

HIMatrix

Safety-Related Controller

System Manual Modular Systems



HIMA Paul Hildebrandt GmbH + Co KG
Industrial Automation

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

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

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

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

© Copyright 2011, HIMA Paul Hildebrandt GmbH + Co KG

All rights reserved

Contact

HIMA Address

HIMA Paul Hildebrandt GmbH + Co KG

P.O. Box 1261

68777 Brühl, Germany

Phone: +49 6202 709-0

Fax: +49 6202 709-107

E-mail: info@hima.com

Revision index	Revisions	Type of Change	
		technical	editorial
1.00	The SILworX programming tool is taken into account The document layout was modified.	X	X
2.00	HIMatrix devices layout 3, SILworX V4, HIMatrix CPU operating system V.8, COM operating system V.13 are taken into account	X	X

Table of Contents

1	Introduction	7
1.1	Structure and Use of the Document.....	7
1.2	Target Audience.....	9
1.3	Formatting Conventions	9
1.3.1	Safety Notes	9
1.3.2	Operating Tips	10
1.4	Service and Training.....	10
2	Safety	11
2.1	Intended Use	11
2.1.1	Scope.....	11
2.1.2	Non-Intended Use.....	11
2.2	Operating Requirements	12
2.2.1	Climatic Requirements.....	12
2.2.2	Mechanical Requirements	13
2.2.3	EMC Requirements.....	13
2.2.4	Power Supply.....	14
2.2.5	ESD Protective Measures.....	14
2.3	Requirements to be met by the operator and the machine and system manufacturers.....	14
2.4	Residual Risk	14
2.5	Safety Precautions.....	14
2.6	Emergency Information.....	15
3	Product Description	16
3.1	Modules	16
3.1.1	Inputs	16
3.1.2	Outputs	17
3.2	Monitoring the Operating Voltage.....	18
3.3	Monitoring the Temperature State	18
3.4	Alarm and Sequence of Events Recording - with L3.....	19
3.4.1	Alarm and Events.....	19
3.4.2	Creating Events	19
3.4.3	Recording Events.....	20
3.4.4	Transfer of Events.....	20
3.5	Product Data.....	20
4	Communication	21
4.1	Ethernet	21
4.1.1	safeethernet.....	22
4.1.2	Maximum Communication Time Slice.....	23
4.1.3	Terminals for safeethernet/Ethernet	23
4.1.4	Communication with the PADT	25
4.2	Fieldbus Communication.....	26

5	Operating System	27
5.1	Functions of the Processor Operating System	27
5.2	Behavior in the Event of Faults.....	27
5.2.1	Permanent Faults on Inputs or Outputs	27
5.2.2	Temporary Faults on Inputs or Outputs.....	28
5.2.3	Internal Faults.....	28
5.3	The Processor System.....	28
5.3.1	Modes of Operation for the Processor System.....	28
5.3.2	Programming.....	29
6	User Program	30
6.1	Modes of Operation for the User Program.....	30
6.2	Multitasking - with L3.....	30
6.2.1	Multitasking Mode	33
6.3	Reload - with L3.....	36
6.4	General Information about Forcing	39
6.5	Forcing - CPU-OS Version7 and Newer.....	40
6.5.1	Forcing with Layout 3	40
6.5.2	Forcing with Layout 2	41
6.5.3	Restricting the Use of Forcing.....	42
6.6	Forcing - CPU-OS Versions Prior to 7	42
6.6.1	Time Limits	43
6.6.2	Configuration Parameters for Forcing	43
6.6.3	Force Allowed - CPU Switch	44
7	Start-Up.....	45
7.1	Installation and Mounting.....	45
7.1.1	Mounting.....	45
7.1.2	Mounting on a Flat Base	45
7.1.3	Mounting and Removing the Modules.....	45
7.1.4	Connecting the Input and Output Circuits.....	46
7.1.5	Earthing.....	47
7.1.6	Connecting the Operating Voltage	47
7.1.7	Using the Reset Key with the CPU 01 Module.....	48
7.2	Configuring a Resource with SILworX - CPU-OS Version 7 and Newer.....	48
7.2.1	Configuring the Resource.....	48
7.2.1.1	Resource Properties.....	49
7.2.1.2	Hardware System Variables.....	51
7.2.1.3	Hardware System Variables for Reading the Parameters.....	52
7.2.2	Configuring the Ethernet Interfaces.....	54
7.2.3	Configuring the User Program.....	54
7.2.4	Configuring the Inputs and Outputs.....	55
7.2.5	Generating the Resource Configuration	57
7.2.6	Configuring the System ID and the Connection Parameters.....	58
7.2.7	Loading a Resource Configuration after a Reset	58

7.2.8	Loading a Resource Configuration from the PADT.....	59
7.2.9	Loading a Resource Configuration from the Flash Memory of the Communication System.....	59
7.2.10	To clean-up a resource configuration in the flash memory of the communication system.....	60
7.2.11	Setting the Date and the Time	60
7.3	User Management in SILworX - CPU-OS Version 7 and Newer.....	61
7.3.1	User Management for SILworX Projects.....	61
7.3.2	User Management for the Controller.....	61
7.3.3	Parameters for User Accounts.....	63
7.3.4	Setting Up User Accounts.....	63
7.4	Configuring Communication with SILworX - CPU-OS Version 7 and Newer .	64
7.4.1	Configuring the Ethernet Interfaces	64
7.5	Configuring the Sequence of Events Recording - with L3.....	65
7.6	Configuring a Resource Using ELOP II Factory - CPU-OS Versions Prior to 7	68
7.6.1	Configuring the Resource	68
7.6.2	Configuring the User Program	69
7.6.3	Configuring the Inputs and Outputs	71
7.6.4	Generating the Code for the Resource Configuration.....	72
7.6.5	Configuring the System ID and the Connection Parameters	72
7.6.6	Loading a Resource Configuration after a Reset.....	73
7.6.7	Loading a Resource Configuration from the PADT.....	73
7.6.8	Loading a Resource Configuration from the Flash Memory of the Communication System.....	74
7.6.9	Deleting a Resource Configuration from the Flash Memory of the Communication System.....	75
7.7	Configuring Communication with ELOP II Factory - CPU-OS Versions Prior to 7	75
7.7.1	Configuring the Ethernet Interfaces	75
7.7.2	System Signals of safe ethernet Communication	78
7.7.3	Configuring the safe ethernet connection	80
7.7.4	Configuring the Signals for safe ethernet Communication.....	81
7.8	Handling the User Program	82
7.8.1	Setting the Parameters and the Switches.....	82
7.8.2	Starting the Program from STOP/VALID CONFIGURATION	82
7.8.3	Restarting the Program after Errors.....	83
7.8.4	Stopping the Program	83
7.8.5	Program Test Mode	83
7.8.6	Online Test.....	83
8	Operation	84
8.1	Handling.....	84
8.2	Diagnosis.....	84
8.2.1	Light-Emitting Diode Indicators	84
8.2.2	Diagnostic History	84

8.2.3	Diagnosis in SILworX- CPU-OS Version 7 and Newer.....	86
8.2.4	Diagnosis in ELOP II Factory - CPU-OS Versions Prior to 7.....	86
9	Maintenance	87
9.1	Disturbances.....	87
9.2	Back-up Battery	87
9.3	Replacing Fans	87
9.4	Loading Operating Systems.....	88
9.4.1	Loading the Operating System with SILworX.....	88
9.4.2	Loading the Operating System with ELOP II Factory	88
9.4.3	Switching between ELOP II Factory and SILworX - not with L3.....	89
9.4.3.1	Upgrading from ELOP II Factory to SILworX.....	89
9.4.3.2	Downgrading from SILworX to ELOP II Factory	90
10	Decommissioning	91
11	Transport	92
12	Disposal	93
	Appendix	95
	Glossary	95
	Index of Figures.....	96
	Index of Tables	97
	Declaration of Conformity	99
	Index	100

1 Introduction

The modular HIMatrix F60 system described in this manual is safety-related and can be used for various purposes. The following conditions must be met to safely install and start up the HIMatrix automation devices, and to ensure safety during their operation and maintenance:

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

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

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

1.1 Structure and Use of the Document

This System Manual is composed of the following chapters:

Safety	Information on how to safely use the HIMatrix system. Allowed applications and environmental requirements for operation of HIMatrix systems.
Product Description	Basic structure of the HIMatrix system.
Communication	Brief description of the HIMatrix modular systems' communication among each other and with other systems. Detailed information can be found in the communication manuals.
Operating System	Functions of the operating systems.
User Program	Basic information on the user program.
Start-up, operation, maintenance, placing out of operation, transport, disposal	Phases of a HIMatrix system's life cycle.
Appendix	<ul style="list-style-type: none"> ▪ Glossary ▪ Index of tables and index of figures ▪ Declaration of Conformity ▪ Index

This manual distinguishes between the following variants of the HIMatrix system:

Programming tool	Processor operating system	Communication operating system	Hardware Layout
SILworX	CPU-OS Versions 8 and newer	COM-OS Version 13 and newer	L3
SILworX	CPU-OS Version 7 and newer	COM-OS Version 12 and newer	L2
ELOP II Factory	CPU- OS Versions prior to 7	COM-OS Versions prior to 12	L2

Table 1: HIMatrix System Variants

Operating systems for devices with hardware layout 3 cannot be used for devices with hardware layout 2, and vice versa.

Devices with hardware layout 3 have extended features, e.g., multitasking or reload, with respect to devices with hardware layout 2. In the chapter header or within the various sections of this document, these extended features are referred to as L3.

The manual distinguishes among the different variants using:

- Separated chapters
- Tables differentiating among the versions, e.g., CPU-OS version 7 and newer, or prior to 7

i

Projects created with ELOP II Factory cannot be edited with SILworX, and vice versa!

i

This manual usually refers to compact controllers and remote I/Os as *devices*, and to the plug-in cards of a modular controller as *modules*.

Modules is also the term used in SILworX.

Additionally, the following documents must be taken into account:

Name	Applicable for version	Content	Document number
Himatrix Safety Manual	All versions	Safety functions of the HIMatrix system	HI 800 023 E
HIMatrix Engineering Manual	All versions	Project planning description for HIMatrix systems	HI 800 101 E
Communication Manual	Version 7 and newer	Description of the communication protocols, ComUserTask and their configuration in SILworX	HI 801 101 E
HIMatrix PROFIBUS DP Master/Slave Manual	Versions prior to 7	Description of the PROFIBUS protocol and its configuration in ELOP II Factory	HI 800 009 E
HIMatrix Modbus Master/Slave Manual	Versions prior to 7	Description of the Modbus protocol and its configuration in ELOP II Factory	HI 800 003 E
HIMatrix TCP S/R Manual	Versions prior to 7	Description of the TCP S/R protocol and its configuration in ELOP II Factory	HI 800 117 E
HIMatrix ComUserTask (CUT) Manual	Versions prior to 7	Description of the ComUserTask and its configuration in ELOP II Factory	HI 800 329 E
SILworX Online Help	Version 7 and newer	Instructions on how to use SILworX	-
ELOP II Factory Online Help	Versions prior to 7	Instructions on how to use ELOP II Factory, Ethernet IP protocol, INTERBUS protocol	-
First Steps SILworX	Version 7 and newer	Introduction to SILworX using the HIMax system as an example	HI 801 103 E
First Steps ELOP II Factory	Versions prior to 7	Introduction to ELOP II Factory	HI 800 006 E

Table 2: Additional Relevant Documents

The latest manuals can be downloaded from the HIMA website at www.hima.com. The revision index on the footer can be used to compare the current version of existing manuals with the Internet edition.

In addition to the Table 2 documents, the manuals specific to the F60 modules in use must be taken into account.

1.2 Target Audience

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

1.3 Formatting Conventions

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

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

Safety notes and operating tips are particularly marked.

1.3.1 Safety Notes

The safety notes are represented as described below.

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

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

SIGNAL WORD



Type and source of danger!
Consequences arising from the danger
Danger prevention

The signal words have the following meanings:

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

- Notice indicates a hazardous situation which, if not avoided, could result in property damage.

NOTE

Type and source of damage!
Damage prevention

1.3.2 Operating Tips

Additional information is structured as presented in the following example:

i

The text corresponding to the additional information is located here.

Useful tips and tricks appear as follows:

TIP

The tip text is located here.

1.4 Service and Training

Deadlines and the extent of actions for commissioning, testing and modifying controller systems can be agreed with the service department.

HIMA holds training, usually in-house, for software programs and the hardware of the controllers. Additionally, customer training can be offered on-site.

Refer to the HIMA website at www.hima.com for the current training program and dates. Offers for specialized, on-site training can also be provided upon request.

2 Safety

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

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

2.1 Intended Use

2.1.1 Scope

The safety-related HIMatrix controllers can be used in applications up to SIL 3 in accordance with IEC 61508, in case of railway applications also up to SIL 4 in accordance with EN 50126, EN 50128, and EN 50129, see Safety Manual for Railway Applications.

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

When implementing safety-related communications between the various devices, ensure that the system's overall response time does not exceed the fault tolerance time. All calculations must be performed in accordance with the rules given in Safety Manual HI 800 023.

Only devices with safe electrical isolation may be connected to the communications interfaces.

The-Energize to Trip Principle/ Energize to Trip Principle

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

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

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

The HIMatrix controllers can be used in applications that operate in accordance with the 'energize to trip' principle.

A system operating in accordance with the *energize to trip* principle requires power (such as electrical or pneumatic power) to perform its safety function.

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

Use in Fire Alarm Systems

The HIMatrix systems with detection of short-circuits and open-circuits are tested and certified for used in fire alarm systems in accordance with DIN EN 54-2 and NFPA 72. To contain the hazard, these systems must be able to adopt an active state on demand.

The operating requirements must be observed!

2.1.2 Non-Intended Use

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

Fieldbus interfaces cannot ensure safety-related communication.

2.2 Operating Requirements

The use of the HIMatrix systems is only permitted under the environmental conditions specified in the following section.

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

Standard	Content
EC/EN 61131-2: 2006	Programmable controllers, Part 2 Equipment requirements and tests
IEC/EN 61000-6-2: 2005	EMC Generic standards, Parts 6-2 Immunity for industrial environments
IEC/EN 61000-6-4: 2006	Electromagnetic Compatibility (EMC) Generic emission standard, industrial environments

Table 3: Standards for EMC, Climatic and Environmental Requirements

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

Requirement type	Requirement content
Protection class	Protection class II in accordance with IEC/EN 61131-2
Pollution	Pollution degree II in accordance with IEC/EN 61131-2
Altitude	< 2000 m
Housing	Standard: IP20 If required by the relevant application standards (e.g., EN 60204, EN 15849), the HIMatrix system must be installed in an enclosure of the specified protection class (e.g., IP54).

Table 4: General requirements

2.2.1 Climatic Requirements

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

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

Table 5: Climatic Requirements

Deviating climatic requirements are specified in the manuals of the devices or modules.

2.2.2 Mechanical Requirements

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

IEC/EN 61131-2	Mechanical tests
	Vibration immunity test: 5...9 Hz / 3.5 mm 9...150 Hz, 1 g, EUT in operation, 10 cycles per axis
	Shock immunity test: 15 g, 11 ms, EUT in operation, 3 shocks per axis (18 shocks)

Table 6: Mechanical Tests

2.2.3 EMC Requirements

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

IEC/EN 61131-2	Interference immunity tests	Criterion FS
IEC/EN 61000-4-2	ESD test: 6 kV contact, 8 kV air discharge	6 kV, 8 kV
IEC/EN 61000-4-3	RFI test (10 V/m): 80 MHz...2 GHz, 80 % AM RFI test (3 V/m): 2 GHz...3 GHz, 80 % AM RFI test (20 V/m): 80 MHz...1 GHz, 80 % AM	- - 20 V/m
IEC/EN 61000-4-4	Burst test Power lines: 2 kV and 4 kV Signal lines: 2 kV	4 kV 2 kV
IEC/EN 61000-4-12	Damped oscillatory wave test 2.5 kV L-,L+ / PE 1 kV L+ / L-	- -
IEC/EN 61000-4-6	High frequency, asymmetrical 10 V, 150 kHz...80 MHz, 80% AM 20 V, ISM frequencies, 80 % AM	10 V -
IEC/EN 61000-4-3	900 MHz pulses	-
IEC/EN 61000-4-5	Surge: Power lines: 2 kV CM, 1 kV DM Signal lines: 2 kV CM, 1 kV DM at AC I/O	2 kV /1 kV 2 kV

Table 7: Interference Immunity Tests

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

Table 8: Noise Emission Tests

2.2.4 Power Supply

The following table lists the key tests and thresholds for the HIMatrix systems' power supply:

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

Table 9: Review of the DC Supply Characteristics

2.2.5 ESD Protective Measures

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

NOTE



Electrostatic discharge can damage the electronic components within the HIMatrix systems!

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

2.3 Requirements to be met by the operator and the machine and system manufacturers.

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

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

2.4 Residual Risk

No imminent danger results from a modular HIMatrix F60 system itself.

Residual risk may result from:

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

2.5 Safety Precautions

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

2.6 Emergency Information

A HIMatrix system is a part of the safety equipment of a site. If a device or a module fails, the site adopts the safe state.

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

3 Product Description

The modular HIMatrix F60 system is composed of a subrack and various modules inserted in the subrack slots.

Slots 1 and 2 are reserved, all remaining slots can be freely occupied:

- The wide slot 1 is always occupied with the power supply module (PS 01).
- Slot 2 is always occupied with the CPU module (CPU 01).
- The remaining slots can be freely occupied with all remaining module types.

The modular system F60 can also be connected to HIMatrix compact systems via **safeethernet**.

3.1 Modules

All modules of the HIMatrix F60 system are 6 rack units (RU) high, which correspond to 262 mm.

Specific slots are only reserved for the power supply module and the CPU module, see above.

NOTE



Controller damage!

Inserting and removing the modules during operation is not permitted.

Prior to modifying the assembly, ensure that the controller is shut down!

After modifying a controller assembly, its user program must be adapted to the change and reloaded.

Pluggable clamps located on the module's front plate are used to connect sensors and actuators. The modules indicate the status of digital signals via LEDs located next to the clamps.

3.1.1 Inputs

The module's input channels are used to transmit and adjust the signals between the sensors and the microprocessor systems on the CPU module.

The controller continuously tests the safety-related modules. Depending on the fault, it switches either only the affected channel off or the entire module, in which case the module is indicated as faulty. The controller's operating system then sets the system user program's input value to the safe value 0 (or the initial value).

Signal sources with their own voltage may also be connected instead of contacts. To this end, the signal source ground must be connected to the input ground.

Surges on Digital Inputs

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

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

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

Caution: This action increases the system's response time!

i

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

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

Line Control

Line control is used to detect short-circuits or open-circuits, e.g., on EMERGENCY STOP inputs complying with Cat. 4 in accordance with EN 954-1. Line control can be configured for the F60 system.

Application example: The outputs DO 1 through DO 8 of the DIO 24/16 01 module are connected to the digital inputs (DI) of the same module or of another module as follows:

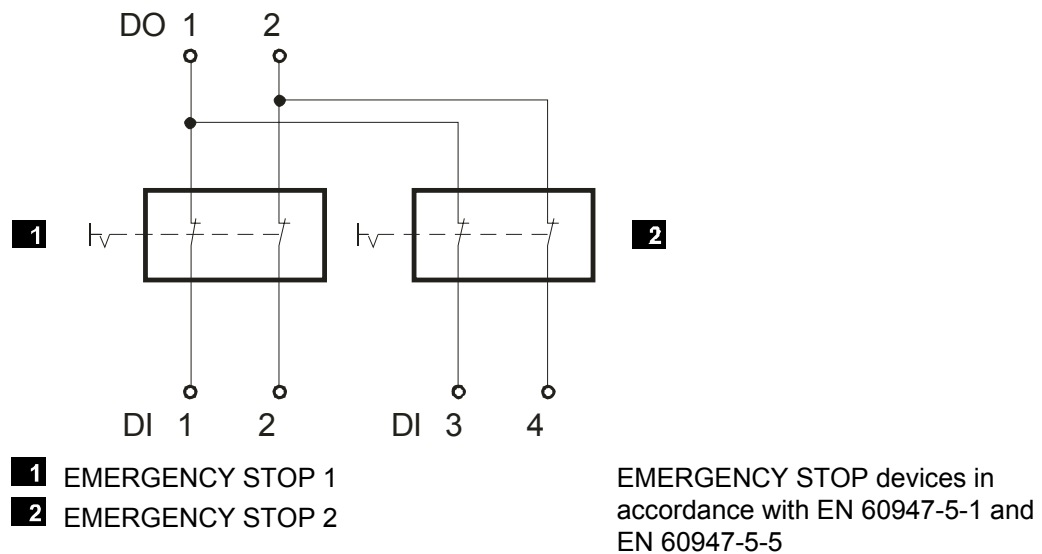


Figure 1: Line Control

The digital outputs are pulsed. This allows monitoring of the wires to the digital inputs of the F60 DI 32 01 or F60 DIO 24/16 01 module.

