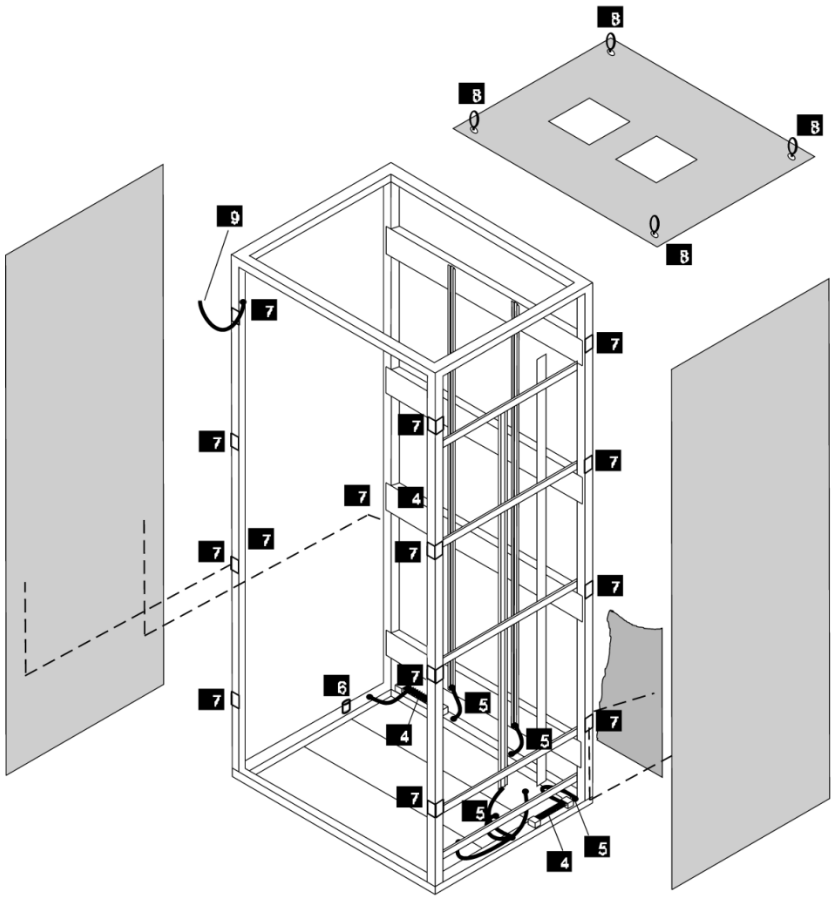
1 Pivoting frame
2 Screws, washers and cage nuts
3 Cabinet frame
4 2 x M 2500 busbar, see Figure 35
5 16 mm^2 or 25 mm^2 grounding strap

Figure 34: Grounding Connectors for Racks

The roof sheeting is secured to the cabinet frame with four lifting eyes [8] (see Figure X). The cabinet frame is electrically connected to the side panels and the backplane through grounding claw fasteners [7] and to the floor panel through screws.

Two M 2500 busbars [4] are installed in the cabinet as standard equipment and connected to the cabinet frame through 25 mm^2 grounding straps [5]. The busbars can also be used as a separate potential (e.g., to connect to the field cable shielding) if the grounding straps between the busbars and the control cabinet are removed.

An M8 bolt is located on the cabinet frame to allow customers to connect the protective ground cable [6].