A fault reaction is triggered if one of the following faults occurs:

- Cross-circuit between two parallel wires.
- Improper connections of two lines (e.g., TO 2 to DI 3).
- Earth fault of a line (with earthed ground only).
- Open-circuit or open contacts, i.e., including when one of the two EMERGENCY STOP switches mentioned above has been engaged.

The fault reaction includes the following actions:

- The *ERROR* LED on the controller's front plate blinks.
- The inputs are set to 0.
- An evaluable error code is created.

For more information on how to configure line control in the user program, refer to the HIMatrix Engineering Manual (HI 800 101 E).

3.1.2 Outputs

The module's output channels are used to transmit and adjust the signals between the microprocessor systems on the central module and the actuators.

The controller continuously tests the modules. Depending on the fault, it switches either only the affected channel off or the entire module, in which case the module is indicated as faulty. The outputs are set to the safe de-energized state.

NOTE



Controller damage!

Do not plug the terminals for output circuits if a load is connected.

- **If short-circuits are present, the resulting high current may damage the terminals.**

Inductive loads may be connected with no free-wheeling diode on the actuator. However, HIMA recommends connecting a protective diode directly to the actuator.

For more information on line control, refer to Chapter 3.1.1

3.2 Monitoring the Operating Voltage

The device monitors the 24 VDC voltage during operation. Reactions occur in accordance with the specified voltage level:

Voltage level	Reaction of the device
19.3...28.8 V	Normal operation
< 18.0 V	Alarm states (internal variables are written and provided to the inputs or outputs)
< 12.0 V	Switching off the inputs and outputs

Table 10: Operating Voltage Monitoring

The *Power Supply State* system variable is used to evaluate the operating voltage state with the programming tool or from within the user program.

3.3 Monitoring the Temperature State

One or multiple sensors are used to measure the temperature at relevant positions within the device or system.

If the measured temperature exceeds the defined temperature threshold, the value of the *Temperature State* system variable changes as follows:

Temperature	Temperature range	<i>Temperature State</i> [BYTE]
< 60 °C	Normal	0x00
60 °C...70 °C	High temperature	0x01
> 70 °C	Very high temperature	0x03
Back to 64 °C...54 °C ¹⁾	High temperature	0x01
Back to < 54 °C ¹⁾	Normal	0x00

¹⁾ The hysteresis of sensors is 6 °C.

Table 11: Temperature Monitoring

If no or insufficient air circulates within a control cabinet and natural convection is not enough, the threshold associated with the *High Temperature* in the HIMA controller can already be exceeded at ambient temperatures of less than 35 °C.

This can be due to local heating or to a bad heat conduction. In particular with digital outputs, the heat levels strongly depend on their load.

The *Temperature State* system variable allows the user to read the temperature. If the state *Very high temperature* often occurs, HIMA recommends improving the system heat dissipation, e.g., by taking additional ventilation or cooling measures, such that the long life time of the HIMA systems can be maintained.

i

The safety of the system is not compromised if the state *High Temperature* or *Very High Temperature* is entered.

3.4 Alarm and Sequence of Events Recording - with L3

The HIMatrix system is able to record alarms and sequence of events (sequence of events recording, SOE) .

3.4.1 Alarm and Events

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

Alarms are events that signalize an increasing risk potential.

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

HIMatrix differentiates between Boolean and scalar events.

Boolean Events:

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

Scalar Events:

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

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

3.4.2 Creating Events

The processor system is able to create events.

The processor system uses global variables to create the events and stores them in the buffer, see Chapter 3.4.3. The events are created in the user program cycle.

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

System Events

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

- **Overflow:** Some events were not stored due to buffer overflow. The timestamp of the overflow event corresponds to that of the event causing the overflow.
- **Init:** The event buffer was initialized.

System events contain the SRS identifier of the device causing the events.

Status Variables

Status variables provide the user program with the state of scalar events. Each of the following states is connected to a status variable and can be assigned a global variable of type BOOL:

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

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

3.4.3 Recording Events

The processor system collects the events:

The processor system stores all the events in its buffer. The buffer is part of the non-volatile memory and has a capacity of 1 000 events.

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

3.4.4 Transfer of Events

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

3.5 Product Data

Designation	Value, range of values
Power supply Power supply module	24 VDC, -15 %...+20 %, $r_P \leq 15\%$, externally fused with 32 A
Backup battery	In the power supply module (PS01) Part no. 44.000 0019 In the CPU module (CPU 01): Goldcap (for buffering date/time only)
Operating temperature	0 °C...+60 °C
Storage temperature	-40 °C...+85 °C
Type of protection	IP20 (unused slots covered, cover part no.: 60.528 2106)
Dimensions	260 mm x 312 mm x 245 mm (W x H x D)
Weight	max. 10 kg (completely assembled)

Table 12: Specifications of F60

The specifications for the input module are detailed in the corresponding manuals.

4 Communication

HIMatrix controllers use the following protocols to communicate:

- **safeethernet**
safety-related protocol enabling controllers and remote I/O to communicate with one another
- Fieldbus protocols for connecting external devices or systems
- Communication with the PADT

The communication system is connected to the safety-related microprocessor system via a dual port RAM. It controls the controller's communication with other systems via efficient interfaces:

Interface	Protocols
RJ-45 connector	safeethernet EtherNet/IP (only with ELOP II Factory Online Help - versions prior to 7) OPC Programming tool (PADT) TCP SR SNTP Modbus TCP ComUserTask L3: PROFINET and PROFI-safe
D-sub connector	PROFIBUS DP Modbus ComUserTask INTERBUS (only with ELOP II Factory Online Help - versions prior to 7)

Table 13: Communication Protocols and Interfaces

4.1 Ethernet

The HIMatrix controllers and remote I/Os are equipped with Ethernet switches with RJ-45 connectors to which Ethernet cables can be attached for connecting to other devices:

- In contrast to a hub, a switch can analyze and temporarily store data packets, thus allowing a temporary targeted connection to be established between two communication partners (sender/receiver) for transmitting the data. This prevents collisions (typical of hubs) and reduces the load on the network. To ensure targeted data forwarding, each switch must have an address/port assignment table. This table is automatically generated by the switch during a self-learning process. In this table, MAC addresses are assigned to a specific port on the switch. Inbound data packets are forwarded directly to the corresponding port in accordance with this table.
- The switch automatically switches between transfer rates of 10 and 100 MBit/s and between full and half duplex connections. In this way, the full bandwidth is available in both directions (full duplex operation).
- A switch controls communication among multiple final devices. The switch can address up to 1000 absolute MAC addresses.
- The 'autocrossing' function recognizes when crossover cables are attached and the switch automatically adjusts itself accordingly.

NOTE

- **When configuring safety-related communication, observe the instructions specified in the Safety Manual.**

4.1.1 safeethernet

Requirements as determinism, reliability, interchangeability, extendibility and above all safety, are central issues for automation technology.

safeethernet provides a transfer protocol for transmitting safety-related data up to SIL 3 based upon Ethernet technology.

safeethernet implements mechanism that can detect and safely react to the following faults:

- Corruptions of the transmitted data (duplicated, lost and changed bits)
- Invalid message addressing (transmitter, receiver)
- Wrong data sequence (repetition, loss, exchange)
- Invalid timing (delay, echo)

safeethernet is based on the IEEE802.3 standard.

The standard Ethernet protocol frame is used to transmit safety-related data.

safeethernet uses unsafe data transfer channels (Ethernet) in accordance with the black channel approach and monitors them on the transmitter and receiver side by using safety-related protocol mechanism. This allows the users to use normal Ethernet network components such as hubs, switches, routers within a safety-related network.

safeethernet uses the abilities of standard Ethernet so that security and real time ability are made possible. A special protocol mechanism ensures a deterministic behavior even if faults occur or new communication participants join the network. The system automatically integrates new components in the running system. All network components can be replaced during operation. Transmission times can be clearly defined using switches. Ethernet is thus real-time capable.

Copper lines and fiber optic cables can be used as transmission media.

safeethernet allows both connections to the company Intranet and to the Internet. Therefore, just one network is required for both safety-related and non-safety-related data transmission.

i

The network may be shared with other participants if sufficient transfer capacity is available.

⚠ WARNING

Manipulation of safety-related data transfer!

Physical injury

The operator is responsible for ensuring that the Ethernet used for safeEthernet is sufficiently protected against manipulations (e.g., from hackers).

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

safeEthernet allows the user to create flexible system structures for decentralize automation with defined reaction times. Depending on the requirements, the intelligence can be distributed to the network participants in a centralized or decentralized manner.

4.1.2 Maximum Communication Time Slice

The maximum communication time slice is the time period in milliseconds (ms) and per cycle assigned to the processor system for processing the communication tasks. If not all upcoming communication tasks can be processed within one cycle, the whole communication data is transferred over multiple cycles (number of communication time slices > 1).

i

When calculating the maximum response times allowed, the number of communication time slices must be equal to 1, see Communication Manual (HI 801 101 E). The duration of the communication time slice must be set such that, when using the communication time slice, the cycle cannot exceed the watchdog time specified by the process.

4.1.3 Terminals for safeEthernet/Ethernet

For networking via safeEthernet/Ethernet, the compact systems are equipped with 2 or 4 terminals, depending on the model, which are located on the housing's lower and upper sides.

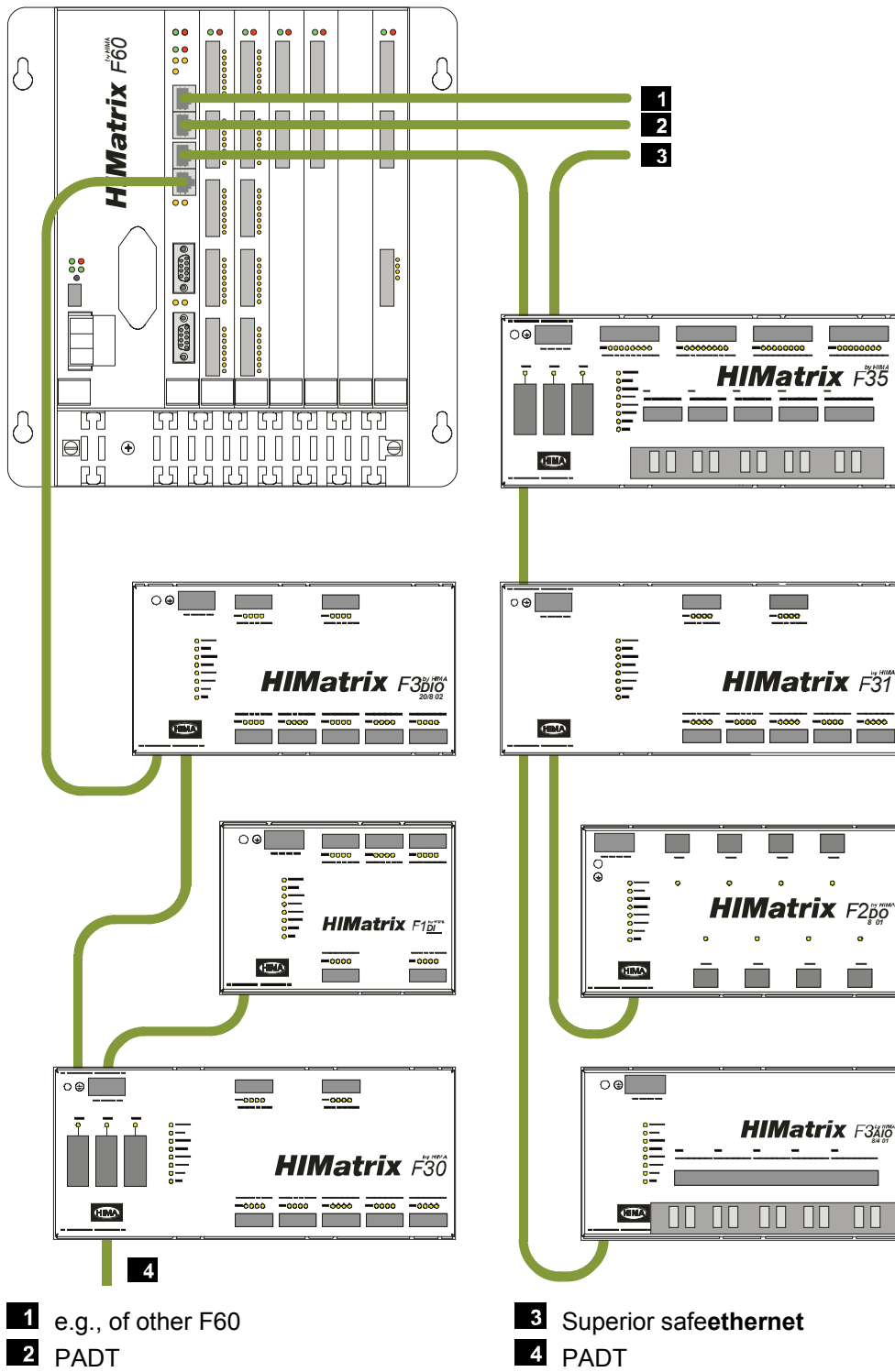


Figure 2: safeEthernet/Ethernet Networking Example

The different systems can be connected to one another via Ethernet in any configuration (e.g., star or linear network); the PADT may also be connected at any location.

NOTE

Ethernet operation may be disturbed!

Ensure that no network rings result from interconnecting the controllers. Data packets may only reach a system over a single path.

With the connection of controllers and remote I/Os with different versions of operating systems via **safeethernet**, the following cases must be observed:

Operating system of controller	Operating system of remote I/O	safeethernet connection possible?
Version 7 and newer	Versions 7 and newer	Yes
Versions prior to 7	Versions prior to 7	Yes
Versions prior to 7	Version 7 and newer	Yes
Version 7 and newer	Versions prior to 7	No

Table 14: Connection of Controllers and Remote I/Os with Different Operating Systems

Controllers with different operating system versions (versions 7 and newer and versions prior to 7) can be connected with cross-project communication, refer to the Communication Manual HI 801 101 E.

The connectors for Ethernet communication are located on the CPU module's front plate. They consist of 4 RJ-45 connectors, individually labeled with 1...4.

4.1.4 Communication with the PADT

A HIMatrix controller communicates with a PADT via Ethernet. A PADT is a computer that is installed with a programming tool, either SILworX or ELOP II Factory. The programming tool must comply with the processor operating system version of the controller, either version 7 and newer (SILworX) or prior to 7 (ELOP II Factory).

The computer must be able to reach the controller via Ethernet.

A controller 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 programming tools can only read information. If they try to establish a writing connection, the controller only allows them a read-only access.

4.2 Fieldbus Communication

The CPU module is equipped with two fieldbus communication terminals. These 9-pole D-sub connectors are located on the front plate of the module.

The two interfaces can operate simultaneously.

NOTE



Fieldbus operation of other devices may be disturbed!

If the controller is reset, do not connect the fieldbus interfaces to an operational fieldbus to prevent potential disturbances.

The two fieldbus interfaces FB1 and FB2 can be equipped with fieldbus submodules. The fieldbus submodules are optional and must be mounted by the manufacturer.

The fieldbus interfaces are not operational without fieldbus submodules.

The available fieldbus submodules are described in the SILworX communication manual (HI 801 101 E). Refer to the Engineering Manual (HI 800 101 E) for further details.

5 Operating System

The operating system includes all basic functions of the HIMatrix controller (programmable electronic system, PES).

Which application functions the PES should perform is specified in the user program. A code generator translates the user program into a machine code. The programming tool transfers this machine code to the controller's flash memory.

5.1 Functions of the Processor Operating System

The following table specifies all basic functions of the operating system for a processor system and the connections to the user program:

Functions of the operating system	Connections to the user program
Cyclic processing the user program	It affects variables, function blocks
Automation device configuration	Defined by selecting the controller
Processor tests	- - -
I/O module tests	Depending on the type
Reactions in the event of a fault:	Defined by default The user program is responsible for the process reaction
Processor system and I/O diagnosis	Use of system signals/variables for error messages
Safe communication: peer-to-peer Non-safe communication: PROFIBUS DP, Modbus, INTERBUS	To define the use of communication signals/variables
PADT interface: Actions allowed	Defined in the programming tool: Configuration of protective functions, User log-in

Table 15: Functions of the Processor Operating System

Each operating system is inspected by the TÜV in charge and approved for operation in the safety-related controller. The valid versions of the operating system and corresponding signatures (CRCs) are documented in a list maintained by HIMA in co-operation with the TÜV.

5.2 Behavior in the Event of Faults

The reaction to faults detected during tests is important. The distinction between the following types of faults is made:

- Permanent faults on inputs or outputs
- Temporary faults on inputs or outputs
- Internal Faults

5.2.1 Permanent Faults on Inputs or Outputs

A fault that occurs on an input or output channel has not effect on the controller. The operating system only considers the defective channel as faulty, and not the entire controller. The remaining safety functions are not affected and remain active.

With faulty input channels, the operating system sends the safe value 0 or the initial value for processing.

Faulty output channels are set to the de-energized state by the operating system. If it is not possible to only switch off a single channel, the entire output module is considered as faulty.

The operating system sets the fault status signal and reports the type of fault to the user program.

If the controller is not able to switch off the corresponding output and the secondary switch-off procedure also has no effect, the controller enters the STOP state. The watchdog of the processor system then switches the outputs off.

If faults are present in the I/O modules for longer than 24 hours, only the affected I/O modules are permanently switched off by the controller.

5.2.2 Temporary Faults on Inputs or Outputs

If a fault occurs in an input or output module and disappears by itself, the operating system resets the fault status and resumes normal operation.

The operating system statistically evaluates the frequency with which a fault occurs. If the specified fault frequency is exceeded, it permanently sets the module status to faulty. In this way, the module no longer operates, even if the fault disappears. The module is released and the fault statistics are reset when the controller operating state switches from STOP to RUN. This change acknowledges the module fault.

5.2.3 Internal Faults

In the rare case of an internal fault within the HIMatrix controller, the fault reaction depends on the version of the operating system loaded into the controller:

- Versions of the processor operating system prior to V.6.44 for controllers, and prior to V.6.42 for remote I/Os:
The HIMatrix controller enters the ERROR STOP state, and all outputs adopt the safe (de-energized) state. The HIMatrix controller must be restarted manually, e.g., using the programming tool.
- For controllers, processor operating system version V.6.44 and beyond, and for remote I/Os V.6.42 and beyond:
The HIMatrix controller is automatically started. Should an internal fault be detected again within the first minute after start up, the HIMatrix controller will remain in the STOP/INVALID CONFIGURATION state.

5.3 The Processor System

The processor system is the central component of the controller and communicates with the I/O modules of the controller via the I/O bus.

The processor system exchanges data with the communication system via a dual-ported RAM.

The processor system monitors the sequence and the proper, logical execution of the operating system and user program. The following functions are time monitored:

- Hardware and software self-tests of the processor system
- RUN cycle of the processor system (including the user program)
- I/O tests and processing of I/O signals

5.3.1 Modes of Operation for the Processor System

LEDs located on the front plate of the controller indicate the operating state of the processor system. The latter can also be reported by the PADT, together with other parameters specific to processor module and user program.

Stopping the processor interrupts the execution of the user program and sets the outputs of the controller and all remote I/Os to safe values.

Setting the EMERGENCY STOP system parameter to TRUE using a program logic brings the processor system to enter the STOP state.

The following table specifies the most important operating states:

Mode of operation	Description
INIT	Safe state of the processor system during the initialization phase. Hardware and software tests are performed.
STOP/INVALID CONFIGURATION	Safe state of the processor system with no execution of the user program All outputs of the controller are reset. Hardware and software tests are performed.
STOP/INVALID CONFIGURATION	Safe state of the processor system without a configuration loaded or after a system fault. All controller's outputs are reset, the hardware watchdog has not triggered. The processor system can only be rebooted using the PADT.
RUN	The processor system is active. The user program is run, I/O signals are processed. The processor system ensures safety-related and non-safety-related communication (if configured). Hardware and software tests, and test for configured I/O modules are performed.

Table 16: Modes of Operation for the Processor System

5.3.2 Programming

A PADT (programming and debugging tool) is used to program the HIMatrix controllers. The PADT is a PC equipped with one of the programming tools:

- SILworX for HIMatrix systems with operating system version 7 and newer.
- ELOP II Factory for HIMatrix systems with operating system version prior to 7.

The programming tools supports the following programming languages in accordance with IEC 61131-3:

- Function block diagrams (FBD)
- Sequential function charts (SFC)

The programming tools are suitable for developing safety-related programs and for operating the controllers.

For more details on the programming tools, refer to the manuals 'First Steps ELOP II Factory' (HI 800 006 E) and 'First Steps SILworX' (HI 801 103 E), and to the corresponding online help.

6 User Program

In accordance with the IEC 61131-3 requirements, a PADT with installed programming tool, i.e., ELOP II Factory or SILworX, must be used to create and load the user program for the PES.

First, use the PADT to create and configure the user program for the controller's safety-related operation. To this end, observe the instructions specified in the Safety Manual (HI 800 023 E) and ensure that the requirements specified in the report to the certificate are met.

Once the compiling is complete, the programming device loads the user program (logic) and its configuration (connection parameters such as IP address, subnet mask and system ID) into the controller and starts them.

The PADT can be used to perform the following actions while the controller is operating:

- Starting and stopping the user program.
- Displaying and forcing variables or signals using the Force Editor.
- In test mode, executing the user program in single steps - not suitable for safety-related operation.
- Reading the diagnostic history.

This is possible, provided that the PADT and the controller are loaded with the same user program.

6.1 Modes of Operation for the User Program

Only one user program at a time can be loaded into a given controller. For this user program, the following modes of operations are allowed:

Mode of operation	Description
RUN	The processor system is in RUN. The user program is run cyclically. I/O signals are processed.
Test mode (single step)	The processor system is in RUN. The user is run cyclically, if previously set by the user. I/O signals are processed. Do not use this function in safety-related operation!
STOP	The processor system is in STOP. The user program is not or no longer run, the outputs are reset.
ERROR	A loaded user program has been stopped due to a failure. The outputs are reset. Note: The program can only be restarted using the PADT.

Table 17: User Program Modes of Operation

6.2 Multitasking - with L3

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

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

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

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

The overview does not include special tasks that might be executed within a CPU cycle such as reload.

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

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

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

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

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

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

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

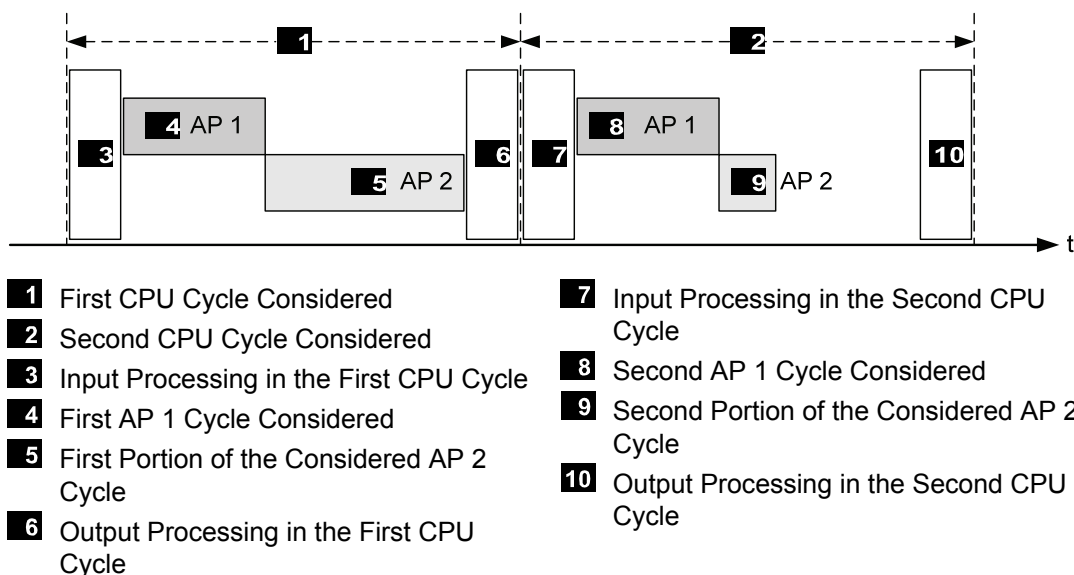


Figure 3: CPU Cycle Sequence with Multitasking

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

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

Upon completion of the first part **5** of the AP 2 cycle under consideration, AP 2 processing is **completely** aborted and only resumed when **9** starts. During its cycle, AP 2 processes

the data provided by the system at **3**. The results of AP 2 are available to the system at **10** (e.g., for process output). The data that the system exchanges with the user program are always consistent.

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

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

i

A license is required to use the multitasking feature.

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

Table 18: Parameters Configurable for Multitasking

Observe the following rules when setting the parameters:

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

- The sum of the *Max. Duration for Each Cycle [μs]* parameters in all user programs must be large enough to ensure that sufficient reserve is available to maintain the target cycle time.
- The *Program IDs* of all user programs must be unique.

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

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

i

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

Usually, the individual user programs run concurrently in a non-reactive manner. However, reciprocal influence can be caused by:

- Use of the same global variables in several user programs.
- Unpredictably long runtimes can occur in individual user programs if a limit is not configured with *Max Duration for Each Cycle*.

NOTE



Reciprocal influence of the user programs is possible!

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

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

i

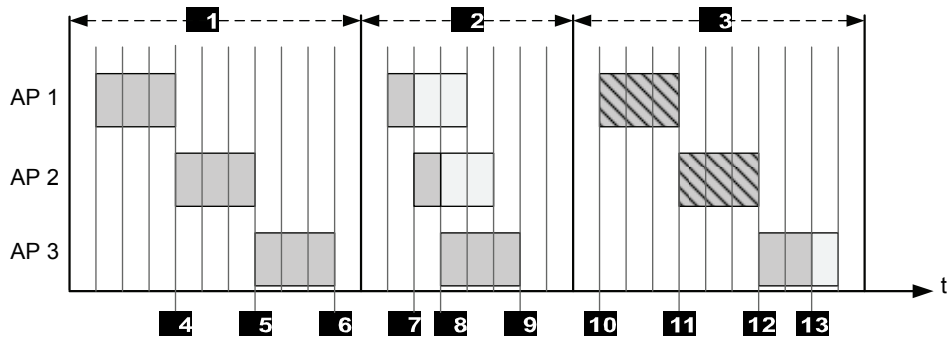
HIMA recommends to set the *Max. Duration for each Cycle [μs]* parameter to an appropriate value ≠ 0. This ensures that a user program with an excessively long runtime is stopped during the current CPU cycle and resumed in the next CPU cycle without affecting the other user programs.

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

6.2.1 Multitasking Mode

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

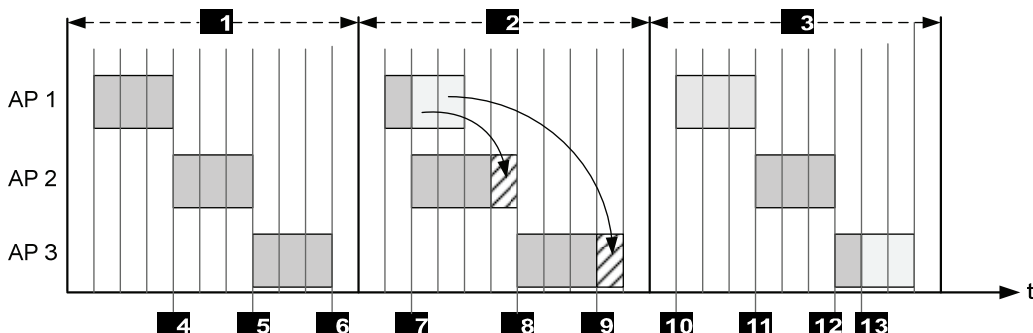
1. **Multitasking Mode 1** uses the unneeded time to reduce the CPU cycle. If the user program is completely processed, processing of the next user program begins immediately. In total, this results in a shorter cycle.
Example: 3 user programs (AP 1, AP 2 and AP 3) that allow a user program cycle to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



- 1** First CPU Cycle Considered.
- 2** Second CPU Cycle Considered.
- 3** Third CPU Cycle Considered.
- 4** The *Max. Duration for Each Cycle [μs]* of AP 1 has Expired, AP 2 Starts.
- 5** The *Max. Duration for Each Cycle [μs]* of AP 2 has Expired, AP 3 Starts.
- 6** The *AP 3 Max. Duration for Each Cycle [μs]* has Expired, Completion of the First CPU Cycle.
- 7** Completion of the AP 1 Cycle, AP 2 Resumes.
- 8** Completion of the AP 2 Cycle, AP 3 Resumes.
- 9** The *AP 3 Max. Duration for Each Cycle [μs]* has Expired, Completion of the Second CPU Cycle.
- 10** The next User Program Cycle of AP 1 Starts.
- 11** *Max. Duration for Each Cycle [μs]* of AP 1 has Expired. The next User Program Cycle of AP 2 Starts.
- 12** The *Max. Duration for Each Cycle [μs]* of AP 2 has Expired, AP 3 Starts.
- 13** Completion of the AP 3 Cycle.

Figure 4: Multitasking Mode 1

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



- 1** First CPU Cycle Considered.
- 2** Second CPU Cycle Considered.
- 3** Third CPU Cycle Considered.
- 4** The *Max. Duration for Each Cycle [μs]* of AP 1 has Expired, AP 2 Starts.
- 5** The *Max. Duration for Each Cycle [μs]* of AP 2 has Expired, AP 3 Starts.
- 6** The AP 3 *Max. Duration for Each Cycle [μs]* has Expired, Completion of the First CPU Cycle.
- 7** Completion of the AP 1 Cycle, AP 2 Resumes. The Remaining Duration is Distributed Among the AP 2 and AP 3 *Max. Duration for Each Cycle [μs]* (Arrows).
- 8** AP 2 *Max. Duration for Each Cycle [μs]* + Proportional Remaining Duration of AP 1 have Expired, AP 3 Resumes.
- 9** AP 3 *Max. Duration for Each Cycle [μs]* + Proportional Remaining Duration of AP 1 have Expired, Completion of the Second CPU Cycle.
- 10** The next User Program Cycle of AP 1 Starts.
- 11** The *Max. Duration for Each Cycle [μs]* of AP 1 has Expired, AP 2 Starts.
- 12** Completion of the AP 2 Cycle, AP 3 Resumes.
- 13** Completion of the AP 3 Cycle.

Figure 5: Multitasking Mode 2

- i** The unused execution time of user programs that do not run cannot be exploited as residual time by other user programs. User programs do not run if they are in one of the following states:
- STOP
 - ERROR
 - TEST_MODE

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

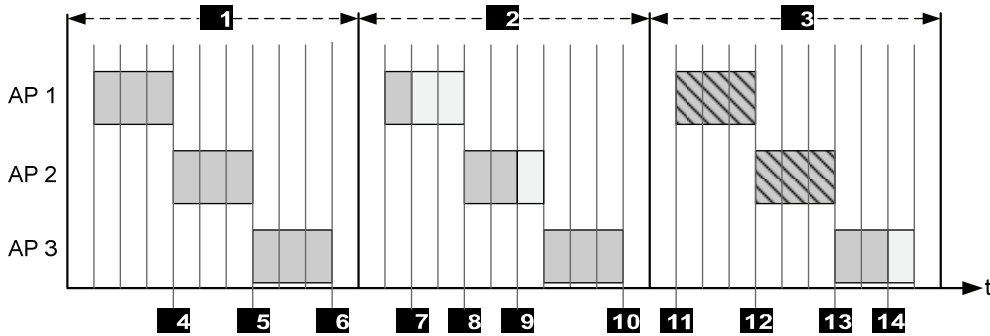
In such a case, if the value set for *Maximum Cycle Count* is too low, the maximum time for processing a user program can be exceeded and result in an error stop!
Maximum processing time = *Max. Duration for Each Cycle [μs]* * *Maximum Number of Cycles*

Use multitasking mode 3 to verify the parameter setting!

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

Multitasking mode 3 allows users to verify if multitasking mode 2 ensures proper program execution, even in the worst case scenario.

Example:



- 1** First CPU Cycle Considered.
- 2** Second CPU Cycle Considered.
- 3** Third CPU Cycle Considered.
- 4** The *Max. Duration for Each Cycle [μs]* of AP 1 has Expired, AP 2 Starts.
- 5** The *Max. Duration for Each Cycle [μs]* of AP 2 has Expired, AP 3 Starts.
- 6** The AP 3 *Max. Duration for Each Cycle [μs]* has Expired, Completion of the First CPU Cycle. AP 1 Resumes.
- 7** Completion of the AP 1 Cycle. Waiting for the Remaining Duration.
- 8** The AP 1 *Max. Duration for Each Cycle [μs]* has Expired, AP 2 Resumes.
- 9** Completion of the AP 2 Cycle. Waiting for the Remaining Duration.
- 10** The AP 3 *Max. Duration for Each Cycle [μs]* has Expired. Completion of the Second CPU Cycle.
- 11** The next User Program Cycle of AP 1 Starts.
- 12** The next *Max. Duration for Each Cycle [μs]* of AP 1 has Expired. The next User Program Cycle of AP 2 starts .
- 13** *Max. Duration for Each Cycle [μs]* of AP 2 has Expired.
- 14** Completion of the AP 3 Cycle. Standby Time Until the AP 3 *Max. Duration for Each Cycle [μs]* has Expired. Completion of the Third CPU Cycle.

Figure 6: Multitasking Mode 3

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

6.3 Reload - with L3

If user programs were modified, the changes can be transferred to the PES during operation. The operating system, checks and activates the modified user program which then assumes the control task.

i**Take the following point into account when reloading step chains:**

The reload information for step sequences does not take the current sequence status into account. The step sequence can be accordingly 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 step of the step chain has the *active* state.
 - Renaming the initial step while another step is active.
As a result, a step chain has two active steps!
-

i**Take the following point into account when reloading actions:**

During the reload, actions are loaded with their corresponding 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 outputs 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.
 - Deleting a *P0* action qualifier set to TRUE actuates the trigger.
-

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

i**The controller can interrupt a running reload process.**

A successful reload is ensured by planning a sufficient reserve for the reload when determining the watchdog time or temporarily increasing the controller watchdog time by a reserve.

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

Also exceeding the target cycle time can result in a reload interruption.

The reload can only be performed if the Reload Allowed system parameter is set to ON and the Reload Deactivation system variable is set to OFF.

i

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

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

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

This procedure has the following consequences, if names are changed and loaded into the PES by performing a reload:

- 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 is also the case for retain variables. The variables lose their current value.
- Renaming a function block instance results in initializing all variables, even retain variables, and all function block instances.
- Renaming a program results in initializing all contained variables and all function block instances.

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

Conditions for Using the Reload Function

A license is required to use the reload feature.

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

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

Changes to	Type of change			
	Add	Delete	Change of the initial value	Assignment of other variables
Assigning global variables to				
User programs	•	•	•	•
System variables	•	•	•	•
I/O channels	•	•	•	•
Communication protocols	-	-	-	-
safeethernet	-	-	•	-
SOE	-	-		
Communication protocols	-	-	n.a.	n.a.
User programs	•	•**	n.a.	n.a.
System ID, rack ID	-			
IP addresses	-			
User accounts and licenses	•			
• Reload possible - Reload impossible ** Reload possible, but the controller must still contain at least one user program n.a. non-applicable				

Table 19: Reloading after Changes - With L3

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

- TIP** Proceed as described below to be able to perform a reload even if global variable assignments have been added:
- While creating the user program, assign unused global variables to communication protocols.
 - Assign safe value as initial value to unused global variables.
- To a later time point, this assignment must only be changed and not added ensuring the possibility to perform a reload.

6.4 General Information about Forcing

Forcing is the procedure by which a variable's current value is replaced with a force value. The current value of a variable is assigned from one of the following sources:

- a physical input
- communication
- a logic operation.

When a variable is being forced, its value is defined by the user.

Forcing is used for the following purposes:

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

WARNING



Physical injury due to forced values is possible!

- **Only force values after receiving consent from the test authority responsible for the final system acceptance test.**
- **Only remove existing forcing restrictions with the consent of the test authority.**

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 to setting a time limit for the forcing procedure, see below.

NOTE



Use of forced values can disrupt the safety integrity!

- **Forced value may lead to incorrect output values.**
- **Forcing prolongates the cycle time. This can cause the watchdog time to be exceeded.**
- **Forcing is only permitted after receiving consent from the test authority responsible for the final system acceptance test.**

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

This document is available on the TÜV homepage:

<http://www.tuv-fs.com> or

<http://www.tuvasi.com>.

6.5 Forcing - CPU-OS Version7 and Newer

Forcing is the procedure by which a variable's current value is replaced with a force value. The current value of a variable is assigned from one of the following sources:

- a physical input
- communication
- a logic operation.

When a variable is being forced, its value is defined by the user.

Forcing is used for the following purposes:

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

⚠ WARNING



Physical injury due to forced values is possible!

- **Only force values after receiving consent from the test authority responsible for the final system acceptance test.**
- **Only remove existing forcing restrictions with the consent of the test authority.**

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 to setting a time limit for the forcing procedure, see below.

NOTE



Use of forced values can disrupt the safety integrity!

- **Forced value may lead to incorrect output values.**
- **Forcing prolongates the cycle time. This can cause the watchdog time to be exceeded.**
- **Forcing is only permitted after receiving consent from the test authority responsible for the final system acceptance test.**

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

This document is available on the TÜV homepage:

<http://www.tuv-fs.com> or
<http://www.tuvasi.com>.

Forcing can operate at two levels:

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

6.5.1 Forcing with Layout 3

To force a global or local variable, the following conditions must be met:

- The corresponding force switch is set.
- Forcing was started.

If forcing was started, a change to the force switch has an immediate effect.

If forcing was started and the force switch is set, a change to the force value has an immediate effect.

Local forcing can be started and stopped individually for each user program.

Time Limits

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

It is possible to define how the HIMatrix system should behave upon expiration of the time limit:

- With global forcing, the following settings can be selected:
 - The resource stops.
 - The resource continues to operate.
- With local forcing, the following settings can be selected:
 - The user program stops.
 - The user program continues to run.

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

If a variable is no longer forced, the process value is used again for the variable.

Force Editor

The SILworX Force Editor displays all the variables for which forcing is allowed. Global and local variables are grouped into two different tabs.

Use these tabs to configure the force values and set the force switches.

6.5.2 Forcing with Layout 2

Forcing in HIMatrix L2 systems is subjected to the restrictions described in the following section.

i

Absolutely take the following restrictions into account when forcing or evaluating online tests performed with forced global variables!

Global Forcing

To force a global variable, the following conditions must be met:

- The corresponding force switch is set.
- Forcing was started.

If forcing was started, a change to the force switch has an immediate effect.

If forcing was started and the force switch is set, a change to the force value has an immediate effect.

Global forced variables have the following characteristics:

- Outputs and communication protocols receive the force value as long as the variable is being forced.
- The following conditions apply for a user program reading and writing the variable:
 - The force value is used until the user program writes a new process value. After that moment, the process value applies for the remaining duration of the user program cycle. The force value applies then again in the following user program cycle.
 - If the user program does not write any process value, the force value continues to be used as the new process value, even after the end of the forcing process! The previous process value is no longer valid.

For global forcing, a time limit can be defined. Once the defined time has expired, the controller stops forcing values.

It is possible to define how the HIMatrix system should behave upon expiration of the time limit:

- The resource stops.

- The resource continues to operate.

Local Forcing

Local variable forcing is limited to the **Edit Local Process Value** command. This command directly changes the value of variables without the need to set a force switch or to start forcing. Additionally, no time limit can be configured for defining the validity of a used value.

The new process value set with this command (i.e., the force value) applies until one of the following events occurs:

- The user program overwrites the value with a new process value.
- A new value is entered.
- The user program is stopped.
- The user program is restarted.

Force Editor

The SILworX Force Editor displays all the variables for which forcing is allowed. Global and local variables are grouped into two specific tabs.

The tab for global variables can be used to configure the force values and set the force switches.

The tab for local variables can be used to enter the local process value.

6.5.3 Restricting the Use of Forcing

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

- Configuring different user profiles with or without forcing authorization.
- Prohibit global forcing for a resource.
- Prohibit local forcing or entering new process values.
- Forcing can also be stopped immediately using a key switch.
To do so, the *Force Deactivation* system variable must be assigned to a digital input connected to a key switch.

This system variable is not always enabled, see the following table.