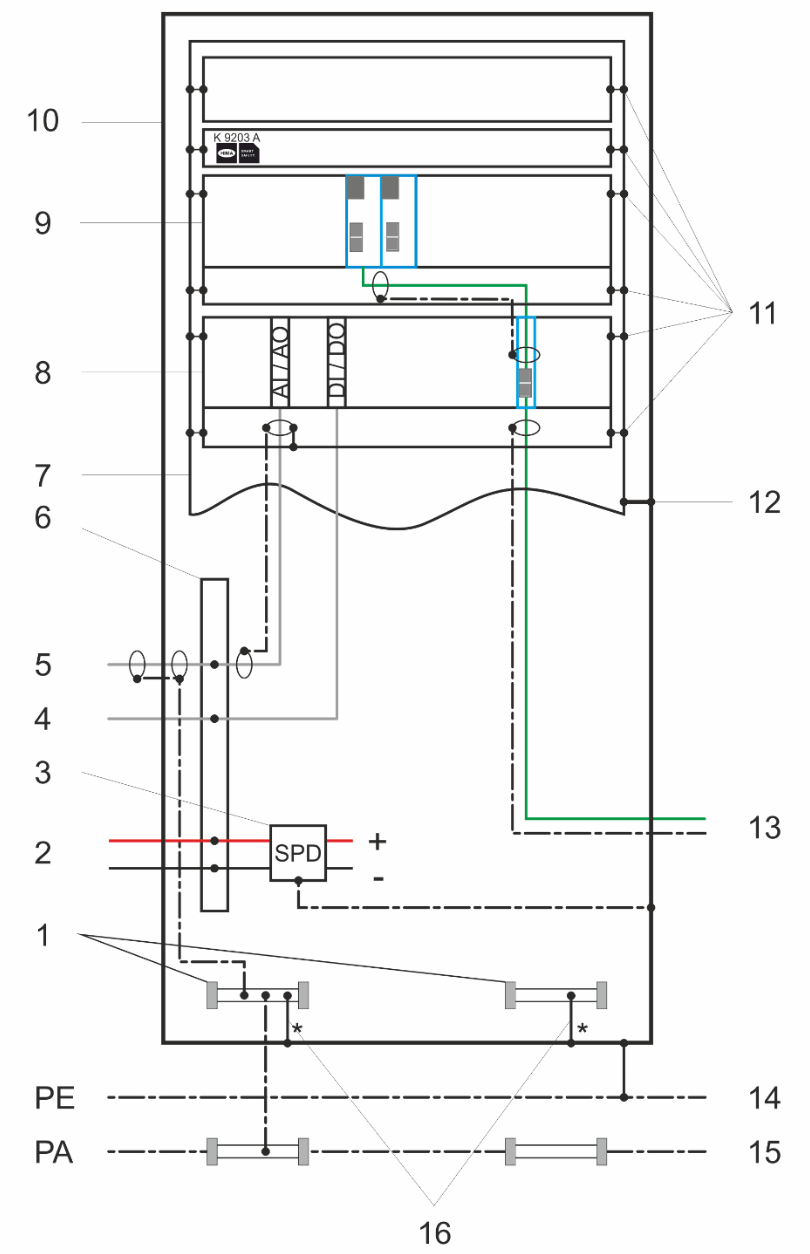


4 2 x M 2500 busbar.
5 DIN rails grounded with 16 mm^2 grounding strap.
6 Central grounding point for the cabinet frame (M8 bolts).
7 Mechanical parts are grounded by standard fasteners to the cabinet frame.
8 Roof sheeting with 4 lifting eyes and fan exhausts for fan inserts.

Figure 35: Grounding Connections in the Control Cabinet

9.1.5 Grounding and Shielding Concept of HIMA Control Cabinets

The following figure shows the grounding and shielding concept of a HIMA control cabinet:



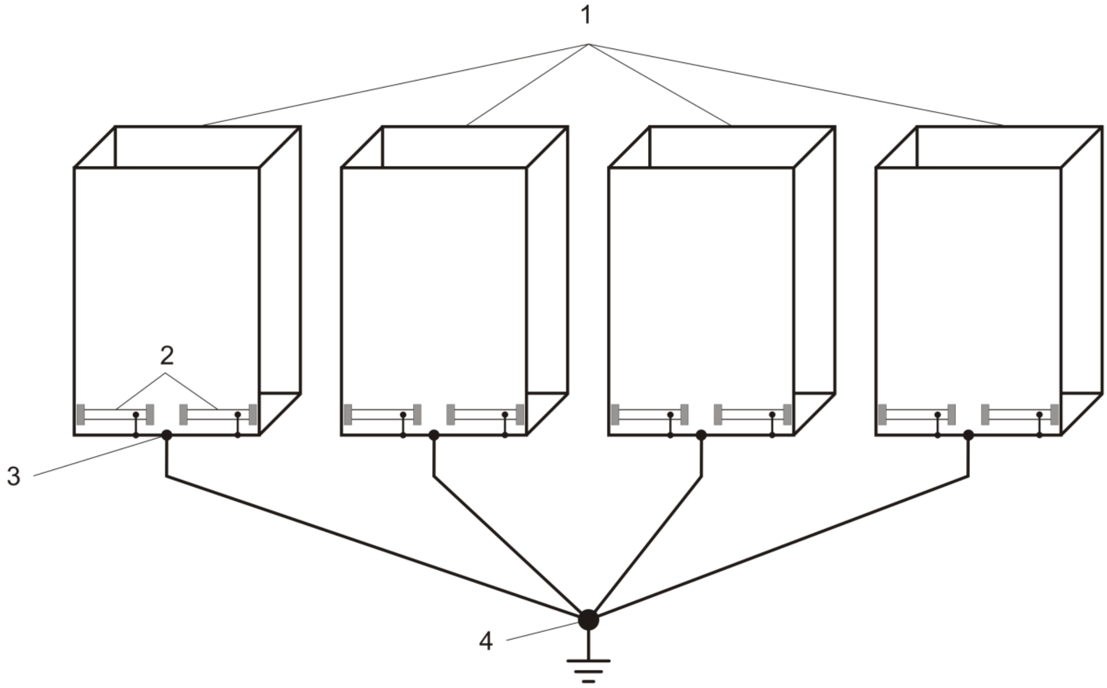
1	M 2500 busbar.	9	Base rack.
2	24 VDC supply.	10	Cabinet frame.
3	Filter (surge protective device).	11	Cage nuts and cage clamps.
4	Digital signals.	12	25 mm grounding connector.
5	Analog signals.	13	Shielded bus cable.
6	Inline terminals.	14	Protective ground.
7	Pivoting or fixed frame.	15	Equipotential bonding.
8	Extension rack.	16	Standard connection to HIMA control cabinets, to delete if equipotential bonding is applied.

Figure 36: Grounding and Shielding Concept of the HIMA Standard Cabinet

9.1.6 Grounding Several Control Cabinets

Connect several control cabinets to a central, interference-free ground; individually ground the control cabinets of a controller, if required.

Ensure at least 16 mm² cross-section for the connection between the central grounding points of the control cabinets and the common central ground.



1	Cabinet frame.	3	Central grounding point.
2	M 2500 busbar.	4	Central ground.

Figure 37: Control Cabinets with Central Ground

9.1.7 Ungrounded Operation

In ungrounded operation, a single ground fault does not affect the safety and availability of the controller.

If several undetected ground faults occur, faulty control signals can be triggered. For this reason, HIMA recommends using ground fault monitoring for ungrounded operation. Some application standards, e.g., DIN EN 50156-1:2005, prescribe the use of ground fault monitoring. Only use ground fault monitoring devices approved by HIMA.

9.1.8 Grounded Operation

Requirements for grounded operation are proper ground conditions and, whenever possible, separate ground connection, in which no parasitic currents may flow. Only the negative pole L- may be grounded. The positive pole L+ must not be grounded since a potential ground fault on the sensor wire would bridge the affected sensor.

L- can only be grounded at one point within the system. L- is usually grounded directly behind the power supply unit (e.g., on the busbar). Grounding should be easy to access and well separate. The grounding resistance must be ≤ 2 Ω.

9.1.9 Shielding within the Input and Output Areas

Lay field cables for sensors and actuators separately from the power supply lines and at a sufficient distance from electromagnetically active devices (electric motors, transformers).

To avoid interferences, ensure that the field cables are provided with continuous shielding. To this end, connect the shielding on both ends of the field cables. This applies, in particular, to field cables of analog inputs and proximity switches. Exception: The shield in the F 6217 may only be connected to the rack.

If high compensation currents are expected, the shielding must be applied on at least one end. Further measures, e.g., galvanic separation, must be implemented to avoid compensation currents. Additionally, the requirements specified in the module-specific manuals must be observed.

9.1.10 Lightning Protection for Data Lines in HIMA Communication Systems

Lightning protection for data lines can be improved by implementing the following measures:

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

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

9.1.11 Cable Colors

The cable colors used in the HIQuad X systems comply with international standards.

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

9.1.12 Connecting the Supply Voltage

Connect the supply voltage infeed lines to the clamp terminal blocks of the base racks (L1+, L2+, L1-, L2-).

Attach the supply voltage infeed lines of the system fan to the screw terminals.

9.2 Start-Up

Only power up the HIQuad X 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 supply voltage, check if all cables are properly connected, thus ensuring that no risk exists for controller and system.

9.2.1.1 Test of All Inputs and Outputs

Impermissible parasitic voltage (in particular with 230 VAC against ground or L-) can be measured using a universal measuring instrument.