Layout	Effect description
L3	<i>Force Deactivation</i> prevents global and local forcing from being started and stops an on-going forcing process.
L2	<i>Force Deactivation</i> prevents global forcing from being started and stops an on-going forcing process. <i>Force Deactivation</i> inhibits the Edit Local Process Values command, but it does not reset changed local variables to their previous process value.

Table 20: Effect of the *Force Deactivation* System Variable

6.6 Forcing - CPU-OS Versions Prior to 7

The force value is stored in the controller. If the CPU switches from RUN to STOP, the forcing procedure is deactivated to ensure that the controller does not accidentally start with active force signals.

i **Absolutely take the following facts into account when forcing or evaluating tests performed with forced global variables:**

Signal force values may be overwritten by the user program!
 Online test fields associated with forced signals may therefore show the forced value, even if a value generated by the user program has already been used in the ensuing calculations or is effective on an output.
 However, if the user program does not overwrite the values, e.g., in case of constants, the force value is used as a process value in the ensuing calculations.

6.6.1 Time Limits

It is possible to set a time limit for the forcing procedure. A configuration parameter defines how a controller should behave once the force time has expired:

- The processor enters the STOP state.
- The force value is no longer valid and the controller continues its normal operation.

In any case, exceeding the force time affects the user program and thus the process.

Forcing is terminated upon expiration of the force time or when forcing is intentionally stopped.

Provided that **Stop at Force Timeout** is set in the resource's properties (see also message in the info field), the controller enters the STOP state after the force time has expired and the user program continues to run with the process values.

If **Stop at Force Timeout** is not set, the controller is not stopped after the force time has expired. Forcing is deactivated and the values previously forced (R force values) are replaced with their process values.

This may have unintentional effects on the overall system.

To manually stop forcing, click the **Stop** button in the Force Editor. By doing so, the controller maintains the RUN state since the timeout has not been attained and the Stop at Force Timeout reaction was not defined.

6.6.2 Configuration Parameters for Forcing

The following table specifies the force switches and parameters :

Switch	Function	Default Value	Setting for safe operation	
Forcing allowed	A force function is enabled	FALSE	FALSE /ON ¹⁾	
Stop at Force Timeout	It stops the controller upon expiration of the force time	ON	ON	
Parameter	Function	Default Value	Indicators	
Forcing activated	Forcing Active	FALSE	FALSE	ON
Remaining force time	Time-limit for the force value, time (in seconds)	0	0	Remaining force time or -1
¹⁾ See also the warning message above: The <i>Force Allowed</i> and <i>Stop at Force Timeout</i> switches cannot be changed when a controller is operating and 'locked', i.e., define these settings prior to locking the controller.				

Table 21: Force Switches and Parameters Prior to V.7

Enter the value -1 for forcing without time limit.

6.6.3 Force Allowed - CPU Switch

- Not set:
 - Forcing is not possible (default setting).
 - The entered force values are kept, but are not effective.
- Set:
 - Forcing is allowed
 - The entered force values only become effective if the corresponding force switch has also been set for the data source.

Forcing Using Force Markers

Force markers are an additional option to force signals, e.g., for finding faults. Force markers are function blocks that can be used in the user program to force individual signals. Refer to the ELOP II Factory online help for more details.

WARNING



Physical injury due to forced signals is possible!

Remove all force markers from the user program prior to starting safety-related operation or before an acceptance test is performed by a test institute!

7 Start-Up

Commissioning of modular HIMatrix system comprises the following phases:

- Mounting the subrack in a suitable location and its assembly with modules.
- Electrical connection of power supply, earthing, sensors, and actuators
- Configuration
 - Writing the user program
 - Definition of safety, communication and other parameters

7.1 Installation and Mounting

This chapter describes how to install the controller mechanically and electrically.

7.1.1 Mounting

The location for installing a HIMatrix F60 subrack must be chosen observing the operating requirements (see Chapter 2.2) to ensure a smooth operation.

Observe the following points:

- Mount the subrack on horizontal DIN rails to ensure effective cooling.
- A distance of at least 100 mm above and below the subrack must be maintained.
- Do not mount the subrack above heating equipment or any heat source.
- Only mount the controller with unconnected connectors.

To mount the modules and observe the maximum operating temperature, follow the instructions specified in the HIMatrix Engineering Manual (HI 800 101 E).

7.1.2 Mounting on a Flat Base

The F60 subrack is equipped with two perpendicular joint bars each of which has two oblong holes for fastening. These joint bars must be secured on a flat base.

The mounting bolts must not exceed 6 mm in diameter and 13 mm head diameter. The screws and the selected base must be suitable to hold the weight of the controller.

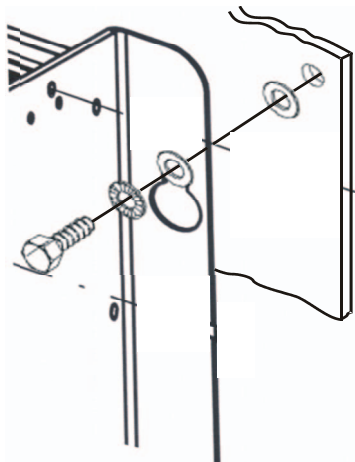


Figure 7: Securing the F60 Subrack

7.1.3 Mounting and Removing the Modules

To mount and remove the modules, the connection cable clamp terminals must be unplugged.

Additionally, personnel must be protected from electrostatic discharge.

NOTE



Electrostatic discharge!

Failure to comply with these instructions can damage the electronic components.

- Prior to working with HIMA components, touch an earthed object.
- Make sure that the workspace is free of static and wear an ESD wrist strap.
- If not used, ensure that the module is protected from electrostatic discharge, e.g., by storing it in its packaging.

Mounting the Modules

To mount a module into the subrack

1. Insert the module as far as it can go – without jamming it – into the two guiding rails which are located on the upper and lower part of the housing.
2. Apply pressure to the upper and lower extremity of the front plate until the module plugs snap into the backplane socket.
3. Secure the module with the screws located on upper and lower extremity of the front plate.

The module is mounted.

Removing the Modules

To remove a module from the subrack

1. Remove the plugs from the module front plate.
2. Release the locking screws located on the upper and lower extremity of the front plate.
3. Loosen the module using the handle located on the lower part of the front plate and remove it from the guiding rails.

The module is removed.

7.1.4 Connecting the Input and Output Circuits

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

NOTE



Electrostatic discharge!

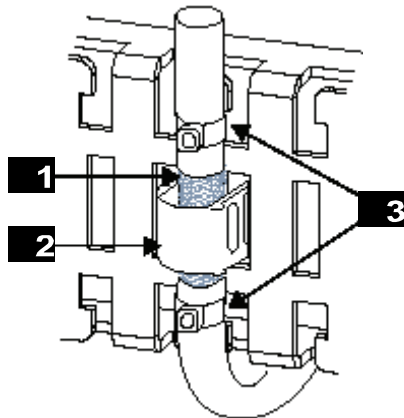
Failure to comply with these instructions can damage the electronic components.

- Prior to working with HIMA components, touch an earthed object.
- Make sure that the workspace is free of static and wear an ESD wrist strap.
- If not used, ensure that the module is protected from electrostatic discharge, e.g., by storing it in its packaging.

To attach the cables and connected the shielding

1. Led the cables vertically downwards and secure them with 2 cable straps to the earth grid guide.
2. Use a clamp to connect the shielding of a cable (if present) to the earth grid. To this end, place the clamp over the surface of the stripped cable shielding and press it from both sides into the oblong holes of the earth grid until it snaps into position.

The cables are attached and the shielding is connected.



- 1** Cable Shielding
- 2** Shield Clamp
- 3** Cable Straps

Figure 8: Securing the Cables and Connecting the Shielding

NOTE



Cable failure is possible under tensile load!
Do not use the shield clamp as a strain relief for the connected cable!

7.1.5 Earthing

The earthing screw is located on the front, left side of the earth grid and is labeled with the earthing symbol.

- i**
 - For improved EMC, earth the housing.
 - The connection to the next grounding point must be as short as possible to achieve a low earthing resistance.

The F60 controller can be operated with earthed ground L- or unearthed. With unearthed operation, earth fault monitoring must be used (see also VDE 0116).

7.1.6 Connecting the Operating Voltage

The contact is electrically connected using a detachable three-pole connector located on the front plate of the power supply units. The connector can accept wires of up to 6 mm².

Connection	Function
L+ DC 24 V	Power supply L+ (24 VDC)
L- DC 24 V	Power supply L- (24 VDC ground)
	Earth / Shielding

Table 22: Connectors for Operating Voltage

if shielded wires are used for the power supply, the shielding must also be connected to the power supply connector via the earth contact.

The power supply unit must meet the IEC/EN 61131-2 requirements and comply with the low voltage directives SELV (Safety Extra Low Voltage) or PELV (Protective Extra Low Voltage).

7.1.7 Using the Reset Key with the CPU 01 Module

The module is equipped with a reset key. The key is only required if the user name or password for administrator account is not known. If only the IP address set for the controller does not match the PADT (PC), the connection can be established with a routing entry on the PC (`Route add` command).

The key can be accessed through a small round hole located on the front plate. The key is engaged using a suitable pin made of insulating material to avoid short-circuits.

The reset is only effective if the controller is rebooted (switched off and on) while the key is simultaneously engaged for at least 20 seconds. Engaging the key during operation has no effect.

NOTE



Fieldbus communication may be disturbed!

Prior to switching on the controller with the reset key engaged, unplug the controller's fieldbus connectors to ensure that the fieldbus communication among other stations is not disturbed.

The fieldbus plugs may only be plugged in again when the controller is in the RUN or STOP state.

Properties and behavior of the controller after a reboot *with* engaged reset key:

- Connection parameters (IP address and system ID) are set to the default values.
- Only the Administrator default user account with an empty password exists.
- With COM operating system version 10.42 and beyond, loading a user program or operating system with default connection parameters is inhibited!

The loading procedure is only allowed once the connection parameters and the account have been configured on the controller and the controller has been rebooted.

After a new reboot without the reset key engaged, the following applies:

- The user-defined connection parameters (IP address and system ID) and user accounts are effective.
- If no changes were performed, the connection parameters and accounts that were valid prior to the reboot with the reset key engaged, apply.

7.2 Configuring a Resource with SILworX - CPU-OS Version 7 and Newer

This chapter describes how to configure resources using SILworX for operating system version 7 **and newer**.

7.2.1 Configuring the Resource

The resource properties and the hardware output variables are changed at this level.

7.2.1.1 Resource Properties

These parameters define how the controller behaves during operation and are configured for the resource in the *Properties* dialog box in SILworX.

Parameter / Switch	Description	Default value	Setting for safe operation
Name	Name of the resource		Any
System ID [SRS]	System ID of the resource 1...65 535 The system ID must have a value different from the default value, otherwise the project cannot be executed!	60 000	Unique value within the controller network.
Safety time [ms]	Safety time in milliseconds 20...22 500 ms	600 ms	Application-specific
Watchdog Time [ms]	Watchdog time in milliseconds 8...5 000 ms	200 ms/ 100 ms1)	Application-specific
Main Enable	<p>ON: The following switches/parameters can be changed during operation (= RUN) using the PADT.</p> <ul style="list-style-type: none"> ▪ <i>System ID</i> ▪ <i>Resource Watchdog Time</i> ▪ <i>Safety Time</i> ▪ <i>Target Cycle Time</i> ▪ <i>Target Cycle Time Mode</i> ▪ <i>Autostart</i> ▪ <i>Global Forcing Allowed</i> ▪ <i>Global Force Timeout Reaction</i> ▪ <i>Load Allowed</i> ▪ <i>Reload Allowed</i> ▪ <i>Start Allowed</i> <p>OFF: The parameters cannot be changed during operation.</p> <p>1 <i>Main Enable</i> can only be set to ON if the PES is stopped.</p>	ON	OFF is recommended
Autostart	<p>ON: If the processor system is connected to the supply voltage, the user program starts automatically</p> <p>OFF: The user program does not start automatically after connecting the supply voltage.</p>	OFF	Application-specific
Start Allowed	<p>ON: A cold start or warm start using the PADT is permitted in the states RUN or STOP</p> <p>OFF: Start not allowed</p>	ON	Application-specific
Load Allowed	<p>ON: Download of the user program permitted</p> <p>OFF: Download of the user program not permitted</p>	ON	Application-specific
Reload Allowed	<p>Only applicable with L3!</p> <p>ON: Reload of the user program permitted</p> <p>OFF: A user program reload is not allowed A running reload process is not aborted when switching to OFF</p>	ON	-
Global Forcing Allowed	<p>ON: Global forcing permitted for this resource</p> <p>OFF: Global forcing not permitted for this resource</p>	ON	Application-specific
Global Force Timeout Reaction	<p>Specifies how the resource should behave when the global force time-out has expired:</p> <ul style="list-style-type: none"> ▪ Stop Forcing ▪ Stopping the Resource 	Stop Forcing	Application-specific

Parameter / Switch	Description	Default value	Setting for safe operation
Max.Com. Time Slice ASYNC [ms]	Highest value in ms for the time slice used for communication during a resource cycle, see the Communication Manual (HI 801 101 E), 2...5000 ms	10 ms	Application-specific
Max. Duration of Configuration Connections [ms]	Only applicable with L3! It defines how much time within a CPU cycle is available for process data communication. 6...5 000	6 ms	
Target Cycle Time [ms]	Targeted or maximum cycle time, see <i>Target Cycle Time Mode</i> , 0...7500 ms. The maximum target cycle time value may not exceed the defined watchdog time (6 ms); otherwise it is rejected by the PES.	0 ms	-
Multitasking Mode	Only applicable with L3! Mode 1 The duration of a CPU cycle is based on the required execution time of all user programs. Mode 2 The processor provides user programs with a higher priority the execution time not needed by user programs with a lower priority. Operation mode for high availability. Mode 3 The processor waits for the unneeded execution time of user programs to expire and thus increases the cycle.	Mode 1	-
Target Cycle Time Mode	Use of <i>Target Cycle Time [ms]</i> . For L3, all values can be used, for L2, only <i>Fixed</i> can be used! Fixed The PES maintains the target cycle time and extends the cycle if necessary. This does not apply if the processing time of the user programs exceeds the target cycle time. Fixed-tolerant Similar to <i>Fixed</i> , but the target cycle time is not taken into account during the first reload activation cycle, with L3. Dynamic-tolerant Similar to <i>Dynamic</i> , but the target cycle time is not taken into account during the first reload activation cycle, with L3. Dynamic HIMax maintains the target cycle time as well as possible, but it also executes the cycle as quickly as possible.	Fixed	-
Minimum Configuration Version	The configuration files and code generation are structured as in the specified SILworX version (except for newest functions). SILworX V2 The code is generated as in SILworX version 4. This setting does not cause the CRC of a project created with SILworX V2 to change. SILworX V3 The code is generated as in SILworX version 3. This setting ensures the compatibility with future versions. SILworX V4 The code is generated as in SILworX version 4. This setting ensures the compatibility with future versions.	SILworX V4	-
Maximum System Bus Latency [µs]	Not applicable for HIMatrix controllers!	0 ms	Application-specific

Parameter / Switch	Description	Default value	Setting for safe operation
safeethernet CRC	SILworX V.2.36.0	Current Version	Application-specific
	Current Version		
¹⁾ 200 ms with controllers, 100 ms with remote I/Os.			

Table 23: System Parameters of the Resource - CPU-OS Version 7 and Newer

7.2.1.2 Hardware System Variables

These variables are used to change the behavior of the controller while it is operating in specific states. These variables can be set in the hardware Detail View located in the SILworX Hardware Editor.

Parameter / Switch	Function	Default setting	Setting for safe operation
Force Deactivation	Used to prevent forcing and to stop it immediately	FALSE	Application-specific
Spare 0 ... Spare 16	No function	-	-
Emergency Stop 1 ... Emergency Stop 4	Emergency stop switch to shutdown the controller if faults are detected by the user program	FALSE	Application-specific
Read-only in RUN	After starting the controller, no operating action such as stop, start or download is permitted in SILworX , except for forcing and reload.	FALSE	Application-specific
Relay Contact 1 ... Relay Contact 4	Only applicable with L3! Activates the corresponding relay contacts, if any.	FALSE	Application-specific
Reload Deactivation	Only applicable with L3! Prevents execution of reload.	FALSE	Application-specific
User LED 1 ... User LED 2	Only applicable with L3! Activates the corresponding LEDs, if any.	FALSE	Application-specific

Table 24: Hardware System Variables - CPU-OS V.7 and Newer

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

7.2.1.3 Hardware System Variables for Reading the Parameters

These system variables can be accessed in the SILworX Hardware Editor.

Select the gray background outside the (yellow) subrack representation and double-click or use the context menu to open the detail view.

Variable	Description	Data type
Number of IO Errors	Number of current I/O errors	UDINT
IO Error Historic Count	Counted number of I/O errors (counter resettable)	UDINT
IO Warning Count	Number of current I/O warnings	UDINT
IO Warning Historic Count	Counted number of I/O warnings (counter resettable)	UDINT
Communication Error Count	Number of current communication errors	UDINT
Communication Error Historic Count	Counted number of communication errors (counter resettable)	UDINT
Communication Warning Count	Number of current communication warnings	UDINT
Communication Warnings Historic Count	Counted number of communication warnings (counter resettable)	UDINT
System Error Count	Number of current system errors	UDINT
System Error Historic Count	Counted number of system errors (counter resettable)	UDINT
System Warning Count	Number of current system warnings	UDINT
System Warning Historic Count	Counted number of system warnings (counter resettable)	UDINT
Autostart CPU Release	ON: When the processor system is connected to the supply voltage, it automatically starts the user program OFF: When the processor system is connected to the supply voltage, it enters the STOP state	BOOL
OS Major	Operating system version contained in processor system	UINT
OS Minor		UINT
CRC	Project configuration checksum	UDINT
Date/time [ms part]	System date and time in s and ms since 1970-01-01	UDINT
Date/time [sec. part]		UDINT
Force Deactivation	ON: Forcing is deactivated. OFF: Forcing is possible.	BOOL
Forcing Active	ON: Global or local forcing is active. OFF: Global and local forcing are not active.	BOOL
Force Switch State	State of the force switches 0xFFFFFFFF No force switches are set 0xFFFFFFFF At least one force switch is set	UDINT
Global Forcing Started	ON: Global forcing is active. OFF: Global forcing is not active.	BOOL
Spare 0 ... Spare 16	Reserved	USINT
Spare in17		BOOL
Last IO Warning [ms]	Date and time of the last I/O warning in s and ms since 1970-01-01	UDINT
Last IO Warning [s]		UDINT

Variable	Description	Data type
Last Communication Warning [ms]	Date and time of the last communication warning in s and ms since 1970-01-01	UDINT
Last Communication Warning [s]		UDINT
Last System Warning [ms]	Date and time of the last system warning in s and ms since 1970-01-01	UDINT
Last System Warning [s]		UDINT
Last IO Error [ms]	Date and time of the last I/O error in s and ms since 1970-01-01	UDINT
Last IO Error [s]		UDINT
Last Communication Error [ms]	Date and time of the last communication error in s and ms since 1970-01-01	UDINT
Last Communication Error [s]		UDINT
Last System Error [ms]	Date and time of the last system error in s and ms since 1970-01-01	UDINT
Last System Fault [s]		UDINT
Fan State	0x00 Fan functioning 0x01 Fan defective 0xFF Not available	BYTE
Major CPU Release	Main enable switch of the processor system ON: The subordinate enable switches can be changed. OFF: The subordinate enable switches cannot be changed.	BOOL
Read-only in RUN	ON: The operator actions: Stop, Start, Download are locked. OFF: The operator actions: Stop, Start, Download are not locked.	BOOL
CPU Safety Time [ms]	Safety time set for the controller in ms	UDINT
Start CPU Release	ON: Start of processor system using PADT is allowed OFF: Start of processor system using PADT is not allowed	BOOL
Start Cycle	TRUE during the first cycle after starting, otherwise FALSE	BOOL

Variable	Description	Data type	
Power Supply State	Bit-coded state of the power supply units. Compact controllers and remote I/Os:	BYTE	
	Value		State
	0x00		Normal
	0x01		Undervoltage with 24 V supply voltage
	0x02		(Battery undervoltage) <i>not used</i>
	0x04		Undervoltage of internally generated 5 V
	0x08		Undervoltage of internally generated 3.3 V
	0x10		Overvoltage with internally generated 3.3 V
	F60 modular controller:		
	Value		State
	0x00		Normal
	0x01		Error with 24 V supply voltage
	0x02		Battery failure
	0x04		Error with 5 V of power supply
	0x08		Error with 3.3 V of power supply
0x10	Undervoltage at 5 V		
0x20	Overvoltage at 5 V		
0x40	Undervoltage at 3.3 V		
0x80	Overvoltage at 3.3 V		
System ID	System ID of the controller, 1...65535	UINT	
Systemtick HIGH	Circular millisecond counter (64 bit)	UDINT	
Systemtick LOW		UDINT	
Temperature State	Bit-coded temperature state of the processor system	BYTE	
	Value		State
	0x00		Normal temperature
	0x01		Temperature threshold 1 exceeded
	0x03		Temperature threshold 2 exceeded
0xFF	Not available		
Remaining Global Force Duration [ms]	Time in ms until the time limit set for global forcing expires.	DINT	
CPU Watchdog Time [ms]	Maximum permissible duration of a RUN cycle in ms.	UDINT	
Cycle Time, last [ms]	Current cycle time in ms	UDINT	
Cycle Time, max [ms]	Maximum cycle time in ms	UDINT	
Cycle Time, min [ms]	Minimum cycle time in ms	UDINT	
Cycle Time, average [ms]	Average cycle time in ms	UDINT	

Table 25: Hardware System Variables for Reading the Parameters

7.2.2 Configuring the Ethernet Interfaces

Ethernet interfaces are configured in the detail view of the communication system. If the remote I/Os have no communication system, the Ethernet interfaces are configured in the detail view of the processor system. Refer to the manuals of the HIMatrix controllers and remote I/Os for more details.

7.2.3 Configuring the User Program

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

Switch / Parameter	Function	Default value	Setting for safe operation	
Name	Name of the user program		Arbitrary	
Safety Integrity Level	Safety integrity level: SIL0, SIL3 (for purposes of documentation only)	SIL3	Application-specific	
Start Allowed	ON: The PADT may be used to start the user program. OFF: The PADT may not be used to start the user program	ON	Application-specific	
Program Main Enable	It enables changes of other user program switches: Only the enable switch of the resource is relevant!	ON	-	
Autostart	Enabled type of Autostart: Cold Start, Warm Start, Off	Cold start	Application-specific	
Test Mode Allowed	ON The user program is allowed to operate in test mode. OFF The user program is not allowed to operate in test mode.	OFF	Application-specific	
Local Forcing Allowed	ON: Forcing Allowed at Program Level OFF: Forcing not Allowed at Program Level	OFF	OFF is recommended	
Reload Allowed	ON: User program reload is permitted OFF: User program reload is not permitted	ON	Application-specific	
Program's Maximum Number of CPU Cycles	Maximum number of CPU cycles that a user program cycle may encompass. A value greater than 1 is only allowed for HIMatrix controllers L3!	1	Application-specific	
Max. Duration for Each Cycle [μ s]	Maximum time in each processor module cycle for executing the user program: 1...7 500 000 μ s, no limitations A value other than 0 μ s is only allowed for HIMatrix controllers L3!	0 μ s	0 μ s	
Local Force Timeout Reaction	Behavior of the user program after the forcing time has expired: <ul style="list-style-type: none"> ▪ Stop Forcing Only. ▪ Stop Program. 	Stop Forcing Only.	-	
Program ID	ID for identifying the program as displayed within SILworX, 1...32	1	Application-specific	
Watchdog Time [ms] (calculated)	Monitoring time of the user program, calculated from the <i>Program's Maximum Number of Cycles</i> and the watchdog time of the resource Not changeable! <ul style="list-style-type: none"> • 1 For HIMatrix L3 systems with counter inputs, ensure that the user program's watchdog time is less than or equal to 5 000 ms. 		-	
Code Generation Compatibility	SILworX V4	Code generation is compatible with SILworX version 4.	SILworX V4	SILworX V2 with V.7 SILworX V3, SILWorX V4 with V.8 and beyond (L3)
	SILworX V3	Code generation is compatible with SILworX version 3.		
	SILworX V2	Code generation is compatible with SILworX version 2.		

Table 26: System Parameters of the User Program - CPU-OS Version 7 and Newer

7.2.4 Configuring the Inputs and Outputs

In the Hardware Editor, the inputs and outputs are configured by allocating global variables to the system variables for input and output channels.

To access the system variables for input and output channels

1. Display the desired resource in the Hardware Editor.
2. Double-click the required input or output module to open the corresponding detail view.
3. In the detail view, open the tab with the required channels.

The system variables for the channels appear.

Use of Digital Inputs

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

1. Define a global variable of type BOOL.
2. Enter an appropriate initial value, when defining the global variable.
3. Assign the global variable to the channel value of the input.
4. In the user program, program a safety-related fault reaction using the error code -> *Error Code [Byte]*.

The global variable provides values to the user program.

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

To get additional options for programming fault reactions in the user program, assign global variable to *DI.Error Code* and *Module Error Code*. For more information on the error codes, refer to the manual of the corresponding compact system or module.

Use of Analog Inputs

Analog input channels convert the measured input currents into a value of type INT (double integer). This value is then made available to the user program. With analog inputs of type FS1000, the range of values is 0...1 000, with analog inputs of type FS2000, the range of values is 0...2 000.

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

Perform the following steps to use the value of an analog input in the user program

1. Define a global variable of type INT.
2. Enter an appropriate initial value, when defining the global variable.
3. Assign the global variable to the channel value -> *Value [INT]* of the input.
4. In the user program, define a global variable of the type needed.
5. In the user program, program a suitable conversion function to convert the raw value into a used type and consider the measurement range.
6. In the user program, program a safety-related fault reaction using the error code -> *Error Code [Byte]*.

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

If the value 0 for a channel is **within the valid measuring range**, the user program must, at a minimum, evaluate the parameter *Error Code [Byte]* in addition to the process value.

To get additional options for programming fault reactions in the user program, assign global variable to *AI.Error Code* and *Module Error Code*. For more information on the error codes, refer to the manual of the corresponding compact system or module.

Use of Safety-Related Counter Inputs

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

In the following sections, xx refers to the corresponding channel number.

Perform the following steps to use the integer value

1. Define a global variable of type UDINT.
2. Enter an appropriate initial value, when defining the global variable.
3. Assign the global variable to the integer value *Counter[xx].Value* of the input.
4. In the user program, program a safety-related fault reaction using the error code *Counter[xx].Error Code*.

The global variable provides values to the user program.

To get additional options for programming fault reactions in the user program, assign global variable to *Counter.Error Code* and *Module Error Code*. Refer to the manual of the compact system or module for more details on how to use the error codes and other parameters of the counter input.

Use of Digital Outputs**Perform the following steps to write a value in the user program to a digital output:**

1. Define a global variable of type BOOL containing the value to be output.
2. Enter an appropriate initial value, when defining the global variable.
3. Assign the global variable the *Value [BOOL]* -> channel value of the output.
4. In the user program, program a safety-related fault reaction using the error code -> *Error Code [Byte]*.

The global variable provides values to the digital output.

To get additional options for programming fault reactions in the user program, assign global variable to *DO.Error Code* and *Module Error Code*. Refer to the manual of the compact system or module for more details.

Use of Analog Outputs**Perform the following steps to write a value in the user program to an analog output**

1. Define a global variable of type INT containing the value to be output
2. Enter an appropriate initial value, when defining the global variable.
3. Assign the global variable the *Value [INT]* -> channel value of the output:
5. In the user program, program a safety-related fault reaction using the error code -> *Error Code [Byte]*.

The global variable provides values to the analog output.

To get additional options for programming fault reactions in the user program, assign global variable to *AO.Error Code* and *Module Error Code*. Refer to the manual of the compact system or module for more details.

7.2.5 Generating the Resource Configuration

To generate the code for the resource configuration

1. Select the resource in the structure tree.
2. Click the **Code Generation** button located on the Action Bar or select **Code Generation** from the context menu.
 - The *Start Code Generation* dialog box appears.
3. In the Start Code Generation dialog box, click OK.
 - An additional *Start Code Generation* opens, shows the code generation progress and disappears. The logbook contains a row informing about the code generation result.
4. With the resource still selected, select the **Version Comparison:** function on the **Extras** menu.
 - The *Version Overview* dialog box opens. It displays the CRC for the generated code.
5. Click **Export**.

- The *Archive* dialog box is displayed in which the user can add a comment to the project status and give a name to the archive file.
 - 6. Generate the code once again, follow the instruction for steps 2 and 3.
 - 7. With the resource still selected, select the **Version Comparison**: function on the **Extras** menu.
 - The *Version Overview* dialog box opens.
 - 8. Click **Import** and in the *Restore* dialog box, import the archive file exported in step 5.
 - The *Version Overview* dialog box specifies details on the project status generated last and on the project status that should be imported.
 - 9. Click **OK**.
 - The result of the version comparison is displayed in the workspace. If OK appears in the CRC Comparison column, the codes generated for the two project statuses are identical and suitable for safety-related operation. Divergences are marked in red.
- The code for the resource configuration is generated.

NOTE



Failures during the code generation may occur due to the non-safe PC!

For safety-related applications, the code generator must generate the code two times and the checksums (CRCs) resulting from the two code generations must be identical. Only if this is the case, an error-free code is ensured.

Refer to the Safety Manual (HI 800 023 E) for further details.

7.2.6 Configuring the System ID and the Connection Parameters

Configuring the System ID and the Connection Parameters

1. Select the resource in the structure tree.
2. Click the **Online** button located on the Action Bar or select **Online** from the context menu.
 - The *System Login* dialog box is displayed.
3. Click **Search**.
 - The *Search per MAC* dialog box appears.
4. Enter the MAC address valid for the controller - see the label on the housing - and click **Search**.
 - In the dialog box, the values set for IP address, subnet mask and S:R:S are displayed.
5. If the values for the project are not correct, click Change.
 - The *Write via MAC* dialog box appears.
6. Type correct values for the connection parameters and the S.R.S., and enter the access data for a user account with administrator rights valid on the controller. Click **Write**.

Connection data and S.R.S are configured and it is now possible to log in.

For further details, refer to the SILworX manual First Steps (HI 801 103 E).

7.2.7 Loading a Resource Configuration after a Reset

If the compact system is switched on with engaged reset key, it restarts and resets the connections parameters and user account to the default values (only in case of a controller). After a new restart with disengaged reset key, the original values are used.

If the connection parameters were modified in the user program, they can be configured in the compact systems such as described in Chapter 7.2.6.

Logging in as Default User

After configuring the connection parameters and prior to loading the user program, the default user (administrator with empty password) must be used in the following cases:

- The password for the user account is no longer known.
- A new user account should be used in the project.

To log in as default user

1. Select the resource in the structure tree.
2. Click the **Online** button located on the Action Bar or select **Online** from the context menu.
 - The *System Login* dialog box opens.
3. In the *IP Address* field, select the correct address or use the MAC address.
4. Enter *Administrator* in the *User Group* field.
5. Let the *Password* field empty or delete the password.
6. Select **Administrator** in the *Access Mode* field.
7. Click **Log-in**.

SILworX is disconnected to the HIMatrix controller with default user rights.

Use <Ctrl>+A in the *System Login* dialog box to skip steps 4-6!

7.2.8 Loading a Resource Configuration from the PADT

Before a user program can be loaded with the connection parameters (IP address, subnet mask and system ID) into the controller, the code must have been generated for the resource, and the connection parameters for PADT and resource must be valid, see Chapter 7.2.6.

To load a resource configuration from the PADT

1. Select a resource in the structure tree.
2. Click the **Online** button located on the Action Bar or select **Online** from the context menu.
3. In the *System Login* dialog box, enter a user group with administrator rights or write access.
 - The Control Panel appears in the workspace and displays the controller state.
4. In the **Online** menu, select **Resource Download**.
 - The *Resource Download* dialog box appears.
5. Click **OK** to confirm the download.
 - SILworX loads the configuration into the controller.
6. Upon completion of the loading procedure, click the Resource Cold Start function on the Online menu to start the user program.
 - After the cold start, *System State* and *Program Status* enter the RUN state.

The resource configuration is loaded from the PADT.

The Start, Stop and Load functions are also available as symbols on the Symbol Bar.

7.2.9 Loading a Resource Configuration from the Flash Memory of the Communication System

If data errors were detected in the NVRAM thus causing the watchdog time to be exceeded, it can be useful to load the resource configuration from the flash memory for the communication system instead of from the PADT:

If the Control Panel (CP) is no longer accessible, the connection parameters for the project must be reset in the controller, see Chapter 7.2.6.

If the controller adopts the STOP/VALID CONFIGURATION state after restarting, the user program can again be started.

If the controller adopts the STOP/INVALID CONFIGURATION state after restarting, the user program must be reloaded into the NVRAM.

Use the **Load Configuration from Flash** command to read a backup copy of the last executable configuration from the flash memory for the communication system transfer it to the processor's NVRAM. At this point, select **Online -> Start (Cold Start)** to restart the user program with no need for performing a Download of the project.

Loading a Resource Configuration from the Flash Memory of the Communication System

1. Log in to the required resource.
2. In the **Online** menu, select **Maintenance/Service -> Load Configuration from Flash**.
3. A dialog box appears. Confirm the action.

The controller loads the resource configuration from the flash memory for the communication system into the NVRAM.

7.2.10 To clean-up a resource configuration in the flash memory of the communication system

After temporary hardware faults, the flash memory for the communication system could contain residual invalid configuration parts.

The **Clean Up Configuration** command is used to delete these residual parts,

To clean up the resource configuration

1. Select a resource in the structure tree.
 2. Click the **Online** button located on the Action Bar or select **Online** from the context menu.
 3. In the *System Login* dialog box, enter a user group with administrator rights or write access.
 - The Control Panel appears in the workspace and displays the controller state.
 4. In the **Online** menu, select **Maintenance/Service -> Clean Up Configuration**.
 5. The *Clean Up Configuration* dialog box appears. Click **OK** to confirm the action.
- The configuration within the flash memory of the communication system has been cleaned up.

The clean-up function is not frequently needed.

A valid configuration is not affected by the clean-up process.

7.2.11 Setting the Date and the Time

To set the controller's time and date

1. Select a resource in the structure tree.
2. Click the **Online** button located on the Action Bar or select **Online** from the context menu.
3. In the *System Login* dialog box, enter a user group with administrator rights or write access.
 - The Control Panel appears in the workspace and displays the controller state.
4. In the **Online** menu, select **Start-Up -> Set Date/Time**.
 - The *Set Date/Time* dialog box appears.

5. Select one of the following options:
 - **Use the PADT date and time** - to transfer the time and date displayed for the PADT into the controller.
 - **User-defined** - to transfer the date and time from the two input boxes into the controller. Make sure that the format used for date and time is correct!
 6. Click **OK** to confirm the action.
- The time and the date are set for the controller.

7.3 User Management in SILworX - CPU-OS Version 7 and Newer

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

7.3.1 User Management for SILworX Projects

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

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

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

Table 27: Authorization Types for the PADT User Management Scheme

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

Characteristics of user groups:

- The name must be unique within the project and must contain 1...31 characters.
- A user group is assigned an authorization type.
- A user group may be assigned an arbitrary number of user accounts.
- A project may contain up to 100 user groups.

User account properties

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

7.3.2 User Management for the Controller

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

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

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

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

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

Default User

The factory user settings apply as long as no user accounts have been created for a resource. The factory settings also apply after starting a controller with the reset pushbutton activated.

Factory settings

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

i

If user accounts are defined, the default settings cannot be maintained.

7.3.3 Parameters for User Accounts

Define the following parameters to create user accounts:

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

Table 28: Parameters for User Accounts in the PES User Management Scheme

7.3.4 Setting Up User Accounts

A user with administrator rights can access all user accounts.

Observe the following points when setting up user accounts:

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

7.4 Configuring Communication with SILworX - CPU-OS Version 7 and Newer

This chapter describes how to configure communication using SILworX for processor operating system version 7 **and newer**.

Depending on the application, the following elements must be configured:

- Ethernet/safeethernet.
- Standard protocols

Refer to the Communication Manual (HI 801 101 E) for more information on how to configure the standard protocols.

7.4.1 Configuring the Ethernet Interfaces

Ethernet interfaces are configured in the Detail View of the communication module. If the remote I/Os have no communication module, the Ethernet interfaces are configured in the Detail View of the processor module. Refer to the manuals of the HIMatrix systems for more details.

i

SILworX represents the processor system and the communication system within a device or module as processor module and communication module.

For HIMatrix systems, set the *Speed Mode [Mbit/s]* and *Flow Control Mode* to **Autoneg** in the Ethernet switch settings.

The parameters *ARP Aging Time*, *MAC Learning*, *IP Forwarding*, *Speed [Mbit/s]* and *Flow Control* are explained in details in the SILworX online help.

i

Replacement of one controller with identical IP address:

If a controller has its *ARP Aging Time* set to 5 minutes and its *MAC Learning* set to **Conservative**, its communication partner does not adopt the new MAC address until a period of 5 to 10 minutes after the controller is replaced. During this time period, no communication is possible with the replaced controller.

The port settings of the integrated Ethernet switch on a HIMatrix resource can be configured individually. In the **Ethernet Switch** tab, an entry can be created for each switch port.

Name	Explanation
Port	Port number as printed on the housing; per port, only one configuration may exist. Range of values: 1...n, depending on the resource
Speed [Mbit/s]	10 Mbit/s: Data rate 10 Mbit/s 100 Mbit/s: Data rate 100 Mbit/s Autoneg (10/100): Automatic baud rate setting Standard: Autoneg
Flow Control	Full duplex: Simultaneous communication in both directions Half duplex: Communication in both directions, but only one direction at a time Autoneg: Automatic communication control Standard: Autoneg
Autoneg also with Fixed Values	The "Advertising" function (forwarding the Speed and Flow control properties) is also performed if <i>Speed</i> and <i>Flow Control</i> have fixed values. This allows other devices with ports set to Autoneg to recognize the HIMatrix ports' settings.
Limit	Limit the inbound multicast and/or broadcast packets. Off: No limitation Broadcast: Limit broadcast packets (128 kbit/s) Multicast and Broadcast: Limit multicast and broadcast packets (1024 kbit/s) Default: Broadcast

Table 29: Parameters of the Port Configuration - CPU-OS Version 7 and Newer

To modify and enter these parameters in the communication system's configuration, double-click each table cell. The parameters become operative for HIMatrix communication, once they have been re-compiled with the user program and transferred to the controller.

The properties of the communication system and Ethernet switch can also be changed online using the Control Panel. These settings become operative immediately, but they are not adopted by the user program.

Refer to the Communication Manual (HI 801 101 E) for more details about configuring **safeethernet**.

7.5 Configuring the Sequence of Events Recording - with L3

Defining Events

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

The events are defined.

For further information, refer to the SILworX online help.