HIMA recommends testing every individual terminal for impermissible parasitic voltage.

When testing external cables for isolation resistance, potential short-circuits or open-circuits, no cable ends must be connected to prevent potential damage or destruction of modules caused by high voltage.

To check for ground faults, unplug the voltage connection plugs from the power distributor and disconnect the supply voltage for sensors and the negative pole of actuators.

If the negative pole is grounded during operation, the ground connection must be interrupted for the duration of the ground fault check. The same applies to the ground connection of ground fault measuring facility, which may be connected to the system.

A megohmmeter or a special measuring instrument must be used to check each connection against ground.

9.2.1.2 Voltage Connection

All the HIQuad X modules are inserted in the racks and the cable plugs are screwed on the I/O modules. Check proper polarity, voltage and ripple for the 24 VDC supply voltage.

9.2.2 Starting up the PES with Processor Modules (F-CPU 01)

Requirements for start-up:

- The hardware is installed.
- The racks are interconnected.
- The PADT network connection is configured such that the modules of the HIQuad X base plate can be reached. If required, enter a routing for the interface card in use.
- A suitable project is available with configured rack ID, IP address and system ID.

To start up the controller with processor modules (F-CPU 01)

1. Connect the supply voltage.
2. Set the system ID and IP address of the left F-CPU 01 processor module:
   - Establish a direct physical connection between PADT and processor module.
   - In the structure tree, select **Hardware** within the resource element, and click **Online** on the Action Bar.
     The *Online Hardware* tab and the *System Login* window appear.
   - Click the **To Module Login** button.
   - In the *Online Hardware,* log in to the processor module (double-click the processor module, the module login window appears).
     Use the MAC address (see the label on the module) to read the IP address and the SRS (**Browse** button in the login window).

- Use the **Change** button on the *Search via MAC* dialog box to display the *Write via MAC* window. This window can be used to set the system ID and IP address on the processor module.

3. Use patch cables to connect the base rack to the extension rack as described in Chapter 3.2 and Chapter 3.3.

   ☑ The LEDs *UP* and *DOWN* as well as the *Red.* LEDs on the associated processor and I/O processing modules are lit, see the F-CPU 01 manual (HI 803 215 E) and the F-IOP 01 manual (HI 803 219 E).

4. Prepare the left processor module:
   - Log in to the processor module: Double click the processor module symbol in the online image.

---

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 resetting to the factory settings (master reset) when using processor modules with an unknown history.

   - Reset the processor module to the factory settings (master reset).
   - If the system is only equipped with one processor module (mono system), set mono operation. To do so, click **Set Mono/Redundancy Operation** in the *Online->Start-up* menu.
     This setting only takes effect if a mono project is loaded. Otherwise, the system automatically resets the switch.

5. Set the mode switch of the left processor module to *Stop* and wait until the processor module indicates to be running in system operation.

   ☑ The *Stop* LED is lit or blinking, the *Init* LED is off.

6. Log in to the system.
7. Set the mode switch of the right processor module to *Stop*.

   ☑ The right processor module starts operating redundantly. The *Stop* LED is lit and the *Init* LED is off.

8. Load the existing configuration to the processor modules by performing a **download** (menu functions: **Online -> Resource Download**)

   ☑ The processor modules enter the STOP/VALID CONFIGURATION state.

9. The mode switches on all the processor modules are set to *Run*.
10. Perform a resource cold start.

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

For further details on how to start up the system, refer to the first steps manual (HI 801 103 E).

9.2.3 Start with Only one Processor Module Set to *Responsible.*

A redundant system does not start up after the supply voltage is switched on unless both responsible processor modules set to *Responsible* (F-CPU 01) can be reached and are functional.

This limitation of the availability must be rectified before starting, for example, by replacing the defective module.

Start Emergency Mono System Operation serves to start a system with only one responsible module in exceptional cases. Starting the mono system operation by setting the *Mono Startup* switch or using *Start Emergency Mono System Operation* for a processor module is only permitted if there is no processor module present in system operation and if no other processor module apart from the selected one is ready to commence system operation. Otherwise, there is no guarantee that the selected processor module will start the system operation first.

For details, refer to the SILworX online help.

9.2.3.1 Faults

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

9.3 Maintenance and Repairs

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

i For a safety-related application, the controller must be subject to a proof test at regular intervals. For further details, refer to the safety manual (HI 803 209 E).