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

Column	Description	Range of values
Name	Name for the event definition; it must be unique within the resource.	Text, max. 32 characters.
Global Variable	Name of the assigned global variable (added using a drag&drop operation)	
Data type	Data type of the global variable; it cannot be modified.	BOOL
Event source	CPU event The processor module creates the timestamp. It creates all the events in each of its cycle. Auto event Same as CPU event. Default value: Auto	CPU, Auto
Alarm when FALSE	Activated If the global variable value changes from TRUE to FALSE, an event is triggered. Deactivated If the global variable value changes from FALSE to TRUE, an event is triggered. Default value: Deactivated	Checkbox activated, deactivated
Alarm Text	Text specifying the alarm state	Text
Alarm priority	Priority of the alarm state Default value: 500	0...1000
Alarm Acknowledgment Successful	Activated The alarm state must be confirmed by the user (acknowledgement) Deactivated The alarm state may not be confirmed by the user Default value: Deactivated	Checkbox activated, deactivated
Return to Normal Text	Text specifying the alarm state	Text
Return to Normal Severity	Priority of the normal state	0...1000
Return to Normal Ack Required	The normal state must be confirmed by the user (acknowledgement) Default value: Deactivated	Checkbox activated, deactivated

Table 30: Parameters for Boolean Events

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

Column	Description	Range of values
Name	Name for the event definition; it must be unique within the resource.	Text, max. 32 characters.
Global Variable	Name of the assigned global variable (added using a drag&drop operation)	
Data type	Data type of the global variable; it cannot be modified.	depending on the global variable type
Event source	CPU event The processor module creates the timestamp. It creates all the events in each of its cycle. Auto event Same as CPU event. Default value: Auto event	CPU, Auto
HH Alarm Text	Text specifying the alarm state of the upper limit.	Text
HH Alarm Value	Upper limit triggering an event. Condition: (HH Alarm Value - Hysteresis) > H Alarm Value or HH Alarm Value = H Alarm Value	depending on the global variable type
HH Alarm Priority	Priority of the upper limit; default value: 500	0...1000
HH Alarm Acknowledgment Required	Activated The user must confirm that the upper limit has been exceeded (acknowledgment). Deactivated The user may not confirm that the upper limit has been exceeded. Default value: Deactivated	Checkbox activated, deactivated
H Alarm Text	Text specifying the alarm state of the upper limit.	Text
H Alarm Value	Upper limit triggering an event. Condition: (H Alarm Value - Hysteresis) > (L Alarm Value + Hysteresis) or H Alarm Value = L Alarm Value	depending on the global variable type
H Alarm Priority	Priority of the upper limit; default value: 500	0...1000
H Alarm Acknowledgment Required	Activated The user must confirm that the upper limit has been exceeded (acknowledgment). Deactivated The user may not confirm that the upper limit has been exceeded. Default value: Deactivated	Checkbox activated, deactivated
Return to Normal Text	Text specifying the normal state	Text
Return to Normal Severity	Priority of the normal state; default value: 500	0...1000
Return to Normal Ack Required	The normal state must be confirmed by the user (acknowledgement); default value: Deactivated	Checkbox activated, deactivated
L Alarm Text	Text specifying the alarm state of the lower limit.	Text
L Alarm Value	Lower limit triggering an event. Condition: (L Alarm Value + Hysteresis) < (H Alarm Value - Hysteresis) or L Alarm Value = H Alarm Value	depending on the global variable type
L Alarm Priority	Priority of the lower limit; default value: 500	0...1000
L Alarm Acknowledgment Required	Activated The user must confirm that the lower limit has been exceeded (acknowledgment). Deactivated The user may not confirm that the lower limit has been exceeded. Default value: Deactivated	Checkbox activated, deactivated
LL Alarm Text	Text specifying the alarm state of the lowest limit.	Text
LL Alarm Value	Lower limit triggering an event. Condition: (LL Alarm Value + Hysteresis) < (L Alarm Value) or LL Alarm Value = L Alarm Value	depending on the global variable type
LL Alarm Priority	Priority of the lowest limit; default value: 500	0...1000

Column	Description	Range of values
LL Alarm Acknowledgment Required	<p>Activated The user must confirm that the lowest limit has been exceeded (acknowledgment).</p> <p>Deactivated The user may not confirm that the lowest limit has been exceeded.</p> <p>Default value: Deactivated</p>	Checkbox activated, deactivated
Alarm Hysteresis	The hysteresis avoids that many events are continuously created when the process value often oscillate around a limit.	depending on the global variable type

Table 31: Parameters for Scalar Events

NOTE

Faulty events recording due invalid parameter settings possible!

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

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

7.6 Configuring a Resource Using ELOP II Factory - CPU-OS Versions Prior to 7

This chapter describes how to configure a resource using ELOP II Factory for processor operating system versions **prior to 7**.

7.6.1 Configuring the Resource

The first step is to configure the resource. The parameter and switch settings associated with the configuration are stored to the NVRAM of the processor system and to the flash memory of the communication system.

The following system parameters can be set for a resource:

Parameter / Switch	Range	Description	Default Value
System ID [SRS]	1...65 535	System ID within the network	0 (invalid)
Safety Time [ms]	20...50 000 ms	Safety time of the controller (not of the entire process)	2. Watchdog Time
Watchdog Time [ms]	≥ 10 ms $\leq (\text{Safety Time}) / 2$ ≤ 5000 ms	Maximum time allowed for a PES RUN cycle. If the cycle time has been exceeded, the controller enters the STOP state.	Controller: 50 ms Remote I/O: 10 ms
Main Enable	On/Off	The main enable switch can only be set to ON if the controller is in the STOP state. It allows the user to modify the settings for the following switches and the parameters <i>Safety Time</i> and <i>Watchdog Time</i> in the RUN state.	On
Autostart	On/Off	Automatic start of the controller after its powering ON (automatic transition from STOP to RUN)	Off
Start/Restart Allowed	On/Off	Start command for the controller On: Start (cold start) or restart (warm start) commands accepted by the controller Off: Start/restart not allowed	On
Load allowed	On/Off	User program load On: Load allowed Off: Load not allowed	On
Test mode allowed	On/Off	Test mode On: Test mode allowed Off: Test mode not allowed	Off
Variables may be changed in the online test	On/Off	Changing variables in the online test On: Allowed Off: Not Allowed	On
Forcing allowed	On/Off	On: Forcing allowed Off: Forcing not allowed	Off
Stop at Force Timeout	On/Off	On: STOP upon expiration of the force time. Off: No STOP upon expiration of the force time.	On
Max. Com. Time Slice [ms]	2...5 000 ms	Time for performing the communication tasks	10 ms

Table 32: Resource Configuration Parameters - CPU-OS Versions Prior to 7

Refer to the Safety Manual for the HIMatrix system (HI 800 023 E) for more details on how to configure a resource for safety-related operation.

7.6.2 Configuring the User Program

General system signals and parameters

Signal	[Data type], Unit, value	R/W	Description
System ID high/low	[USINT]	R	CPU system ID (the first part of the SRS) [not safe] ¹⁾
OS major version OS major high OS major low	[USINT]	R	Major version of CPU operation system (OS) Example: OS version 6.12, major version: 6 OS version 6, valid if system ID ≠ 0 [not safe]
OS minor version OS minor high OS minor low	[USINT]	R	Minor version of CPU operation system (OS) Example: OS version 6.12, minor version: 6 OS version 6, valid if system ID ≠ 0 [not safe]
Configuration signature CRC byte 1 through 4	[USINT]	R	CRC of the configuration loaded, only valid in the states RUN and STOP VALID CONFIGURATION. OS version 6, valid if system ID ≠ 0 [not safe]
Date/Time [sec part] and [ms part]	[USINT] s ms	R	Seconds since 1970, and ms Changing automatically between Winter and Summer time is not supported. [not safe]
Remaining force time	[DINT] Ms	R	Remaining time during Forcing; 0 ms if forcing is not active. [not safe]
Fan State	[BYTE] 0x00 0x01	R	Normal (fan ON) Fan defective [not safe]
Power supply state	[BYTE] 0x00 0x01 0x02 0x04 0x08 0x10 0x20 0x40 0x80	R	Normal Undervoltage 24 V [not safe] Low battery voltage [not safe] Fault power supply 5 V [not safe] Fault power supply 3.3 V [not safe] Undervoltage 5 V [safe] Overvoltage 5 V [safe] Undervoltage at 3.3 V [safe] Overvoltage 3.3 V [safe]
Systemtick HIGH/LOW	[UDINT] ms	R	64-bit ring counter Each UDINT includes 32 bits. [safe]
Temperature State	[BYTE] 0x00 0x01 0x02 0x03	R	Normal High Defective Very high [not safe]
Cycle time	[UDINT] ms	R	Duration of the last cycle [safe]
Emergency stop 1, 2, 3, 4	TRUE, FALSE	W	TRUE: Emergency stop of the system [safe]
¹⁾ System signals with the <i>[not safe]</i> characteristic may only be combined with signals defined as <i>[safe]</i> to trigger a safety shut-down.			

Table 33: General System Signals and Parameters - CPU-OS Versions Prior to 7

The following table specifies the parameters for configuring the user program:

Parameter	Range	Description	Default Value
Execution Time	0 ms	For future applications in which a resource is able to process multiple program instances simultaneously. It determines the maximum cycle time portion that must not be exceeded by the program instance. If this time portion is exceeded, the program enters the STOP state. Note: Maintain the default setting 0 (no special cycle time monitoring).	0 ms
Autostart Enable	Off, Cold Start, Warm Start	The user program starts automatically after powering on	Cold Start
Memory model	SMALL, BIG	Structure of the resource memory required and expected for performing a code generation.	SMALL
		SMALL Compatibility with previous controller versions is ensured.	
		BIG Compatibility with future controller versions.	

Table 34: User program Parameters - CPU-OS Versions Prior to 7

The parameters specified above can be accessed via the ELOP II Factory Hardware Management.

To change the user program parameters

1. Right-click the resource and select **Properties** on the context menu. The Properties dialog box appears.
Enter the values in the input boxes or check the corresponding checkboxes.
2. Define the values for *Autostart* (**Off, Cold Start, Warm Start**) in the **Properties** menu for the type instance of the corresponding resource. With cold start, the system initializes all signal values, with warm start, it reads the signal values of retain variables from the non-volatile memory.

The settings for the user program are thus defined.

7.6.3 Configuring the Inputs and Outputs

The *Signal Connections* pane for an I/O module or a remote I/O in the Hardware Management is used to connect the signals previously defined in the Signal Editor to the individual hardware channels (inputs and outputs).

To configure the inputs or outputs

1. Click the **Signals** menu to open the **Signal Editor**.
2. Right-click the module or the remote I/O and select **Connect Signals** on the context menu.
 The **Signal Connections** pane appears. It contains the Inputs and Outputs tabs.
3. Position the two dialog boxes adjacently to get a better overview.
4. Drag the signals onto the inputs located in the Signal Connections pane.
5. To connect the signals for the outputs, select the **Outputs** tab and proceed as described for the inputs.

The inputs and outputs are now connected and thus effective in the user program.

Refer to the manual for the individual modules or remote I/Os, Chapter *Signals and Error Codes for the Inputs and Outputs* for a description of the signals available for configuring the corresponding module or remote I/O.

With the **Inputs** and **Outputs** tabs of the *Signal Connections* pane, observe the following points:

- The signals for the error codes associated with the hardware channels are always located in the **Input** tab.
- The signals for setting the parameters or configuring the hardware channels are located in the **Outputs** tab, for physical inputs or outputs too.
- The hardware channel value for a physical input is always located in the **Input** tab, the channel value for a physical output in the **Output** tab.

7.6.4 Generating the Code for the Resource Configuration

To generate the code for the resource configuration

1. Move to the ELOP II Factory Project Management and select the HIMatrix resource in the project window.
2. Right-click the HIMatrix resource and select **Code Generation** on the context menu.
3. After a successful code generation, i.e., no red messages or texts in the Status Viewer, note down the created checksum.
4. Move to the ELOP II Factory Hardware Management, right-click the HIMatrix resource and select **Configuration Information** on the context menu.
5. Note down the checksum displayed in the CRC PADT column for *root.config*.
6. Generate once again the code.
7. Compare the checksum of the second code generation with the checksum previously noted down. Only if the checksums are identical, the code may be used for safety-related operation.

The code for the resource configuration is generated.

NOTE



Failures during the code generation may occur due to the non-safe PC!

For safety-related applications, the code generator must generate the code two times and the checksums (CRCs) resulting from the two code generations must be identical. Only if this is the case, an error-free code is ensured.

Refer to the Safety Manual (HI 800 023 E) for further details.

7.6.5 Configuring the System ID and the Connection Parameters

Prior to loading the resource configuration using the Control Panel, the system ID and the connection parameters must be configured in the controller.

To configuring the system ID and the connection parameters

1. Move to the ELOP II Factory Hardware Management.
2. Select and right click the required resource..
 - The context menu for the resource appears.
3. Click **Online -> Connection Parameters**.
 - The overview for the PES connection parameters appears.
4. Enter the MAC address valid for the controller in the MAC Address input box and click **Set via MAC**.

The connection parameters and the system/rack ID configured in the project are set.

For further details, refer to the ELOP II Factory manual First Steps (HI 800 006 E).

7.6.6 Loading a Resource Configuration after a Reset

If the compact system is switched on with engaged reset key, it restarts and resets the connections parameters and, with controllers, the user account to the default values. After a new restart with disengaged reset key, the original values are used.

If the connection parameters were modified in the user program, they can be configured in the controller or remote I/O such as described in Chapter 7.6.5.

For further information on the reset key, refer to the manual of the corresponding controller and to the ELOP II Factory manual 'First Steps' (HI 800 006 E).

Loading a Resource with Communication Operating System Version 10.42 and Beyond

After configuring the connection parameters and prior to loading the user program, the default user (administrator with empty password) must be used in the following cases:

- The password for the user account is no longer known.
- A new user account should be used in the project.

To set the default user

1. Right-click the resource and select **Online -> User Management** on the context menu.
2. Click the **Connect** button to establish the connection.
3. Click the **Default Settings** button.

The user management contained in the controller is deleted and the Administrator default user with empty password is set.

The user program can now be loaded into the controller.

User Management with Communication Operating System Version 6.0 and Beyond

To create new users

1. Right-click the required resource and select **New -> User Management**.
 - A new element, User Management, is added to the structure tree associated with the resource.
 2. Right-click the user management and select **New -> User** to create a new user.
- A new user has been created.


Right-click the user and select Properties on the context menu to configure the new user (user name, password, etc.). Additional users are created accordingly.

Upon completion of the code generation, perform a download of the resource configuration to transfer the new user management to the controller. Afterwards, a user from the new user lists can log-in to the controller.

7.6.7 Loading a Resource Configuration from the PADT

Before a user program can be loaded with the connection parameters (IP address, subnet mask and system ID) into the controller, the machine code must have been generated for the resource, and the connection parameters for PADT and resource must be valid.

To load a resource configuration from the PADT

1. Right click the resource and select **Online -> Control Panel**.
2. Log in to the controller as administrator or at least as user with write access.
3. Load the user program. The controller must be in the STOP state. If required, use the **Resource -> Stop** menu functions.
4. Click the Load  button. A confirmation prompt is displayed.
5. Click **Yes** to confirm the prompt and start the loading process.

6. Upon completion of the loading process, click the Resource Cold Start button to start the user program.

- After a cold start, *CPU State*, *COM State* and *Program State* are set to RUN.

The resource configuration is loaded from the PADT.

The functions Start, Stop and Load can also be performed using the Resource menu.

The controller's mode of operation 'STOP' is divided as follows:

Mode of operation	Meaning with remote I/Os	Meaning with controllers
STOP/LOAD CONFIGURATION	A configuration can be loaded into the remote I/O.	A configuration with user program can be loaded into the controller.
STOP/INVALID CONFIGURATION	The configuration was loaded into the remote I/O properly.	The configuration with user program was loaded into the controller properly. A command from the PADT can set the controller into the RUN state. This causes a loaded user program to start.
STOP/INVALID CONFIGURATION	No configuration available or the loaded configuration is corrupted.	
		In this mode of operation, the controller is not able to enter the RUN state.

Table 35: Sub-States Associated with STOP - CPU-OS Versions Prior to 7

Loading a new configuration with or without user program automatically overwrites all objects previously loaded.

7.6.8 Loading a Resource Configuration from the Flash Memory of the Communication System

In certain cases, it can be useful to load the resource configuration from the flash memory for the communication system instead of from the PADT:

- After replacing the back-up battery - with controllers with layout 0 or 1 only.
- With data errors within the NVRAM and associated watchdog time overrun:

If the Control Panel (CP) is no longer accessible, the connection parameters for the project must be reset in the controller, see Chapter 7.6.5. After this action, the Control Panel can be accessed again. Select **Extra -> Reboot Resource** to restart the controller.

If the controller adopts the STOP/VALID CONFIGURATION state after restarting, the user program can again be started.

If the controller adopts the STOP/INVALID CONFIGURATION state after restarting, the user program must be reloaded into the NVRAM of the processor system.

Use the **Load Resource Configuration from Flash** command to read a backup copy of the last executable configuration from the flash memory of the communication system and transfer it to the NVRAM of the processor system. At this point, select **Resource -> Start (Cold Start)** to restart the user program with no need for performing a Download of the project.

To load a resource configuration from the flash memory of the communication system

1. Move to the ELOP II Factory Hardware Management to load the resource configuration.
2. Select and right click the required resource.
3. Select **Online -> Control Panel**. The Control Panel appears.
4. To restore the configuration and user program from the flash memory of the communication system, click the **Extra -> Load Resource Configuration from Flash** menu function. The user program is thus transferred from the flash memory of the user

program into the working memory of the processor system and the configuration into the NVRAM.

The resource configuration is thus restored.

7.6.9 Deleting a Resource Configuration from the Flash Memory of the Communication System

Delete Resource Configuration is generally used to remove the user program from the controller.

To do this, the processor system must be in STOP.

To delete a resource configuration from the flash memory of the communication system

1. In ELOP II Factory Hardware Management, select and right click the required resource.
2. Select **Online -> Control Panel** on the context menu. The Control Panel appears.
3. Select **Extra -> Delete Resource Configuration** to remove the configuration and user program from the flash memory of the communication system..

Deleting the configuration has the following effects:

- The controller adopts the STOP/INVALID CONFIGURATION state.
- The access to the user program in the working memory of the processor system is inhibited in this state.
- System ID, IP address and user management still exist in the NVRAM of the processor system such that a connection to the PADT can still be established.

Upon deletion, the controller can immediately be loaded with a new program. This action deletes the previous program from the working memory of the processor system.

Refer to the ELOP II Factory manual First Steps (HI 800 006 E) for further details about communication between PADT and controller.

7.7 Configuring Communication with ELOP II Factory - CPU-OS Versions Prior to 7

This chapter describes how to configure communication using ELOP II Factory for processor operating system versions prior to 7.

Depending on the application, the following elements must be configured:

- Ethernet/safeethernet, also referred to as peer-to-peer communication
- Standard protocols

For more details on how to configure the standard protocols, refer to the corresponding communication manuals:

- Send/Receive TCP (HI 800 117 E)
- Modbus Master/Slave (HI 800 003 E)
- PROFIBUS DP Master/Slave (HI 800 009 E)
- INTERBUS in ELOP II Factory Online Help
- EtherNet/IP in ELOP II Factory Online Help

7.7.1 Configuring the Ethernet Interfaces

Operating system version of the communication system 8.32 and lower:

For all Ethernet ports on the integrated Ethernet switch, the *Speed Mode* and *Flow Control Mode* parameters are set to Autoneg. No other setting is allowed, i.e., the system rejects settings other than Autoneg while loading the configuration.

The 10/100 BaseT Ethernet interface on the HIMatrix controllers and remote I/Os have the following parameters:

Speed Mode Autoneg
Flow Control Mode Autoneg

External devices that should communicate with HIMatrix controllers, must have the following network settings:

Parameter	Alternative 1	Alternative 2	Alternative 3	Alternative 4
<i>Speed Mode</i>	Autoneg	Autoneg	10 Mbit/s	100 Mbit/s
<i>Flow Control Mode</i>	Autoneg	Half Duplex	Half Duplex	Half Duplex

Table 36: Permissible Communication Settings for External Devices - CPU-OS Versions Prior to 7

The following network settings are not allowed:

Parameter	Alternative 1	Alternative 3	Alternative 4
<i>Speed Mode</i>	Autoneg	10 Mbit/s	100 Mbit/s
<i>Flow Control Mode</i>	Full Duplex	Full Duplex	Full Duplex

Table 37: Invalid Communication Settings for External Devices - CPU-OS Versions Prior to 7

With operating systems versions for the communication system higher than 8.32 (COM OS) and version 7.56.10 for ELOP II Hardware Management:

The operating parameters of each Ethernet port on the integrated Ethernet switch can be set individually.

For HIMatrix controllers and remote I/Os with extended settings, set the *Speed Mode* and *Flow Control Mode* to **Autoneg**. To ensure that the parameters of this dialog box become effective, the option *Activate Extended Settings* must be selected, see Figure 9.

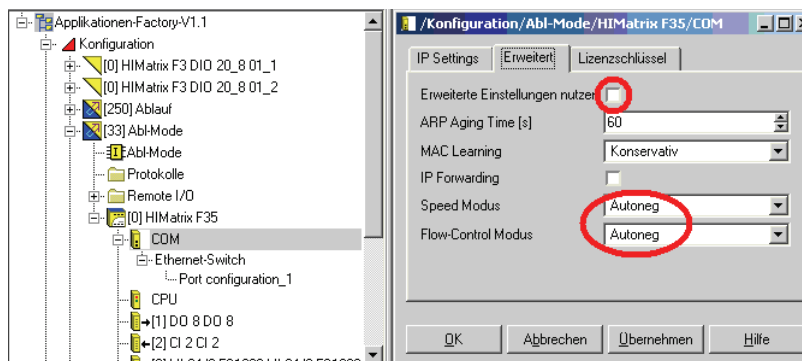


Figure 9: Communication System Properties - CPU-OS Versions Prior to 7

The parameters *ARP*, *MAC Learning*, *IP Forwarding*, *Speed Mode* and *Flow Control Mode* are explained in details in the ELOP II Factory online help.

i

Replacement of one controller with identical IP address:

If a controller has its *ARP Aging Time* set to 5 minutes and its *MAC Learning* set to **Conservative**, its communication partner does not adopt the new MAC address until a period of 5 to 10 minutes after the controller is replaced. During this time period, no communication is possible with the replaced controller.