NOTICE

Malfunction due to electrostatic discharge!

Damage to 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!

9.3.1 Connecting the Power Supply after a Service Interruption

After connecting to the power supply, the HIQuad X system modules start in random order. This applies to the HIQuad X modules as well as to the connected remote I/Os.

9.3.2 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 redundant power source!

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

9.3.3 Loading Operating Systems

Refer to the release notes for the corresponding operating system version for details on how to load the operating system.

The HIQuad X system modules F-CPU 01 and F-IOP 01 contain processors and an operating system controlling 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.

10 HIQuad X Documentation

The following documents are available:

Document	Document number	Topic HIQuad X	File format
System manual	HI 803 211 E	Description of the system	PDF
Safety Manual	HI 803 209 E	Safe use of the HIQuad X system	PDF
F-CPU 01	HI 803 215 E	Processor module, SIL 3	PDF
F-COM 01	HI 803 223 E	Communication module	PDF
F-IOP 01	HI 803 219 E	I/O processing module, SIL 3	PDF
F-PWR 01	HI 803 225 E	24 VDC / 5 VDC power supply unit, 50 W	PDF
F-PWR 02	HI 803 227 E	Buffer module	PDF
F-FAN 01		Description of the system fans	
SILworX first steps manual	HI 801 103 E	Introduction for engineering HIMA controllers using SILworX	PDF
SILworX online help	-		CHM
Communication manual	HI 801 101 E	Communication protocols and their application	PDF
HIPRO-S V2 manual	HI 800 723 E	Safety-related HIPRO-S V2 communication protocol	PDF
Document	**Document number**	**Topic Fuse and power distribution module**	**File format**
K 7205	HI 800 273 E	63 A, 18 circuit breakers for SELV/PELV	PDF
K 7206	HI 800 275 E	63 A, supply with decoupling for SELV/PELV	PDF
K 7207	HI 800 277 E	Diode on heat sink, 25 A, for SELV/PELV	PDF
K 7212	HI 800 287 E	35 A, 12 circuit breakers with decoupling for SELV/PELV	PDF
K 7213	HI 800 289 E	35 A, 12 circuit breakers for SELV/PELV	PDF
K 7214	HI 800 291 E	150 A, 18 circuit breakers for SELV/PELV	PDF

Table 41: Overview of the HIQuad X Documentation

Appendix

Glossary

Term	Description
AI	Analog input
AO	Analog output
ARP	Address resolution protocol, network protocol for assigning the network addresses to hardware addresses
COM	Communication Module
CRC	Cyclic redundancy check
DI	Digital input
DO	Digital output
EMC	Electromagnetic compatibility
EN	European standard
ESD	Electrostatic discharge
FB	Fieldbus
FBD	Function block diagrams
ICMP	Internet control message protocol, network protocol for status or error messages
IEC	International electrotechnical commission
Interference-free	In this context, interference-free refers to safety-related and non-safety-related modules, which may be operated within a rack, if they are marked as interference-free. In terms of functional safety, the non-safety-related-module has no influence on the safety-related modules.
MAC Address	Media access control address, hardware address of one network connection
PADT	Programming and debugging tool (in accordance with IEC 61131-3), PC with SILworX
PCB	Backplane
PELV	Protective extra low voltage
PES	Programmable electronic system
R	Read: The system variable or signal provides a value, e.g., to the user program
R/W	Read/Write (column title for system variable/signal type)
Rack ID	Rack identification (number)
r_{PP}	Peak-to-peak value of a total AC component
SELV	Safety extra low voltage
SFF	Safe failure fraction, portion of faults that can be safely controlled
SIL	Safety integrity level (in accordance with IEC 61508)
SILworX	Programming tool for HIMA systems
SNTP	Simple network time protocol (RFC 1769)
SRS	System.Rack.Slot addressing of a module
SW	Software
TMO	Timeout
t_{WDT}	Watchdog time
W	Write, the system variable or signal receives a value, e.g., from the user program
WD	Watchdog, device for monitoring the system's correct operation. Signal for fault-free process

Index of Figures

Index of Tables

Index

MANUAL
HIQuad X system manual

HI 803 211 E

For further information, please contact:

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

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

Learn more about HIMA solutions online:

www.hima.com/en//

www.hima.com