The port settings of the integrated Ethernet switch on a HIMatrix resource can be configured individually starting with the following versions.

- Version 8.32 of the communication operating system and
- version 7.56.10 of ELOP II Hardware Management

Select **Ethernet Switch -> New -> Port Configuration** to define the configuration parameters for each switch port.

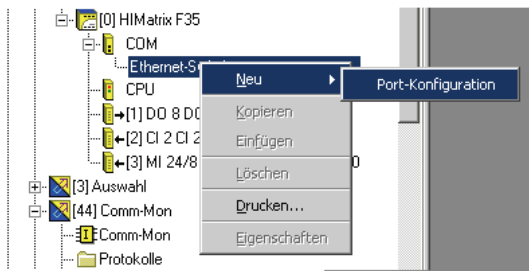


Figure 10: Creating a Port Configuration - CPU-OS Versions Prior to 7

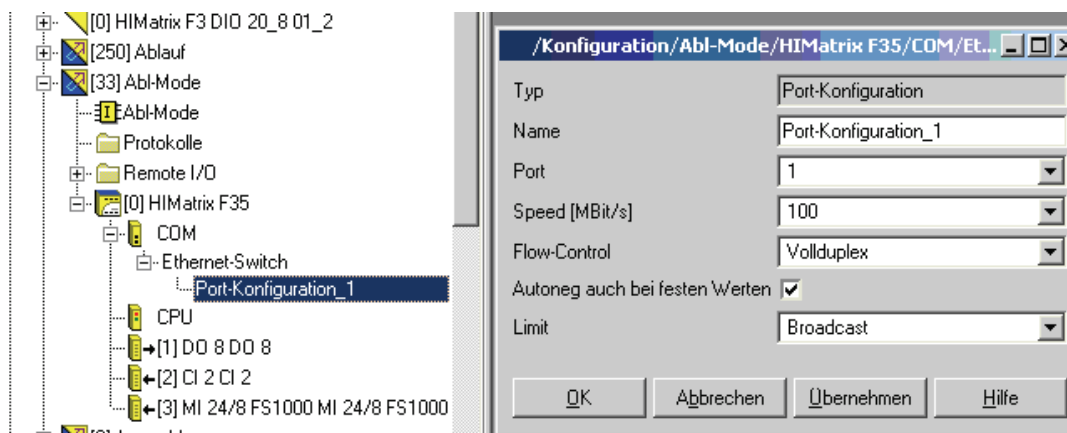


Figure 11: Parameters of a Port Configuration - CPU-OS Versions Prior to 7

Name	Explanation
Port	Port number as printed on the housing; per port, only one configuration may exist. Range of values: 1...n, depending on the resource
Speed [Mbit/s]	10 Mbit/s: Data rate 10 Mbit/s 100 Mbit/s: Data rate 100 Mbit/s Autoneg (10/100): Automatic baud rate setting Standard: Autoneg
Flow Control	Full duplex: Simultaneous communication in both directions Half duplex: Communication in both directions, but only one direction at a time Autoneg: Automatic communication control Standard: Autoneg
Autoneg also with Fixed Values	The "Advertising" function (forwarding the Speed and Flow Control properties) is also performed if <i>Speed</i> and <i>Flow Control</i> have fixed values. This allows other devices with ports set to Autoneg to recognize the HIMatrix ports' settings.
Limit	Limit the inbound multicast and/or broadcast packets. Off: No limitation Broadcast: Limit broadcast packets (128 kbit/s) Multicast and Broadcast: Limit multicast and broadcast packets (1024 kbit/s) Default: Broadcast

Table 38: Parameters of a Port Configuration - CPU-OS Versions Prior to 7

Click the **Apply** option to transfer the parameters to the communication system's configuration. The parameters set in the properties of the communication system and Ethernet switch (configuration) become operative for the HIMatrix communication, once they have been re-compiled with the user program and transferred to the controller.

The properties of the communication system and Ethernet switch can also be changed online using the Control Panel. These settings become operative immediately, but they are not adopted by the user program.

7.7.2 System Signals of safeethernet Communication

The user program can use system signals to read the status of the **safeethernet** communication (peer-to-peer communication) and of some time parameters. It can control peer-to-peer communication via the *Connection Control* system parameter.

The following signals are available for **safeethernet** communication:

Input Signals	[Data type], Unit/Value	Description
Receive Timeout	[UDINT] ms	Maximum time in ms that may elapse between the reception of two valid messages
Response Time	[UDINT] ms	Time in ms that elapsed while waiting for a response to the last message sent
Connection state	[UINT] 0 (CLOSED) 1 (TRY_OPEN) 2 (CONNECTED)	CLOSED: No connection TRY_OPEN: Attempt t establish the connection (state valid for active and passive sides) CONNECTED: The connection is established (active data exchange and time monitoring).
Version	[WORD]	Communication version signature

Table 39: System Signals of a **safeethernet** Connection for Reading the Status - CPU-OS Versions Prior to 7

Output signal	[Data type], Unit/Value	Description
Connection control	[WORD] 0x0000 0x0100 0x0101 0x8000	Commands: AUTOCONNECT TOGGLE_MODE_0 TOGGLE_MODE_1 DISABLED Used by a user program to close a safety-related protocol or enable it for operation. Refer to the following table for the corresponding description.

Table 40: System Signal of a safeethernet Connection for Setting the Connection Control - CPU-OS Versions Prior to 7

The following commands can be used for the *Connection Control* signal:

Command	Description
AUTOCONNECT	After a peer-to-peer communication loss, the controller attempts to reestablish communication in the following cycle. This is the default setting.
TOGGLE_MODE_0 TOGGLE_MODE_1	After a communication loss, the user program can re-establish the connection by changing the TOGGLE MODE. If TOGGLE MODE 0 is active and the communication is lost (Connection State = CLOSED), a reconnection is only attempted after the user program switched the TOGGLE MODE to TOGGLE MODE_1. If TOGGLE MODE 1 is active and the communication is lost, a reconnection is only attempted after the user program switched the TOGGLE MODE to TOGGLE MODE_0.
DISABLED	Peer-to-peer communication is off No attempt made to establish the connection

Table 41: The *Connection Control* Parameter - CPU-OS Versions Prior to 7

To evaluate system signals in the user program

1. Right-click the resource in the ELOP II Factory Hardware Management and select **P2P Editor** on the context menu to open it.
2. Select the row for the required resource.
3. Click the **Connect System Signals** button. The *P2P System Signals* window opens. Select the **Inputs** tab

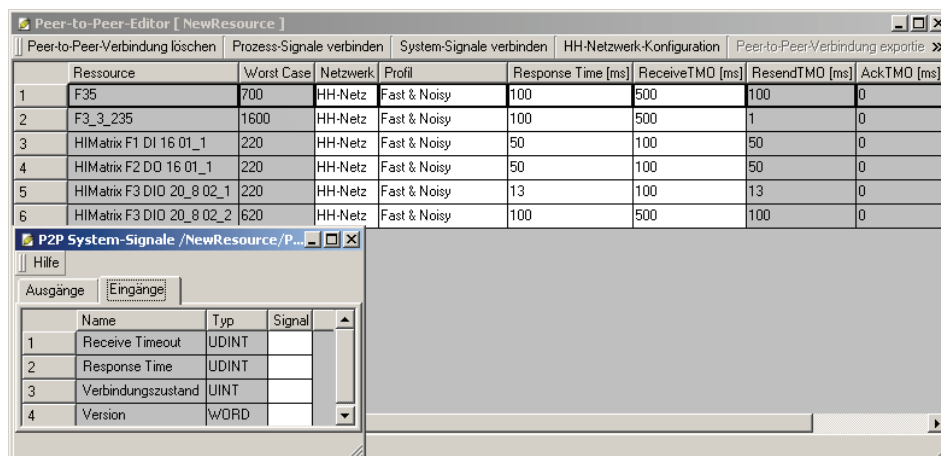


Figure 12: Peer-to-Peer Parameters in the **Inputs** Tab - CPU-OS Version Prior to 7

- The system parameters *Receive Timeout*, *Response Time*, *Connection State* and *Version* can be evaluated in the user program based on the signal assignment performed in the Signal Editor.

The status signals can be evaluated in the user program.

To set a system signal from the user program

- Right-click the resource in the ELOP II Factory Hardware Management and select **P2P Editor** on the context menu to open it.
- Select the row for the required resource.
- Click the **Connect System Signals** button. The *P2P System Signals* window opens. Select the **Inputs** tab

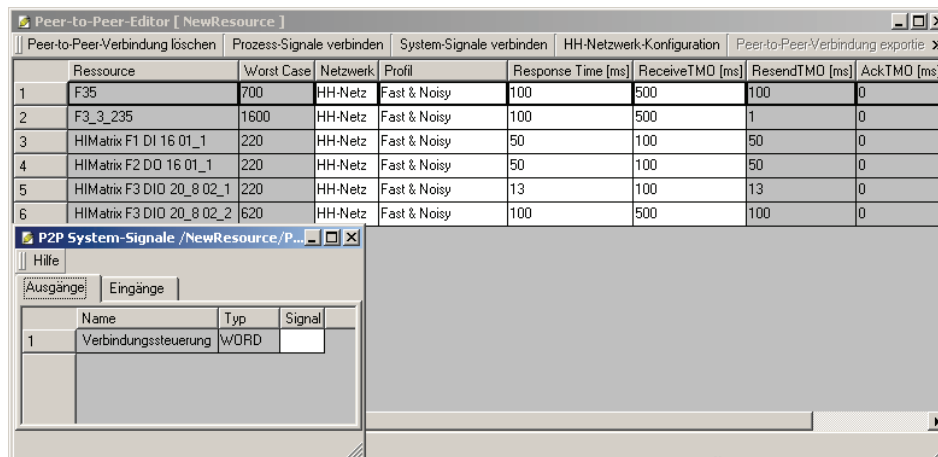


Figure 13: *Connection Control* System Signal in the **Outputs** Tab - CPU-OS Versions Prior to 7

The user program can set the *Connection Control* system signal.

7.7.3 Configuring the safeethernet connection

The following parameters can be set for a resource in the **P2P Editor**:

- Profile - see below
- Response Time

The response time is the time period that elapses until the sender of the message receives acknowledgement from the recipient.
- Receive TMO

ReceiveTMO is the monitoring time of PES1 within which a correct response from PES2 must be received.

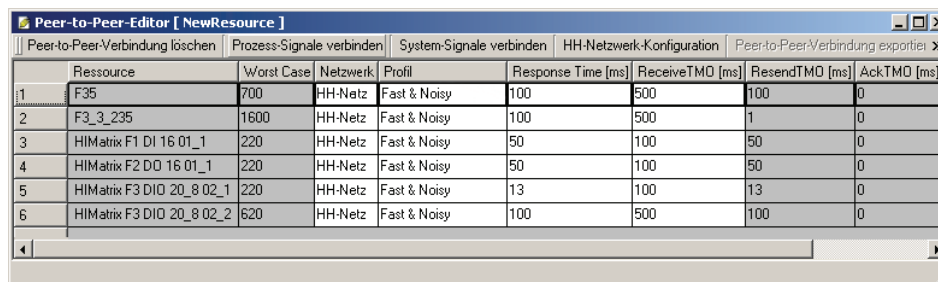


Figure 14: Setting the Parameters in the P2P Editor - CPU-OS Versions Prior to 7

The parameters specified above determine the data throughput and the fault and collision tolerance of the safeethernet connection.

Refer to the HIMatrix Safety manual (HI 800 023 E), Chapter *Configuring Communication*, for details on how to calculate the ReceiveTMO, Response Time and Worst Case Reaction Time.

Profile

Due to the high number of parameters, the manual network configuration is very complex and requires a thorough knowledge of the parameters and their mutual influence.

To simplify the parameter settings, six peer-to-peer profiles are available, among which the user can select the most suitable for his application and network.

The profiles are combinations of parameters compatible with one another that are automatically set when selecting the profile.

Profiles I through VI are described in details in the ELOP II Factory Hardware Management online help.

7.7.4 Configuring the Signals for safeethernet Communication

To be able to configure signals, a network (token group) must have been created, see ELOP II Factory manual First Steps (HI 800 006 E).

To configure the signals for safeethernet communication

1. In the P2P Editor, click a line number in the left-hand column to select the resource with which data should be exchanged.
2. In the P2P Editor, click **Connect Process Signals**.
 When the *Process Signals* opens for the first time, it is empty.
3. In the **Signals** menu, select **Editor** to open the Signal Editor.
4. Arrange the Signal Editor and P2P Process Signals windows adjacent to one another.
5. In the P2P Process Signals window, select the tab corresponding to the desired data transfer direction, e.g., from the resource selected in the structure tree to the resource selected in the P2P Editor.
6. Drag a signal name from the Signal Editor onto the desired row in the *P2P Process Signals* window.

As an option, use the **Add Signal** button. A row is created where the name of the signal can be entered; the signal name is case sensitive.

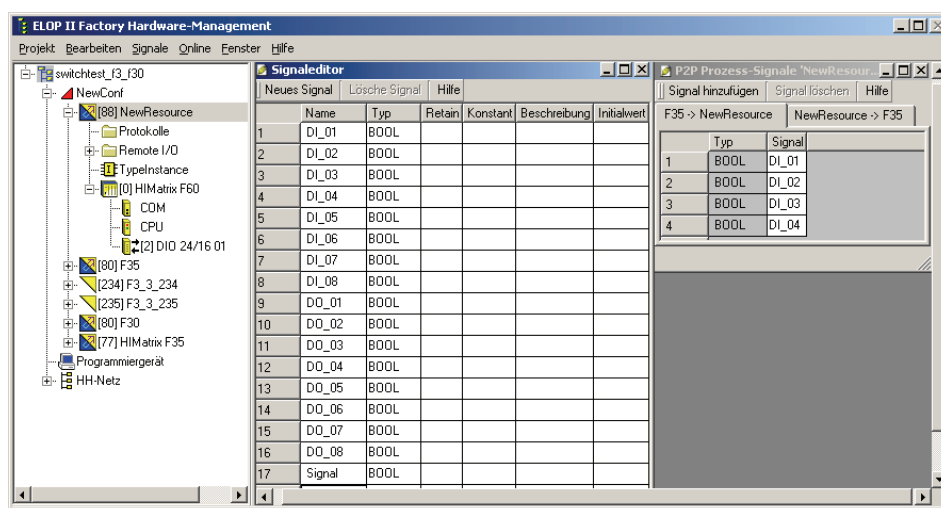


Figure 15: Assigning Process Signals per Drag&Drop - CPU-OS Versions Prior to 7

i By sending a signal value from a controller to another controller (PES₁ -> PES₂), the value is available in the second controller PES₂. To be able to use the value, use the same signals in the PES₁ and PES₂ logic.

7. Select the other tab in the P2P Process Signals window to switch the direction of the data exchange and define the signals for the other transfer direction.

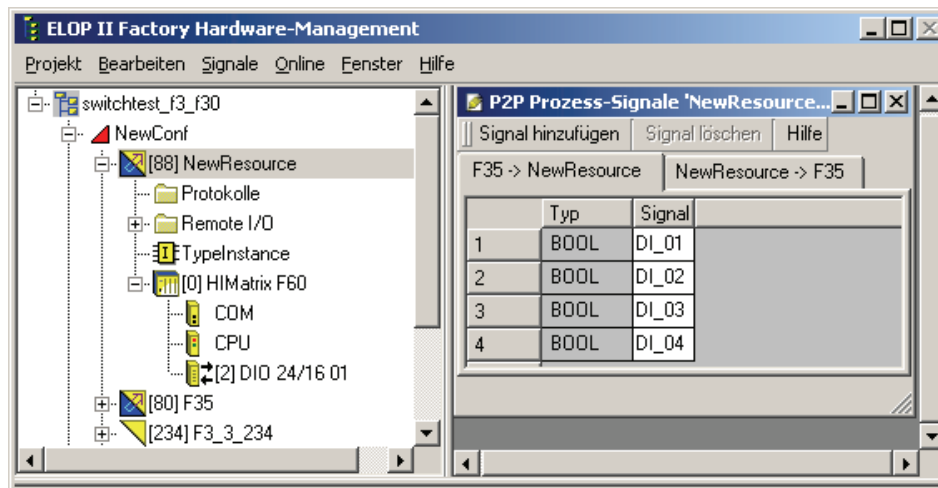


Figure 16: Example of Process Signals - CPU-OS Versions Prior to 7

The signals for safe**ethernet** communication are defined.

Monitoring the Transmitted Signals

Whenever a data packet is sent, the signal values currently available in the controller are used.

Since the PES cycle can be faster than packets are sent, it may be not possible to transfer all values if this is the case. To ensure the transfer and reception of a value, the monitoring time (ReceiveTMO) on the sending side must still be running to allow reception of the acknowledgment from the receiving side.

As an alternative, it is also possible to program an active acknowledgment signal within the application on the receiving side.

7.8 Handling the User Program

The PADT can be used to influence the program's function within the controller as follows:

7.8.1 Setting the Parameters and the Switches

During the user program's configuration, the parameters and the switches are set to offline and are loaded into the controller with the code-generated program. The parameters and the switches can also be set when the controller is in the STOP or RUN state, provided that the *main enable* switch has been activated. Only the elements in the NVRAM can be modified, all remaining elements are activated during the load procedure.

7.8.2 Starting the Program from STOP/VALID CONFIGURATION

Starting the program has the same effect as switching the controller's mode of operation from STOP/VALID CONFIGURATION to RUN; the program enters the RUN state too. The program enters the test mode if the test mode is active while starting the program. In accordance with IEC 61131, a cold or a warm start can also be performed in addition the starting in test mode.

i

The program can only be started if the *Start/Restart Allowed* switch was activated.

7.8.3 Restarting the Program after Errors

If the program enters the STOP/INVALID CONFIGURATION state, e.g., due to unauthorized access to operating system areas, it restarts. If the user program enters the STOP/INVALID CONFIGURATION state again within roughly one minute since the restart, it remains in this state. If this is the case, it can be restarted using the Control Panel's start button. After a restart, the operating system checks the entire program.

7.8.4 Stopping the Program

If the user program is stopped, the switches from RUN to STOP/VALID CONFIGURATION.

7.8.5 Program Test Mode

The test mode is started from the Control Panel, selecting Test Mode -> Test Mode with Hot Start (...Cold Start, ...Warm Start). Each Single Cycle command is used to activate a single cycle (one complete logic cycle).

Behavior of variable/signal values in test mode

The selection of cold, warm or hot start determines which variable values are used during the first cycle in test mode.

Cold start: all variables/signals are set to their initial values.

Warm start: retain signals retain their value, the remaining signals are set to their initial value.

Hot start: All variables/signals retain their current values.

Finally, the Cycle Step command can be used to start the user program in single step mode. All current values are retained for the following cycle (freezed state).

DANGER



**Property damage or physical injury possible due to actuators in unsafe state!
Do not use the test mode function in safety-related operation!**

7.8.6 Online Test

The Online Test function is used to add online test fields (OLT fields) to the program logic, to display and to force signals/variables while the controller is operating.

If the *Online Test Allowed* switch is on, the values of signals/variables can be manually entered in the corresponding OLT fields and thus forced. However, the forced values only apply until they are overwritten by the program logic.

If the *Online Test Allowed* switch is off, the values of the signals/variables in OLT fields are only displayed and cannot be modified.

For more information on how to use OLT fields, enter `OLT field` in the online help of the programming tool.

8 Operation

This chapter describes how to handle and diagnose the controller during its operation.

8.1 Handling

The controller needs not be handled during its normal operation. Only if problems arise, an intervention with the PADT may be required.

8.2 Diagnosis

A first, rough diagnosis can be performed via the light-emitting diodes (LEDs). The diagnostic history that can be displayed using the programming tool provides a more detailed analysis of the operating or error state.

8.2.1 Light-Emitting Diode Indicators

The light emitting diodes (LEDs) indicate the module state. The functionality and meaning of LEDs depend on the used version of the processor operating system. Details are described in the corresponding device or module manuals.:

The functionality and meaning of the fieldbus LEDs is described in the communication manuals.

CPU-OS Versions	Manual	Document number
Version 7 and newer	SILworX Communication Manual	HI 801 101 E
Versions prior to 7	HIMatrix PROFIBUS DP Master/Slave Manual	HI 800 009 E
	HIMatrix Modbus Master/Slave Manual	HI 800 003 E
	HIMatrix TCP S/R Manual	HI 800 117 E
	HIMatrix ComUserTask (CUT) Manual	HI 800 329 E

Table 42: Manuals Describing the Communication LEDs

8.2.2 Diagnostic History

The diagnostic history records the various states of the processor system and communication system and stores them in a non-volatile memory. Both systems include a short term and a long term diagnosis. The number of entries in the diagnostic history depends on the operating system version (prior to 7 or beyond 7).

	CPU	COM
Number of entries in the long term diagnosis	700	300
Number of entries in the short term diagnosis	700	700

Table 43: Number of Entries in the Diagnostic History - L3

	CPU	COM
Number of entries in the long term diagnosis	935	230
Number of entries in the short term diagnosis	468	655

Table 44: Number of Entries in the Diagnostic History - CPU-OS Version7 and Newer

	CPU	COM
Number of entries in the long term diagnosis	1000	200/250 ¹⁾
Number of entries in the short term diagnosis	500	700/800 ¹⁾
1) Higher value for COM operating system version 4 and beyond		

Table 45: Maximum Number of Entries in the Diagnostic History - CPU-OS Versions Prior to 7

The long-term diagnosis of the processor system includes the following events:

- Rebooting,
- Changing the mode of operation (INIT, RUN, STOP/VALID CONFIGURATION, STOP/INVALID CONFIGURATION),
- Changing the program mode of operation (START, RUN, ERROR, TEST MODE),
- Loading or deleting a configuration,
- Setting and resetting switches,
- Processor system failures,
- Downloading an operating system,
- Forcing (setting and resetting the force switch is allowed),
- I/O module diagnostics,
- Power supply and temperature diagnostics.

The long-term diagnosis of the communication system includes the following events:

- Rebooting the communication system,
- Changing the mode of operation (INIT, RUN, STOP/VALID CONFIGURATION, STOP/INVALID CONFIGURATION),
- User log-in,
- Loading the operating system.

If the memory for the long term diagnosis is full, all data older than three days is deleted allowing new entries to be stored. If no data is older than three days, the new entries cannot be stored and get lost. A message in the long-term diagnosis warns that it was not possible to store the data.

The short-term diagnosis of the processor system includes the following events:

- Processor system diagnostics (setting the force switches and force values),
- User program diagnostics (cyclic operation),
- Communication diagnostics,
- Power supply and temperature diagnostics,
- I/O module diagnostics.

The short-term diagnosis of the communication system includes the following events:

- **safeethernet**-related events
- Start / stop while writing the flash memory
- Faults that can occur while loading a configuration from the flash memory
- Unsuccessful time synchronization between the communication system and the processor system

Parameter errors associated with the inputs or outputs are possibly not detected during the code generation. If a parameter error occurs, the message INVALID CONFIG with the error

source and code are displayed in the feedback box for the diagnosis. This message helps analyzing errors due to an incorrect configuration of the inputs or outputs.

If the memory for the short-term diagnosis is full, the oldest entries are deleted to allow new data to be saved. No message appears warning that old entries are being deleted.

Diagnostic data recording is not safety-related. To read the data recorded in chronological order, use the programming tool. Reading does not delete the data stored in the controller. The programming tool is capable of storing the contents of the diagnostic window.

8.2.3 Diagnosis in SILworX- CPU-OS Version 7 and Newer

Use the Online View in the SILworX Hardware Editor to access to the diagnostic panel.

To open the diagnostic panel

1. Select the **Hardware** branch located beneath the required resource.
2. Click **Online** on the context menu or on the Action Bar.
 - The system log-in window opens.
3. In the system log-in window, select or enter the following information:
 - IP address of the controller
 - User name and password.
 - The Hardware Editor's Online View opens.
4. In the Online View, select the required module, usually the processor or the communication module.
5. Select **Diagnosis** on the context menu or on the **Online** menu.

The diagnostic panel for the required module appears.

With an operating controller, messages about the state of the processor system, communication system and I/O modules are displayed at specific, user-defined time intervals.

8.2.4 Diagnosis in ELOP II Factory - CPU-OS Versions Prior to 7

Select the corresponding resource in the ELOP II Factory's Hardware Management to access the diagnostic panel.

To open the diagnostic panel

1. Select and right click the required resource.
2. The context menu opens. Select **Online**, and then select **Diagnosis**.
3. If not yet already done, log in to the resource in the corresponding window.

The diagnostic panel appears.

With an operating controller, messages about the state of the processor system, communication system and I/O modules are displayed at specific, user-defined time intervals.

9 Maintenance

The maintenance of HIMatrix systems is restricted to the following:

- Removing disturbances
- Replacing the back-up battery, if required.
- Loading operating systems

9.1 Disturbances

Disturbances in the CPU 01 processor system mostly result in the complete shut-down of the controller and are indicated via the *ERR* LED on the CPU 01.

For possible reasons for the *ERR* LED to light, refer to the CPU 01 manual (HI 800 189 E).

To turn off the indicator, start the **Reboot Resource** command located in the **Extra** menu associated with the Control Panel. The controller is booted and re-started.

The system automatically detects disturbances in the input and output channels during operation and displays them via the *FAULT* LED on the front plate of the corresponding module.

Even if the controller is stopped, the PADT diagnostic history can be used to read out detected faults, provided that communication was not disturbed as well.

Prior to replacing a module, check whether an external line disturbance exists and whether the corresponding sensor and/or actuator is ok.

9.2 Back-up Battery

The PS 01 power supply is equipped with a back-up battery of type CR 1/2 AA 3 V Lithium to store data and operate the clock when the 24 V supply voltage has failed.

The back-up battery must be replaced every 4 years.

The battery may be replaced during operation. When doing so, the system ID and the IP address in the CPU NVRAM are not lost and need not be reloaded into the controller.

Refer to the PS 01 power supply manual (HI 800 211 E) for more information on how to replace the back-up battery.

9.3 Replacing Fans

HIMA recommends replacing the fans of the HIMatrix F60 on a regular basis to prevent the fans to fail:

- At normal temperatures (< 40 °C): every 5 years
- At increased temperatures (> 40 °C): every 3 years

The fans must be replaced by HIMA service.

9.4 Loading Operating Systems

The processor and communication systems have different operating systems that are stored in the rewritable flash memories and can be replaced, if necessary.

NOTE



Disruption of the safety-related operation!

The controller must be in the STOP state to enable the programming tool to load new operating systems.

During this time period, the operator must ensure the plant safety, e.g., by taking organizational measures.

i

- The programming tool prevents controllers from loading the operating systems in the RUN state and reports this as such.
- Interruption or incorrect termination of the loading process has the effect that the controller is no longer functional. However, it is possible to reload operating system.

The operating system for the processor system (processor operating system) must be loaded before that for the communication system (communication operating system).

Operating systems for controllers differ from those for remote I/Os.

To be able to load a new operating system, it must be stored in a directory that can be accessed by the programming tool.

9.4.1 Loading the Operating System with SILworX

Use SILworX if the processor operating system version loaded in the controller is **7 or beyond**.

To load the new operating system

1. Set the controller to the STOP state, if it has not already been done.
2. Open the Online View of the hardware and log in to the controller with administrator rights.
3. Right-click the module, processor or communication module.
4. The context menu opens. Click **Maintenance/Service->Load Module Operating System**.
5. In the *Load Module Operating System* dialog box, select the type of operating system that should be loaded.
6. A dialog box for selecting a file opens. Select the file with the operating system that should be loaded and click Open.

SILworX loads the new operating system into the controller.

9.4.2 Loading the Operating System with ELOP II Factory

Use the ELOP II Factory programming tool if the processor operating system version loaded in the controller is **prior to 7**.

To load the new operating system

1. Set the controller to the STOP state, if it has not already been done.
2. Log in to the controller with administrator rights.
3. In ELOP II Factory Hardware Management, right click the required resource.
4. The context menu opens. On the **Online** submenu, select **Control Panel**.
 - The Control Panel opens.

- 5 On the **Extra** menu, **OS Update** submenu, select the type of operating system that should be loaded (processor operating system, communication operating system).
 A dialog box for selecting a file opens.
- 6 In this dialog box, move to the directory in which the operating system is stored and select it.
7. Click **OK** to load the operating system.

The operating system is loaded into the controller. The controller restarts and enters the STOP state.

After an operating system has been loaded, the controller also enters the STOP state if a program is loaded with the *Autostart* safety parameter set to TRUE.

The following is possible:

- Repeating the described sequence, further operating systems can be loaded, e.g., the operating system for the communication system, after the operating system for the processor system.
- The controller can be set to the RUN state.

9.4.3 Switching between ELOP II Factory and SILworX - not with L3

HIMatrix controllers (except for hardware layout 3) can either be programmed with ELOP II Factory or with SILworX, if the appropriate version for the processor operating system is installed. The combinations of programming tool and operating system version are specified in the table.

Operating system	Version for ELOP II Factory	Version for SILworX
Processor system	Versions prior to 7	Version 7 and newer
Communication system	Versions prior to 12	Versions 12 and newer
OS loader	Versions prior to 7	Versions 7 and newer

Table 46: Operating System Versions and Programming Tools

9.4.3.1 Upgrading from ELOP II Factory to SILworX

This upgrade may only be used for HIMatrix controllers and remote I/Os with newer layouts. Any attempt to use it with controllers and remote I/Os with previous layouts leads to failures that can only be removed by HIMA.

i

- HIMatrix controllers that can be programmed with SILworX, are only compatible with remote I/Os that can also be programmed with SILworX. For this reason, also ensure that the appropriate remote I/O is used.
- For F60 systems, no upgrade other than that of the processor module is required. The operating system of the processor module determines the programming tool.
- The user program cannot be converted from ELOP II Factory to SILworX and vice-versa.
- Please contact HIMA service if it is not clear whether a given controller or remote I/O may be upgraded.

Update the operating system loader (OSL) when performing an upgrade.

To prepare a HIMatrix controller for being programmed with SILworX

1. Use ELOP II Factory to load the processor operating system into the controller, versions beyond 7.
2. Use ELOP II Factory to load the communication operating system into the controller, version 12 and beyond.
3. Use SILworX to load the OSL into the controller, versions beyond 7.

The controller must be programmed with SILworX.

9.4.3.2 Downgrading from SILworX to ELOP II Factory

In rare cases, it can be necessary changing a controller or remote I/O to be programmed using ELOP II Factory instead of SILworX.

To prepare a HIMatrix controller for being programmed with ELOP II Factory

- 1 Use SILworX to load the OSL into the controller, version prior to 7.
2. Use SILworX to load the processor operating system into the controller, for CPU versions prior to 7.
3. Use SILworX to load the communication operating system into the controller, for COM versions prior to 12.

The controller must be programmed with ELOP II Factory.

i

Controllers with layout 3 and operating system version beyond 8 cannot be adapted to be programmed using ELOP III!

10 Decommissioning

Remove the supply voltage to decommission the modular controller. Afterwards it is possible to pull out the pluggable screw terminal connector blocks for inputs and outputs and the Ethernet cables, and to remove the module.

11 Transport

To avoid mechanical damage, HIMatrix components must be transported in packaging.

Always store HIMatrix components in their original product packaging. This packaging also provides protection against electrostatic discharge. Note that the product packaging alone is not suitable for transmission.

12 Disposal

Industrial customers are responsible for correctly disposing of decommissioned HIMatrix hardware. Upon request, a disposal agreement can be arranged with HIMA.

All materials must be disposed of in an ecologically sound manner.

Appendix

Glossary

Term	Description
ARP	Address Resolution Protocol: Network protocol for assigning the network addresses to hardware addresses
AI	Analog Input
COM	COMmunication module
CRC	Cyclic Redundancy Check
DI	Digital Input
DO	Digital Output
ELOP II Factory	Programming tool for HIMatrix systems
EMC	ElectroMagnetic Compatibility
EN	European Norm
ESD	ElectroStatic Discharge
FB	FieldBus
FBD	Function Block Diagrams
FTA	Field Termination Assembly
FTT	Fault Tolerance Time
ICMP	Internet Control Message Protocol: Network protocol for status or error messages
IEC	International Electrotechnical Commission
MAC address	Media Access Control address: Hardware address of one network connection
PADT	Programming And Debugging Tool (in accordance with IEC 61131-3), PC with SILworX or ELOP II Factory
PE	Protective Earth
PELV	Protective Extra Low Voltage
PES	Programmable Electronic System
PFD	Probability of Failure on Demand, probability of failure on demand of a safety function
PFH	Probability of Failure per Hour, probability of a dangerous failure per hour
R	Read: The system variable or signal provides value, e.g., to the user program
Rack ID	Base plate identification (number)
Non-reactive	Supposing that two input circuits are connected to the same source (e.g., a transmitter). An input circuit is termed <i>non-reactive</i> if it does not distort the signals of the other input circuit.
R/W	Read/Write (column title for system variable/signal type)
SB	System Bus (module)
SELV	Safety Extra Low Voltage
SFF	Safe Failure Fraction, portion of safely manageable faults
SIL	Safety Integrity Level (in accordance with IEC 61508)
SILworX	Programming tool for HIMatrix systems
SNTP	Simple Network Time Protocol (RFC 1769)
S.R.S	System.Rack.Slot addressing of a module
SW	Software
TMO	TiMeOut
W	Write: System variable/signal is provided with value, e.g., from the user program
WD	WatchDog: Time monitoring for modules or programs. If the watchdog time is exceeded, the module or program enters the ERROR STOP state.
WDT	WatchDog Time

Index of Figures

Figure 1:	Line Control	17
Figure 2:	safeethernet/Ethernet Networking Example	24
Figure 3:	CPU Cycle Sequence with Multitasking	31
Figure 4:	Multitasking Mode 1	34
Figure 5:	Multitasking Mode 2	35
Figure 6:	Multitasking Mode 3	36
Figure 7:	Securing the F60 Subrack	45
Figure 8:	Securing the Cables and Connecting the Shielding	47
Figure 9:	Communication System Properties - CPU-OS Versions Prior to 7	76
Figure 10:	Creating a Port Configuration - CPU-OS Versions Prior to 7	77
Figure 11:	Parameters of a Port Configuration - CPU-OS Versions Prior to 7	77
Figure 12:	Peer-to-Peer Parameters in the Inputs Tab - CPU-OS Version Prior to 7	79
Figure 13:	<i>Connection Control</i> System Signal in the Outputs Tab - CPU-OS Versions Prior to 7	80
Figure 14:	Setting the Parameters in the P2P Editor - CPU-OS Versions Prior to 7	80
Figure 15:	Assigning Process Signals per Drag&Drop - CPU-OS Versions Prior to 7	81
Figure 16:	Example of Process Signals - CPU-OS Versions Prior to 7	82

Index of Tables

Table 1:	HiMatrix System Variants	7
Table 2:	Additional Relevant Documents	8
Table 3:	Standards for EMC, Climatic and Environmental Requirements	12
Table 4:	General requirements	12
Table 5:	Climatic Requirements	12
Table 6:	Mechanical Tests	13
Table 7:	Interference Immunity Tests	13
Table 8:	Noise Emission Tests	13
Table 9:	Review of the DC Supply Characteristics	14
Table 10:	Operating Voltage Monitoring	18
Table 11:	Temperature Monitoring	18
Table 12:	Specifications of F60	20
Table 13:	Communication Protocols and Interfaces	21
Table 14:	Connection of Controllers and Remote I/Os with Different Operating Systems	25
Table 15:	Functions of the Processor Operating System	27
Table 16:	Modes of Operation for the Processor System	29
Table 16:	User Program Modes of Operation	30
Table 18:	Parameters Configurable for Multitasking	32
Table 19:	Reloading after Changes - With L3	38
Table 20:	Effect of the <i>Force Deactivation</i> System Variable	42
Table 21:	Force Switches and Parameters Prior to V.7	43
Table 20:	Connectors for Operating Voltage	47
Table 23:	System Parameters of the Resource - CPU-OS Version 7 and Newer	51
Table 24:	Hardware System Variables - CPU-OS V.7 and Newer	51
Table 25:	Hardware System Variables for Reading the Parameters	54
Table 26:	System Parameters of the User Program - CPU-OS Version 7 and Newer	55
Table 25:	Authorization Types for the PADT User Management Scheme	61
Table 26:	Parameters for User Accounts in the PES User Management Scheme	63
Table 29:	Parameters of the Port Configuration - CPU-OS Version 7 and Newer	65
Table 29:	Parameters for Boolean Events	66
Table 30:	Parameters for Scalar Events	68
Table 32:	Resource Configuration Parameters - CPU-OS Versions Prior to 7	69
Table 31:	General System Signals and Parameters - CPU-OS Versions Prior to 7	70
Table 34:	User program Parameters - CPU-OS Versions Prior to 7	71
Table 35:	Sub-States Associated with STOP - CPU-OS Versions Prior to 7	74
Table 36:	Permissible Communication Settings for External Devices - CPU-OS Versions Prior to 7	76
Table 37:	Invalid Communication Settings for External Devices - CPU-OS Versions Prior to 7	76
Table 38:	Parameters of a Port Configuration - CPU-OS Versions Prior to 7	78

Table 39:	System Signals of a safeethernet Connection for Reading the Status - CPU-OS Versions Prior to 7	78
Table 40:	System Signal of a safeethernet Connection for Setting the Connection Control - CPU-OS Versions Prior to 7	79
Table 41:	The <i>Connection Control</i> Parameter - CPU-OS Versions Prior to 7	79
Table 44:	Manuals Describing the Communication LEDs	84
Table 46:	Number of Entries in the Diagnostic History - L3	84
Table 46:	Number of Entries in the Diagnostic History - CPU-OS Version7 and Newer	84
Table 47:	Maximum Number of Entries in the Diagnostic History - CPU-OS Versions Prior to 7	85
Table 46:	Operating System Versions and Programming Tools	89

Declaration of Conformity

For the HIMatrix system, declarations of conformity exist for the following directives:

- EMC Directive
- Low Voltage Directive
- EX Directive

The current declarations of conformity are available on the HIMA website www.hima.com.

Index

Alarm (see events) - L3.....	19	defining - L3	65
analog inputs		recording - L3	20
use - CPU-OS version 7.0 and newer ..	56	faults	
analog outputs		internal	28
use - CPU-OS version 7 and newer	57	permanent on I/O	27
analoge inputs		reaction to	27
use - CPU-OS versions prior to 7.0.....	71	temporary on I/Os	28
analoge outputs		forcing	
use - CPU-OS versions prior to 7.0.....	71	CPU-OS version 7.0 and beyond.....	40
communication		CPU-OS versions prior to 7	42
configuration - CPU-OS version 7 and		restrictions CPU-OS version 7 and newer	
newer	64	42
configuration - CPU-OS versions prior to		switches and parameters - CPU-OS	
7	75	versions prior to 7	43
configuring the Ethernet interfaces -		forcing	39, 40
CPU-OS version 7 and newer	64	Forcing with L2	41
communication time slice		Forcing with L3	40
maximum	23	Hardware Editor.....	51
counter inouts		Online Test	83
use - CPU-OS version 7.0 and newer ..	56	operating requirements	
counter inputs		climatic	12
use- CPU-OS versions prior to 7.0.....	71	EMC	13
de-energize to trip principle.....	11	ESD protection.....	14
device monitoring.....	18	mechanical.....	13
diagnosis		power supply	14
ELOP II Factory.....	86	operating system	27
SILworX.....	86	PADT user management.....	61
diagnostic history	84	PES user management	61
digital inputs		processor system	
use with CPU-OS version 7 and newer	56	modes of operation	28
digital outputs		processor system	28
use - CPU-OS version 7.0 and beyond	57	safeethernet.....	22
digitale inputs		configuring signals - CPU-OS versions	
use - CPU-OS versions prior to 7.0.....	71	prior to 7	81
digitale outputs		monitoring the signals- CPU-OS versions	
use - CPU-OS versions prior to 7.0.....	71	prior to 7	82
energize to trip principle.....	11	profile - CPU-OS versions prior to 7 ...	81
Ethernet	21	system signals - CPU-OS versions prior	
switch	21	to 7	78
terminals.....	23	temperature monitoring	18
Ethernet interfaces		user account.....	61
configuration - CPU-OS versions prior to		user group	61
7	75	user program	30
event		restart after errors	83
in general - L3	19	stop	83
events		test mode	83
creating - L3	19		



SAFETY
NONSTOP

HIMA Paul Hildebrandt GmbH + Co KG

P.O. Box 1261

68777 Brühl, Germany

Phone: +49 6202 709-0

Fax: +49 6202 709-107

E-mail: info@hima.com Internet: www.hima.com

(1130)